

BXJS ドキュメントクラス

ZR

v0.3a [2012/05/01]

1 はじめに

この文書は「BXJS ドキュメントクラス」の DocStrip 形式のソースである。インストール時のモジュール指定は以下のようなものである。

<code><article></code>	<code>bxjsarticle.cls</code>	短いレポート（章なし）
<code><report></code>	<code>bxjsreport.cls</code>	長いレポート（章あり）
<code><book></code>	<code>bxjsbook.cls</code>	書籍用
<code><slide></code>	<code>bxjsslide.cls</code>	スライド用

本ドキュメントクラスは奥村晴彦氏の「p_{La}T_EX 2_ε 新ドキュメントクラス」に改変を加えたものである。本ドキュメントクラスに関する説明は全てこの形式の枠の中に記す。枠の外にあるものは奥村氏による原版に対する解説である。

これは L_AT_EX 3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいて奥村が改変したものです。権利については両者のものに従います。奥村は何の権利も主張しません。

[2009-02-22] 田中琢爾氏による up_{La}T_EX 対応パッチを取り込みました。

以下では実際のコードに即して説明します。

■BXJS クラス特有の設定 ☆

```
<!*ptex>
\RequirePackage{calc}
\RequirePackage{keyval}
```

エンジン（T_EX の種類）の判定。`\bxjs@engine` の値：`n` = 8 ビット欧文 T_EX、`j` = pT_EX または upT_EX、`J` = NTT jT_EX、`x` = XeT_EX、`p` = pdfT_EX、`l` = LuaT_EX。pT_EX か upT_EX の判別はスイッチ `\if@jsc@uplatex` で表される。

```
\let\bxjs@engine=n
% 念のためプリミティブテストに変えた
```

```

\def\bxjs@test@engine#1#2{%
  \edef\bxjs@tmpa{\string#1}%
  \edef\bxjs@tmpb{\meaning#1}%
  \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
\bxjs@test@engine\kanjiskip{\let\bxjs@engine=j}
\bxjs@test@engine\jintercharskip{\let\bxjs@engine=J}
\bxjs@test@engine\XeTeXversion{\let\bxjs@engine=x}
\bxjs@test@engine\pdftexversion{\let\bxjs@engine=p}
\bxjs@test@engine\luatexversion{\let\bxjs@engine=l}
\newif\if@jsc@uplatex
% 内部 Unicode 動作の時だけ upTeX とみなす
\ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
  \@jsc@uplatextrue
\fi\fi
\let\jsArticle=a
\let\jsBook=b
\let\jsReport=r
\let\jsSlide=s
<article>\let\jsDocClass\jsArticle
<article>\def\bxjs@clsname{bxjsarticle}
<book>\let\jsDocClass\jsBook
<book>\def\bxjs@clsname{bxjsbook}
<report>\let\jsDocClass\jsReport
<report>\def\bxjs@clsname{bxjsreport}
<slide>\let\jsDocClass\jsSlide
<slide>\def\bxjs@clsname{bxjsslide}

```

LuaTeX の場合、原版のコード中のコントロールワード中に現れる日本語文字のカテゴリコードを 11 に変更する。(元は 12。)ただし、読込終了時点で元に戻しているため、日本語文字のコントロールワードのマクロはそのままでは使える状態になっていない。

```

\@onlypreamble\bxjs@restore@jltrcc
\let\bxjs@restore@jltrcc\@empty
\if 1\bxjs@engine
\def\bxjs@change@jltrcc#1{%
  \xdef\bxjs@restore@jltrcc{%
    \bxjs@restore@jltrcc
    \catcode'#1=\the\catcode'#1\relax}%
  \catcode'#1=11\relax}
\@tfor\bxjs@x:=西暦\do
  {\expandafter\bxjs@change@jltrcc\bxjs@x}
\fi

```

\jsInhibitGlue は \inhibitglue が定義されていればそれを実行し、未定義ならば何もしない。

```
\DeclareRobustCommand*\jsInhibitGlue}{%
  \ifx\inhibitglue\@undefined\else \inhibitglue \fi}
```

2 オプション

これらのクラスは `\documentclass{jsarticle}` あるいは `\documentclass[オプション]{jsarticle}` のように呼び出します。

まず、オプションに関連するいくつかのコマンドやスイッチ（論理変数）を定義します。

```
\if@restonecol 段組のときに真になる論理変数です。
  \newif\if@restonecol

\if@titlepage これを真にすると表題，概要を独立したページに出力します。
  \newif\if@titlepage

\if@openright \chapter, \part を奇数ページ起こしにするかどうかです。書籍では真が標準です。
  (book | report) \newif\if@openright

\if@mainmatter 真なら本文，偽なら前付け・後付けです。偽なら \chapter で章番号が出ません。
  (book | report) \newif\if@mainmatter \@mainmattertrue

\if@enablejfam 和文フォントを数式フォントとして登録するかどうかを示すスイッチです。
  \newif\if@enablejfam \@enablejfamtrue
```

以下で各オプションを宣言します。

■用紙サイズ JIS や ISO の A0 判は面積 1 m^2 ，縦横比 $1:\sqrt{2}$ の長方形の辺の長さを mm 単位に切り捨てたものです。これを基準として順に半截しては mm 単位に切り捨てたものが A1, A2, …です。

B 判は JIS と ISO で定義が異なります。JIS では B0 判の面積が 1.5 mm^2 ですが，ISO では B1 判の辺の長さが A0 判と A1 判の辺の長さの幾何平均です。したがって ISO の B0 判は $1000\text{ mm} \times 1414\text{ mm}$ です。このため， \LaTeX 2_ϵ の `b5paper` は $250\text{ mm} \times 176\text{ mm}$ ですが， $\text{p}\text{\LaTeX 2}_\epsilon$ の `b5paper` は $257\text{ mm} \times 182\text{ mm}$ になっています。ここでは $\text{p}\text{\LaTeX 2}_\epsilon$ にならって JIS に従いました。

デフォルトは `a4paper` です。

`b5var` (B5 変形, $182\text{ mm} \times 230\text{ mm}$)，`a4var` (A4 変形, $210\text{ mm} \times 283\text{ mm}$) を追加しました。

BXJS クラスではページレイアウト設定に `geometry` パッケージを用いる。用紙サイズ設定は `geometry` に渡すオプションの指定と扱われる。

```
\def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
\DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
\DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
\DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
```

```

\DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
\DeclareOption{b4paper}{\bxjs@setpaper{{257truemm}{364truemm}}}
\DeclareOption{b5paper}{\bxjs@setpaper{{182truemm}{257truemm}}}
\DeclareOption{b6paper}{\bxjs@setpaper{{128truemm}{182truemm}}}
\DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
\DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
\DeclareOption{b4j}{\bxjs@setpaper{{257truemm}{364truemm}}}
\DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
\DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
\DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
\DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
\DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
\DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

\newif\if@landscape
\@landscapefalse
\DeclareOption{landscape}{\@landscapetrue}

```

■slide オプション slide を新設しました。

```

\newif\if@slide
(!slide)\@slidefalse
(slide)\@slidetrue

```

■サイズオプション 10pt, 11pt, 12pt のほかに, 8pt, 9pt, 14pt, 17pt, 21pt, 25pt, 30pt, 36pt, 43pt を追加しました。これは等比数列になるように選んだものです (従来の 20pt も残しました)。`\@ptsize` の定義が変だったのでご迷惑をおかけしましたが, 標準的なドキュメントクラスと同様にポイント数から 10 を引いたものに直しました。

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

`\@ptsize` は 10pt, 11pt, 12pt が指定された時のみ従来と同じ値とし、それ以外は `\jsUnusualPtSize` (= -20) にする。

```

\newcommand{\@ptsize}{0}
\def\bxjs@param@basefontsize{10pt}
\def\jsUnusualPtSize{-20}
\def\bxjs@setbasefontsize#1{%
  \setlength\@tempdima{#1}%
  \edef\bxjs@param@basefontsize{\the\@tempdima}%
  \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
  \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
  \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
  \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi}
\DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
\DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}

```

```

\DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
\DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
\DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
\DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
\DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
\DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
\DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
\DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
\DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
\DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
\DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
\DeclareOption{12Q}{\bxjs@setbasefontsize{3mm}}
\DeclareOption{14Q}{\bxjs@setbasefontsize{3.5mm}}

```

■ トンボオプション

欧文 L^AT_EX のカーネルではサポートされないため削除。

■ 面付け

欧文 L^AT_EX のカーネルではサポートされないため削除。

- 両面, 片面オプション `twoside` で奇数ページ・偶数ページのレイアウトが変わります。
[2003-04-29] `vartwoside` でどちらのページも傍注が右側になります。

```

\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
\DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

- 二段組 `twocolumn` で二段組になります。

```

\DeclareOption{onecolumn}{\@twocolumnfalse}
\DeclareOption{twocolumn}{\@twocolumntrue}

```

- 表題ページ `titlepage` で表題・概要を独立したページに出力します。

```

\DeclareOption{titlepage}{\@titlepagetrue}
\DeclareOption{notitlepage}{\@titlepagefalse}

```

- 右左起こし 書籍では章は通常は奇数ページ起こしになりますが, `openany` で偶数ページからでも始まるようになります。

```

(book | report) \DeclareOption{openright}{\@openrighttrue}
(book | report) \DeclareOption{openany}{\@openrightfalse}

```

- `eqnarray` 環境と数式の位置 森本さんのご教示にしたがって前に移動しました。

`eqnarray` L^AT_EX の `eqnarray` 環境では `&` でできるアキが大きすぎるようですので, 少し小さくしま

す。また、中央の要素も `\displaystyle` にします。

```
\def\eqnarray{%
  \stepcounter{equation}%
  \def\@currentlabel{\p@equation\theequation}%
  \global\@eqnswtrue
  \m@th
  \global\@eqcnt\z@
  \tabskip\@centering
  \let\@eqncr
  $$\everycr{}\halign to\displaywidth\bgroup
    \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
    &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
    &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
    &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
    \tabskip\z@skip
  \cr}
```

`leqno` で数式番号が左側になります。`fleqn` で数式が本文左端から一定距離のところに出力されます。森本さんにしたがって訂正しました。

```
\DeclareOption{leqno}{\input{leqno.clo}}
\DeclareOption{fleqn}{\input{fleqn.clo}}
% fleqn 用の eqnarray 環境の再定義
\def\eqnarray{%
  \stepcounter{equation}%
  \def\@currentlabel{\p@equation\theequation}%
  \global\@eqnswtrue\m@th
  \global\@eqcnt\z@
  \tabskip\mathindent
  \let\@eqncr
  \setlength\abovedisplayskip{\topsep}%
  \ifvmode
    \addtolength\abovedisplayskip{\partopsep}%
  \fi
  \addtolength\abovedisplayskip{\parskip}%
  \setlength\belowdisplayskip{\abovedisplayskip}%
  \setlength\belowdisplayshortskip{\abovedisplayskip}%
  \setlength\abovedisplayshortskip{\abovedisplayskip}%
  $$\everycr{}\halign to\linewidth% $$
  \bgroup
    \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
    &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
    &\global\@eqcnt\tw@
      $\displaystyle{##}$\hfil \tabskip\@centering
    &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
  \tabskip\z@skip\cr
}
```

■文献リスト 文献リストを open 形式（著者名や書名の後に改行が入る）で出力します。
これは使われることはないのでコメントアウトしてあります。

```
% \DeclareOption{openbib}{%
%   \AtEndOfPackage{%
%     \renewcommand\@openbib@code{%
%       \advance\leftmargin\bibindent
%       \itemindent -\bibindent
%       \listparindent \itemindent
%       \parsep \z@}%
%     \renewcommand\newblock{\par}}}
```

■数式フォントとして和文フォントを登録しないオプション

ここは和文処理コンポーネントの管轄。

■ドラフト draft で overfull box の起きた行末に 5pt の罫線を引きます。

```
\newif\ifdraft
\DeclareOption{draft}{\drafttrue \setlength\overfullrule{5pt}}
\DeclareOption{final}{\draftfalse \setlength\overfullrule{0pt}}
```

■和文フォントメトリックの選択

ここは和文処理コンポーネントの管轄。

■papersize スペシャルの利用

geometry パッケージが行う。

■英語化 オプション english を新設しました。

```
\newif\if@english
\@englishfalse
\DeclareOption{english}{\@englishttrue}
```

■jsreport 相当 オプション report を新設しました。

BXJS では ‘report’ 相当のものは別に bxjsreport クラスとして用意する。

■BXJS 特有のオプション ☆

-
- base=*<dimen>* : 基底フォントサイズを直接指定する。(xxpt オプションの代用な

ので、既定値は 10 pt である。)

- `scale=<real>` : 和文フォントのスケールを表すマクロ `\jsScale` の値を設定する。
もちろんこの値を何らかの方法で和文処理モジュールに渡さないと意味を成さない。
既定値は 0.924715 (= 13 Q/10 pt)。
- `mag=<int>` : `\mag` 値の直接設定。既定は `base` から算出する。
- `paper={<dimen:width>}{<dimen:height>}` : 用紙サイズ設定。用紙サイズオプションの代用で、既定値は `a4paper` 相当。
- `platex, uplatex` : (u)pT_EX 用の連携モジュールを読み込む。
- `CJK` : CJK パッケージ用の連携モジュールを読み込む。

`\bxjs@invscale` は T_EX における「長さのスケール」の逆関数を求めるもの。例えば `\bxjs@invscale\dimX{1.3}` は `\dimX=1.3\dimX` の逆の演算を行う。

```
\mathchardef\bxjs@csta=259
\def\bxjs@invscale#1#2{%
  \bgroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
  \@tempcnta\@tempdima \multiply\@tempcnta\ccclvi
  \divide\@tempcnta\@tempdimb \multiply\@tempcnta\ccclvi
  \@tempcntb\p@ \divide\@tempcntb\@tempdimb
  \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
  \@tempdimb\@tempcnta\@ne
  \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
  \advance\@tempcnta\bxjs@csta \@tempdimc\@tempcnta\@ne
  \@whiledim\@tempdimb<\@tempdimc\do{%
    \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
    \advance\@tempcntb\@ne \divide\@tempcntb\tw@
    \ifdim #2\@tempcntb>\@tempdima
      \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
    \else \@tempdimb=\@tempcntb\@ne \fi}%
  \xdef\bxjs@gtmpa{\the\@tempdimb}%
  \egroup #1=\bxjs@gtmpa\relax}
% platex, uplatex オプション
\newif\ifbxjs@ptexmodule
\DeclareOption{platex}{%
  \@tempswafalse
  \if j\bxjs@engine \if jsc@uplatex\else \@tempswatrue \fi\fi
  \bxjs@opt@platex}
\DeclareOption{uplatex}{%
  \@tempswafalse
  \if j\bxjs@engine \if jsc@uplatex \@tempswatrue \fi\fi
  \bxjs@opt@platex}
\def\bxjs@opt@platex{%
  \if@tempswa
    \bxjs@ptexmoduletrue
  \else
    \ClassError\bxjs@clsname%
```



```

        {Option '\CurrentOption' used on wrong engine}\@ehc
    \fi}
% CJK オプション
\newif\ifbxjs@usecjk
\DeclareOption{CJK}{%
    \@tempswafalse
    \if \if p\bxjs@engine T\else\if n\bxjs@engine T\else F\fi\fi T
        \bxjs@usecjktrue
    \else
        \ClassError\bxjs@clsname%
        {Option '\CurrentOption' used on wrong engine}\@ehc
    \fi}
% zw/nozw オプション
\newif\ifbxjs@usezw \bxjs@usezwtrue
\DeclareOption{nozw}{%
    \bxjs@usezwfalse
}
\DeclareOption{zw}{%
    \bxjs@usezwtrue
}
% keyval 型のオプションの処理。
\def\bxjs@setkey{%
    \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
\def\bxjs@setkey@a{\setkeys{bxjs}}
\define@key{bxjs}{base}{\bxjs@setbasefontsize{#1}}
\newif\ifbxjs@scaleset
\def\jsScale{0.924715}
% scale との順序依存を消すため、jbase の実行を遅延させる
\let\bxjs@do@opt@jbase\relax
\define@key{bxjs}{jbase}{%
    \def\bxjs@do@opt@jbase{\bxjs@opt@jbase{#1}}}
\def\bxjs@opt@jbase#1{%
    \setlength\@tempdima{#1}%
    \bxjs@invscale\@tempdima\jsScale
    \bxjs@setbasefontsize{\@tempdima}}
\define@key{bxjs}{scale}{%
    \bxjs@scalesettrue
    \edef\jsScale{#1}}
\let\bxjs@param@mag\relax
\define@key{bxjs}{mag}{\edef\bxjs@param@mag{#1}}
\define@key{bxjs}{paper}{\edef\bxjs@param@paper{#1}}
\DeclareOption*{\bxjs@setkey}

```

■オプションの実行 デフォルトのオプションを実行し、dvi ファイルの先頭に dvips の papersize special を書き込みます。この special は dvips や最近の dviout が対応しています。multicols や url を \RequirePackage するのはやめました。

```

(article)\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
(report)\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}

```

```

\book\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
\slide\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
\ProcessOptions\relax

```

CJK パッケージ連携時の処理。

```
\ifbxjs@usecjk
```

CJK パッケージは和文スケールの連携ができないので、scale を 1 に固定する。

```

\ifbxjs@scaleset\ifdim\jsScale\p@=\p@\else
  \ClassWarning\bxjs@clsname{'scale' value changed to 1}
\fi\fi
\def\jsScale{1}

```

CJK パッケージが読み込まれているかを本文開始時に検査する。

```

\AtBeginDocument{%
  \@ifpackageloaded{CJK}{-}{%
    \ClassWarning\bxjs@clsname{CJK package is not loaded}%
  }%
}
\fi

```

jbase の処理を実行する。

```
\bxjs@do@opt@jbase
```

後処理

```

\if@slide
  \def\maybeblue{\@ifundefined{ver@color.sty}{-}{\color{blue}}}
\fi
\if@landscape
  \setlength\@tempdima {\paperheight}
  \setlength\paperheight{\paperwidth}
  \setlength\paperwidth {\@tempdima}
\fi

```

8bit 欧文 T_EX の場合は、高位バイトをアクティブ化しておく。(和文を含むマクロ定義を通用させるため。)

```

\if \if p\bxjs@engine T\else\if n\bxjs@engine T\else F\fi\fi T
  \@tempcnta="80 \loop \ifnum\@tempcnta<"100
    \catcode\@tempcnta\active
    \advance\@tempcnta\@ne
  \repeat
\fi

```

■papersize スペシャルの出力

geometry パッケージが行う。

■基準となる行送り

`\n@baseline` 基準となる行送りをポイント単位で表したものです。

```
<slide>\def\n@baseline{13}%  
<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%  
<!slide>\else \def\n@baseline{16}\fi
```

■拡大率の設定 サイズの変更は T_EX のプリミティブ `\mag` を使って行います。9 ポイントについては行送りも若干縮めました。サイズについては全面的に見直しました。

[2008-12-26] 1000 / `\mag` に相当する `\inv@mag` を定義しました。truein を使っていたところを `\inv@mag in` に直しましたので、geometry パッケージと共存できると思います。なお、新ドキュメントクラス側で 10pt 以外にする場合の注意：

- geometry 側でオプション `trueedimen` を指定してください。
- geometry 側でオプション `mag` は使えません。

設定すべき `\mag` 値を (基底サイズ)/(10 pt) × 1000 と算出。BXJS クラスでは、`\mag` を直接指定したい場合は、geometry 側ではなくクラスのオプションで行うものとする。

```
\ifx\bxjs@param@mag\relax  
  \@tempdima=\bxjs@param@basefontsize  
  \advance\@tempdima.001pt \multiply\@tempdima25  
  \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax  
  \edef\bxjs@param@mag{\the\@tempcmta}  
\else  
  \let\c@bxjs@cmta\@tempcmta  
  \setcounter{bxjs@cmta}{\bxjs@param@mag}  
  \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi  
  % 有効な mag 値の範囲は 1--32768  
  \edef\bxjs@param@mag{\the\@tempcmta}  
  \advance\@tempcmta100000  
  \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}  
  \expandafter\bxjs@tmpa\the\@tempcmta\@nil  
  \edef\bxjs@param@basefontsize{\the\@tempdima}  
\fi  
  
\let\jsBaseFontSize\bxjs@param@basefontsize  
%\typeout{\string\jsDocClass: \meaning\jsDocClass}  
%\typeout{\string\bxjs@engine: \meaning\bxjs@engine}  
%\typeout{\string\jsBaseFontSize: \jsBaseFontSize}  
%\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
```

ここで pT_EX の `zw` に相当する単位として用いる長さ変数 `\jsZw` を作成する。約束によ

り、これは $\text{\jsScale} \times (\text{指定フォントサイズ})$ に等しい。
nozw 非指定時は \zw を \jsZw と同義にする。

```
\newdimen\jsZw
\jsZw=10pt \jsZw=\jsScale\jsZw
\ifbxjs@usezw
  \providecommand*\zw{\jsZw}
\fi
```

■pagesize スペシャルの出力

削除。

3 和文フォントの変更

和文フォントの設定は和文処理モジュールの管轄。

\@ 欧文といえば、 \LaTeX の $\text{\def\@{\spacefactor\@m}}$ という定義 (\@m は 1000) では I watch TV\@. と書くと V とピリオドのペアカーニングが効かなくなります。そこで、次のような定義に直し、 I watch TV.\@ と書くことにします。

```
\chardef\bxjs@periodchar='\.
\def\@{\spacefactor\sffcode\bxjs@periodchar}
```

4 フォントサイズ

フォントサイズを変える命令 (\normalsize , \small など) の実際の挙動の設定は、三つの引数をとる命令 \@setfontsize を使って、たとえば

```
\@setfontsize{\normalsize}{10}{16}
```

のようにして行います。これは

\normalsize は 10 ポイントのフォントを使い、行送りは 16 ポイントである

という意味です。ただし、処理を速くするため、以下では 10 と同義の \LaTeX の内部命令 \@xpt を使っています。この \@xpt の類は次のものがあり、 \LaTeX 本体で定義されています。

\@vpt	5	\@vipt	6	\@viipt	7
\@viiipt	8	\@ixpt	9	\@xpt	10
\@xipt	10.95	\@xiipt	12	\@xivpt	14.4

`\@setfontsize` ここでは `\@setfontsize` の定義を少々変更して、段落の字下げ `\parindent`、和文文字間のスペース `\kanjiskip`、和文・欧文間のスペース `\xkanjiskip` を変更しています。

`\kanjiskip` は p \LaTeX 2 ϵ で `0pt plus .4pt minus .5pt` に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

`\xkanjiskip` については、四分つまり全角の 1/4 を標準として、追い出すために三分あるいは二分まで延ばすのが一般的ですが、ここでは Times や Palatino のスペースがほぼ四分であることに着目して、これに一致させています。これなら書くときにスペースを空けても空けなくても同じ出力になります。

`\parindent` については、0 (以下) でなければ全角幅 (1zw) に直します。

[2008-02-18] `english` オプションで `\parindent` を 1em にしました。

`\@setfontsize` の中で `\jsZw` を再設定している。またユーザ定義用のフック `\jsResetDimen` を用意した。

```
\def\@setfontsize#1#2#3{%
% \@nomath#1%
\ifx\protect\@typeset@protect
\let\@currsize#1%
\fi
\fontsize{#2}{#3}\selectfont
\jsZw=\f@size pt\relax
\jsZw=\jsScale \jsZw\relax
\ifdim\parindent>\z@
\if@english \parindent=1em
\else \parindent=1\jsZw
\fi
\fi\relax
\jsResetDimen}
\def\jsResetDimen{}
```

これらのグルーをもってしても行分割ができない場合は、`\emergencystretch` に訴えます。

これはフォントサイズ非依存なので `\Cwd` で書くのが適当だが、`\Cwd` はまだ定義されていない。

```
\emergencystretch 3\jsZw
```

`\ifnarrowbaselines` 欧文用に行間を狭くする論理変数と、それを真・偽にするためのコマンドです。

`\narrowbaselines` [2003-06-30] 数式に入るところで `\narrowbaselines` を実行しているので

`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対し

て、しっぽ愛好家さんが次の修正を教えてくださいました。

[2008-02-18] `english` オプションで最初の段落のインデントをしないようにしました。

```
\newif\ifnarrowbaselines
\if@english
  \narrowbaselinestrue
\fi
\def\narrowbaselines{%
  \narrowbaselinestrue
  \skip0=\abovedisplayskip
  \skip2=\abovedisplayshortskip
  \skip4=\belowdisplayskip
  \skip6=\belowdisplayshortskip
  \@currsize\selectfont
  \abovedisplayskip=\skip0
  \abovedisplayshortskip=\skip2
  \belowdisplayskip=\skip4
  \belowdisplayshortskip=\skip6\relax}
\def\widebaselines{\narrowbaselinesfalse\@currsize\selectfont}
```

`\normalsize` 標準のフォントサイズと行送りを選ぶコマンドです。

本文 10 ポイントのときの行送りは、欧文の標準クラスファイルでは 12 ポイント、アスキーの和文クラスファイルでは 15 ポイントになっていますが、ここでは 16 ポイントにしました。ただし `\narrowbaselines` で欧文用の 12 ポイントになります。

公称 10 ポイントの和文フォントが約 9.25 ポイント（アスキーのものの 0.961 倍）であることもあり、行送りがかなりゆったりとしたと思います。実際、 $16/9.25 \approx 1.73$ であり、和文の推奨値の一つ「二分四分」（1.75）に近づきました。

```
\renewcommand{\normalsize}{%
  \ifnarrowbaselines
    \@setfontsize\normalsize\@xpt\@xipt
  \else
    \@setfontsize\normalsize\@xpt{\n@baseline}%
  \fi}
```

数式の上のアキ (`\abovedisplayskip`), 短い数式の上のアキ (`\abovedisplayshortskip`), 数式の下のアキ (`\belowdisplayshortskip`) の設定です。

[2003-02-16] ちょっと変えました。

```
\abovedisplayskip 10\p@ \@plus3\p@ \@minus4\p@
\abovedisplayshortskip \z@ \@plus3\p@
\belowdisplayskip \abovedisplayskip
\belowdisplayshortskip \belowdisplayskip
```

最後に、リスト環境のトップレベルのパラメータ `\@listI` を、`\@listi` にコピーしておきます。`\@listI` の設定は後で出てきます。

```
\let\@listi\@listI}
```

ここで実際に標準フォントサイズで初期化します。

```
\normalsize
```

`\Cht` 基準となる長さの設定をします。pL^AT_EX 2_ε カーネル (`plfonts.dtx`) で宣言されているパラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (1zw) です。

`\Cwd`

`\Cvs` まず `\Cwd` 等の変数を定義する。

`\Chs`

```

\ifx\Cht\@undefined \newdimen\Cht \fi
\ifx\Cdp\@undefined \newdimen\Cdp \fi
\ifx\Cwd\@undefined \newdimen\Cwd \fi
\ifx\Cvs\@undefined \newdimen\Cvs \fi
\ifx\Chs\@undefined \newdimen\Chs \fi

```

規約上、現在の `\jsZw` の値が `\Cwd` である。`\Cht` と `\Cdp` は単純に `\Cwd` の 88% と 12% の値とする。

```

\setlength\Cht{0.88\jsZw}
\setlength\Cdp{0.12\jsZw}
\setlength\Cwd{1\jsZw}
\setlength\Cvs{\baselineskip}
\setlength\Chs{1\jsZw}

```

`\small` `\small` も `\normalsize` と同様に設定します。行送りは、`\normalsize` が 16 ポイントなら、割合からすれば $16 \times 0.9 = 14.4$ ポイントになりますが、`\small` の使われ方を考えて、ここでは和文 13 ポイント、欧文 11 ポイントとします。また、`\topsep` と `\parsep` は、元はそれぞれ 4 ± 2 , 2 ± 1 ポイントでしたが、ここではゼロ (`\z@`) にしました。

```

\newcommand{\small}{%
  \ifnarrowbaselines
    \!kiyou) \setfontsize\small\@ixpt{11}%
    \!kiyou) \setfontsize\small{8.8888}{11}%
  \else
    \!kiyou) \setfontsize\small\@ixpt{13}%
    \!kiyou) \setfontsize\small{8.8888}{13.2418}%
  \fi
  \abovedisplayskip 9\p@ \@plus3\p@ \@minus4\p@
  \abovedisplayshortskip \z@ \@plus3\p@
  \belowdisplayskip \abovedisplayskip
  \belowdisplayshortskip \belowdisplayskip
  \def\@listi{\leftmargin\leftmargini
    \topsep \z@
    \parsep \z@
    \itemsep \parsep}}

```

`\footnotesize` `\footnotesize` も同様です。`\topsep` と `\parsep` は、元はそれぞれ 3 ± 1 , 2 ± 1 ポイントでしたが、ここではゼロ (`\z@`) にしました。

```

\newcommand{\footnotesize}{%
  \ifnarrowbaselines

```

```

<!kiyou) \setfontsize\footnotesize\@viipt{9.5}%
<kiyou) \setfontsize\footnotesize{8.888}{11}%
\else
<!kiyou) \setfontsize\footnotesize\@viipt{11}%
<kiyou) \setfontsize\footnotesize{8.888}{13.2418}%
\fi
\abovedisplayskip 6\p@ \@plus2\p@ \@minus3\p@
\abovedisplayshortskip \z@ \@plus2\p@
\belowdisplayskip \abovedisplayskip
\belowdisplayshortskip \belowdisplayskip
\def\@listi{\leftmargin\leftmargin
\topsep \z@
\parsep \z@
\itemsep \parsep}}

```

`\scriptsize` それ以外のサイズは、本文に使うことがないので、単にフォントサイズと行送りだけ変更します。特に注意すべきは `\large` で、これは二段組のときに節見出しのフォントとして使
`\large` い、行送りを `\normalsize` と同じにすることによって、節見出しが複数行にわたっても段
`\Large` 間で行が揃うようにします。

```

\scriptsize [2004-11-03] \HUGE を追加。
\huge \newcommand{\scriptsize}{\setfontsize\scriptsize\@viipt\@viipt}
\Huge \newcommand{\tiny}{\setfontsize\tiny\@vpt\@vpt}
\HUGE \if@twocolumn
<!kiyou) \newcommand{\large}{\setfontsize\large\@xipt{\n@baseline}}
<kiyou) \newcommand{\large}{\setfontsize\large{11.111}{\n@baseline}}
\else
<!kiyou) \newcommand{\large}{\setfontsize\large\@xipt{17}}
<kiyou) \newcommand{\large}{\setfontsize\large{11.111}{17}}
\fi
<!kiyou) \newcommand{\Large}{\setfontsize\Large\@xivpt{21}}
<kiyou) \newcommand{\Large}{\setfontsize\Large{12.222}{21}}
\newcommand{\LARGE}{\setfontsize\LARGE\@xviipt{25}}
\newcommand{\huge}{\setfontsize\huge\@xxpt{28}}
\newcommand{\Huge}{\setfontsize\Huge\@xxvpt{33}}
\newcommand{\HUGE}{\setfontsize\HUGE{30}{40}}

```

別行立て数式の中では `\narrowbaselines` にします。和文の行送りのままでは、行列や
場合分けの行送り、連分数の高さなどが不釣合いに大きくなるためです。

本文中の数式の中では `\narrowbaselines` にしていません。本文中ではなるべく行送り
が変わるような大きいものを使わず、行列は `amsmath` の `smallmatrix` 環境を使うのがい
いでしょう。

```
\everydisplay=\expandafter{\the\everydisplay \narrowbaselines}
```

しかし、このおかげで別行数式の上下のスペースが少し違ってしまいました。とりあえず
`amsmath` の `equation` 関係は `okumacro` のほうで逃げていますが、もっとうまい逃げ道が
あれば教えてください。

見出し用のフォントは `\bfseries` 固定ではなく、`\headfont` という命令で定めることにします。これは太ゴシックが使えるときは `\sffamily \bfseries` でいいと思いますが、通常の中ゴシックでは単に `\sffamily` だけのほうがよさそうです。『*LaTeX 2_ε 美文書作成入門*』（1997 年）では `\sffamily \fontseries{sbc}` として新ゴ M と合わせましたが、`\fontseries{sbc}` はちょっと幅が狭いように感じました。

```
% \newcommand{\headfont}{\bfseries}
\newcommand{\headfont}{\sffamily}
% \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}
```

5 レイアウト

■二段組

`\columnsep` `\columnsep` は二段組のときの左右の段間の幅です。元は 10pt ですが、2zw にしました。
`\columnseprule` このスペースの中央に `\columnseprule` の幅の罫線が引かれます。

```
(!kiyou)\setlength\columnsep{2\Cwd}
(kiyou)\setlength\columnsep{28truebp}
\setlength\columnseprule{0\p@}
```

■段落

`\lineskip` 上下の行の文字が `\lineskiplimit` より接近したら、`\lineskip` より近づかないようにします。元は 0pt ですが 1pt に変更しました。normal... の付いた方は保存用です。
`\normallineskip`
`\lineskiplimit` `\setlength\lineskip{1\p@}`
`\normallineskiplimit` `\setlength\normallineskip{1\p@}`
`\setlength\lineskiplimit{1\p@}`
`\setlength\normallineskiplimit{1\p@}`

`\baselinestretch` 実際の行送りが `\baselineskip` の何倍かを表すマクロです。たとえば

```
\renewcommand{\baselinestretch}{2}
```

とすると、行送りが通常の 2 倍になります。ただし、これを設定すると、たとえ `\baselineskip` が伸縮するように設定しても、行送りの伸縮ができなくなります。行送りの伸縮はしないのが一般的です。

```
\renewcommand{\baselinestretch}{}
```

`\parskip` `\parskip` は段落間の追加スペースです。元は 0pt plus 1pt になっていましたが、ここではゼロにしました。`\parindent` は段落の先頭の字下げ幅です。

```
\setlength\parskip{0\p@}
\if@slide
  \setlength\parindent{0\p@}
\else
  \setlength\parindent{1\Cwd}
\fi
```

`\@lowpenalty` `\nopagebreak`, `\nolinebreak` は引数に応じて次のペナルティ値のうちどれかを選ぶよう
`\@medpenalty` になっています。ここはオリジナル通りです。
`\@highpenalty` `\@lowpenalty` 51
 `\@medpenalty` 151
 `\@highpenalty` 301

`\interlinepenalty` 段落中の改ページのペナルティです。デフォルトは 0 です。
 % `\interlinepenalty` 0

`\brokenpenalty` ページの最後の行がハイフンで終わる際のペナルティです。デフォルトは 100 です。
 % `\brokenpenalty` 100

5.1 ページレイアウト

`geometry` パッケージに任せる。

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```

\mag=\bxjs@param@mag
\setlength{\topskip}{10pt}

```

`\bxjs@param@paper` が長さ指定 (`{W}{H}`) の場合、`geometry` の形式 (`papersize={W,H}`) に変換する。

```

\def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
\def\bxjs@read@b{%
  \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
  \else \expandafter\bxjs@read@d \fi}
\def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
\def\bxjs@read@d#1\@nil{}
\expandafter\bxjs@read@a\bxjs@param@paper\@nil

```

`\bxjs@layout@paper` は `geometry` の用紙設定のオプション。

```

\edef\bxjs@layout@paper{%
  truedimen,\if@landscape landscape,\fi \bxjs@param@paper}

```

`\bxjs@layout` は `geometry` のページレイアウトのオプション列であり、文書クラス毎に異なる。

`<{*article | report}>`

```

\def\bxjs@layout{%
  headheight=\topskip,footskip=0.03367\paperheight,%
  headsep=\footskip-\topskip,includeheadfoot,%
  hscale=0.76,,hmarginratio=1:1,%
  vscale=0.83,vmarginratio=1:1,%
}
</article|report>
<*book>
\def\bxjs@layout{%
  headheight=\topskip,headsep=6mm,nofoot,includeheadfoot,%
  hmargin=36mm,hmarginratio=1:1,%
  vscale=0.83%,vmarginratio=1:1,%
}
</book>
<*slide>
\def\bxjs@layout{%
  noheadfoot,%
  hscale=0.9,hmarginratio=1:1,%
  vscale=0.95,vmarginratio=1:1,%%
}
</slide>

```

geometry の後処理。

```

\newdimen\fullwidth
\def\bxjs@postproc@layout{%
% textwidth 調整
  \@tempdimb=\textwidth
  \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
  \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
  \advance\@tempdimb-\textwidth
  \advance\oddsidemargin 0.5\@tempdimb
  \advance\evensidemargin 0.5\@tempdimb
% textheight 調整
  \@tempdimb=\textheight
  \advance\textheight-\topskip
  \divide\textheight\baselineskip \multiply\textheight\baselineskip
  \advance\textheight\topskip
  \advance\@tempdimb-\textheight
  \advance\topmargin0.5\@tempdimb
% headheight 調整
  \@tempdima=\topskip
  \advance\headheight\@tempdima
  \advance\topmargin-\@tempdima
% marginpar 設定
  \setlength\marginparsep{\columnsep}
  \setlength\marginparpush{\baselineskip}
  \setlength\marginparwidth{\evensidemargin-1truein-\marginparsep}
}

```

```

\divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
% 連動する変数
\maxdepth=.5\topskip
\fullwidth=\textwidth
}

```

ここで `geometry` を読み込む。`geometry` は `\topskip` が標準の行高 (`\ht\strutbox`) より小さくならないようにする自動調整を行うが、これをどうするかは未検討。今のところ、単純に回避（無効化）している。

```

\@onlypreamble\bxjs@revert
\let\bxjs@revert\empty
\edef\bxjs@tmpa{\the\ht\strutbox}
\ht\strutbox=10pt
\g@addto@macro\bxjs@revert{\ht\strutbox=\bxjs@tmpa\relax}

```

エンジンが XeTeX である場合の処理。

- `geometry` 4.2 版の場合：TeX Live の一部の版に用意されている設定ファイル `geometry.cfg` では、`geometry` のドライバ自動検出の処理を乗っ取って、強制的に `pdftex` を使用するようになっている。ところが、XeTeX では `\pdfhorigin` など未サポートの pdfTeX プリミティブがあり、`\mag ≠ 1000` の場合にエラーになる。そこで、`geometry.cfg` の読込の箇所にフックをかけて、`geometry.cfg` での処理をさらに乗っ取ることにする。
 - XeTeX が `\pdfpagewidth` をサポートする場合は、ダミーの `\pdfhorigin`、`\pdfvorigin` を用意してドライバを `pdftex` に強制的に変更する。
 - そうでない場合は、ドライバを `dvipdfm` に強制的に変更する。
 - `geometry` 5.x 版の場合：この版の `geometry` は XeTeX に正式に対応（ドライバ名 `xetex`）していて、また XeTeX で実行した場合にドライバ自動検出の結果は `xetex` となる。従って、特に何もする必要がない。さらに、ドライバ自動検出処理のマクロ名を 4.2 版のもの (`\Gm@checkdrivers`) から変えているので、上述の `geometry.cfg` のフックおよびここで行っているフックはともに無力化されている。
-

```

\if x\bxjs@engine
\ifx\pdfpagewidth\@undefined
\def\bxjs@Gm@driver{dvipdfm}
\else
\def\bxjs@Gm@driver{pdftex}
\ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
\ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
\fi
\let\bxjs@InputIfFileExists\InputIfFileExists
\g@addto@macro\bxjs@revert{\let\InputIfFileExists\bxjs@InputIfFileExists}

```

```

\def\InputIfFileExists#1#2#3{%
  \bxjs@InputIfFileExists{#1}{#2}{#3}%
  \edef\Gm@checkdrivers{\noexpand\Gm@setdriver{\bxjs@Gm@driver}}}%
\fi
\edef\bxjs@nxt{%
  \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
\bxjs@nxt \bxjs@revert
\bxjs@postproc@layout

```

- `\setpagelayout{<text>}` : 現在の geometry の設定の一部を修正する。
- `\setpagelayout*{<text>}` : 用紙以外の設定をリセットして、改めて設定を行う。

どちらも設定の後で後処理 `\bxjs@postproc@layout` を実行する。

```

\def\setpagelayout{\@ifstar
  {\bxjs@reset@layout}{\bxjs@modify@layout}}
\def\bxjs@modify@layout#1{%
  \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
  \bxjs@nxt\bxjs@postproc@layout}
\def\bxjs@reset@layout#1{%
  \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
  \bxjs@nxt\bxjs@postproc@layout}

```

ここからのコードは以下の点を除いて JS クラスのものを踏襲する。

- `zw` の代わりに `\jsZw` を用いる。
 - `article/report/book/slide` の切り分けの処理が異なる。
-

■脚注

`\footnotesep` 各脚注の頭に入る支柱 (strut) の高さです。脚注間に余分のアキが入らないように、`\footnotesize` の支柱の高さ (行送りの 0.7 倍) に等しくします。

```

{\footnotesize\global\setlength\footnotesep{\baselineskip}}
\setlength\footnotesep{0.7\footnotesep}

```

`\footins` `\skip\footins` は本文の最終行と最初の脚注との間の距離です。標準の 10 ポイントクラスでは 9 plus 4 minus 2 ポイントになっていますが、和文の行送りを考えてもうちょっと大きくします。

```

\setlength{\skip\footins}{16\p@ \@plus 5\p@ \@minus 2\p@}

```

■フロート関連 フロート (図, 表) 関連のパラメータは $\text{\LaTeX 2}_{\epsilon}$ 本体で定義されていますが、ここで設定変更します。本文ページ (本文とフロートが共存するページ) ちなみに、カウンタは内部では `\c@` を名前に冠したマクロになっています。とフロートだけのページで

設定が異なります。

`\c@topnumber` `topnumber` カウンタは本文ページ上部のフロートの最大数です。
[2003-08-23] ちょっと増やしました。
`\setcounter{topnumber}{9}`

`\topfraction` 本文ページ上部のフロートが占有できる最大の割合です。フロートが入りやすいように、元の値 0.7 を 0.8 [2003-08-23: 0.85] に変えてあります。
`\renewcommand{\topfraction}{.85}`

`\c@bottomnumber` `bottomnumber` カウンタは本文ページ下部のフロートの最大数です。
[2003-08-23] ちょっと増やしました。
`\setcounter{bottomnumber}{9}`

`\bottomfraction` 本文ページ下部のフロートが占有できる最大の割合です。元は 0.3 でした。
`\renewcommand{\bottomfraction}{.8}`

`\c@totalnumber` `totalnumber` カウンタは本文ページに入りうるフロートの最大数です。
[2003-08-23] ちょっと増やしました。
`\setcounter{totalnumber}{20}`

`\textfraction` 本文ページに最低限入らなければならない本文の割合です。フロートが入りやすいように元の 0.2 を 0.1 に変えました。
`\renewcommand{\textfraction}{.1}`

`\floatpagefraction` フロートだけのページでのフロートの最小割合です。これも 0.5 を 0.8 に変えてあります。
`\renewcommand{\floatpagefraction}{.8}`

`\c@dbltopnumber` 二段組のとき本文ページ上部に出力できる段抜きフロートの最大数です。
[2003-08-23] ちょっと増やしました。
`\setcounter{dbltopnumber}{9}`

`\dbltopfraction` 二段組のとき本文ページ上部に出力できる段抜きフロートが占めうる最大の割合です。0.7 を 0.8 に変えてあります。
`\renewcommand{\dbltopfraction}{.8}`

`\dblfloatpagefraction` 二段組のときフロートだけのページに入るべき段抜きフロートの最小割合です。0.5 を 0.8 に変えてあります。
`\renewcommand{\dblfloatpagefraction}{.8}`

`\floatsep` `\floatsep` はページ上部・下部のフロート間の距離です。`\textfloatsep` はページ上部・下部のフロートと本文との距離です。`\intextsep` は本文の途中に出力されるフロートと本文との距離です。

```
\setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}  
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}  
\setlength\intextsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
```

`\dblfloatsep` 二段組のときの段抜きのフロートについての値です。

`\dbltextfloatsep` `\setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}`
`\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}`

`\@fptop` フロートだけのページに入るグルーです。`\@fptop` はページ上部, `\@fpbot` はページ下部,
`\@fpsep` `\@fpsep` はフロート間に入ります。

`\@fpbot` `\setlength\@fptop{0\p@ \@plus 1fil}`
`\setlength\@fpsep{8\p@ \@plus 2fil}`
`\setlength\@fpbot{0\p@ \@plus 1fil}`

`\@dblfpsep` 段抜きフロートについての値です。

`\@dblfpsep` `\setlength\@dblfpsep{0\p@ \@plus 1fil}`
`\@dblfpbot` `\setlength\@dblfpsep{8\p@ \@plus 2fil}`
`\setlength\@dblfpbot{0\p@ \@plus 1fil}`

6 ページスタイル

ページスタイルとして, L^AT_EX 2_ε (欧文版) の標準クラスでは `empty`, `plain`, `headings`, `myheadings` があります。このうち `empty`, `plain` スタイルは L^AT_EX 2_ε 本体で定義されています。

アスキーのクラスファイルでは `headnombre`, `footnombre`, `bothstyle`, `jpl@in` が追加されていますが, ここでは欧文標準のものだけにしました。

ページスタイルは `\ps@...` の形のマクロで定義されています。

`\@evenhead` `\@oddhead`, `\@oddfoot`, `\@evenhead`, `\@evenfoot` は偶数・奇数ページの柱 (ヘッダ, フッタ) を出力する命令です。これらは `\fullwidth` 幅の `\hbox` の中で呼び出されます。

`\@evenfoot` `\ps@...` の中で定義しておきます。

`\@oddfoot` 柱の内容は, `\chapter` が呼び出す `\chaptermark{何々}`, `\section` が呼び出す `\sectionmark{何々}` で設定します。柱を扱う命令には次のものがあります。

`\markboth{左}{右}` 両方の柱を設定します。
`\markright{右}` 右の柱を設定します。
`\leftmark` 左の柱を出力します。
`\rightmark` 右の柱を出力します。

柱を設定する命令は, 右の柱が左の柱の下位にある場合は十分まともに動作します。たとえば左マークを `\chapter`, 右マークを `\section` で変更する場合はこれにあたります。しかし, 同一ページに複数の `\markboth` があると, おかしな結果になることがあります。

`\tableofcontents` のような命令で使われる `\mkboth` は, `\ps@...` コマンド中で `\markboth` か `\gobbletwo` (何もしない) に `\let` されます。

`\ps@empty` `empty` ページスタイルの定義です。L^AT_EX 本体で定義されているものをコメントアウトした形で載せておきます。

```

% \def\ps@empty{%
%   \let\@mkboth\@gobbletwo
%   \let\@oddhead\@empty
%   \let\@oddfoot\@empty
%   \let\@evenhead\@empty
%   \let\@evenfoot\@empty}

\ps@plainhead plainhead はシンプルなヘッダだけのページスタイルです。
\ps@plainfoot plainfoot はシンプルなフッタだけのページスタイルです。
\ps@plain plain は book では plainhead, それ以外では plainfoot になります。

\def\ps@plainfoot{%
  \let\@mkboth\@gobbletwo
  \let\@oddhead\@empty
  \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
  \let\@evenhead\@empty
  \let\@evenfoot\@oddfoot}
\def\ps@plainhead{%
  \let\@mkboth\@gobbletwo
  \let\@oddfoot\@empty
  \let\@evenfoot\@empty
  \def\@evenhead{%
    \if@mparswitch \hss \fi
    \hbox to \fullwidth{\textbf{\thepage}\hfil}%
    \if@mparswitch\else \hss \fi}%
  \def\@oddhead{%
    \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
<book>\let\ps@plain\ps@plainhead
<!book>\let\ps@plain\ps@plainfoot

\ps@headings headings スタイルはヘッダに見出しとページ番号を出力します。ここではヘッダにアンダーラインを引くようにしてみました。
まず article の場合です。
<*article|slide>
\if@twoside
\def\ps@headings{%
  \let\@oddfoot\@empty
  \let\@evenfoot\@empty
  \def\@evenhead{\if@mparswitch \hss \fi
    \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
    \if@mparswitch\else \hss \fi}%
  \def\@oddhead{%
    \underline{%
      \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
  \let\@mkboth\markboth
  \def\sectionmark##1{\markboth{%
    \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
    ##1}{}}%
  \def\subsectionmark##1{\markright{%

```



```

        \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
        ##1}}%
    }
\else % if not twoside
    \def\ps@headings{%
        \let\@oddfoot\@empty
        \def\@oddhead{%
            \underline{%
                \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
        \let\@mkboth\markboth
        \def\sectionmark##1{\markright{%
            \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
            ##1}}}%
\fi
</article | slide>

```

次は book の場合です。

```

<*book | report>
\def\ps@headings{%
    \let\@oddfoot\@empty
    \let\@evenfoot\@empty
    \def\@evenhead{%
        \if@mparswitch \hss \fi
        \underline{\hbox to \fullwidth{\autoxspacing
            \textbf{\thepage}\hfil\leftmark}}}%
        \if@mparswitch\else \hss \fi}%
    \def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
        {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
    \let\@mkboth\markboth
    \def\chaptermark##1{\markboth{%
        \ifnum \c@secnumdepth >\m@ne
            \if@mainmatter
                \@chapapp\thechapter\@chappos\hskip1\jsZw
            \fi
        \fi
        ##1}{}}}%
    \def\sectionmark##1{\markright{%
        \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
        ##1}}}%
</book | report>

```

最後は学会誌の場合です。

```

<*jspf>
\def\ps@headings{%
    \def\@oddfoot{\normalfont\hfil\thepage\hfil}
    \def\@evenfoot{\normalfont\hfil\thepage\hfil}
    \def\@oddhead{\normalfont\hfil \@title \hfil}
    \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
</jspf>

```

`\ps@myheadings` myheadings ページスタイルではユーザが `\markboth` や `\markright` で柱を設定するため、ここでの定義は非常に簡単です。

[2004-01-17] 渡辺徹さんのパッチを適用しました。

```
\def\ps@myheadings{%
  \let\@oddfoot\@empty\let\@evenfoot\@empty
  \def\@evenhead{%
    \if@mparswitch \hss \fi%
    \hbox to \fullwidth{\thepage\hfil\leftmark}%
    \if@mparswitch\else \hss \fi}%
  \def\@oddhead{%
    \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
  \let\@mkboth\@gobbletwo
  (book|report) \let\chaptermark\@gobble
  \let\sectionmark\@gobble
  (!book&!report) \let\subsectionmark\@gobble
}
```

7 文書のマークアップ

7.1 表題

`\title` これらは L^AT_EX 本体で次のように定義されています。ここではコメントアウトした形で示します。

```
\author
\date    % \newcommand*\title{[1]{\gdef\@title{#1}}
          % \newcommand*\author{[1]{\gdef\@author{#1}}
          % \newcommand*\date{[1]{\gdef\@date{#1}}
          % \date{\today}
```

`\etitle` 某学会誌スタイルで使う英語のタイトル、英語の著者名、キーワード、メールアドレスです。

```
\eauthor  <*jspf>
\keywords \newcommand*\etitle{[1]{\gdef\@etitle{#1}}
          \newcommand*\eauthor{[1]{\gdef\@eauthor{#1}}
          \newcommand*\keywords{[1]{\gdef\@keywords{#1}}
          \newcommand*\email{[1]{\gdef\authors@mail{#1}}
          \newcommand*\AuthorsEmail{[1]{\gdef\authors@mail{author's e-mail:\ #1}}
          </jspf>
```

`\plainifnotempty` 従来の標準クラスでは、文書全体のページスタイルを `empty` にしても表題のあるページだけ `plain` になってしまうことがありました。これは `\maketitle` の定義中に `\thispagestyle{plain}` が入っているためです。この問題を解決するために、「全体のページスタイルが `empty` でないならこのページのスタイルを `plain` にする」という次の命令を作ることにします。

```
\def\plainifnotempty{%
  \ifx \@oddhead \@empty
    \ifx \@oddfoot \@empty
      \else
```

```

\thispagestyle{plainfoot}%
\fi
\else
\thispagestyle{plainhead}%
\fi}

```

`\maketitle` 表題を出力します。著者名を出力する部分は、欧文の標準クラスファイルでは `\large`、和文のものでは `\Large` になっていましたが、ここでは `\large` にしました。

```

(*article|book|report|slide)
\if@titlepage
\newcommand{\maketitle}{%
\begin{titlepage}%
\let\footnotesize\small
\let\footnoterule\relax
\let\footnote\thanks
\null\vfil
\if@slide
{\footnotesize \@date}%
\begin{center}
\mbox{} \[\!1\!j\!s\!Z\!w\]
\large
{\maybeblue\hrule height0pt depth2pt\relax}\par
\smallskip
\@title
\smallskip
{\maybeblue\hrule height0pt depth2pt\relax}\par
\vfill
{\small \@author}%
\end{center}
\else
\vskip 60\p@
\begin{center}%
{\LARGE \@title \par}%
\vskip 3em%
{\large
\lineskip .75em
\begin{tabular}[t]{c}%
\@author
\end{tabular}\par}%
\vskip 1.5em
{\large \@date \par}%
\end{center}%
\fi
\par
\@thanks\vfil\null
\end{titlepage}%
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax

```

```

\global\let\thanks\@empty
\global\let\author\@empty
\global\let\date\@empty
\global\let\title\@empty
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
}%
\else
\newcommand{\maketitle}{\par
\begingroup
\renewcommand\thefootnote{\@fnsymbol\c@footnote}%
\def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
\long\def\@makefntext##1{\advance\leftskip 3\jsZw
\parindent 1\jsZw\noindent
\llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
\if@twocolumn
\ifnum \col@number=\@ne
\@maketitle
\else
\twocolumn[\@maketitle]%
\fi
\else
\newpage
\global\@topnum\z@ % Prevents figures from going at top of page.
\@maketitle
\fi
\plainifnotempty
\thanks
\endgroup
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax
\global\let\thanks\@empty
\global\let\author\@empty
\global\let\date\@empty
\global\let\title\@empty
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
}

```

`\@maketitle` 独立した表題ページを作らない場合の表題の出力形式です。

```

\def\@maketitle{%
\newpage\null
\vskip 2em
\begin{center}%

```

```

\let\footnote\thanks
{\LARGE \@title \par}%
\vskip 1.5em
{\large
\lineskip .5em
\begin{tabular}[t]{c}%
\@author
\end{tabular}\par}%
\vskip 1em
{\large \@date}%
\end{center}%
\par\vskip 1.5em
<article | slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
}
\fi
</article | book | report | slide>
<*jspf>
\newcommand{\maketitle}{\par
\begingroup
\renewcommand\thefootnote{\@fnsymbol\c@footnote}%
\def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
\long\def\@makefntext##1{\advance\leftskip 3\jsZw
\parindent 1\jsZw\noindent
\llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
\twocolumn[\@maketitle]%
\plainifnotempty
\@thanks
\endgroup
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax
\global\let\@thanks\@empty
\global\let\@author\@empty
\global\let\@date\@empty
% \global\let\@title\@empty % \@title は柱に使う
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
\ifx\authors@mail\@undefined\else%
\def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
\footnotetext[0]{\itshape\authors@mail}%
}\fi
\global\let\authors@mail\@undefined}
\def\@maketitle{%
\newpage\null
\vskip 6em % used to be 2em
\begin{center}
\let\footnote\thanks

```

```

\ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
\lineskip .5em
\ifx\@author\@undefined\else
  \vskip 1em
  \begin{tabular}[t]{c}%
    \@author
  \end{tabular}\par
\fi
\ifx\@etitle\@undefined\else
  \vskip 1em
  {\large \@etitle \par}%
\fi
\ifx\@eauthor\@undefined\else
  \vskip 1em
  \begin{tabular}[t]{c}%
    \@eauthor
  \end{tabular}\par
\fi
\vskip 1em
\@date
\end{center}
\vskip 1.5em
\centerline{\box\@abstractbox}
\ifx\@keywords\@undefined\else
  \vskip 1.5em
  \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small \@keywords}}
\fi
\vskip 1.5em}
</jspf>

```

7.2 章・節

■構成要素 \@startsection マクロは 6 個の必須引数と、オプションとして * と 1 個のオプション引数と 1 個の必須引数をとります。

```

\@startsection{名}{レベル}{字下げ}{前アキ}{後アキ}{スタイル}
               *[別見出し]{見出し}

```

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

レベル 見出しの深さを示す数値です (chapter=1, section=2, ...)。この数値が secnumdepth 以下のとき見出し番号を出力します。

字下げ 見出しの字下げ量です。

前アキ この値の絶対値が見出し上側の空きです。負の場合は、見出し直後の段落をインデントしません。

後アキ 正の場合は、見出しの下空きです。負の場合は、絶対値が見出しの右の空きです
(見出しと同じ行から本文を始めます)。

スタイル 見出しの文字スタイルの設定です。

* この * 印がないと、見出し番号を付け、見出し番号のカウンタに 1 を加算します。

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

見出しの命令は通常 `\@startsection` とその最初の 6 個の引数として定義されます。

次は `\@startsection` の定義です。情報処理学会論文誌スタイルファイル (`ipsjcommon.sty`) を参考にさせていただきましたが、完全に行送りが `\baselineskip` の整数倍にならなくてもいいから前の行と重ならないようにしました。

```
\def\@startsection#1#2#3#4#5#6{%
  \if@noskipsec \leavevmode \fi
  \par
  % 見出し上の空きを \@tempskipa にセットする
  \@tempskipa #4\relax
  % \@afterindent は見出し直後の段落を字下げするかどうかを表すスイッチ
  \if@english \@afterindentfalse \else \@afterindenttrue \fi
  % 見出し上の空きが負なら見出し直後の段落を字下げしない
  \ifdim \@tempskipa <\z@
    \@tempskipa -\@tempskipa \@afterindentfalse
  \fi
  \if@nobreak
    % \everypar{\everyparhook}% これは間違い
    \everypar{}%
  \else
    \addpenalty\@secpenalty
  % 次の行は削除
  % \addvspace\@tempskipa
  % 次の \noindent まで追加
  \ifdim \@tempskipa >\z@
    \if@slide\else
      \null
      \vspace*{-\baselineskip}%
    \fi
    \vskip\@tempskipa
  \fi
  \fi
  \noindent
  % 追加終わり
  \@ifstar
    {\@ssect{#3}{#4}{#5}{#6}}%
    {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%
  \@sect と \@xsect は、前のアキがちょうどゼロの場合にもうまくいくように、多少変えてあります。
  \everyparhook も挿入しています。
  \def\@sect#1#2#3#4#5#6[#7]#8{%
```

```

\ifnum #2>\c@secnumdepth
  \let\@svsec\@empty
\else
  \refstepcounter{#1}%
  \protected@edef\@svsec{\@secntformat{#1}\relax}%
\fi
% 見出し後の空きを \@tempskipa にセット
\@tempskipa #5\relax
% 条件判断の順序を入れ替えました
\ifdim \@tempskipa<\z@
  \def\@svsechd{%
    #6{\hskip #3\relax
      \@svsec #8}%
    \csname #1mark\endcsname{#7}%
    \addcontentsline{toc}{#1}{%
      \ifnum #2>\c@secnumdepth \else
        \protect\numberline{\csname the#1\endcsname}%
      \fi
      #7}}% 目次にフルネームを載せるなら #8
\else
  \begingroup
    \interlinepenalty \@M % 下から移動
    #6{%
      \@hangfrom{\hskip #3\relax\@svsec}%
%      \interlinepenalty \@M % 上に移動
      #8\@@par}%
  \endgroup
  \csname #1mark\endcsname{#7}%
  \addcontentsline{toc}{#1}{%
    \ifnum #2>\c@secnumdepth \else
      \protect\numberline{\csname the#1\endcsname}%
    \fi
    #7}% 目次にフルネームを載せるならここは #8
\fi
\@xsect{#5}}

```

二つ挿入した `\everyparhook` のうち後者が `\paragraph` 類の後で 2 回実行され、それ以降は前者が実行されます。

```

\def\@xsect#1{%
% 見出しの後ろの空きを \@tempskipa にセット
\@tempskipa #1\relax
% 条件判断の順序を変えました
\ifdim \@tempskipa<\z@
  \nobeckfalse
  \global\@noskipsectrue
  \everypar{%
    \if@noskipsec
      \global\@noskipsecfalse
      {\setbox\z@\lastbox}%
    }
  }

```



```

\clubpenalty\@M
\begingroup \@svsechd \endgroup
\unskip
\@tempskipa #1\relax
\hskip -\@tempskipa
\else
\clubpenalty \@clubpenalty
\everypar{\everyparhook}%
\fi\everyparhook}%
\else
\par \nobreak
\vskip \@tempskipa
\@afterheading
\fi
\if@slide
{\vskip-6pt\maybeblue\hrule height0pt depth1pt\vskip7pt\relax}%
\fi
\par % 2000-12-18
\ignorespaces}
\def\@ssect#1#2#3#4#5{%
\@tempskipa #3\relax
\ifdim \@tempskipa<\z@
\def\@svsechd{#4{\hskip #1\relax #5}}%
\else
\begingroup
#4{%
\hangfrom{\hskip #1}%
\interlinepenalty \@M #5\@@par}%
\endgroup
\fi
\@xsect{#3}}

```

■柱関係の命令

`\chaptermark` `\...mark` の形の命令を初期化します (第 6 節参照)。`\chaptermark` 以外は L^AT_EX 本体で定義済みです。

```

\subsectionmark \newcommand*\chaptermark[1]{}
\subsubsectionmark % \newcommand*\sectionmark[1]{}
\paragrammark % \newcommand*\subsectionmark[1]{}
\subparagrammark % \newcommand*\subsubsectionmark[1]{}
% \newcommand*\paragrammark[1]{}
% \newcommand*\subparagrammark[1]{}

```

■カウンタの定義

`\c@secnumdepth` `secnumdepth` は第何レベルの見出しまで番号を付けるかを定めるカウンタです。

```

(!book&!report)\setcounter{secnumdepth}{3}
(book | report)\setcounter{secnumdepth}{2}

```

<code>\c@chapter</code>	見出し番号のカウンタです。 <code>\newcounter</code> の第 1 引数が新たに作るカウンタです。これは
<code>\c@section</code>	第 2 引数が増加するたびに 0 に戻されます。第 2 引数は定義済みのカウンタです。
<code>\c@subsection</code>	<code>\newcounter{part}</code>
<code>\c@subsubsection</code>	<code>\book report \newcounter{chapter}</code>
<code>\c@paragraph</code>	<code>\book report \newcounter{section}[chapter]</code>
<code>\c@subparagraph</code>	<code>\!book&\!report \newcounter{section}</code>
	<code>\newcounter{subsection}[section]</code>
	<code>\newcounter{subsubsection}[subsection]</code>
	<code>\newcounter{paragraph}[subsubsection]</code>
	<code>\newcounter{subparagraph}[paragraph]</code>
<code>\thepart</code>	カウンタの値を出力する命令 <code>\the</code> 何々 を定義します。
<code>\thechapter</code>	カウンタを出力するコマンドには次のものがあります。
<code>\thesection</code>	<code>\arabic{COUNTER}</code> 1, 2, 3, ...
<code>\thesubsection</code>	<code>\roman{COUNTER}</code> i, ii, iii, ...
<code>\thesubsubsection</code>	<code>\Roman{COUNTER}</code> I, II, III, ...
<code>\theparagraph</code>	<code>\alph{COUNTER}</code> a, b, c, ...
<code>\thesubparagraph</code>	<code>\Alph{COUNTER}</code> A, B, C, ...
	<code>\kansuji{COUNTER}</code> 一, 二, 三, ...

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

\renewcommand{\thepart}{\@Roman\c@part}
\!book&\!report % \renewcommand{\thesection}{\@arabic\c@section}
\!book&\!report \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
\!book&\!report \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
(*book | report)
\renewcommand{\thechapter}{\@arabic\c@chapter}
\renewcommand{\thesection}{\thechapter.\@arabic\c@section}
\renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
(/book | report)
\renewcommand{\thesubsubsection}{%
  \thesubsection.\@arabic\c@subsubsection}
\renewcommand{\theparagraph}{%
  \thesubsubsection.\@arabic\c@paragraph}
\renewcommand{\thesubparagraph}{%
  \theparagraph.\@arabic\c@subparagraph}

\@chapapp \@chapapp の初期値は \prechaptername (第) です。
\@chappos \@chappos の初期値は \postchaptername (章) です。
\appendix は \@chapapp を \appendixname に, \@chappos を空に再定義します。
[2003-03-02] \@secapp は外しました。
(book | report) \newcommand{\@chapapp}{\prechaptername}
(book | report) \newcommand{\@chappos}{\postchaptername}

```

■前付, 本文, 後付 本のうち章番号があるのが「本文」、それ以外が「前付」「後付」です。

`\frontmatter` ページ番号をローマ数字にし、章番号を付けないようにします。

```
<(*book|report)
\newcommand\frontmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse
  \pagenumbering{roman}}
```

`\mainmatter` ページ番号を算用数字にし、章番号を付けるようにします。

```
\newcommand\mainmatter{%
% \if@openright
%   \cleardoublepage
% \else
%   \clearpage
% \fi
  \@mainmattertrue
  \pagenumbering{arabic}}
```

`\backmatter` 章番号を付けないようにします。ページ番号の付け方は変わりません。

```
\newcommand\backmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse}
</book|report)
```

■部

`\part` 新しい部を始めます。

`\secdef` を使って見出しを定義しています。このマクロは二つの引数をとります。

```
\secdef{星なし}{星あり}
```

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

```
\def\chapter { ... \secdef \CMDA \CMDB }
\def\CMDA    [#1]#2{...} % \chapter[...]{...} の定義
\def\CMDB    #1{...}    % \chapter*{...} の定義
```

まず book クラス以外です。

```
<(*!book&!report)
```

```

\newcommand\part{%
  \if@noskipsec \leavevmode \fi
  \par
  \addvspace{4ex}%
  \if@english \@afterindentfalse \else \@afterindenttrue \fi
  \secdef\@part\@spart}
\end{book}&!report)

```

book スタイルの場合は、少し複雑です。

```

\begin{book} \report
\newcommand\part{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \thispagestyle{empty}% 欧文用標準スタイルでは plain
  \if@twocolumn
    \onecolumn
    \@restonecoltrue
  \else
    \@restonecolfalse
  \fi
  \null\vfil
  \secdef\@part\@spart}
\end{book} \report)

```

\@part 部の見出しを出力します。 \bfseries を \headfont に変えました。

book クラス以外では secnumdepth が -1 より大きいとき部番号を付けます。

```

\end{book}&!report)
\def\@part[#1]#2{%
  \ifnum \c@secnumdepth >\m@ne
    \refstepcounter{part}%
    \addcontentsline{toc}{part}{%
      \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
  \else
    \addcontentsline{toc}{part}{#1}%
  \fi
  \markboth{}{}%
  f\parindent\z@
  \raggedright
  \interlinepenalty \@M
  \normalfont
  \ifnum \c@secnumdepth >\m@ne
    \Large\headfont\prepartname\thepart\postpartname
    \par\nobreak
  \fi
  \huge \headfont #2%
  \markboth{}{}\par}%

```

```

\ nobreak
\ vskip 3ex
\ @afterheading}
</!book&!report>

```

book クラスでは secnumdepth が -2 より大きいとき部番号を付けます。

```

<*book | report>
\ def\ @part[#1]#2{%
  \ ifnum \ c@secnumdepth >-2\ relax
    \ refstepcounter{part}%
    \ addcontentsline{toc}{part}{%
      \ prepartname\ thepart\ postpartname\ hspace{1\ jsZw}#1}%
    \ else
      \ addcontentsline{toc}{part}{#1}%
    \ fi
    \ markboth{}{}%
    { \ centering
      \ interlinepenalty \ @M
      \ normalfont
      \ ifnum \ c@secnumdepth >-2\ relax
        \ huge\ headfont \ prepartname\ thepart\ postpartname
        \ par\ vskip20\ p@
      \ fi
      \ Huge \ headfont #2\ par}%
    \ @endpart}
</book | report>

```

\@spart 番号を付けない部です。

```

<!*book&!report>
\ def\ @spart#1{%
  \ parindent \ z@ \ raggedright
  \ interlinepenalty \ @M
  \ normalfont
  \ huge \ headfont #1\ par}%
\ nobreak
\ vskip 3ex
\ @afterheading}
</!book&!report>
<*book | report>
\ def\ @spart#1{%
  \ centering
  \ interlinepenalty \ @M
  \ normalfont
  \ Huge \ headfont #1\ par}%
\ @endpart}
</book | report>

```

\@endpart \@part と \@spart の最後で実行されるマクロです。両面印刷のときは白ページを追加します。二段組のときには、二段組に戻します。

```

(*book | report)
\def\@endpart{\vfil\newpage
  \if@twoside
    \null
    \thispagestyle{empty}%
  \newpage
\fi
\if@restonecol
  \twocolumn
\fi}
(/book | report)

```

■ 章

`\chapter` 章の最初のページスタイルは、全体が `empty` でなければ `plain` にします。また、`\@topnum` を 0 にして、章見出しの上に図や表が来ないようにします。

```

(*book | report)
\newcommand{\chapter}{%
  \if@openright\cleardoublepage\else\clearpage\fi
  \plainifnotempty % 元: \thispagestyle{plain}
  \global\@topnum\z@
  \if@english \@afterindentfalse \else \@afterindenttrue \fi
  \secdef\@chapter\@schapter}

```

`\@chapter` 章見出しを出力します。`secnumdepth` が 0 以上かつ `\@mainmatter` が真のとき章番号を出力します。

```

\def\@chapter[#1]#2{%
  \ifnum \c@secnumdepth > \m@ne
    \if@mainmatter
      \refstepcounter{chapter}%
      \typeout{\@chapapp\thechapter\@chappos}%
      \addcontentsline{toc}{chapter}%
        {\protect\numberline
          {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}}%
      #1}%
    \else\addcontentsline{toc}{chapter}{#1}\fi
  \else
    \addcontentsline{toc}{chapter}{#1}%
  \fi
  \chaptermark{#1}%
  \addtocontents{lof}{\protect\addvspace{10\p@}}%
  \addtocontents{lot}{\protect\addvspace{10\p@}}%
  \if@twocolumn
    \@topnewpage[\@makechapterhead{#2}]%
  \else
    \@makechapterhead{#2}%
    \@afterheading
  \fi}

```

`\@makechapterhead` 実際に章見出しを組み立てます。`\bfseries` を `\headfont` に変えました。

```
\def\@makechapterhead#1{%
  \vspace*{2\Cvs}% 欧文は 50pt
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \huge\headfont \@chapapp\thechapter\@chappos
        \par\nobreak
        \vskip \Cvs % 欧文は 20pt
      \fi
    \fi
    \interlinepenalty\@M
    \Huge \headfont #1\par\nobreak
    \vskip 3\Cvs}} % 欧文は 40pt
```

`\@schapter` `\chapter*{...}` コマンドの本体です。`\chaptermark` を補いました。

```
\def\@schapter#1{%
  \chaptermark{#1}%
  \if@twocolumn
    \@topnewpage[\@makeschapterhead{#1}]%
  \else
    \@makeschapterhead{#1}\@afterheading
  \fi}
```

`\@makeschapterhead` 番号なしの章見出しです。

```
\def\@makeschapterhead#1{%
  \vspace*{2\Cvs}% 欧文は 50pt
  {\parindent \z@ \raggedright
    \normalfont
    \interlinepenalty\@M
    \Huge \headfont #1\par\nobreak
    \vskip 3\Cvs}} % 欧文は 40pt
</book | report>
```

■下位レベルの見出し

`\section` 欧文版では `\@startsection` の第 4 引数を負にして最初の段落の字下げを禁止していますが、和文版では正にして字下げするようにしています。

段組のときはなるべく左右の段が狂わないように工夫しています。

```
\if@twocolumn
  \newcommand{\section}{%
    \jpf\ifx\maketitle\relax\else\maketitle\fi
    \@startsection{section}{1}{\z@}%
    \!kiyou {0.6\Cvs}{0.4\Cvs}%
    \kiyou { \Cvs}{0.5\Cvs}%
    % { \normalfont\large\headfont\@secapp}%
    { \normalfont\large\headfont\raggedright}}
\else
```

```

\newcommand{\section}{%
  \if@slide\clearpage\fi
  \@startsection{section}{1}{\z@}%
  {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
  {.5\Cvs \@plus.3\Cdp}% 後アキ
%   {\normalfont\Large\headfont\@secapp}%
   {\normalfont\Large\headfont\raggedright}}
\fi

```

\subsection 同上です。

```

\if@twocolumn
  \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
    {\z@}{\z@}%
    {\normalfont\normalsize\headfont}}
\else
  \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
    {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
    {.5\Cvs \@plus.3\Cdp}% 後アキ
    {\normalfont\large\headfont}}
\fi

```

\subsubsection

```

\if@twocolumn
  \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
    {\z@}{\z@}%
    {\normalfont\normalsize\headfont}}
\else
  \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
    {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
    {\z@}%
    {\normalfont\normalsize\headfont}}
\fi

```

\paragraph 見出しの後ろで改行されません。

```

\if@twocolumn
  \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
    {\z@}{-1\jsZw}% 改行せず 1\jsZw のアキ
    \jpf {\normalfont\normalsize\headfont}}
    \!jpf {\normalfont\normalsize\headfont ■}}
\else
  \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
    {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
    {-1\jsZw}% 改行せず 1\jsZw のアキ
    \jpf {\normalfont\normalsize\headfont}}
    \!jpf {\normalfont\normalsize\headfont ■}}
\fi

```

\subparagraph 見出しの後ろで改行されません。

```

\newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%

```


7.3	リスト環境
	第 k レベルのリストの初期化をするのが <code>\@listk</code> です ($k = \text{i, ii, iii, iv}$)。 <code>\@listk</code> は <code>\leftmargin</code> を <code>\leftmargink</code> に設定します。
<code>\leftmargini</code>	二段組であるかないかに応じてそれぞれ 2em, 2.5em でしたが、ここでは全角幅の 2 倍にしました。 [2002-05-11] 3zw に変更しました。 [2005-03-19] 二段組は 2zw に戻しました。
	<pre> \if@slide \setlength\leftmargini{1\jsZw} \else \if@twocolumn \setlength\leftmargini{2\jsZw} \else \setlength\leftmargini{3\jsZw} \fi \fi </pre>
<code>\leftmarginii</code>	ii, iii, iv は <code>\labelsep</code> とそれぞれ ‘(m)’, ‘vii.’, ‘M.’ の幅との和より大きくすること
<code>\leftmarginiii</code>	になっています。ここでは全角幅の整数倍に丸めました。
<code>\leftmarginiv</code>	
<code>\leftmarginv</code>	
<code>\leftmarginvi</code>	
	<pre> \if@slide \setlength\leftmarginii {1\jsZw} \setlength\leftmarginiii{1\jsZw} \setlength\leftmarginiv {1\jsZw} \setlength\leftmarginv {1\jsZw} \setlength\leftmarginvi {1\jsZw} \else \setlength\leftmarginii {2\jsZw} \setlength\leftmarginiii{2\jsZw} \setlength\leftmarginiv {2\jsZw} \setlength\leftmarginv {1\jsZw} \setlength\leftmarginvi {1\jsZw} \fi </pre>
<code>\labelsep</code>	<code>\labelsep</code> はラベルと本文の間の距離です。 <code>\labelwidth</code> はラベルの幅です。これは二分
<code>\labelwidth</code>	に変えました。
	<pre> \setlength \labelsep {0.5\jsZw} % .5em \setlength \labelwidth{\leftmargini} \addtolength\labelwidth{-\labelsep} </pre>
<code>\partopsep</code>	リスト環境の前に空行がある場合、 <code>\parskip</code> と <code>\topsep</code> に <code>\partopsep</code> を加えた値だけ縦方向の空白ができます。0 に改変しました。
	<pre> \setlength\partopsep{\z@} % {2\p@ \@plus 1\p@ \@minus 1\p@} </pre>

`\@beginparpenalty` リストや段落環境の前後，リスト項目間に挿入されるペナルティです。

```
\@endparpenalty    \@beginparpenalty -\@lowpenalty
\@itempenalty       \@endparpenalty    -\@lowpenalty
                   \@itempenalty       -\@lowpenalty
```

`\@listi` `\@listi` は `\leftmargin`, `\parsep`, `\topsep`, `\itemsep` などのトップレベルの定義をします。この定義は，フォントサイズコマンドによって変更されます（たとえば `\small` の中では小さい値に設定されます）。このため，`\normalsize` がすべてのパラメータを戻せるように，`\@listI` で `\@listi` のコピーを保存します。元の値はかなり複雑ですが，ここでは簡素化してしまいました。特に最初と最後に行送りの半分の空きが入るようにしてあります。アスキーの標準スタイルではトップレベルの `itemize`, `enumerate` 環境でだけ最初と最後に行送りの半分の空きが入るようになっていました。

[2004-09-27] `\topsep` のグルー $+0.2_{-0.1}$ `\baselineskip` を思い切って外しました。

```
\def\@listi{\leftmargin\leftmargini
\parsep \z@
\topsep 0.5\baselineskip
\itemsep \z@ \relax}
\let\@listI\@listi
```

念のためパラメータを初期化します（実際には不要のようです）。

```
\@listi
```

`\@listii` 第 2～6 レベルのリスト環境のパラメータの設定です。

```
\@listiii \def\@listii{\leftmargin\leftmarginii
\labelwidth\leftmarginii \advance\labelwidth-\labelsep
\@listiv \topsep \z@
\@listv \parsep \z@
\@listvi \itemsep\parsep}
\def\@listiii{\leftmargin\leftmarginiii
\labelwidth\leftmarginiii \advance\labelwidth-\labelsep
\topsep \z@
\parsep \z@
\itemsep\parsep}
\def\@listiv {\leftmargin\leftmarginiv
\labelwidth\leftmarginiv
\advance\labelwidth-\labelsep}
\def\@listv {\leftmargin\leftmarginv
\labelwidth\leftmarginv
\advance\labelwidth-\labelsep}
\def\@listvi {\leftmargin\leftmarginvi
\labelwidth\leftmarginvi
\advance\labelwidth-\labelsep}
```

■**enumerate 環境** `enumerate` 環境はカウンタ `enumi`, `enumii`, `enumiii`, `enumiv` を使います。`enumn` は第 n レベルの番号です。

```
\theenumi 出力する番号の書式を設定します。これらは LATEX 本体 (ltlists.dtx 参照) で定義済み
\theenumii
\theenumiii
\theenumiv
```

ですが、ここでは表し方を変えています。`\@arabic`, `\@alph`, `\@roman`, `\@Alph` はそれぞれ算用数字, 小文字アルファベット, 小文字ローマ数字, 大文字アルファベットで番号を出力する命令です。

```
\renewcommand{\theenumi}{\@arabic\c@enumi}
\renewcommand{\theenumii}{\@alph\c@enumii}
\renewcommand{\theenumiii}{\@roman\c@enumiii}
\renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

`\labelenumi` `enumerate` 環境の番号を出力する命令です。第 2 レベル以外は最後に欧文のピリオドが付きますが、これは好みに応じて取り払ってください。第 2 レベルの番号のかっこは和文用に換え、その両側に入る余分なグルーを `\jsInhibitGlue` で取り除いています。

```
\labelenumi \newcommand{\labelenumi}{\theenumi.}
\labelenumii \newcommand{\labelenumii}{\jsInhibitGlue (\theenumii) \jsInhibitGlue}
\labelenumiii \newcommand{\labelenumiii}{\theenumiii.}
\labelenumiv \newcommand{\labelenumiv}{\theenumiv.}
```

`\p@enumii` `\p@enumn` は `\ref` コマンドで `enumerate` 環境の第 n レベルの項目が参照されるときに書式です。これも第 2 レベルは和文用かっこにしました。

```
\p@enumii \renewcommand{\p@enumii}{\theenumi}
\p@enumiii \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
\p@enumiv \renewcommand{\p@enumiv}{\p@enumiii\theenumiii}
```

■itemize 環境

`\labelitemi` `itemize` 環境の第 n レベルのラベルを作るコマンドです。

```
\labelitemi \newcommand\labelitemi{\textbullet}
\labelitemii \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv \newcommand\labelitemiv{\textperiodcentered}
```

■description 環境

`description` 本来の `description` 環境では、項目名が短いと、説明部分の頭がそれに引きずられて左に出てしまいます。これを解決した新しい `description` の実装です。

```
\newenvironment{description}{%
\list{}{%
\labelwidth=\leftmargin
\labelsep=1\jsZw
\advance \labelwidth by -\labelsep
\let \makelabel=\descriptionlabel}}{\endlist}
```

`\descriptionlabel` `description` 環境のラベルを出力するコマンドです。好みに応じて #1 の前に適当な空き (たとえば `\hspace{1\jsZw}`) を入れるのもいいと思います。

```
\newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}
```

■概要

abstract 概要（要旨，梗概）を出力する環境です。book クラスでは各章の初めにちょっとしたことを書くのに使います。titlepage オプション付きの article クラスでは，独立したページに出力されます。abstract 環境は元は quotation 環境で作られていましたが，quotation 環境の右マージンをゼロにしたので，list 環境で作り直しました。

JSPF スタイルでは実際の出力は \maketitle で行われます。

```
(*book | report)
\newenvironment{abstract}{%
  \begin{list}{}{%
    \listparindent=1\jsZw
    \itemindent=\listparindent
    \rightmargin=0pt
    \leftmargin=5\jsZw\item[]\end{list}\vspace{\baselineskip}}
  (/book | report)
(*article | slide)
\newbox\@abstractbox
\if@titlepage
  \newenvironment{abstract}{%
    \titlepage
    \null\vfil
    \@beginparpenalty\@lowpenalty
    \begin{center}%
      \headfont \abstractname
      \@endparpenalty\@M
    \end{center}}%
  {\par\vfil\null\endtitlepage}
\else
  \newenvironment{abstract}{%
    \if@twocolumn
      \ifx\maketitle\relax
        \section*{\abstractname}%
      \else
        \global\setbox\@abstractbox\hbox\bgroup
        \begin{minipage}[b]{\textwidth}
          \small\parindent1\jsZw
          \begin{center}%
            {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
          \end{center}%
          \list{}{%
            \listparindent\parindent
            \itemindent \listparindent
            \rightmargin \leftmargin}%
          \item\relax
        \end{minipage}
      \fi
    \else
      \small
    \fi
  }
```

```

\begin{center}%
  {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
\end{center}%
\list{}{%
  \listparindent\parindent
  \itemindent \listparindent
  \rightmargin \leftmargin}%
\item\relax
\fi}{\if@twocolumn
  \ifx\maketitle\relax
  \else
    \endlist\end{minipage}\egroup
  \fi
\else
  \endlist
\fi}
\fi
</article | slide>
<*jspf>
\newbox\@abstractbox
\newenvironment{abstract}{%
  \global\setbox\@abstractbox\hbox\bgroup
  \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
  \small
  \if@english \parindent6mm \else \parindent1\jsZw \fi}%
  {\end{minipage}\egroup}
</jspf>

```

■キーワード

keywords キーワードを準備する環境です。実際の出力は `\maketitle` で行われます。

```

<*jspf>
%\newbox\@keywordsbox
%\newenvironment{keywords}{%
% \global\setbox\@keywordsbox\hbox\bgroup
% \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
% \small\parindent0\jsZw}%
% {\end{minipage}\egroup}
</jspf>

```

■verse 環境

verse 詩のための `verse` 環境です。

```

\newenvironment{verse}{%
  \let \=\@centercr
  \list{}{%
    \itemsep \z@
    \itemindent -2\jsZw % 元: -1.5em

```

```

\listparindent\itemindent
\rightmargin \z@
\advance\leftmargin 2\jsZw}% 元: 1.5em
\item\relax}{\endlist}

```

■quotation 環境

quotation 段落の頭の字下げ量を 1.5em から \parindent に変えました。また、右マージンを 0 にしました。

```

\newenvironment{quotation}{%
\list{}{%
\listparindent\parindent
\itemindent\listparindent
\rightmargin \z@}%
\item\relax}{\endlist}

```

■quote 環境

quote quote 環境は、段落がインデントされないことを除き、quotation 環境と同じです。

```

\newenvironment{quote}%
{\list{}{\rightmargin\z@}\item\relax}{\endlist}

```

■定理など ltthm.dtx 参照。たとえば次のように定義します。

```

\newtheorem{definition}{定義}
\newtheorem{axiom}{公理}
\newtheorem{theorem}{定理}

```

[2001-04-26] 定理の中はイタリック体になりましたが、これでは和文がゴシック体になってしまうので、\itshape を削除しました。

```

\def\@begintheorem#1#2{\trivlist
\item[\hskip \labelsep{\bfseries #1\ #2}]}
\def\@opargbegintheorem#1#2#3{\trivlist
\item[\hskip \labelsep{\bfseries #1\ #2\ (#3)}]}

```

titlepage タイトルを独立のページに出力するのに使われます。

```

\newenvironment{titlepage}{%
<book | report> \cleardoublepage
\if@twocolumn
\@restonecoltrue\onecolumn
\else
\@restonecolfalse\newpage
\fi
\thispagestyle{empty}%
\setcounter{page}\@ne
}%
{\if@restonecol\twocolumn \else \newpage \fi
\if@twoside\else

```

```

\setcounter{page}\@ne
\fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```

(*!book&!report)
\newcommand{\appendix}{\par
\setcounter{section}{0}%
\setcounter{subsection}{0}%
\gdef\presectionname{\appendixname}%
\gdef\postsectionname{}}%
% \gdef\thesection{\@Alph\c@section}% [2003-03-02]
\gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
\gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
(/!book&!report)
(*book | report)
\newcommand{\appendix}{\par
\setcounter{chapter}{0}%
\setcounter{section}{0}%
\gdef\@chapapp{\appendixname}%
\gdef\@chappos{}}%
\gdef\thechapter{\@Alph\c@chapter}}
(/book | report)

```

7.4 パラメータの設定

■array と tabular 環境

`\arraycolsep` array 環境の列間には `\arraycolsep` の 2 倍の幅の空が入ります。

```
\setlength\arraycolsep{5\p@}
```

`\tabcolsep` tabular 環境の列間には `\tabcolsep` の 2 倍の幅の空が入ります。

```
\setlength\tabcolsep{6\p@}
```

`\arrayrulewidth` array, tabular 環境内の罫線の幅です。

```
\setlength\arrayrulewidth{.4\p@}
```

`\doublerulesep` array, tabular 環境での二重罫線間のアキです。

```
\setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \’ コマンドで入るアキです。

```
\setlength\tabbingsep{\labelsep}
```

■minipage 環境

`\@mpfootins` minipage 環境の脚注の `\skip\@mpfootins` は通常のページの `\skip\footins` と同じ働きをします。

```
\skip\@mpfootins = \skip\footins
```

■framebox 環境

`\fboxsep` `\fbox`, `\framebox` で内側のテキストと枠との間の空きです。

`\fboxrule` `\fbox`, `\framebox` の罫線の幅です。

```
\setlength\fboxsep{3\p@}
\setlength\fboxrule{.4\p@}
```

■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

```
(!book&!report)\renewcommand \theequation {\@arabic\c@equation}
(*book|report)
\@addtoreset{equation}{chapter}
\renewcommand\theequation
{\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
(/book|report)
```

`\jot` `eqnarray` の行間に余分に入るアキです。デフォルトの値をコメントアウトして示しておきます。

```
% \setlength\jot{3pt}
```

`\@eqnnum` 数式番号の形式です。デフォルトの値をコメントアウトして示しておきます。

`\jsInhibitGlue` (`\theequation`) `\jsInhibitGlue` のように和文かっこを使うことも可能です。

```
% \def\@eqnnum{(\theequation)}
```

`amsmath` パッケージを使う場合は `\tagform@` を次のように修正します。

```
% \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip\@italiccorr)}}
```

7.5 フロート

タイプ `TYPE` のフロートオブジェクトを扱うには、次のマクロを定義します。

`\fps@TYPE` フロートを置く位置 (float placement specifier) です。

`\ftype@TYPE` フロートの番号です。2 の累乗 (1, 2, 4, ...) でなければなりません。

`\ext@TYPE` フロートの目次を出力するファイルの拡張子です。

`\fnum@TYPE` キャプション用の番号を生成するマクロです。

`\@makecaption<num><text>` キャプションを出力するマクロです。`<num>` は `\fnum@...` の生成する番号, `<text>` はキャプションのテキストです。テキストは適当な幅の `\parbox` に入ります。

■figure 環境

`\c@figure` 図番号のカウンタです。

`\thefigure` 図番号を出力するコマンドです。

```
(*!book&!report)
\newcounter{figure}
\renewcommand \thefigure {\@arabic\c@figure}
(!book&!report)
(*book | report)
\newcounter{figure}[chapter]
\renewcommand \thefigure
  {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
(/book | report)
```

`\fps@figure` figure のパラメータです。`\figurename` の直後に ~ が入っていましたが、ここでは外しました。

```
\ext@figure \def\fps@figure{tbp}
\ftype@figure \def\ftype@figure{1}
\fnum@figure \def\ext@figure{lof}
\def\figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* \newenvironment{figure}%
        {\@float{figure}}%
        {\end@float}
\newenvironment{figure*}%
        {\@dblfloat{figure}}%
        {\end@dblfloat}
```

■table 環境

`\c@table` 表番号カウンタと表番号を出力するコマンドです。アスキー版では `\thechapter.` が `\thetable \thechapter{}` になっていますが、ここではオリジナルのままにしています。

```
(*!book&!report)
\newcounter{table}
\renewcommand \thetable {\@arabic\c@table}
(!book&!report)
(*book | report)
\newcounter{table}[chapter]
\renewcommand \thetable
  {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
(/book | report)
```

`\fps@table` `table` のパラメータです。`\tablename` の直後に `~` が入っていましたが、ここでは外しました。

```
\ext@table    \def\fps@table{tbp}
\ftype@table   \def\ftype@table{2}
\fnun@table    \def\ext@table{lot}
               \def\fnun@table{\tablename\nobreak\thetable}
```

```
table * は段抜きのフロートです。
table*    \newenvironment{table}%
           {\float{table}}%
           {\endfloat}
           \newenvironment{table*}%
           {\dblfloat{table}}%
           {\enddblfloat}
```

7.6 キャプション

`\@makecaption` `\caption` コマンドにより呼び出され、実際にキャプションを出力するコマンドです。第 1 引数はフロートの番号、第 2 引数はテキストです。

`\abovecaptionskip` それぞれキャプションの前後に挿入されるスペースです。`\belowcaptionskip` が 0 になっていたので、キャプションを表の上につけた場合にキャプションと表がくっついてしまうのを直しました。

```
\newlength\abovecaptionskip
\newlength\belowcaptionskip
\setlength\abovecaptionskip{5\p@} % 元: 10\p@
\setlength\belowcaptionskip{5\p@} % 元: 0\p@
```

実際のキャプションを出力します。オリジナルと異なり、文字サイズを `\small` にし、キャプションの幅を 2cm 狭くしました。

[2003-11-05] ロジックを少し変えてみました。

```
<*\jspf>
% \long\def\@makecaption#1#2{{\small
%   \advance\leftskip1cm
%   \advance\rightskip1cm
%   \vskip\abovecaptionskip
%   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
%   \ifdim \wd\@tempboxa >\hsize
%     #1\hskip1\jsZw\relax #2\par
%   \else
%     \global \@minipagefalse
%     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
%   \fi
%   \vskip\belowcaptionskip}}
\long\def\@makecaption#1#2{{\small
  \advance\leftskip .0628\linewidth
```

```

\advance\rightskip .0628\linewidth
\vskip\abovecaptionskip
\sbbox\@tempboxa{#1\hskip1\jszw\relax #2}%
\ifdim \wd\@tempboxa <\hsize \centering \fi
#1\hskip1\jszw\relax #2\par
\vskip\belowcaptionskip}}
</!jspf>
<*jspf>
\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
  \sbbox\@tempboxa{\small\sffamily #1\quad #2}%
  \ifdim \wd\@tempboxa >\hsize
    {\small\sffamily
      \list{#1}{%
        \renewcommand{\makelabel}[1]{##1\hfil}
        \itemsep \z@
        \itemindent \z@
        \labelsep \z@
        \labelwidth 11mm
        \listparindent\z@
        \leftmargin 11mm}\item\relax #2\endlist}
  \else
    \global \@minipagefalse
    \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
</jspf>

```

8 フォントコマンド

ここでは L^AT_EX 2.09 で使われていたコマンドを定義します。これらはテキストモードと数式モードのどちらでも動作します。これらは互換性のためのもので、できるだけ `\text...` と `\math...` を使ってください。

`\mc` フォントファミリを変更します。

```

\gt \DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm \DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\rm \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sffamily \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\ttfamily \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

```

`\bf` ボールドシリーズにします。通常のミディアムシリーズに戻るコマンドは `\mdseries` です。

```
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

`\it` フォントシェイプを変えるコマンドです。斜体とスモールキャプスは数式中では何もしません（警告メッセージを出力します）。通常のアップライト体に戻るコマンドは `\upshape`

`\sc`

です。

```
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}  
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}  
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
```

`\cal` 数式モード以外では何もしません（警告を出します）。

```
\mit \DeclareRobustCommand*{\cal}{\@fontswitch\relax\mathcal}  
\mit \DeclareRobustCommand*{\mit}{\@fontswitch\relax\mathnormal}
```

9 相互参照

9.1 目次の類

`\section` コマンドは `.toc` ファイルに次のような行を出力します。

```
\contentsline{section}{タイトル}{ページ}
```

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

となります。この「番号」は `\thesection` コマンドで生成された見出し番号です。

`figure` 環境の `\caption` コマンドは `.lof` ファイルに次のような行を出力します。

```
\contentsline{figure}{\numberline{番号}{キャプション}{ページ}}
```

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

`\contentsline{...}` は `\l@...` というコマンドを実行するので、あらかじめ `\l@chapter`, `\l@section`, `\l@figure` などを定義しておかなければなりません。これらの多くは `\@dottedtocline` コマンドを使って定義します。これは

```
\@dottedtocline{レベル}{インデント}{幅}{タイトル}{ページ}
```

という書式です。

レベル この値が `tocdepth` 以下のときだけ出力されます。`\chapter` はレベル 0, `\section` はレベル 1, 等々です。

インデント 左側の字下げ量です。

幅 「タイトル」に `\numberline` コマンドが含まれる場合、節番号が入る箱の幅です。

`\@pnumwidth` ページ番号の入る箱の幅です。

`\@tocrmarg` 右マージンです。`\@tocrmarg ≥ \@pnumwidth` とします。

`\@dotsep` 点の間隔です（単位 `mu`）。

`\c@tocdepth` 目次ページに出力する見出しレベルです。元は `article` で 3, その他で 2 でしたが、ここ

では一つずつ減らしています。

```
\newcommand\@pnumwidth{1.55em}
\newcommand\@tocrmarg{2.55em}
\newcommand\@dotsep{4.5}
<!book&!report>\setcounter{tocdepth}{2}
(book | report)\setcounter{tocdepth}{1}
```

■ 目次

`\tableofcontents` 目次を生成します。

```
\newcommand{\tableofcontents}{%
(*book | report)
  \if@twocolumn
    \@restonecoltrue\onecolumn
  \else
    \@restonecolfalse
  \fi
  \chapter*{\contentsname}%
  \@mkboth{\contentsname}{}%
</book | report>
<*>book&!report>
  \section*{\contentsname}%
  \@mkboth{\contentsname}{\contentsname}%
</!book&!report>
  \@starttoc{toc}%
(book | report) \if@restonecol\twocolumn\fi
}
```

`\l@part` 部の目次です。

```
\newcommand*{\l@part}[2]{%
  \ifnum \c@tocdepth >-2\relax
<!book&!report> \addpenalty\@secpenalty
(book | report) \addpenalty{-\@highpenalty}%
    \addvspace{2.25em \@plus\p@}%
    \begingroup
      \parindent \z@
% \@pnumwidth should be \@tocrmarg
% \rightskip \@pnumwidth
\rightskip \@tocrmarg
\parfillskip -\rightskip
{\leavevmode
  \large \headfont
  \setlength\@lnumwidth{4\jsZw}%
  #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
  \nobreak
(book | report) \global\@nobreaktrue
(book | report) \everypar{\global\@nobreakfalse\everypar{}}%
  \endgroup
\fi}
```

`\l@chapter` 章の目次です。 `\@lnumwidth` を 4.683zw に増やしました。

```
(*book|report)
\newcommand*{\l@chapter}[2]{%
  \ifnum \c@tocdepth > \m@ne
    \addpenalty{-\@highpenalty}%
    \addvspace{1.0em \@plus\p@}
%   \vskip 1.0em \@plus\p@   % book.cls では↑がこうなっている
    \begingroup
      \parindent\z@
%     \rightskip\@pnumwidth
      \rightskip\@tocrmarg
      \parfillskip-\rightskip
      \leavevmode\headfont
      \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
      \advance\leftskip\@lnumwidth \hskip-\leftskip
      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
      \penalty\@highpenalty
    \endgroup
  \fi}
(/book|report)
```

`\l@section` 節の目次です。

```
(*!book&!report)
\newcommand*{\l@section}[2]{%
  \ifnum \c@tocdepth > \z@
    \addpenalty{\@secpenalty}%
    \addvspace{1.0em \@plus\p@}%
    \begingroup
      \parindent\z@
%     \rightskip\@pnumwidth
      \rightskip\@tocrmarg
      \parfillskip-\rightskip
      \leavevmode\headfont
      \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
      \advance\leftskip\@lnumwidth \hskip-\leftskip
      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
    \endgroup
  \fi}
(/!book&!report)
```

インデントと幅はそれぞれ 1.5em, 2.3em でしたが³, 1zw, 3.683zw に変えました。

```
(book|report)\newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\l@subsection` さらに下位レベルの目次項目の体裁です。あまり使ったことがありませんので、要修正かも

`\l@subsubsection` しれません。

```
\l@paragraph (*!book&!report)
% \newcommand*{\l@subsection}{\@dottedtocline{2}{1.5em}{2.3em}}
\l@subparagraph \newcommand*{\l@subsection}{\@dottedtocline{2}{1\jsZw}{3\jsZw}}
% \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
```

```

\newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
% \newcommand*{\l@paragraph}{\@dottedtocline{4}{7.0em}{4.1em}}
\newcommand*{\l@paragraph}{\@dottedtocline{4}{3\jsZw}{3\jsZw}}
% \newcommand*{\l@subparagraph}{\@dottedtocline{5}{10em}{5em}}
\newcommand*{\l@subparagraph}{\@dottedtocline{5}{4\jsZw}{3\jsZw}}
</!book&!report>
<*book | report>
\newcommand*{\l@subsection}{\@dottedtocline{2}{3.8em}{3.2em}}
\newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
\newcommand*{\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
\newcommand*{\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
</book | report>

```

`\numberline` 欧文版 L^AT_EX では `\numberline{...}` は幅 `\@tempdima` の箱に左詰めで出力する命令ですが、アスキー版では `\@tempdima` の代わりに `\@lnumwidth` という変数で幅を決めるように再定義しています。後続文字が全角か半角かでスペースが変わらないように `\hspace` を入れておきました。

```

\newdimen\@lnumwidth
\def\numberline#1{\hbext@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

`\@dottedtocline` L^AT_EX 本体 (ltsect.dtx 参照) での定義と同じですが、`\@tempdima` を `\@lnumwidth` に変えています。

```

\def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
\vskip \z@ \@plus.2\p@
{\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
\parindent #2\relax\@afterindenttrue
\interlinepenalty\@M
\leavevmode
\@lnumwidth #3\relax
\advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
{#4}\nobreak
\leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
mu$}\hfill \nobreak\hbext@\@pnumwidth{#5}
\hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

\newcommand{\listoffigures}{%
<*book | report>
\if@twocolumn\@restonecoltrue\onecolumn
\else\@restonecolfalse\fi
\chapter*{\listfigurename}%
\@mkboth{\listfigurename}{}%
</book | report>
<!*book&!report>
\section*{\listfigurename}%
\@mkboth{\listfigurename}{\listfigurename}%

```

```

</!book&!report)
  \@starttoc{lof}%
<book | report> \if@restonecol\twocolumn\fi
}

```

`\l@figure` 図目次の項目を出力します。

```
\newcommand*{\l@figure}{\@dottedtocline{1}{1}{jsZw}{3.683}{jsZw}}
```

`\listoftables` 表目次を出力します。

```

\newcommand{\listoftables}{%
<*book | report>
  \if@twocolumn\@restonecoltrue\onecolumn
  \else\@restonecolfalse\fi
  \chapter*{\listtablename}%
  \@mkboth{\listtablename}{}%
</book | report>
<!*book&!report>
  \section*{\listtablename}%
  \@mkboth{\listtablename}{\listtablename}%
</!book&!report>
  \@starttoc{lot}%
<book | report> \if@restonecol\twocolumn\fi
}

```

`\l@table` 表目次は図目次と同じです。

```
\let\l@table\l@figure
```

9.2 参考文献

`\bibindent` オープンスタイルの参考文献で使うインデント幅です。元は 1.5em でした。

```

\newdimen\bibindent
\setlength\bibindent{2\jsZw}

```

`thebibliography` 参考文献リストを出力します。

```

\newenvironment{thebibliography}[1]{%
  \global\let\presectionname\relax
  \global\let\postsectionname\relax
<article | slide> \section*{\refname}\@mkboth{\refname}{\refname}%
<*kiyou>
  \vspace{1.5\baselineskip}
  \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
  \vspace{0.5\baselineskip}
</kiyou>
<book | report> \chapter*{\bibname}\@mkboth{\bibname}{}%
<book | report> \addcontentsline{toc}{chapter}{\bibname}%
  \list{\@biblabel{\@arabic\c@enumiv}}%
    {\settowidth\labelwidth{\@biblabel{#1}}%
    \leftmargin\labelwidth

```



```

\advance\leftmargin\labelsep
\@openbib@code
\usecounter{enumiv}%
\let\p@enumiv\@empty
\renewcommand\theenumiv{\@arabic\c@enumiv}%
(kiyou) \small
\slippy
\clubpenalty4000
\@clubpenalty\clubpenalty
\widowpenalty4000%
\sffcode'\.\@m}
{\def\@noitemerr
{\@latex@warning{Empty 'thebibliography' environment}}}%
\endlist}

\newblock \newblock はデフォルトでは小さなスペースを生成します。
\newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

\@openbib@code \@openbib@code はデフォルトでは何もしません。この定義は openbib オプションによっ
て変更されます。
\let\@openbib@code\@empty

\@biblabel \bibitem[...] のラベルを作ります。ltbibl.dtx の定義の半角 [] を全角 [] に変え、余
分なスペースが入らないように \jsInhibitGlue ではさみました。とりあえずコメントア
ウトしておきますので、必要に応じて生かしてください。
% \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

\cite 文献の番号を出力する部分は ltbibl.dtx で定義されていますが、コンマとカッコを和文
\@cite フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必
\@citex 要に応じて生かしてください。かつこの前後に入るグルーを \jsInhibitGlue で取ってい
ますので、オリジナル同様、Knuth~\cite{knu} のように半角空白で囲んでください。
% \def\@citex[#1]#2{%
% \let\@citea\@empty
% \@cite{\@for\@citeb:=#2\do
% {\@citea\def\@citea{, \jsInhibitGlue\penalty\@m\ }%
% \edef\@citeb{\expandafter\@firstofone\@citeb}%
% \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
% \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
% \G@refundefinedtrue
% \@latex@warning
% {Citation '\@citeb' on page \thepage \space undefined}}%
% {\hbox{\csname b@\@citeb\endcsname}}}{#1}}
% \def\cite#1#2{\jsInhibitGlue [{#1}\if@tempswa , #2\fi] \jsInhibitGlue}

引用番号を上ツキの 1) のようなスタイルにするには次のようにします。 \cite の先頭に
\unskip を付けて先行のスペース ( ~ も ) を帳消しにしています。
% \DeclareRobustCommand\cite{\unskip
% \ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}}

```

```
% \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
% , \jsInhibitGlue\ #2\fi}}}$}
```

9.3 索引

theindex 2～3 段組の索引を作成します。最後が偶数ページのときにマージンがずれる現象を直しました (Thanks: 藤村さん)。

```
\newenvironment{theindex}{% 索引を 3 段組で出力する環境
  \if@twocolumn
    \onecolumn\@restonecolfalse
  \else
    \clearpage\@restonecoltrue
  \fi
  \columnseprule.4pt \columnsep 2\jsZw
  \ifx\multicols\@undefined
<book | report> \twocolumn[\@makeschapterhead{\indexname}%
<book | report> \addcontentsline{toc}{chapter}{\indexname}]%
<!book&!report> \def\presectionname{}\def\postsectionname{}%
<!book&!report> \twocolumn[\section*{\indexname}]%
  \else
    \ifdim\textwidth<\fullwidth
      \setlength{\evensidemargin}{\oddsidemargin}
      \setlength{\textwidth}{\fullwidth}
      \setlength{\linewidth}{\fullwidth}
<book | report> \begin{multicols}{3}[\chapter*{\indexname}%
<book | report> \addcontentsline{toc}{chapter}{\indexname}]%
<!book&!report> \def\presectionname{}\def\postsectionname{}%
<!book&!report> \begin{multicols}{3}[\section*{\indexname}]%
    \else
<book | report> \begin{multicols}{2}[\chapter*{\indexname}%
<book | report> \addcontentsline{toc}{chapter}{\indexname}]%
<!book&!report> \def\presectionname{}\def\postsectionname{}%
<!book&!report> \begin{multicols}{2}[\section*{\indexname}]%
    \fi
  \fi
  \ifx\mkboth\@mkboth{\indexname}{\indexname}%
  \ifx\mkboth\@mkboth{\indexname}{\indexname}%
    \plainifnotempty % \thispagestyle{plain}
    \parindent\z@
    \parskip\z@ \@plus .3\p@\relax
    \let\item\@idxitem
    \raggedright
    \footnotesize\narrowbaselines
  }{
    \ifx\multicols\@undefined
      \if@restonecol\onecolumn\fi
    \else
```

```

\end{multicols}
\fi
\clearpage
}

\@idxitem 索引項目の字下げ幅です。 \@idxitem は \item の項目の字下げ幅です。
\subitem   \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
            \newcommand{\subsubitem}{\@idxitem \hspace*{3\jsZw}} % 元 30pt

\indexspace 索引で先頭文字ごとのブロックの間に入るスペースです。
            \newcommand{\indexspace}{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}

\seename   索引の \see, \seealso コマンドで出力されるものです。デフォルトはそれぞれ see,
\alsoname  see also という英語ですが、ここではとりあえず両方とも「→」に変えました。⇒
            ($\rightarrow$) などでもいいでしょう。
            \newcommand\seename{\if@english see\else →\fi}
            \newcommand\alsoname{\if@english see also\else →\fi}

```

9.4 脚注

```

\footnote  和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため、
\footnotemark \jsInhibitGlue を入れることにします。
            \let\footnotes@ve=\footnote
            \def\footnote{\jsInhibitGlue\footnotes@ve}
            \let\footnotemarks@ve=\footnotemark
            \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}

\@makefnmark 脚注番号を付ける命令です。ここでは脚注番号の前に記号 * を付けています。「注
1」の形式にするには \textasteriskcentered を 注\kern0.1em にしてください。
\@xfootnotenext と合わせて、もし脚注番号が空なら記号も出力しないようにしてあ
ります。
[2002-04-09] インプリメントの仕方を変えたため消しました。
% \def\@makefnmark{\hbox{\@textsuperscript{\normalfont
%       \ifx\@thefnmark\empty\else
%       \textasteriskcentered
%       \fi
%       \@thefnmark}}}}

\thefootnote 脚注番号に * 印が付くようにしました。ただし、番号がゼロのときは * 印も脚注番号も付
きません。
[2003-08-15] \textasteriskcentered ではフォントによって下がりすぎるので変更しま
した。
\def\thefootnote{\ifnum\c@footnote>\z@ \leavevmode\lower.5ex\hbox{*}\@arabic\c@footnote\fi}
「注 1」の形式にするには次のようにしてください。
% \def\thefootnote{\ifnum\c@footnote>\z@ 注\kern0.1\jsZw\@arabic\c@footnote\fi}

```

`\footnoterule` 本文と脚注の間の罫線です。

```

\renewcommand{\footnoterule}{%
  \kern-3\p@
  \hrule width .4\columnwidth
  \kern 2.6\p@}

```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```

(book | report) \@addtoreset{footnote}{chapter}

```

`\@footnotetext` 脚注で `\verb` が使えるように改変してあります。Jeremy Gibbons, *TEX and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9

```

\long\def\@footnotetext{%
  \insert\footins\bgroup
  \normalfont\footnotesize
  \interlinepenalty\interfootnotelinepenalty
  \splittopskip\footnotesep
  \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
  \hsize\columnwidth \@parboxrestore
  \protected@edef\@currentlabel{%
    \csname p@footnote\endcsname\@thefnmark
  }%
  \color@begingroup
  \@makefnmark
  \rule{.5\footnotesep}{\footnotesep}%
  \futurelet\next\fo@t
  \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
    \else \let\next\fo@t\fi \next}
  \def\fo@t{\bgroup\aftergroup\@foot\let\next}
  \def\fo@t#1{#1\@foot}
  \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup}

```

`\@makefnmark` 実際に脚注を出力する命令です。`\@makefnmark` は脚注の番号を出力する命令です。ここでは脚注が左端から一定距離に来るようにしてあります。

```

\newcommand\@makefnmark[1]{%
  \advance\leftskip 3\jsZw
  \parindent 1\jsZw
  \noindent
  \llap{\@makefnmark\hskip0.3\jsZw}#1}

```

`\@xfootnotenext` 最初の `\footnotetext{...}` は番号が付きません。著者の所属などを脚注の欄に書くときに便利です。

すでに `\footnote` を使った後なら `\footnotetext[0]{...}` とすれば番号を付けない脚注になります。ただし、この場合は脚注番号がリセットされてしまうので、工夫が必要です。

[2002-04-09] インプリメントの仕方を変えたため消しました。

```

% \def\@xfootnotenext[#1]{%
%   \begingroup

```

```
% \ifnum#1>\z@
% \csname c@\@mpfn\endcsname #1\relax
% \unrestored@protected@xdef\@thefnmark{\thempfn}%
% \else
% \unrestored@protected@xdef\@thefnmark{}%
% \fi
% \endgroup
% \@footnotetext}
```

ここまでのコードは JS クラスを踏襲する。

10 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

```
\def\@inhibitglue{%
  \futurelet\@let@token\@inhibitglue}
\def\@inhibitglue{%
  \ifx\@let@token「
    \jsInhibitGlue
  \else
    \ifx\@let@token (
      \jsInhibitGlue
    \else
      \ifx\@let@token『
        \jsInhibitGlue
      \else
        \ifx\@let@token [
          \jsInhibitGlue
        \fi
      \fi
    \fi
  \fi}
\def\everyparhook{\jsInhibitGlueAtParTop}
\let\jsInhibitGlueAtParTop\@inhibitglue
\AtBeginDocument{\everypar\everyparhook}}
```

これだけではいけないようです。あちこちに `\everypar` を初期化するコマンドが隠されていました。

まず、環境の直後の段落です。

```
\def\@doendpe{%
  \@endpetrue
  \def\par{%
    \restorepar\everypar{\everyparhook}\par\@endpefalse}%
    \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
\item 命令の直後です。
```

```

\def\@item[#1]{%
  \if@noperitem
    \@donoperitem
  \else
    \if@inlabel
      \indent \par
    \fi
    \ifhmode
      \unskip\unskip \par
    \fi
    \if@newlist
      \if@nobreak
        \@nbitem
      \else
        \addpenalty\@beginparpenalty
        \addvspace\@topsep
        \addvspace{-\parskip}%
      \fi
    \else
      \addpenalty\@itempenalty
      \addvspace\itemsep
    \fi
    \global\@inlabeltrue
  \fi
\everypar{%
  \@minipagefalse
  \global\@newlistfalse
  \if@inlabel
    \global\@inlabelfalse
    {\setbox\z@\lastbox
     \ifvoid\z@
       \kern-\itemindent
     \fi}%
    \box\@labels
    \penalty\z@
  \fi
  \if@nobreak
    \@nobreakfalse
    \clubpenalty \@M
  \else
    \clubpenalty \@clubpenalty
    \everypar{\everyparhook}%
  \fi\everyparhook}%
\if@noitemarg
  \@noitemargfalse
  \if@nmbrlist
    \refstepcounter\@listctr
  \fi
\fi

```

```

\sbbox\@tempboxa{\makelabel{#1}}%
\global\setbox\@labels\hbox{%
  \unhbox\@labels
  \hskip \itemindent
  \hskip -\labelwidth
  \hskip -\labelsep
  \ifdim \wd\@tempboxa >\labelwidth
    \box\@tempboxa
  \else
    \hbox to\labelwidth {\unhbox\@tempboxa}%
  \fi
  \hskip \labelsep}%
\ignorespaces}

```

二つ挿入した `\everyparhook` のうち後者が `\section` 類の直後に 2 回、前者が 3 回目以降に実行されます。

```

\def\@afterheading{%
  \@nobraektrue
  \everypar{%
    \if@nobraek
      \@nobraekfalse
      \clubpenalty \@M
      \if@afterindent \else
        {\setbox\z@\lastbox}%
      \fi
    \else
      \clubpenalty \@clubpenalty
      \everypar{\everyparhook}%
    \fi\everyparhook}}

```

`\@gnewline` についてはちょっと複雑な心境です。もともとの $\text{p}\text{A}\text{T}\text{E}\text{X }2_{\epsilon}$ は段落の頭にグルーが入る方で統一されていました。しかし `\` の直後にはグルーが入らず、不統一でした。そこで `\` の直後にもグルーを入れるように直していただいた経緯があります。しかし、ここでは逆にグルーを入れない方で統一したいので、また元に戻してしまいました。

しかし単に戻すだけでも駄目みたいなので、ここでも最後にグルーを消しておきます。

```

\def\@gnewline #1{%
  \ifvmode
    \@nolnerr
  \else
    \unskip \reserved@e {\reserved@f#1}\nobraek \hfil \break \null
    \jsInhibitGlue \ignorespaces
  \fi}

```

11 いろいろなロゴ

とりあえず削除。

12 amsmath との衝突の回避

最近の \LaTeX では該当の問題は対処されているので削除。

13 初期設定

■いろいろな語

```
\prepartname
\postpartname      \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername    \newcommand{\postpartname}{\if@english\else 部\fi}
\postchaptername   <book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\presectionname    <book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\postsectionname   \newcommand{\presectionname}{}% 第
\presectionname    \newcommand{\postsectionname}{}% 節
\contentsname
\listfigurename    \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listfigurename    \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
\listtablename     \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname           \newcommand{\refname}{\if@english References\else 参考文献\fi}
\bibname           \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
\indexname         \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename         \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
\tablename         \newcommand{\tablename}{\if@english Table.~\else 表\fi}

\appendixname
\abstractname      \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
\abstractname      (!book&!report)\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}
```

■今日の日付 \LaTeX で処理した日付を出力します。jarticle などと違って、標準を西暦にし、余分な空白が入らないように改良しました。和暦にするには \和暦 と書いてください。

```
\today
\@tempswafalse
\if p\bxjs@engine \@tempwattrue \fi
\if n\bxjs@engine \@tempwattrue \fi
```



```

\if@tempswa \expandafter\@firstoftwo
\else      \expandafter\@secondoftwo
\fi
{%
% 欧文 8bitTeX の場合
\newif\ifjsSeireki \jsSeirekitrue
\def\西暦{\jsSeirekitrue}
\def\和暦{\jsSeirekifalse}
\def\Seireki{\jsSeirekitrue}
\def\Wareki{\jsSeirekifalse}
\def\bxjs@if@use@seireki{%
  \ifjsSeireki \expandafter\@firstoftwo
  \else \expandafter\@secondoftwo \fi}
}%
\newif\if 西暦 \西暦 true
\def\西暦{\西暦 true}
\def\和暦{\西暦 false}
\def\Seireki{\西暦 true}
\def\Wareki{\西暦 false}
\def\bxjs@if@use@seireki{%
  \if 西暦 \expandafter\@firstoftwo
  \else \expandafter\@secondoftwo \fi}
}
% \bxjs@unxp
\let\bxjs@unxp\@firstofone
\bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
% \bxjs@iai
\if \if p\bxjs@engine T\else\if n\bxjs@engine T\else F\fi\fi T
  \def\bxjs@iai{\noexpand~}
\else \def\bxjs@iai{}
\fi
% \heisei
\newcount\heisei \heisei\year \advance\heisei-1988\relax
% \today
\edef\today{%
  \if@english
    \ifcase\month\or
      January\or February\or March\or April\or May\or June\or
      July\or August\or September\or October\or November\or December\fi
    \space\number\day, \number\year
  \else
    \noexpand\bxjs@if@use@seireki{%
      \number\year\bxjs@iai\bxjs@unxp{年}%
      \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
      \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
    }{%
      \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
      \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
      \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
    }
  }
}

```

```
}%
\fi}
```

■ハイフネーション例外 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ のハイフネーションルールの補足です（ペンディング：
eng-lish）

```
\hyphenation{ado-be post-script ghost-script phe-nom-e-no-log-i-cal man-u-script}
```

■ページ設定 ページ設定の初期化です。

```
<slide>\pagestyle{empty}%
<article|report>\pagestyle{plain}%
<book>\pagestyle{headings}%
\pagenumbering{arabic}
\if@twocolumn
  \twocolumn
  \sloppy
  \flushbottom
\else
  \onecolumn
  \raggedbottom
\fi
<*slide>
% \renewcommand\kanjifamilydefault{\gtdefault}
% \renewcommand\familydefault{\sfdefault}
% \raggedright
% \xkanjiskip=0.1em\relax
</slide>
```

(u)p $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 上で動作している場合は、p $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 用の連携モジュールを読み込む。最後に日本語文字のカテゴリコードを元に戻す。

```
\ifbxjs@ptexmodule
  \input{bxjsptex.def}
\fi
\bxjs@restore@jltrcc
</!ptex>
```

以上です。

付録 A 連携モジュールの仕様 ☆

- 次の命令が BXJS クラス側で定義される。
 - `\jsDocClass` [文字トークンの `let`] 文書クラスの種類を示し、次のいずれかと一致する (`\if` で判定可能)。

<code>\jsArticle</code>	<code>bxjsarticle</code> クラス
<code>\jsBook</code>	<code>bxjsbook</code> クラス
<code>\jsReport</code>	<code>bxjsreport</code> クラス
<code>\jsSlide</code>	<code>bxjsslide</code> クラス
 - `\jsUnusualPtSize` [整数定数を表す文字列のマクロ] 基底フォントサイズが 10pt、11pt、12pt のいずれでもない場合の `\@ptsize` の値。 (`\@ptsize` 自体があまり有用でないと思われる。)
 - `\jsScale` [実数を表す文字列のマクロ] 和文フォントサイズの要求サイズに対するスケール。クラスオプション `scale` で指定される。(既定値は 0.924715。)
 - `\jsInhibitGlue` [マクロ] `\inhibitglue` という命令が定義されていればそれを実行し、そうでなければ何もしない。JS クラスで `\inhibitglue` を用いている箇所は全て `\jsInhibitGlue` に置き換えられている。従って、`\inhibitglue` は未定義でも動作するが、その実装がある場合は BXJS クラスはそれを活用する。
- 「現在の全角幅」を表す長さ変数 `\jsZw` が用意されていて、JS クラスで `zw` 単位で設定されている長さパラメタはこの変数を単位として設定されている。この変数の値は実際に用いられる「和文フォント」のメトリックに基づくのではなく、機械的に `\jsScale × (フォントサイズ)` であると定められている (フォントサイズ変更の度に再設定される)。従って、「和文コンポーネント」はこの設定と辻褄が合うように和文フォントサイズを調整する必要がある。ほとんどの場合、和文フォントを NFSS で規定する際に `\jsScale` の値をスケール値として与えれば上手くいく。
- フォントサイズが変更された時には `\jsResetDimen` というマクロが呼び出される。和文コンポーネントでフォントサイズに依存するパラメタをここで設定することができる。既定の定義は空。

付録 B pTeX 用の連携モジュール ☆

ここでは、`upTeX` 用の連携モジュールのコードを示す。基本的に、JS クラスのコードの中で、「和文コンポーネントの管轄」として BXJS クラスで除外されている部分に相当するが、若干の変更が加えられている。このモジュールは、BXJS クラスのオプションとして `platex` または `uplatex` を指定した時に自動的に読み込まれる。オプションと実際のエンジンが一致しているかは既に検査されている。

■エンジン依存の定義 最初にエンジンに依存する定義を行う。`\if@jsc@uplatex` は BXJS において定義されているスイッチで、エンジンが `upTEX` であるかを表す。

`\jsc@JYn` および `\jsc@JTn` は標準の和文横書きおよび縦書き用エンコーディングを表す。

```
< *ptex>
\edef\jsc@JYn{\if@jsc@uplatex JY2\else JY1\fi}
\edef\jsc@JTn{\if@jsc@uplatex JT2\else JT1\fi}
\edef\jsc@pfx{\if@jsc@uplatex u\fi}
```

`\bxjs@declarefontshape` は標準の和文フォント宣言である。後で `\bxjs@scale` を求めるため一旦マクロにしておく。`\bxjs@sizereference` は全角幅を測定する時に参照するフォント。

まず `upTEX` の場合の定義を示す。JS クラスの `uplatex` オプション指定時の定義と同じである。

```
\@onlypreamble\bxjs@declarefontshape
\if@jsc@uplatex
\def\bxjs@declarefontshape{%
\DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjisr-h}{}%
\DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjisg-h}{}%
\DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjisr-v}{}%
\DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjisg-v}{}%
}
\def\bxjs@sizereference{upjisr-h}
```

`pTEX` の場合の定義を示す。JS クラスのフォント種別オプション非指定時の定義と同じである。

```
\else
\def\bxjs@declarefontshape{%
\DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
\DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
\DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
\DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
}
\def\bxjs@sizereference{jis}
\fi
```

既に使用されている標準和文フォント定義がもしあれば取り消す。(BXJS クラスの中から呼ばれている場合は不要のはずである。)

```
\@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
\jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
{\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined}
```

■和文フォントスケールの補正 実は、`pTEX` の標準的な和文フォント (JFM のこと、例えば `jis`) では、指定された `\jsScale` (この値を s とする) をそのまま使って定義すると期待通りの大きさにならない。これらの JFM では `1zw` の大きさが指定されたサイズではなく既にスケール (この値を f とする; `jis` では 0.962216 倍) が掛けられた値になっているか

らである。そのため、ここでは s/f を求めてその値をマクロ `\bxjs@scale` に保存する。

```
\begingroup
% 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
\font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
\setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
% 幅が丁度 10pt なら補正は不要
\ifdim\wd\z@=10pt
\global\let\bxjs@scale\jsScale
\else
% (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
\edef\bxjs@tmpa{\strip@pt\wd\z@}
\@tempdima=10pt \@tempdima=\jsScale\@tempdima
\bxjs@invscale\@tempdima\bxjs@tmpa
\xdef\bxjs@scale{\strip@pt\@tempdima}
\fi
\endgroup
%\typeout{\string\bxjs@scale : \bxjs@scale}
```

■和文フォント関連定義 `\bxjs@scale` が決まったので先に保存した標準和文フォント宣言を実行する。

```
\bxjs@declarefontshape
フォント代替の明示的定義。
\DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
\DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
\DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
\DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
\DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
\DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
\DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
\DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
\DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
\DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
\DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
\DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
\DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
\DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
```

欧文総称フォント命令で和文フォントが連動するように修正する。その他の和文フォント関係の定義を行う。

```
\DeclareRobustCommand\rmfamily
{\not@math@alphabet\rmfamily\mathrm
\romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
\DeclareRobustCommand\sffamily
{\not@math@alphabet\sffamily\mathsf
\romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
\DeclareRobustCommand\ttfamily
{\not@math@alphabet\ttfamily\mathtt
\romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
```

```
\def\textmc#1{\relax\ifmmode\hbox\fi{\mcfamily #1}}
\def\textgt#1{\relax\ifmmode\hbox\fi{\gtfamily #1}}
```

■パラメタの設定

```
\prebreakpenalty\jis"2147=10000
\postbreakpenalty\jis"2148=10000
\prebreakpenalty\jis"2149=10000
\inhibitxspcode'\overline=2
\xspcode'+=3
\xspcode'\%=3

"80~"FF の範囲の \spcode を 3 に変更。

\@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
  \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
</ptex>
```

以上で終わり。 \jsResetDimen は空のままでよい。

付録 C CJK パッケージ用の連携モジュール ☆

■準備 CJKspace と CJKpunct パッケージを読み込む。

```
< * c j k >
\RequirePackage{CJKspace}
\RequirePackage{CJKpunct}
\RequirePackage{ifpdf}
```

zhmetrics パッケージで提供されるファミリの情報を集約する。 \bxjs@CJKfam@list はファミリ名の \do-リスト、 \bxjs@tfm@<fam> はファミリに対応する TFM 名 (の基底)。

```
\let\bxjs@CJKfam@list\@empty
\def\bxjs@entry#1#2{%
  \let\do\relax
  \edef\bxjs@CJKfam@list{\bxjs@CJKfam@list\do{#1}}%
  \@namedef{bxjs@tfm@#1}{#2}%
}
\bxjs@entry {zh song} {unison g}
\bxjs@entry {zh kai} {unikai}
\bxjs@entry {zh hei} {unihei}
\bxjs@entry {zh fs} {unifs}
\bxjs@entry {zh li} {unili}
\bxjs@entry {zh you} {uniyou}
\def\bxjs@letcs#1#2{%
  \expandafter\let\expandafter#1\csname#2\endcsname
}
```

■フォントマップ設定 \mcdefault は明朝体用のファミリ名。 \gtdefault はゴシック体用のファミリ名。(これらの値は変更禁止。)

```
\newcommand*\mcdefault{zh song}
\newcommand*\gtdefault{zh hei}
```

`\setCJKfontmap{<fam>}{<fontfile>}` : ファミリ `<fam>` に対する物理フォントを `<fontfile>` に設定する。

```
\@onlypreamble\setCJKfontmap
\newcommand*\setCJKfontmap[2]{%
  \bxjs@letcs\bxjs@tmpa{\bxjs@tfm@#1}%
  \ifx\bxjs@tmpa\relax
    \ClassError\bxjs@clsname
      {Unknown CJK family '#1'}\@ehc
  \else
    \expandafter\xdef\csname bxjs@fontmap@#1\endcsname{#2}%
  \fi
}
```

IPA フォントを既定とする。

```
\setCJKfontmap{\mcdefault}{ipam.ttf}
\setCJKfontmap{\gtdefault}{ipag.ttf}
```

フォントマップ指定の実装。

```
\@onlypreamble\bxjs@do@onefam
\def\bxjs@do@onefam#1{%
  \bxjs@letcs\bxjs@tmpa{\bxjs@fontmap@#1}%
  \ifx\bxjs@tmpa\relax\else
    \bxjs@letcs\bxjs@tmpb{\bxjs@tfm@#1}%
    \ifpdf
      \pdfmapline{=\bxjs@tmpb@Unicode@ <\bxjs@tmpa}%
      \pdfmapline{=\bxjs@tmpb sl@Unicode@ <\bxjs@tmpa}%
    \else
      \AtBeginDvi{%
        \special{pdf:mapline \bxjs@tmpb @Unicode@ unicode \bxjs@tmpa}%
        \special{pdf:mapline \bxjs@tmpb sl@Unicode@ unicode \bxjs@tmpa}}%
      \fi
    \fi
  }
\AtBeginDocument{%
  \let\do\bxjs@do@onefam
  \bxjs@CJKfam@list
}
```

■ フォント選択命令の修正 欧文のフォント選択命令について和文を追従させる。

```
\providecommand*\CJKfamilydefault{\mcdefault}
\DeclareRobustCommand\normalfont{%
  \CJKfamily{\CJKfamilydefault}%
  \usefont\encodingdefault\familydefault
    \seriesdefault\shapedefault\relax
}
\let\reset@font\normalfont
\DeclareRobustCommand\rmfamily{%
  \not@math@alphabet\rmfamily\mathrm
  \fontfamily\rmdefault\CJKfamily{\mcdefault}\selectfont
}
```

```

}
\DeclareRobustCommand\sffamily{%
  \not@math@alphabet\sffamily\mathsf
  \fontfamily\rmdefault\CJKfamily{\gtdefault}\selectfont
}
\DeclareRobustCommand\ttfamily{%
  \not@math@alphabet\ttfamily\mathtt
  \fontfamily\rmdefault\CJKfamily{\gtdefault}\selectfont
}

```

■全般設定 本体の全体が自動的に CJK* 環境に入るようにする。

```

\AtEndOfPackage{\CJK@makeActive}
\AtBeginDocument{\begin{CJK*}{UTF8}{\CJKfamilydefault}}
\AtEndDocument{\clearpage\end{CJK*}}
</cjk>

```