

# pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> について

中野 賢 & 日本語 T<sub>E</sub>X 開発コミュニティ

作成日 : 2019/09/29

## 注意 :

この文書では「コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>」について簡単に説明します。株式会社アスキー（現アスキー・メディアワークス<sup>1</sup>）が配布している pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>（以下、アスキー版 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>）とは異なる内容ですので、注意してください。

2010 年以降、アスキー pT<sub>E</sub>X<sup>2</sup>は、国際的に広く使われている T<sub>E</sub>X Live というディストリビューションに取り込まれ、そこで独自の改良や仕様変更が加えられてきました。最近（2011 年以降）の T<sub>E</sub>X Live や W32T<sub>E</sub>X では、pL<sup>A</sup>T<sub>E</sub>X も元々の pT<sub>E</sub>X ではなく、その拡張版 ε-pT<sub>E</sub>X をエンジンに用いるようになっています。また、pL<sup>A</sup>T<sub>E</sub>X のベースである L<sup>A</sup>T<sub>E</sub>X も更新が進められています。

こうした流れにあわせた新しい pL<sup>A</sup>T<sub>E</sub>X として、アスキー版から fork して日本語 T<sub>E</sub>X 開発コミュニティ (Japanese T<sub>E</sub>X Development Community) が配布しているものが、コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X です。開発中の版は GitHub のリポジトリ<sup>3</sup>で管理しています。コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X はアスキー版とは異なりますので、バグレポートはアスキー宛てではなく、日本語 T<sub>E</sub>X 開発コミュニティに報告してください。T<sub>E</sub>X Forum や GitHub の Issue システムが利用できます。

この文書 (platex.pdf) はコミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の概要を説明したのですが、内容はアスキー版（1995 年頃）からほとんど変わっていませんので、今では歴史的な文書ということにしておきます。最近の pL<sup>A</sup>T<sub>E</sub>X の更新内容は pL<sup>A</sup>T<sub>E</sub>X ニュース（アスキー版 : plnews\*.pdf、コミュニティ版 : plnewsc\*.pdf）を参照してください。また、実際の pL<sup>A</sup>T<sub>E</sub>X のソースコードは pldoc.pdf で説明しています。

---

<sup>1</sup>アスキー日本語 T<sub>E</sub>X (pT<sub>E</sub>X)、<http://ascii.asciimw.jp/pb/ptex/>

<sup>2</sup>アスキー日本語 pT<sub>E</sub>X: <https://asciidwango.github.io/ptex/>

<sup>3</sup><https://github.com/texjporg/platex>

## 1 この文書について

この文書は pL<sup>A</sup>T<sub>ε</sub>X の概要を示していますが、使い方のガイドではありません。pL<sup>A</sup>T<sub>ε</sub>X の機能全般については、[1] を参照してください。また、[2] で説明されていた縦組向けの拡張コマンドについては、p1doc.pdf 中の p1ext.dtx の項目を参照してください。

日本語の組版処理については、pT<sub>ε</sub>X (あるいはその前身の「日本語 T<sub>ε</sub>X」) に関する文献 [3] や [4] (英語), [5] (英語) も併せてご参照ください。

L<sup>A</sup>T<sub>ε</sub>X の機能については、[7] や [9] などを参照してください。新しい機能については usrguide.tex を参照してください。pL<sup>A</sup>T<sub>ε</sub>X のコマンド一覧は「pL<sup>A</sup>T<sub>ε</sub>X チートシート」(platexsheet.pdf) またはその jsclasses 版 (platexsheet-jsclasses.pdf) が参考になるでしょう<sup>4</sup>。

この文書の構成は次のようになっています。

**第 1 節** この節です。この文書についての概要を述べています。

**第 2 節** pL<sup>A</sup>T<sub>ε</sub>X で拡張した機能についての概要です。付属のクラスファイルやパッケージファイルについても簡単に説明しています。

**第 3 節** 現在のバージョンの pL<sup>A</sup>T<sub>ε</sub>X と旧バージョン、あるいは元となっている L<sup>A</sup>T<sub>ε</sub>X との互換性について述べています。

**付録 A** この文書ソース (platex.dtx) の DOCSTRIP のためのオプションについて述べています。

**付録 B** pL<sup>A</sup>T<sub>ε</sub>X の dtx ファイルをまとめて、一つのソースコード説明書にするための文書ファイルの説明をしています。

**付録 C** 付録 B で説明した文書ファイルを処理する sh スクリプト (手順)、DOCSTRIP 文書ファイル内の入れ子の対応を調べる perl スクリプトなどについて説明しています。

## 2 pL<sup>A</sup>T<sub>ε</sub>X の機能について

pL<sup>A</sup>T<sub>ε</sub>X が提供するファイルは、次の 3 種類に分類することができます。

- フォーマットファイル
- クラスファイル
- パッケージファイル

---

<sup>4</sup>両者の PDF とも、コマンドラインで `texdoc -l platexcheat` を実行すると表示されます。

フォーマットファイルには、基本的な機能が定義されており、pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> 2<sub>ε</sub> の核となるファイルです。このファイルに定義されているマクロは、実行時の速度を高めるために、あらかじめ T<sub>E</sub>X の内部形式の形で保存されています。

クラスファイルは文書のレイアウトを設定するファイル、パッケージファイルはマクロの拡張を定義するファイルです。前者は `\documentclass` コマンドを用いて読み込み、後者は `\usepackage` コマンドを用いて読み込みます。

古い pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> 2.09 ユーザへの注意：

クラスファイルとパッケージファイルは、従来、スタイルファイルと呼ばれていたものです。L<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> 2<sub>ε</sub> ではそれらを、レイアウトに関するものをクラスファイルと呼び、マクロの拡張をするものをパッケージファイルと呼んで区別するようになりました。

## 2.1 フォーマットファイル

pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> のフォーマットファイルを作成するには、ソースファイル “`platex.ltx`” を  $\epsilon$ -pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> の INI モードで処理します<sup>5</sup>。ただし、T<sub>E</sub>X Live や W32T<sub>E</sub>X ではこの処理を簡単にする `fmtutil-sys` あるいは `fmtutil` というプログラムが用意されています。以下を実行すれば、フォーマットファイル `platex.fmt` が作成されます。

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

次のリストが、`platex.ltx` の内容です。ただし、このバージョンでは、L<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> から pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> への拡張を `plcore.ltx` をロードすることで行ない、`latex.ltx` には直接、手を加えないようにしています。したがって `platex.ltx` はとても短いものとなっています。`latex.ltx` には L<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> のコマンドが、`plcore.ltx` には pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> で拡張したコマンドが定義されています。

```
1 (*plcore)
```

`latex.ltx` の末尾で使われている `\dump` をいったん無効化します。

```
2 \let\orgdump\dump
```

```
3 \let\dump\relax
```

`latex.ltx` を読み込みます。T<sub>E</sub>X Live の標準的インストールでは、この中で Babel 由来のハイフネーション・パターン `hyphen.cfg` が読み込まれるはずですが。

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

`plcore.ltx` を読み込みます。

```
5 \typeout{*****~^J%
```

```
6      *~^J%
```

```
7      * making pLaTeX format~^J%
```

---

<sup>5</sup>2016 年以前は pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> と  $\epsilon$ -pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> のどちらでもフォーマットを作成することができましたが、2017 年に L<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> が  $\epsilon$ -T<sub>E</sub>X 必須となったことに伴い、pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> も  $\epsilon$ -pL<sup>A</sup>T<sub>ε</sub>E<sub>X</sub> が必須となりました。

```

8      *^^J%
9      *****}
10 \makeatletter
11 \input plcore.ltx

```

フォント関連のデフォルト設定ファイルである、pldefs.ltxを読み込みます。T<sub>E</sub>Xの入力ファイル検索パスに設定されているディレクトリにpldefs.cfgファイルがある場合は、そのファイルを使います。

```

12 \InputIfFileExists{pldefs.cfg}
13     {\typeout{*****^^J%
14             * Local config file pldefs.cfg used^^J%
15             *****}}%
16     {\input{pldefs.ltx}}

```

以前のバージョンでは、フォーマット作成時にpL<sup>A</sup>T<sub>E</sub>Xのバージョンがわかるように、端末に表示していましたが、\everyjobにバナー表示以外のコードが含まれる可能性を考慮し、安全のためやめました。

```
17 %\the\everyjob
```

pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>の起動時にplatex.cfgがある場合、それを読み込むようにします。バージョン2016/07/01ではコードをplcore.ltxに入れていましたが、platex.ltxへ移動しました。

```

18 \everyjob\expandafter{%
19   \the\everyjob
20   \IfFileExists{platex.cfg}{%
21     \typeout{*****^^J%
22             * Loading platex.cfg.^^J%
23             *****}}%
24   \input{platex.cfg}}{%
25 }

```

フォーマットファイルにダンプします。

```

26 \let\dump\orgdump
27 \let\orgdump\@undefined
28 \makeatother
29 \dump
30 %\endinput
31 </plcore>

```

実際にpL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>への拡張を行なっているplcore.ltxは、DOCSTRIPプログラムによって、次のファイルの断片が連結されたものです。

- plvers.dtx は、pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>のフォーマットバージョンを定義しています。
- plfonts.dtx は、NFSS2を拡張しています。
- plcore.dtx は、上記以外のコマンドでフォーマットファイルに格納されるコマンドを定義しています。

また、プリロードフォントや組版パラメータなどのデフォルト設定は、`platex.ltx` の中で `pldefs.ltx` をロードすることにより行います<sup>6</sup>。このファイル `pldefs.ltx` も `plfonts.dtx` から生成されます。

注意：

このファイルに記述されている設定を変更すれば  $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}2_{\epsilon}$  をカスタマイズすることができますが、その場合は `pldefs.ltx` を直接修正するのではなく、いったん `pldefs.cfg` という名前でコピーして、そのファイルを編集してください。フォーマット作成時に `pldefs.cfg` が存在した場合は、そちらが `pldefs.ltx` の代わりに読み込まれます。

### 2.1.1 バージョン

$\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}2_{\epsilon}$  のバージョンやフォーマットファイル名は、`plvers.dtx` で定義しています。

### 2.1.2 NFSS2 コマンド

$\text{L}\text{A}\text{T}\text{E}\text{X}2_{\epsilon}$  では、フォント選択機構として NFSS2 を用いています。 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}2_{\epsilon}$  では、オリジナルの NFSS2 と同様のインターフェイスで、和文フォントを選択できるように、`plfonts.dtx` で NFSS2 を拡張しています。

$\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}2_{\epsilon}$  の NFSS2 は、フォントを切替えるコマンドを指定するときに、それが欧文書体か和文書体のいずれかを対象とするものかを、できるだけ意識しないようにする方向で拡張しています。いいかえれば、コマンドが（可能な限りの）判断をします。したがって数多くある英語版のクラスファイルやパッケージファイルなどで書体の変更を行っている箇所を修正する必要はあまりありません。

NFSS2 についての詳細は、 $\text{L}\text{A}\text{T}\text{E}\text{X}2_{\epsilon}$  に付属の `fntguide.tex` を参照してください。

### 2.1.3 出力ルーチンとフロート

`plcore.dtx` は、次の項目に関するコマンドを日本語処理用に修正や拡張をしています。

- プリアンブルコマンド
- 改ページ
- 改行

---

<sup>6</sup>アスキー版では `plcore.ltx` の中でロードしていましたが、2018 年以降の新しいコミュニティ版  $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$  では `platex.ltx` から読み込むことにしました。

- オブジェクトの出力順序
- トンボ
- 脚注マクロ
- 相互参照
- 疑似タイプ入力

## 2.2 クラスファイルとパッケージファイル

p $\LaTeX$  2 $\epsilon$  が提供をするクラスファイルやパッケージファイルは、オリジナルのファイルに基づいています。

p $\LaTeX$  2 $\epsilon$  に付属のクラスファイルは、次のとおりです。

- jarticle.cls, jbook.cls, jreport.cls  
横組用の標準クラスファイル。jclasses.dtx から作成される。
- tarticle.cls, tbook.cls, treport.cls  
縦組用の標準クラスファイル。jclasses.dtx から作成される。
- jltxdoc.cls  
日本語の.dtx ファイルを組版するためのクラスファイル。jltxdoc.dtx から作成される。

また、p $\LaTeX$  2 $\epsilon$  に付属のパッケージファイルは、次のとおりです。

- plect.sty  
縦組用の拡張コマンドなどが定義されているファイル。plext.dtx から作成される。
- ptrace.sty  
 $\LaTeX$  でフォント選択コマンドのトレースに使う tracefnt.sty が再定義してしまう NFSS2 コマンドを、p $\LaTeX$  2 $\epsilon$  用に再々定義するためのパッケージ。plfonts.dtx から作成される。
- pfltrace.sty  
 $\LaTeX$  でフロート関連コマンドのトレースに使う fltrace.sty<sup>7</sup>が再定義してしまうコマンドを、p $\LaTeX$  2 $\epsilon$  用に再々定義するためのパッケージ。plcore.dtx から作成される。

---

<sup>7</sup>p $\LaTeX$  2 $\epsilon$  2014/05/01 で追加されました。参考： $\LaTeX$  2 $\epsilon$  News Issue 21 (ltnews21.tex)

- oldfont.sty

pL<sup>A</sup>T<sub>E</sub>X 2.09 のフォントコマンドを提供するパッケージ。pl209.dtx から作成される。

なお、以前のバージョンに同梱していた ascmac パッケージと nidanfloat パッケージは、別のバンドルとして独立させました。

### 3 他のフォーマット・旧バージョンとの互換性

ここでは、この pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> のバージョンと以前のバージョン、あるいは L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> との互換性について説明をしています。

#### 3.1 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> との互換性

pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> は、L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> の上位互換という形を取っていますが、いくつかの命令の定義やパラメータなども変更しています。したがって英文書など、L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> でも処理できるファイルを pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> で処理しても、完全に同じ結果になるとは限りません。

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 向けに書かれた多くのクラスファイルやパッケージファイルは、そのまま使えると思います。ただし、それらが pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> で拡張しているコマンドと同じ名前のコマンドを再定義している場合は、その拡張の仕方によってはエラーになることもあります。用いようとしているクラスファイルやパッケージファイルがうまく動くかどうかを、完全に確かめる方法は残念ながらありません。一番簡単なのは、動かしてみることです。不幸にもうまく動かない場合は、ログファイルや付属の文書ファイルを参考に原因を調べてください。

なお、いくつかの L<sup>A</sup>T<sub>E</sub>X パッケージについては、pL<sup>A</sup>T<sub>E</sub>X 向けのパッチが用意されています。その一覧は、plautopatch パッケージ (Hironobu Yamashita 作) のドキュメント (日本語版は plautopatch-ja.pdf) に記載されています。

#### 3.2 pL<sup>A</sup>T<sub>E</sub>X 2.09 との互換性

pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> では、文書が使用するクラスを、プリアンブルで `\documentclass` コマンドにより指定します。ここで `\documentclass` の代わりに `\documentstyle` を用いると、pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> は自動的に 2.09 互換モードに入ります。これは L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> が L<sup>A</sup>T<sub>E</sub>X 2.09 互換モードに入るのと同様で、互換モードは古い文書を組版するためだけに作られています。新しく文書を作成する場合は、`\documentclass` コマンドを用いてください。

互換モードでは (p)L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> の新しい機能を利用できず、また古いネイティブな pL<sup>A</sup>T<sub>E</sub>X 2.09 環境と微妙に異なる結果になる可能性もあるという点は、英語版の

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> でも同じです。詳細は、L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> に付属の `usrguide.tex` を参照してください。

### 3.3 latexrelease パッケージへの対応

L<sup>A</sup>T<sub>E</sub>X <2015/01/01>で導入された latexrelease パッケージをもとに、新しい pL<sup>A</sup>T<sub>E</sub>X では platexrelease パッケージを用意しました。platexrelease パッケージを用いると、過去の pL<sup>A</sup>T<sub>E</sub>X をエミュレートしたり、フォーマットを作り直すことなく新しい pL<sup>A</sup>T<sub>E</sub>X を試したりすることができます。詳細は platexrelease のドキュメントを参照してください。

## A DOCSTRIP プログラムのためのオプション

この文書のソース (`platex.dtx`) を DOCSTRIP プログラムで処理することによって、いくつかの異なるファイルを生成することができます。DOCSTRIP プログラムの詳細は、`docstrip.dtx` を参照してください。

この文書の DOCSTRIP プログラムのためのオプションは、次のとおりです。

オプション	意味
<code>plcore</code>	フォーマットファイルを作るためのファイルを生成
<code>pldoc</code>	pL <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> のソースファイルをまとめて組版するための文書ファイル ( <code>pldoc.tex</code> ) を生成
<code>shprog</code>	上記のファイルを作成するための sh スクリプトを生成
<code>plprog</code>	入れ子構造を調べる簡単な perl スクリプトを生成
<code>Xins</code>	上記の sh スクリプトや perl スクリプトを取り出すための DOCSTRIP バッチファイル ( <code>Xins.ins</code> ) を生成

### A.1 ファイルの取り出し方

たとえば、この文書の “`plcore`” の部分を “`platex.ltx`” というファイルにするときの手順はつぎのようになります。

1. `platex docstrip`
2. 入力ファイルの拡張子 (`dtx`) を入力する。
3. 出力ファイルの拡張子 (`ltx`) を入力する。
4. DOCSTRIP オプション (`plcore`) を入力する。

5. 入力ファイル名 (platex) を入力する。
6. platex.ltx が存在する場合は、確認を求めてくるので、“y” を入力する。
7. 別の処理を行なうかを問われるので、“n” を入力する。

これで、platex.ltx が作られます。

あるいは、次のような内容のファイル fmt.ins を作成し、platex fmt.ins することでも platex.ltx を作ることができます。

```
\def\batchfile{fmt.ins}
\input docstrip.tex
\generateFile{platex.ltx}{t}{\from{platex.dtx}{plcore}}
```

## B 文書ファイル

ここでは、このパッケージに含まれている dtx ファイルをまとめて組版し、ソースコード説明書を得るための文書ファイル pldoc.tex について説明をしています。個別に処理した場合と異なり、変更履歴や索引も付きます。全体で、およそ 200 ページ程度になります。

デフォルトではソースコードの説明が日本語で書かれます。もし英語の説明書を読みたい場合は、

```
\newif\ifJAPANESE
```

という内容の platex.cfg を予め用意してから pldoc.tex を処理してください (2016 年 7 月 1 日以降のコミュニティ版 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> が必要)。

filecontents 環境は、引数に指定されたファイルが存在するときは何もみませんが、存在しないときは、環境内の内容でファイルを作成します。pldoc.dic ファイルは、mendex プログラムで索引を処理するときに \西暦、\和暦に対する「読み」を付けるために必要です。

```
32 (*pldoc)
33 \begin{filecontents}{pldoc.dic}
34 西暦    せいれき
35 和暦    われき
36 \end{filecontents}
```

文書クラスには、jltxdoc クラスを用います。plext.dtx の中でサンプルを組み立てていますので、plext パッケージが必要です。

```
37 \documentclass{jltxdoc}
38 \usepackage{plext}
39 \listfiles
40
```

いくつかの TeX プリミティブと plain TeX コマンドを索引に出力しないようにします。

```
41 \DoNotIndex{\def,\long,\edef,\xdef,\gdef,\let,\global}
42 \DoNotIndex{\if,\ifnum,\ifdim,\ifcat,\ifmmode,\ifvmode,\ifhmode,%
43             \iftrue,\iffalse,\ifvoid,\ifx,\ifeof,\ifcase,\else,\or,\fi}
44 \DoNotIndex{\box,\copy,\setbox,\unvbox,\unhbox,\hbox,%
45             \vbox,\vtop,\vcenter}
46 \DoNotIndex{\@empty,\immediate,\write}
47 \DoNotIndex{\egroup,\bgroup,\expandafter,\begingroup,\endgroup}
48 \DoNotIndex{\divide,\advance,\multiply,\count,\dimen}
49 \DoNotIndex{\relax,\space,\string}
50 \DoNotIndex{\csname,\endcsname,\@spaces,\openin,\openout,%
51             \closein,\closeout}
52 \DoNotIndex{\catcode,\endinput}
53 \DoNotIndex{\jobname,\message,\read,\the,\m@ne,\noexpand}
54 \DoNotIndex{\hspace,\vspace,\hskip,\vskip,\kern,\hfil,\hfill,\hss,\vss,\unskip}
55 \DoNotIndex{\m@ne,\z@,\z@skip,\@ne,\tw@,\p@,\@minus,\@plus}
56 \DoNotIndex{\dp,\wd,\ht,\setlength,\addtolength}
57 \DoNotIndex{\newcommand,\renewcommand}
58
```

索引と変更履歴の見出しに `\part` を用いるように設定をします。

```
59 \ifJAPANESE
60 \IndexPrologue{\part*{索引}}%
61             \markboth{索引}{索引}%
62             \addcontentsline{toc}{part}{索引}%
63 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。
64 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。
65 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。}
66 \else
67 \IndexPrologue{\part*{Index}}%
68             \markboth{Index}{Index}%
69             \addcontentsline{toc}{part}{Index}%
70 The italic numbers denote the pages where the corresponding entry
71 is described, numbers underlined point to the definition,
72 all others indicate the places where it is used.}
73 \fi
74 %
75 \ifJAPANESE
76 \GlossaryPrologue{\part*{変更履歴}}%
77             \markboth{変更履歴}{変更履歴}%
78             \addcontentsline{toc}{part}{変更履歴}}
79 \else
80 \GlossaryPrologue{\part*{Change History}}%
81             \markboth{Change History}{Change History}%
82             \addcontentsline{toc}{part}{Change History}}
83 \fi
84
```

標準の \changes コマンドを、複数ファイルの文書に合うように修正しています。

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

コード行では、少しの Overfull を警告無しに許容します。

```
101 \renewcommand*\MacroFont{\fontencoding\encodingdefault
102   \fontfamily\ttdefault
103   \fontseries\mddefault
104   \fontshape\updefault
105   \small
106   \hfuzz 6pt\relax}
```

章番号の桁数が多い場合を考慮し、目次でのスペースを少し増やします。

```
107 \renewcommand*\l@section{\@dottedtocline{2}{1.5em}{2.8em}}
108 \renewcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.4em}}
109 \makeatother
```

変更履歴と 2 段組の索引を作成します。

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

この文書のタイトル・著者・日付を設定します。

```
115 \title{The \pLaTeXe\ Sources}
116 \author{Ken Nakano \& Japanese \TeX\ Development Community}
117
118 % Get the date and patch level from plvers.dtx
119 \makeatletter
120 \let\patchdate=\@empty
121 \begingroup
122   \def\ProvidesFile#1\pfmtversion#2#3\ppatch@level#4{%
123     \date{#2}\xdef\patchdate{#4}\endinput}
124   \input{plvers.dtx}
125 \endgroup
```

```

126
127 % Add the patch version if available.
128 \def\Xpatch{0}
129 \ifx\patchdate\Xpatch\else
130 % number is assumed
131 \ifnum\patchdate>0
132 \edef\@date{\@date\space Patch level\space\patchdate}
133 \else
134 \edef\@date{\@date\space Pre-Release\patchdate}
135 \fi\fi
136
137 % Add the last update info, in case format date unchanged
138 % Note: \@ifl@t@r can be used only in preamble.
139 \def\lastupd@te{0000/00/00}
140 \begingroup
141 \def\ProvidesFile#1[#2 #3]{%
142 \def\@tempd@te{#2}\endinput
143 \@ifl@t@r{\@tempd@te}{\lastupd@te}{%
144 \global\let\lastupd@te\@tempd@te
145 }}
146 \let\ProvidesClass\ProvidesFile
147 \let\ProvidesPackage\ProvidesFile
148 \input{plvers.dtx}
149 \input{plfonts.dtx}
150 \input{plcore.dtx}
151 \input{plext.dtx}
152 \input{pl209.dtx}
153 \input{kinksoku.dtx}
154 \input{jclasses.dtx}
155 \input{jltxdoc.cls}
156 \endgroup
157 \@ifl@t@r{\lastupd@te}{\pfmtversion}{%
158 \edef\@date{\@date\break (last updated: \lastupd@te)}%
159 }{}
160 \makeatother
    ここからが本文ページとなります。
161 \begin{document}
162 \pagenumbering{roman}
163 \maketitle
164 \renewcommand\maketitle{}
165 \tableofcontents
166 \clearpage
167 \pagenumbering{arabic}
168
169 \DocInclude{plvers} % pLaTeX version
170
171 \DocInclude{plfonts} % NFSS2 commands
172
173 \DocInclude{plcore} % kernel commands

```

```

174
175 \DocInclude{plext} % external commands
176
177 \DocInclude{pl209} % 2.09 compatibility mode commands
178
179 \DocInclude{kinsoku} % kinsoku parameter
180
181 \DocInclude{jclasses} % Standard class
182
183 \DocInclude{jltxdoc} % dtx documents class
184

```

ltxdoc.cfg に \AtEndOfClass{\OnlyDescription}が指定されている場合は、ここで終了します。

```

185 \StopEventually{\end{document}}
186

```

変更履歴と索引を組版します。変更履歴ファイルと索引の作り方の詳細については、おまけ C.1 を参照してください。

```

187 \clearpage
188 \pagestyle{headings}
189 % Make TeX shut up.
190 \hbadness=10000
191 \newcount\hbadness
192 \hfuzz=\maxdimen
193 %
194 \PrintChanges
195 \clearpage
196 %
197 \begingroup
198 \def\endash{--}
199 \catcode'\-\active
200 \def-\{\futurelet\temp\indexdash}
201 \def\indexdash{\ifx\temp-\endash\fi}
202
203 \PrintIndex
204 \endgroup

```

ltxdoc.cfg に 2 度目の \PrintIndex が指定されているかもしれません。そこで、最後に、変更履歴や索引が 2 度組版されないように \PrintChanges および \PrintIndex コマンドを何も実行しないようにします。

```

205 \let\PrintChanges\relax
206 \let\PrintIndex\relax
207 \end{document}
208 \pdoc)

```

## C おまけプログラム

### C.1 シェルスクリプト `mkpldoc.sh`

pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> のマクロ定義ファイルをまとめて組版し、変更履歴と索引も付けるときに便利なシェルスクリプトです。このシェルスクリプト<sup>8</sup>の使用方法は次のとおりです。

```
sh mkpldoc.sh
```

#### C.1.1 `mkpldoc.sh` の内容

まず、以前に `pldoc.tex` を処理したときに作成された、目次ファイルや索引ファイルなどを削除します。

```
209 <*shprog>
210 <ja>rm -f pldoc.toc pldoc.idx pldoc.glo
211 <en>rm -f pldoc-en.toc pldoc-en.idx pldoc-en.glo
```

そして、`ltxdoc.cfg` を空にします。このファイルは、`jltxdoc.cls` の定義を変更するものですが、ここでは、変更されたくありません。

```
212 echo "" > ltxdoc.cfg
```

そして、`pldoc.tex` を処理します。

```
213 <ja>platex pldoc.tex
214 <en>platex -jobname=pldoc-en pldoc.tex
```

索引と変更履歴を作成します。このスクリプトでは、変更履歴や索引を生成するのに `mendex` プログラムを用いています。`mendex` は `makeindex` の上位互換のファイル整形コマンドで、索引語の読みを自動的に付けるなどの機能があります。

`-s` オプションは、索引ファイルを整形するためのスタイルオプションです。索引用の `gind.ist` と変更履歴用の `gglo.ist` は、L<sup>A</sup>T<sub>E</sub>X のディストリビューションに付属しています。

`-o` は、出力するファイル名を指定するオプションです。

`-f` は、項目に“読み”がなくてもエラーとしないオプションです。`makeindex` コマンドには、このオプションがありません。

```
215 <ja>mendex -s gind.ist -d pldoc.dic -o pldoc.ind pldoc.idx
216 <en>mendex -s gind.ist -d pldoc.dic -o pldoc-en.ind pldoc-en.idx
217 <ja>mendex -f -s gglo.ist -o pldoc.gls pldoc.glo
218 <en>mendex -f -s gglo.ist -o pldoc-en.gls pldoc-en.glo
```

`ltxdoc.cfg` の内容を `\includeonly{}` にし、`pldoc.tex` を処理します。このコマンドは、引数に指定されたファイルだけを“`\include`”するためのコマンドですが、

---

<sup>8</sup>このシェルスクリプトは UNIX 用です。しかし `rm` コマンドを `delete` コマンドにするなどすれば、簡単に DOS などのバッチファイルに修正することができます。

ここでは何も `\include` したくないので、引数には何も指定をしません。しかし、`\input` で指定されているファイルは読み込まれます。したがって、目次や索引や変更履歴のファイルが処理されます。この処理は、主に、これらでエラーが出るかどうかの確認です。

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

最後に、再び `ltxdoc.cfg` を空にして、`pldoc.tex` を処理をします。本文を 1 ページから開始していますので、この後、もう一度処理をする必要はありません。

```
222 echo "" > ltxdoc.cfg
223 (ja)platex pldoc.tex
224 (en)platex -jobname=pldoc-en pldoc.tex
225 # EOT
226 (/shprog)
```

## C.2 Perl スクリプト `dstcheck.pl`

DOCSTRIP 文書ファイルは、 $\text{\LaTeX}$  のソースとその文書を同時に管理する方法として、とてもすぐれていると思います。しかし、たとえば `jclasses.dtx` のように、条件が多くなると、入れ子構造がわからなくなってしまいがちです。 $\text{\LaTeX}$  で処理すれば、エラーによってわかりますが、文書ファイルが大きくなると面倒です。

ここでは、DOCSTRIP 文書ファイルの入れ子構造を調べるのに便利な、perl スクリプトについて説明をしています。

この perl スクリプトの使用方法は次のとおりです。

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

### C.2.1 `dstcheck.pl` の内容

最初に、この perl スクリプトが何をするのかを簡単に記述したコメントを付けます。

```
227 (*plprog)
228 ##
229 ## DOCSTRIP 文書内の環境や条件の入れ子を調べる perl スクリプト
230 ##
```

このスクリプトは、入れ子の対応を調べるために、次のスタックを用います。〈条件〉あるいは〈環境〉を開始するコードが現れたときに、それらはスタックにプッシュされ、終了するコードでポップされます。したがって、現在の〈条件〉あるいは〈環境〉と、スタックから取り出した〈条件〉あるいは〈環境〉と一致すれば、対応が取れているといえます。そうでなければエラーです。

@dst スタックには、〈条件〉が入ります。条件の開始は、“%<\*(条件)>”です。条件の終了は、“%/<(条件)>”です。〈条件〉には、>文字が含まれません。@env スタックには、〈環境〉が入ります。

先頭を明示的に示すために、ダミーの値を初期値として用います。スタックは、〈条件〉あるいは〈環境〉の名前と、その行番号をペアにして操作をします。

```
231 push(@dst,"DUMMY"); push(@dst,"000");
232 push(@env,"DUMMY"); push(@env,"000");
```

この while ループの中のスクリプトは、文書ファイルの 1 行ごとに実行をします。

```
233 while (<>) {
```

入力行が条件を開始する行なのかを調べます。条件の開始行ならば、@dst スタックに〈条件〉と行番号をプッシュします。

```
234 if (/^%<*\([^>]+\)>/) { # check conditions
235     push(@dst,$1);
236     push(@dst,$.);
```

そうでなければ、条件の終了行なのかを調べます。現在行が条件の終了を示している場合は、@dst スタックをポップします。

```
237 } elsif (/^%<\/\([^>]+\)>/) {
238     $linenum = pop(@dst);
239     $conditions = pop(@dst);
```

現在行の〈条件〉と、スタックから取り出した〈条件〉が一致しない場合、その旨のメッセージを出力します。

なお、DUMMY と一致した場合は、一番外側のループが合っていないことを示しています。このとき、これらのダミー値をスタックに戻します。いつでもスタックの先頭をダミー値にするためです。

```
240     if ($1 ne $conditions) {
241         if ($conditions eq "DUMMY") {
242             print "$ARGV: '</$1>' (l.$.) is not started.\n";
243             push(@dst,"DUMMY");
244             push(@dst,"000");
245         } else {
246             print "$ARGV: '<*$conditions>' (l.$linenum) is ended ";
247             print "by '<*$1>' (l.$.)\n";
248         }
249     }
250 }
```

環境の入れ子も条件と同じように調べます。

verbatim 環境のときに、その内側をスキップしていることに注意をしてください。

```
251 if (/^% *\\begin\{verbatim\}/) { # check environments
252     while(<>) {
253         last if (/^% *\\end\{verbatim\}/);
254     }
```

```

255 } elsif (/^% *\\begin\{([^\}]+\)\}\{(.*)\}/) {
256     push(@env,$1);
257     push(@env,$.);
258 } elsif (/^% *\\begin\{([^\}]+\)\}/) {
259     push(@env,$1);
260     push(@env,$.);
261 } elsif (/^% *\\end\{([^\}]+\)\}/) {
262     $linenum = pop(@env);
263     $environment = pop(@env);
264     if ($1 ne $environment) {
265         if ($environment eq "DUMMY") {
266             print "$ARGV: '\\end{$1}' (l.$.) is not started.\n";
267             push(@env,"DUMMY");
268             push(@env,"000");
269         } else {
270             print "$ARGV: \\begin{$environment} (l.$linenum) is ended ";
271             print "by \\end{$1} (l.$.)\n";
272         }
273     }
274 }

```

ここまでが、最初の while ループです。

```
275 }
```

文書ファイルを読み込んだ後、終了していない条件があるかどうかを確認します。すべての条件の対応がとれていれば、この時点での@dst スタックにはダミー値しか入っていません。したがって、対応が取れている場合は、最初の2つのポップによって、ダミー値が設定されます。ダミー値でなければ、ダミー値になるまで、取り出した値を出力します。

```

276 $linenum = pop(@dst);
277 $conditions = pop(@dst);
278 while ($conditions ne "DUMMY") {
279     print "$ARGV: '<*$conditions>' (l.$linenum) is not ended.\n";
280     $linenum = pop(@dst);
281     $conditions = pop(@dst);
282 }

```

環境の入れ子についても、条件の入れ子と同様に確認をします。

```

283 $linenum = pop(@env);
284 $environment = pop(@env);
285 while ($environment ne "DUMMY") {
286     print "$ARGV: '\\begin{$environment}' (l.$linenum) is not ended.\n";
287     $linenum = pop(@env);
288     $environment = pop(@env);
289 }
290 exit;
291 </plprog>

```

### C.3 DOCSTRIP バッチファイル

ここでは、付録 C.1 と付録 C.2 で説明をした二つのスクリプトを、このファイルから取り出すための DOCSTRIP バッチファイルについて説明をしています。

まず、DOCSTRIP パッケージをロードします。また、実行経過のメッセージを出力しないようにしています。

```
292 <*Xins>
293 \input docstrip
294 \keepsilent
```

DOCSTRIP プログラムは、連続する二つのパーセント記号 (%%) ではじまる行をメタコメントとみなし、条件によらず出力をします。しかし、“%” は  $\text{\TeX}$  ではコメントであっても、sh や perl にとってはコメントではありません。そこで、メタコメントとして出力する文字を “##” と変更します。

```
295 {\catcode'#=12 \gdef\MetaPrefix{## }}
```

そして、プリアンブルに出力されるメッセージを宣言します。ここでは、とくに何も指定していませんが、宣言をしないとデフォルトの記述が ‘%%’ 付きで出力されてしまうため、それを抑制する目的で使用しています。

```
296 \declarepreamble\thispre
297 \endpreamble
298 \usepreamble\thispre
```

ポストアンブルも同様に、宣言をしないと ‘\endinput’ が出力されます。

```
299 \declarepostamble\thispost
300 \endpostamble
301 \usepostamble\thispost
```

`\generate` コマンドで、どのファイルに、どのファイルのどの部分を出力するのかを指定します。

```
302 \generate{
303   \file{dstcheck.pl}{\from{platex.dtx}{plprog}}
304   \file{mkpldoc.sh}{\from{platex.dtx}{shprog,ja}}
305   \file{mkpldoc-en.sh}{\from{platex.dtx}{shprog,en}}
306 }
307 \endbatchfile
308 </Xins>
```

## 参考文献

- [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

## 変更履歴

1995/05/08 v1.0	2016/06/19 v1.0l
・最初のバージョン . . . . . 2	・パッチレベルを <code>plvers.dtx</code> から取得 . . . . . 11
1995/08/25 v1.0a	2016/08/26 v1.0m
・互換性について、DOCSTRIP の使い方、参考文献を追加 . . . . . 2	・ <code>platex.cfg</code> の読み込みを <code>plcore.ltx</code> から <code>platex.ltx</code> へ移動 . . . . . 4
1996/02/01 v1.0b	2016/09/14 v1.0n
・ <code>omake-sh.ins</code> , <code>omake-pl.ins</code> を DOCSTRIP の変更にもなう変更をした . . . . . 18	・ $\LaTeX$ のバナーの保存しかたを改良 3
1997/01/23 v1.0c	2017/09/24 v1.0o
・DOCSTRIP にもなう変更 . . . . . 18	・パッチレベルが負の数の場合を <code>pre-release</code> 扱いへ . . . . . 11
・ <code>gind.ist</code> と <code>gglo.ist</code> を $\$TEXMF/tex/latex2e/base$ ディレクトリからコピーしないようにした . . . . . 14	2017/11/11 v1.0p
1997/01/25 v1.0c	・ $\LaTeX$ のバナーを保存するコードを <code>platex.ltx</code> から <code>plcore.ltx</code> へ移動 . . . . . 3
・ <code>pldoc.dic</code> を <code>filecontents</code> 環境により作成 . . . . . 9	2017/11/29 v1.0q
1997/01/29 v1.0c	・英語版ドキュメントを追加 . . . . . 1
・ <code>pltpatch.ltx</code> を <code>plpatch.ltx</code> に名称変更 . . . . . 11	2017/12/02 v1.0r
2016/01/27 v1.0d	・英語の参考文献も追加 . . . . . 2
・ $\LaTeX 2_{\epsilon}$ に付属するファイルの説明を更新 . . . . . 6	2017/12/05 v1.0s
・ <code>rm</code> コマンド実行前に存在確認するようにした . . . . . 14	・デフォルト設定ファイルの読み込みを <code>plcore.ltx</code> から <code>platex.ltx</code> へ移動 . . . . . 4
2016/02/16 v1.0e	2018/02/07 v1.0t
・ <code>platexrelease</code> の説明を追加 . . . . . 8	・ <code>ascmac</code> パッケージを独立させた . . 7
2016/04/12 v1.0f	2018/02/18 v1.0u
・ドキュメントを更新 . . . . . 1	・ <code>nidanfloat</code> パッケージを独立させた 7
2016/05/07 v1.0g	2018/04/06 v1.0v
・フォーマット作成時に $\LaTeX$ のバナーを一旦保存 . . . . . 3	・最新の <code>source2e</code> への追随 . . . . . 11
2016/05/08 v1.0h	2018/04/08 v1.0w
・ドキュメントから <code>plpatch.ltx</code> を除外 . . . . . 11	・安全のためフォーマット作成時のバナー表示をやめた . . . . . 4
2016/05/12 v1.0i	2018/09/03 v1.0x
・一時コマンド <code>\orgdump</code> を最終的に未定義へ . . . . . 4	・ <code>platexcheat</code> に言及 . . . . . 2
2016/05/20 v1.0j	・ <code>plautopatch</code> に言及 . . . . . 7
・ <code>plftrace</code> の説明を追加 . . . . . 6	・ドキュメントを更新 . . . . . 1
2016/05/21 v1.0k	2018/09/22 v1.0y
・変更履歴も出力するようにした . . 1	・最終更新日を <code>pldoc.pdf</code> に表示 11
	2019/09/29 v1.0z
	・タイポ修正 . . . . . 1