

# BXjscls パッケージ (BXJS 文書クラス集) ソースコード説明書

八登崇之 (Takayuki YATO; aka. “ZR”)

v1.6 [2017/09/09]

---




この文書はソースコード説明書です。一般の文書作成者向けの解説については、ユーザーマニュアル `bxjscls-manual.pdf` を参照してください。

---

## 目次

1	はじめに	3
2	オプション	7
3	和文フォントの変更	35
4	フォントサイズ	36
5	レイアウト	41
5.1	ページレイアウト . . . . .	42
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	55
7	ページスタイル	56
8	文書のマークアップ	60
8.1	表題 . . . . .	60
8.2	章・節 . . . . .	65
8.3	リスト環境 . . . . .	76
8.4	パラメータの設定 . . . . .	84
8.5	フロート . . . . .	85
8.6	キャプション . . . . .	87
9	フォントコマンド	88

10	相互参照	90
10.1	目次の類 . . . . .	90
10.2	参考文献 . . . . .	96
10.3	索引 . . . . .	97
10.4	脚注 . . . . .	99
11	段落の頭へのグルー挿入禁止	101
12	いろいろなロゴ	105
13	amsmath との衝突の回避	105
14	初期設定	106
付録 A	和文ドライバの仕様 罨	110
付録 B	和文ドライバ : minimal 罨	111
B.1	補助マクロ . . . . .	111
B.2	(u)pTeX 用の設定 . . . . .	113
B.3	pdfTeX 用の処理 . . . . .	117
B.4	X <sub>Y</sub> TeX 用の処理 . . . . .	118
B.5	後処理 (エンジン共通) . . . . .	118
付録 C	和文ドライバ : standard 罨	120
C.1	共通処理 (1) . . . . .	121
C.2	pTeX 用設定 . . . . .	127
C.3	pdfTeX 用設定 : CJK + bxcjkatype . . . . .	131
C.4	X <sub>Y</sub> TeX 用設定 : xeCJK + zxjatype . . . . .	133
C.5	LuaTeX 用設定 : LuaTeX-ja . . . . .	135
C.6	共通処理 (2) . . . . .	137
付録 D	和文ドライバ : modern 罨	138
D.1	フォント設定 . . . . .	138
D.2	fixltx2e 読込 . . . . .	138
D.3	和文カテゴリコード . . . . .	139
D.4	完了 . . . . .	139
付録 E	和文ドライバ : pandoc 罨	139
E.1	dupload システム . . . . .	139
E.2	lang 変数 . . . . .	140
E.3	geometry 変数 . . . . .	141
E.4	CJKmainfont 変数 . . . . .	141
E.5	fixltx2e パッケージ . . . . .	141

E.6	cmap パッケージ . . . . .	141
E.7	microtype パッケージ . . . . .	142
E.8	完了 . . . . .	142
付録 F	補助パッケージ一覧 	142
付録 G	補助パッケージ : bxjscompat 	142
G.1	準備 . . . . .	142
G.2	X <sub>Y</sub> TeX 部分 . . . . .	143
G.3	LuaTeX 部分 . . . . .	144
G.4	完了 . . . . .	145
付録 H	補助パッケージ : bxjscjkat 	145
H.1	準備 . . . . .	145
H.2	和文カテゴリコードの設定 . . . . .	146
H.3	ギリシャ・キリル文字の扱い . . . . .	147
H.4	初期設定 . . . . .	153
H.5	完了 . . . . .	153

## 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である。

```

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

```

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「p<sub>Y</sub>TeX<sub>2</sub><sub>ε</sub> 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原版著者による原版に対する解説である。

---

これは L<sup>A</sup>T<sub>E</sub>X3 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<sub>Y</sub>TeX 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。

```

1 %<*cls>
2 %% このファイルは日本語文字を含みます。

```

`\bxjs@clsname` 文書クラスの名前です。エラーメッセージ表示などで使われます。

```
3 %<article>\def\bxjs@clsname{bxjsarticle}
4 %<book>\def\bxjs@clsname{bxjsbook}
5 %<report>\def\bxjs@clsname{bxjsreport}
6 %<slide>\def\bxjs@clsname{bxjsslide}
```

`\ifjsc@needspatch` [2016-08-22] 従来 `jsclasses` では、`pLATEX` や `LATEX` の不都合な点に対して、クラスファイル内で独自に対策を施していました。しかし、2016 年以降、コミュニティ版 `pLATEX` が次第に対策コードをカーネル内に取り込むようになりました。そこで、新しい `pLATEX` カーネルと衝突しないように、日付が古い場合だけパッチをあてる場合があります。この処理に使用するフラグを定義します。

```
7 \newif\ifjsc@needspatch
8 \jsc@needspatchfalse
```

## ■ BXJS クラス特有の設定

---

長さ値の指定で式を利用可能にするため `calc` を読み込む。

```
9 \RequirePackage{calc}
```

クラスオプションで key-value 形式を使用するため `keyval` を読み込む。

```
10 \RequirePackage{keyval}
```

クラスの本体ではこの他に `geometry` パッケージが読み込まれる。

互換性のための補助パッケージを読み込む。

```
11 \IfFileExists{bxjscompat.sty}{%
12   \let\jsAtEndOfClass@gobble
13   \RequirePackage{bxjscompat}%
14 }{}
```

`\jsDocClass` [トークン] 文書クラスの種別。以下の定値トークンの何れかと同等：`\jsArticle=bxjsarticle`、`\jsBook=bxjsbook`、`\jsReport=bxjsreport`、`\jsSlide=bxjsslide`。

```
15 \let\jsArticle=a
16 \let\jsBook=b
17 \let\jsReport=r
18 \let\jsSlide=s
19 %<article>\let\jsDocClass\jsArticle
20 %<book>\let\jsDocClass\jsBook
21 %<report>\let\jsDocClass\jsReport
22 %<slide>\let\jsDocClass\jsSlide
```

`\jsEngine` [暗黙文字トークン] エンジン (`TEX` の種類) の種別：`j` = `pTEX` 系、`x` = `XYTEX`、`p` = `pdfTEX` (含 `DVI` モード)、`l` = `LuaTEX`、`J` = `NTT jTEX`、`0` = `Omega` 系、`n` = 以上の何れでもない。

```
23 \let\jsEngine=n
24 \def\bxjs@test@engine#1#2{%
25   \edef\bxjs@tmpa{\string#1}%
```

```

26 \edef\bxjs@tmpb{\meaning#1}%
27 \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
28 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
29 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
30 \bxjs@test@engine\Omegaversion{\let\jsEngine=0}
31 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
32 \bxjs@test@engine\pdftexversion{\let\jsEngine=p}
33 \bxjs@test@engine\luatexversion{\let\jsEngine=l}

```

`\ifjsWithupTeX` [スイッチ] エンジンが (内部漢字コードが Unicode の) `upTeX` であるか。

```

34 \newif\ifjsWithupTeX
35 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
36 \jsWithupTeXtrue
37 \fi\fi
38 \let\if@jsc@uplatex\ifjsWithupTeX

```

`\ifjsWithpTeXng` [スイッチ] エンジンが `pTeX-ng` であるか。

```

39 \newif\ifjsWithpTeXng
40 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}

```

`\ifjsWitheTeX` [スイッチ] エンジンが  $\varepsilon$ -`TeX` 拡張をもつか。

```

41 \newif\ifjsWitheTeX
42 \bxjs@test@engine\eTeXversion{\jsWitheTeXtrue}

```

非サポートのエンジンの場合は強制終了させる。

※ NTT `jTeX` と Omega 系。

```

43 \let\bxjs@tmpa\relax
44 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
45 \ifx O\jsEngine \def\bxjs@tmpa{Omega}\fi
46 \ifx\bxjs@tmpa\relax \expandafter\@gobble
47 \else
48 \ClassError\bxjs@clsname
49 {The engine in use (\bxjs@tmpa) is not supported}
50 {It's a fatal error. I'll quit right now.}
51 \expandafter\@firstofone
52 \fi{\endinput\@@end}

```

`\bxjs@protected`  $\varepsilon$ -`TeX` 拡張が有効な場合にのみ `\protected` の効果をもつ。

```

53 \ifjsWitheTeX \let\bxjs@protected\protected
54 \else \let\bxjs@protected\@empty
55 \fi

```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

```

56 \ifjsWitheTeX
57 \def\bxjs@robust@def{\protected\def}
58 \else
59 \def\bxjs@robust@def{\DeclareRobustCommand*}
60 \fi

```

`\ifjsInPdfMode` [スイッチ] pdf<sub>T</sub>E<sub>X</sub> / Lua<sub>T</sub>E<sub>X</sub> が PDF モードで動作しているか。  
 ※ Lua<sub>T</sub>E<sub>X</sub> 0.8x 版でのプリミティブ名変更に対応。

```

61 \newif\ifjsInPdfMode
62 \@nameuse{ImposeOldLuaTeXBehavior}
63 \let\bxjs@tmpa\PackageWarningNoLine
64 \let\PackageWarningNoLine\PackageInfo % suppress warning
65 \RequirePackage{ifpdf}
66 \let\PackageWarningNoLine\bxjs@tmpa
67 \@nameuse{RevokeOldLuaTeXBehavior}
68 \let\ifjsInPdfMode\ifpdf

```

`\bxjs@cond` `\bxjs@cond\ifXXX……\fi{⟨真⟩}{⟨偽⟩}`  
<sub>T</sub>E<sub>X</sub> の if-文 (`\ifXXX……⟨真⟩\else⟨偽⟩\fi`) を末尾呼出形式に変換するためのマクロ。

```

69 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
70   #1\expandafter\@firstoftwo
71   \else\expandafter\@secondoftwo
72   \fi}

```

`\bxjs@cslet` `\bxjs@cslet{⟨名前 1⟩}\制御綴 :`

```

73 \def\bxjs@cslet#1{%
74   \expandafter\let\csname#1\endcsname}

```

`\bxjs@csletcs` `\bxjs@csletcs{⟨名前 1⟩}{⟨名前 2⟩} :`

```

75 \def\bxjs@csletcs#1#2{%
76   \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}

```

`\bxjs@catopt` `\bxjs@catopt{⟨文字列 1⟩}{⟨文字列 2⟩} :` 2 つの文字列を , で繋いだ文字列。ただし片方が空の場合は , を入れない。完全展開可能。

```

77 \def\bxjs@catopt#1#2{%
78   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

```

`\jsAtEndOfClass` このクラスの読込終了時に対するフック。(補助パッケージ中で用いられる。)

```

79 \def\jsAtEndOfClass{%
80   \expandafter\g@addto@macro\csname\bxjs@clsname.cls-h@@k\endcsname}

```

Lua<sub>T</sub>E<sub>X</sub> 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一時的に 11 に変更する。クラス読込終了時点で元に戻される。

※現在の Lua<sub>A</sub>T<sub>E</sub><sub>X</sub> では、漢字のカテゴリコードは最初から 11 になっているので、この処理は特段の意味を持たない。しかし、昔は 12 になっていて、この場合、日本語文字のコントロールワードの命令を使用するには、カテゴリコードを 11 に変更する必要がある。

```

81 \@onlypreamble\bxjs@restore@jltrcc
82 \let\bxjs@restore@jltrcc\@empty
83 \if 1\jsEngine
84 \def\bxjs@change@jltrcc#1{%
85   \xdef\bxjs@restore@jltrcc{%
86     \bxjs@restore@jltrcc
87     \catcode`#1=\the\catcode`#1\relax}%

```

```

88 \catcode`#1=11\relax}
89 \@tfor\bxjs@x:=西暦\do
90 {\expandafter\bxjs@change@jltrcc\bxjs@x}
91 \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。

```

92 \bxjs@robust@def\jsInhibitGlue{%
93 \ifx\inhibitglue\undefined\else \inhibitglue \fi}

```

万が一「2.09 互換モード」になっていた場合は、これ以上進むと危険なので強制終了させる。

```

94 \if@compatibility
95 \ClassError\bxjs@clsname
96 {Something went chaotic!\MessageBreak
97 (How come '\string\documentstyle' is there?)\MessageBreak
98 I cannot go a single step further...}
99 {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
100 then there'll still be hope....}
101 \expandafter\@firstofone
102 \else \expandafter\@gobble
103 \fi{\typeout{Farewell!}}\endinput\@@end}

```

## 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。

```
104 \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。

```
105 \newif\if@titlepage
```

\if@openright \chapter, \part を右ページ起こしにするかどうかです。横組の書籍では真が標準で、要するに片起こし、奇数ページ起こしになります。

```
106 %<book|report>\newif\if@openright
```

\if@openleft [2017-02-24] \chapter, \part を左ページ起こしにするかどうかです。

```
107 %<book|report>\newif\if@openleft
```

\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。

BXJS では report 系でも定義されることに注意。

```
108 %<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。
```

---

JS クラスと異なり、初期値は偽とする。

---

```
109 \newif\if@enablejfam \@enablejfamfalse
```

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  $1\text{m}^2$ ，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  $1.5\text{m}^2$  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， $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$  の `b5paper` は  $250\text{mm} \times 176\text{mm}$  ですが， $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$  の `b5paper` は  $257\text{mm} \times 182\text{mm}$  になっています。ここでは  $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$  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。

```
110 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
111 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
112 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
113 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
114 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
115 \DeclareOption{b4paper}{\bxjs@setpaper{{257trueem}{364trueem}}}
116 \DeclareOption{b5paper}{\bxjs@setpaper{{182trueem}{257trueem}}}
117 \DeclareOption{b6paper}{\bxjs@setpaper{{128trueem}{182trueem}}}
118 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
119 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
120 \DeclareOption{b4j}{\bxjs@setpaper{{257trueem}{364trueem}}}
121 \DeclareOption{b5j}{\bxjs@setpaper{{182trueem}{257trueem}}}
122 \DeclareOption{a4var}{\bxjs@setpaper{{210trueem}{283trueem}}}
123 \DeclareOption{b5var}{\bxjs@setpaper{{182trueem}{230trueem}}}
```

※…`var` を Pandoc で使えるように後ろに `paper` をつけた形を用意する。

```
124 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210trueem}{283trueem}}}
125 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182trueem}{230trueem}}}
126 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
127 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
128 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}
```

---



■横置き 用紙の縦と横の長さを入れ換えます。

```
129 \newif\if@landscape
130 \@landscapefalse
131 \DeclareOption{landscape}{\@landscapetrue}
```

■slide オプション slide を新設しました。

[2016-10-08] slide オプションは article 以外では使い物にならなかったの、簡単のため article のみで使えるオプションとしました。

```
132 \newif\if@slide
```

---

BXJS ではスライド用のクラス `bxjsslide` を用意しているので、本来はこのスイッチは不要なはずである。しかし、JS クラスの一部のコードをそのまま使うために保持している。※この `\if@slide` という制御綴は、ユニークでないにも関わらず、衝突した場合に正常動作が保たれない、という問題を抱えている。

```
133 %<!slide>\@slidefalse
134 %<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。

[2016-07-08] `\mag` を使わずに各種寸法をスケールさせるためのオプション `nomag` を新設しました。`usemag` オプションの指定で従来通りの動作となります。デフォルトは `usemag` です。

[2016-07-24] オプティカルサイズを調整するために NFSS ヘパッチを当てるオプション `nomag*` を新設しました。

---

`\@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。

```
135 \newcommand{\@ptsize}{0}
136 \def\bxjs@param@basefontsize{10pt}
137 \def\jsUnusualPtSize{-20}
```

`\bxjs@setbasefontsize` 基底フォントサイズを実際に変更する。

```
138 \def\bxjs@setbasefontsize#1{%
139   \bxjs@setbasefontlength\@tempdima{#1}%
140   \edef\bxjs@param@basefontsize{\the\@tempdima}%
141   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
142   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
```

```

143 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
144 \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi}

```

`\bxjs@setbasefontlength` `base`、`jbase` で指定される長さ（式）のための特別な `\setlength`。与えられた式が“(実数)Q”の形の場合、Q 単位の長さを代入する（この場合“式”は使えない）。  
 ※クラスオプションのトークン列の中に展開可能なトークンがある場合、 $\text{\LaTeX}$  はファイルの読込の前にそれを展開しようとする。このため、この位置で `\jQ` をサポートすることは原理的に不可能である。

```

145 \def\bxjs@setbasefontlength#1#2{%

```

ここで `true` の長さが使われるのは不合理なので、式が“`true`”を含む場合には警告を出す。

```

146 \bxjs@setbasefontlength@a#2true\@nnil

```

式の末尾が“Q”である時は特別に扱い、それ以外は `\setlength` に移譲する。

```

147 \ifx j\jsEngine \setlength#1{#2}%
148 \else
149 \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
150 \ifx\bxjs@tmpa\relax \setlength#1{#2}%
151 \else \@tempdimc0.25mm #1=\bxjs@tmpa\@tempdimc
152 \fi
153 \fi}
154 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
155 \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
156 \else \def\bxjs@tmpa{#1}%
157 \fi}
158 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
159 \ifx\@nnil#2\@nnil\else
160 \ClassWarningNoLine\bxjs@clsname
161 {You should not use 'true' lengths here}%
162 \fi}

```

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

`\ifjsc@mag@xreal` は「NFSS にパッチを当てるか」を表すスイッチ。

```

163 \newif\ifjsc@mag
164 \newif\ifjsc@mag@xreal
165 %\let\jsc@magscale\@undefined
166 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
167 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
168 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
169 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
170 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
171 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
172 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
173 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
174 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
175 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
176 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
177 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}

```

```

178 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
179 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
180 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
181 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
182 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
183 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
184 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

185 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
186 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
187 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}

```

---

■トンボオプション トンボ (crop marks) を出力します。実際の処理は p<sub>La</sub>T<sub>E</sub>X 2<sub>ε</sub> 本体で行います (plcore.dtx 参照)。オプション `tombow` で日付付きのトンボ、オプション `tombo` で日付なしのトンボを出力します。これらはアスキー版のままです。カウンタ `\hour`, `\minute` は p<sub>La</sub>T<sub>E</sub>X 2<sub>ε</sub> 本体で宣言されています。

---

取りあえず、p<sub>T</sub>E<sub>X</sub> 系の場合に限り、JS クラスのトンボ関連のコードをそのまま活かしておく。正常に動作する保証はない。

---

```

188 \if j\jsEngine
189 \hour\time \divide\hour by 60\relax
190 \@tempcnta\hour \multiply\@tempcnta 60\relax
191 \minute\time \advance\minute-\@tempcnta
192 \DeclareOption{tombow}{%
193   \tombowtrue \tombowdatetrue
194   \setlength{\@tombowwidth}{.1\p}%
195   \@bannertoken{%
196     \jobname\space(\number\year-\two@digits\month-\two@digits\day
197     \space\two@digits\hour:\two@digits\minute)}}%
198   \maketombowbox}
199 \DeclareOption{tombo}{%
200   \tombowtrue \tombowdatefalse
201   \setlength{\@tombowwidth}{.1\p}%
202   \maketombowbox}
203 \fi

```

■面付け オプション `mentuke` で幅ゼロのトンボを出力します。面付けに便利です。これもアスキー版のままです。

```

204 \if j\jsEngine
205 \DeclareOption{mentuke}{%
206   \tombowtrue \tombowdatefalse
207   \setlength{\@tombowwidth}{\z@}%
208   \maketombowbox}
209 \fi

```

■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。

```
210 \DeclareOption{oneside}{\@twosidefalse \mparswitchfalse}
211 \DeclareOption{twoside}{\@twosidetrue \mparswitchtrue}
212 \DeclareOption{vartwoside}{\@twosidetrue \mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
213 \DeclareOption{onecolumn}{\@twocolumnfalse}
214 \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。

```
215 \DeclareOption{titlepage}{\@titlepagetrue}
216 \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、横組ではこれを `openright` と表すことにしてあります。 `openany` で偶数ページからでも始まるようになります。

[2017-02-24] `openright` は横組では奇数ページ起こし、縦組では偶数ページ起こしを表します。ややこしいですが、これは  $\text{\LaTeX}$  の標準クラスが西欧の横組事情しか考慮せずに、奇数ページ起こしと右起こしを一緒にしてしまったせいです。縦組での奇数ページ起こしと横組での偶数ページ起こしも表現したいので、`jsclasses` では新たに `openleft` も追加しました。

```
217 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
218 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
219 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}
```

■`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`  $\text{\LaTeX}$  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。

```
220 \def\eqnarray{%
221   \stepcounter{equation}%
222   \def\@currentlabel{\p@equation\theequation}%
223   \global\@eqnswtrue
224   \m@th
225   \global\@eqcnt\z@
226   \tabskip\@centering
227   \let\\\@eqnocr
228   $$$\everycr{}\halign to\displaywidth\bgroup
229     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
230     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
231     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
232     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
233     \tabskip\z@skip
234     \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。

```

235 \DeclareOption{leqno}{\input{leqno.clo}}
236 \DeclareOption{fleqn}{\input{fleqn.clo}}
237 % fleqn 用の eqnarray 環境の再定義
238 \def\eqnarray{%
239   \stepcounter{equation}%
240   \def\@currentlabel{\p@equation\theequation}%
241   \global\@eqnswtrue\m@th
242   \global\@eqcnt\z@
243   \tabskip\mathindent
244   \let\=\@eqnocr
245   \setlength\abovedisplayskip{\topsep}%
246   \ifvmode
247     \addtolength\abovedisplayskip{\partopsep}%
248   \fi
249   \addtolength\abovedisplayskip{\parskip}%
250   \setlength\belowdisplayskip{\abovedisplayskip}%
251   \setlength\belowdisplayshortskip{\abovedisplayskip}%
252   \setlength\abovedisplayshortskip{\abovedisplayskip}%
253   $$\everycr{}\halign to\linewidth% $$
254   \bgroup
255     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
256     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
257     &\global\@eqcnt\tw@
258     $\displaystyle{##}$\hfil \tabskip\@centering
259     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
260   \tabskip\z@skip\cr
261 }

```

■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。

```

262 % \DeclareOption{openbib}{%
263 %   \AtEndOfPackage{%
264 %     \renewcommand\@openbib@code{%
265 %       \advance\leftmargin\bibindent
266 %       \itemindent -\bibindent
267 %       \listparindent \itemindent
268 %       \parsep \z@}%
269 %     \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 数式中では 16 通りのフォントしか使えません。AMSFonTS や mathptmx パッケージを使って数式フォントをたくさん使うと “Too many math alphabets ...” というエラーが起こってしまいます。`disablejfam` オプションを付ければ、明朝・ゴシックを数式用フォントとして登録するのをやめますので、数式用フォントが二つ節約できます。いずれにしても `\textmc` や `\mbox` や `amsmath` パッケージの `\text` を使えば数式中で和文フォントが使えますので、この新ドキュメントクラ

スでは標準で和文フォントを数式用に登録しないことにしていたのですが、従来のドキュメントクラスの仕様に合わせることにしました。

---

`\bxjs@enablejfam` [暗黙文字トークン] `enablejfam` オプションの状態 :

```
270 %\let\bxjs@enablejfam\undefined
```

`enablejfam` オプションの処理。

```
271 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
272 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
273 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
274 \define@key{bxjs}{enablejfam}[true]{%
275   \bxjs@set@keyval{enablejfam}{#1}{}}
```

JS クラスとの互換のため `disablejfam` オプションを定義する。

```
276 \DeclareOption{disablejfam}{\let\bxjs@enablejfam=f}
```

※実際に何らかの設定を行うのは和文ドライバである。和文ドライバとエンジンの組合せにより、`enablejfam` が `default` である場合に「数式和文ファミリ」が有効と無効の選択は異なるし、またそもそも有効と無効の一方しか選択できない場合もある。

---

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

[2016-07-13] `\ifdraft` を定義するのをやめました。

---

`\ifjsDraft` JS クラスは `\ifdraft` という公開名のスイッチを用いているが、これは `ifdraft` パッケージと衝突するので、代わりに `\ifjsDraft` の名前を用い、本文開始時に `\ifdraft` が未定義の場合に限り、`\ifjsDraft` を `\ifdraft` にコピーする処理にする。

※ JS クラスの `\ifdraft` が廃止されたので、`\ifdraft` は 2.0 版で廃止を予定する。

```
277 \let\ifjsDraft\iffalse
278 \@onlypreamble\bxjs@draft
279 \def\bxjs@draft#1{%
280   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
281 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
282 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
283 \AtBeginDocument{%
284   \expandafter\ifx\csname ifdraft\endcsname\relax
285     \expandafter\let\csname ifdraft\expandafter\endcsname
286     \csname ifjsDraft\endcsname
287   \fi}
```

---

■和文フォントメトリックの選択 ここでは新しい `jis` フォントメトリックを標準で使いますが、古い `min10`, `goth10` などを使いたいときは `mingoth` というオプションを指定します。また、`winjis` オプションで `winjis` メトリックが使えます。

---

BXJS クラスではここは和文ドライバの管轄。

---

■papersize スペシャルの利用 dvips や dviout で用紙設定を自動化するにはオプション papersize を与えます。

---

BXJS クラスでは geometry パッケージがこの処理を行う。

`\ifbxjs@papersize` [スイッチ] papersize スペシャルを出力するか。既定で有効であるが、nopapersize オプションで無効にできる。

※ JS クラスでは `\ifpapersize` という制御綴だが、これは採用しない。

```
288 \newif\ifbxjs@papersize
289 \bxjs@papersizetrue
290 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
291 \DeclareOption{papersize}{\bxjs@papersizetrue}
```

---

■英語化 オプション english を新設しました。

```
292 \newif\if@english
293 \@englishfalse
294 \DeclareOption{english}{\@englishttrue}
```

■jsbook を jsreport もどきに オプション report を新設しました。

[2017-02-13] 従来は「jsreport 相当」を jsbook の report オプションで提供していましたが、新しく jsreport クラスも作りました。どちらでも好きな方を使ってください。

---

BXJS では当初から bxjsreport クラスが用意されている。

---


■jslogo パッケージの読み込み L<sup>A</sup>T<sub>E</sub>X 関連のロゴを再定義する jslogo パッケージを読み込まないオプション nojslogo を新設しました。jslogo オプションの指定で従来どおりの動作となります。デフォルトは jslogo で、すなわちパッケージを読み込みます。

---

BXJS クラスでは、nojslogo を既定とする。

---

```
295 \newif\if@jslogo \@jslogofalse
296 \DeclareOption{jslogo}{\@jslogotrue}
297 \DeclareOption{nojslogo}{\@jslogofalse}
```

■BXJS 特有のオプションの一覧 

---

- エンジンオプション：xelatex 等。

- ドライバオプション：dvipdfmx 等。
- 複合設定オプション：pandoc 等。
- nopapersize：papersize（既定で有効）の否定。
- zw / nozw：\jsZw と等価な命令として \zw を定義する／しない。
- js / nojs：JS クラスを読込済として扱う／扱わない。
- precisetext / noprecisetext：X<sub>Y</sub>TeX の “generateactualtext” を有効／無効にする。
- simplejasetup / nosimplejasetup：X<sub>Y</sub>TeX の “linebreaklocale” を有効／無効にする。
- bigcode / nobigcode：upTeX で CMap として UTF8-UCS2 の代わりに UTF8-UTF16 を使う／使わない。
- oldfontcommands / nooldfontcommands：古い “二文字フォント命令” に対する警告を抑止する／しない。
- 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。）
- jbase=*<dimen>*：基底フォントサイズを “和文規準で” 直接指定する。
- 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)。
- noscale：scale=1 と等価。
- 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。
- ja=*<name>*：使用する和文ドライバの指定。
- jafont=*<name>*：和文フォントプリセットの指定。
- japaram=*<name>*：和文フォントパラメタの指定。
- magstyle=*<name>*：“版面拡大” の実現方法の選択。
- dvi=*<name>*：DVI モードの時のみに参照されるドライバ指定。
- geometry=*<{class | user}>*：geometry パッケージの読込を自動的に行うかユーザに任せるか。
- fancyhdr=*<bool>*：fancyhdr パッケージ用の調整を行うか。
- layout=*<name>*：レイアウト変種の指定。
- textwidth-limit=*<number>*：bxjsbook における、\textwidth の上限の全角単位での値。
- paragraph-mark=*<char>*：パラグラフのマーク。
- whole-zw-lines=*<bool>*：行長を全角単位に丸めるか。
- hyperref-enc=*<bool>*：hyperref の文字コード指定補正を行うか。

\bxjs@invscale \bxjs@invscale は T<sub>E</sub>X 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。



※局所化の `\begingroup~\endgroup` について、以前は `\group~\egroup` を使っていたが、これだと数式モード中では空のサブ数式を生み出してしまうため修正した。

※元の長さが 128 pt 以上の場合でも動作するように修正した。

```

298 \mathchardef\bxjs@isc@ll=128
299 \mathchardef\bxjs@isc@sl=259
300 \def\bxjs@isc@sl@h{65539 }
301 \def\bxjs@invscale#1#2{%
302   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
303     \ifdim\@tempdima<\bxjs@isc@ll\p@
304       \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
305       \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
306     \else
307       \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
308       \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
309     \fi
310     \@tempcntb\p@ \divide\@tempcntb\@tempdimb
311     \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
312     \@tempdimb\@tempcnta\@ne
313     \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
314     \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
315     \@whiledim\@tempdimb<\@tempdimc\do{%
316       \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
317       \advance\@tempcntb\@ne \divide\@tempcntb\@tw@
318       \ifdim #2\@tempcntb>\@tempdima
319         \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
320       \else \@tempdimb=\@tempcntb\@ne \fi}%
321     \xdef\bxjs@gtmpa{\the\@tempdimb}%
322   \endgroup #1=\bxjs@gtmpa\relax}

```

---

## ■ 複合設定オプション

---

複合設定オプションとは、「エンジンやドライバや和文ドライバの設定を含む、複数の設定を一度に行うオプション」のことである。ある特定の設定を短く書く必要が高いと判断される場合に用意される。

`\bxjs@composite@proc` 複合設定オプションのための遅延処理マクロ。

```

323 \let\bxjs@composite@proc\relax

```

`pandoc` オプションは、Pandoc で L<sup>A</sup>T<sub>E</sub>X 用の既定テンプレートを用いて他形式から L<sup>A</sup>T<sub>E</sub>X (および PDF) 形式に変換する用途に最適化した設定を与える。

```

324 \DeclareOption{pandoc}{%

```

和文ドライバを `pandoc` に、エンジン指定を `autodetect-engine` に変更する。

※実際の和文ドライバ・エンジン設定より優先される。

```

325   \def\bxjs@composite@proc{%
326     \bxjs@oldfontcommandstrue

```

```

327 \setkeys{bxjs}{ja=pandoc}%
328 \let\bxjs@engine@given=*\%

```

ドライバオプションを dvi=dvipdfmx 相当に変更する。

※これは実際のドライバ設定で上書きできる（オプション宣言順に注意）。

**TODO:** できない気がする…。

```

329 \def\bxjs@driver@opt{dvipdfmx}%
330 \bxjs@dvi@opttrue}

```

---

## ■エンジン・ドライバオプション

---

`\bxjs@engine@given` オプションで明示されたエンジンの種別。

```

331 %\let\bxjs@engine@given\undefined

```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```

332 %\let\bxjs@engine@opt\undefined

```

エンジン明示指定のオプションの処理。

※ 0.9pre 版の暫定仕様と異なり、エンジン名は `...latex` に限定する。`xetex` や `pdftex` は一般的な L<sup>A</sup>T<sub>E</sub>X の慣習に従って「ドライバの指定」とみなすべきだから。

```

333 \DeclareOption{autodetect-engine}{%
334 \let\bxjs@engine@given=*}
335 \DeclareOption{latex}{%
336 \def\bxjs@engine@opt{latex}%
337 \let\bxjs@engine@given=n}
338 \DeclareOption{platex}{%
339 \def\bxjs@engine@opt{platex}%
340 \let\bxjs@engine@given=j}
341 \DeclareOption{uplatex}{%
342 \def\bxjs@engine@opt{uplatex}%
343 \let\bxjs@engine@given=u}
344 \DeclareOption{xelatex}{%
345 \def\bxjs@engine@opt{xelatex}%
346 \let\bxjs@engine@given=x}
347 \DeclareOption{pdflatex}{%
348 \def\bxjs@engine@opt{pdflatex}%
349 \let\bxjs@engine@given=p}
350 \DeclareOption{lualatex}{%
351 \def\bxjs@engine@opt{lualatex}%
352 \let\bxjs@engine@given=l}
353 \DeclareOption{platex-ng}{%
354 \def\bxjs@engine@opt{platex-ng}%
355 \let\bxjs@engine@given=g}
356 \DeclareOption{platex-ng*}{%
357 \def\bxjs@engine@opt{platex-ng*}%
358 \let\bxjs@platexng@nodrv=t%

```

```
359 \let\bxjs@engine@given=g}
```

`\bxjs@driver@given` オプションで明示されたドライバの種別。

```
360 %\let\bxjs@driver@given\undefined
361 \let\bxjs@driver@@dvimode=0
362 \let\bxjs@driver@@dvipdfmx=1
363 \let\bxjs@driver@@pdfmode=2
364 \let\bxjs@driver@@xetex=3
365 \let\bxjs@driver@@dvips=4
366 \let\bxjs@driver@@none=5
```

`\bxjs@driver@opt` 明示された「ドライバ指定」のオプション名。

```
367 %\let\bxjs@driver@opt\undefined

368 \DeclareOption{dvips}{%
369   \def\bxjs@driver@opt{dvips}%
370   \let\bxjs@driver@given\bxjs@driver@@dvips}
371 \DeclareOption{dviout}{%
372   \def\bxjs@driver@opt{dviout}%
373   \let\bxjs@driver@given\bxjs@driver@@dvimode}
374 \DeclareOption{xdvi}{%
375   \def\bxjs@driver@opt{xdvi}%
376   \let\bxjs@driver@given\bxjs@driver@@dvimode}
377 \DeclareOption{dvipdfmx}{%
378   \def\bxjs@driver@opt{dvipdfmx}%
379   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
380 \DeclareOption{nodvidriver}{%
381   \def\bxjs@driver@opt{nodvidriver}%
382   \let\bxjs@driver@given\bxjs@driver@@none}
383 \DeclareOption{pdftex}{%
384   \def\bxjs@driver@opt{pdftex}%
385   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
386 \DeclareOption{luatex}{%
387   \def\bxjs@driver@opt{luatex}%
388   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
389 \DeclareOption{xetex}{%
390   \def\bxjs@driver@opt{xetex}%
391   \let\bxjs@driver@given\bxjs@driver@@xetex}
```

「もし DVI モードであればドライバを dvipdfmx にする」というオプション。

※ 1.2 版で dvi オプションが新設されたが、互換性のためこのオプションも残す。

```
392 \DeclareOption{dvipdfmx-if-dvi}{%
393   \setkeys{bxjs}{dvi=dvipdfmx}}
```

---

## ■その他の BXJS 独自オプション

---

`\ifbxjs@usezw` `\jsZw` の同義語として `\zw` を使えるようにするか。既定は真。

```

394 \newif\ifbxjs@usezw \bxjs@usezwtrue

    zw、nozw オプションの定義。
395 \DeclareOption{nozw}{%
396   \bxjs@usezwfalse}
397 \DeclareOption{zw}{%
398   \bxjs@usezwtrue}

\ifbxjs@disguise@js JS クラスの派生クラスのふりをするか。既定は真。
399 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue

    nojs、js オプションの定義。
400 \DeclareOption{nojs}{%
401   \bxjs@disguise@jsfalse}
402 \DeclareOption{js}{%
403   \bxjs@disguise@jstrue}

\ifbxjs@precisetext XeLaTeX の “generateactualtext” を有効にするか。既定は偽。
404 \newif\ifbxjs@precisetext

    noprecisetext / precisetext オプションの定義。
405 \DeclareOption{noprecisetext}{%
406   \bxjs@precisetextfalse}
407 \DeclareOption{precisetext}{%
408   \bxjs@precisetexttrue}

\ifbxjs@simplejasetup XeLaTeX の “linebreaklocale” を有効にするか。既定は真（であるが多くの場合は後に無効化される）。
409 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

    nosimplejasetup / simplejasetup オプションの定義。
410 \DeclareOption{nosimplejasetup}{%
411   \bxjs@simplejasetupfalse}
412 \DeclareOption{simplejasetup}{%
413   \bxjs@simplejasetuptrue}

\ifbxjs@bigcode upTeX で有効化する ToUnicode CMap として「UTF8-UCS2」の代わりに「UTF8-UTF16」を使うか。BMP 外の文字に対応できる「UTF8-UTF16」の方が望ましいのであるが、このファイルが利用可能かの確実な判定が困難であるため、オプションで指定することとする。
414 \newif\ifbxjs@bigcode \bxjs@bigcodefalse

    その上で、「TeX 環境がある程度新しければ利用可能であろう」と判断し bxjs@bigcode の既定値を真とする。具体的な判断基準として、「TeX のバージョンが 3.14159265（2014 年 1 月）以上であるか」を採用する。
415 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
416 \def\bxjs@tmpb#1 #2#3\@nil{%
417   \ifx1#2\bxjs@bigcodetrue \fi}
418 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil

```

nobigcode / bigcode オプションの定義。

```
419 \DeclareOption{nobigcode}{%
420   \bxjs@bigcodefalse}
421 \DeclareOption{bigcode}{%
422   \bxjs@bigcodetrue}
```

\ifbxjs@oldfontcommands \allowoldfontcommands を既定で有効にするか。

```
423 \newif\ifbxjs@oldfontcommands
```

nooldfontcommands、oldfontcommands オプションの定義。

※ oldfontcommands オプションの名前は memoir クラスに倣った。ちなみに KOMA-Script では enabledeprecatedfontcommands であるがこれはチョットアレなので避けた。

```
424 \DeclareOption{nooldfontcommands}{%
425   \bxjs@oldfontcommandsfalse}
426 \DeclareOption{oldfontcommands}{%
427   \bxjs@oldfontcommandstrue}
```

---

## ■keyval 型のオプション

---

```
428 \def\bxjs@setkey{%
429   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
430 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
431 \DeclareOption*{\bxjs@setkey}
```

\bxjs@safe@setkeys 未知のキーに対してエラー無しで無視する \setkeys。

```
432 \def\bxjs@safe@setkeys#1#2{%
433   \let\bxjs@KV@errx\KV@errx
434   \let\KV@errx\bxjs@safe@setkeys@a
435   \setkeys{#1}{#2}%
436   \let\KV@errx\bxjs@KV@errx}
437 \def\bxjs@safe@setkeys@a#1{}
```

\bxjs@set@keyval \bxjs@set@keyval{<key>}{<value>}{<error>}

\bxjs@kv@<key>@<value> が定義済ならそれを実行し、未定義ならエラーを出す。

```
438 \def\bxjs@set@keyval#1#2#3{%
439   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
440   \ifx\bxjs@next\relax
441     \bxjs@error@keyval{#1}{#2}%
442     #3%
443   \else \bxjs@next
444   \fi}
445 \@onlypreamble\bxjs@error@keyval
446 \def\bxjs@error@keyval#1#2{%
447   \ClassError\bxjs@clsname
448   {Invalid value '#2' for option #1}\@ehc}
```

\ifbxjs@scaleset 和文スケール値が指定されたか。

```

449 \newif\ifbxjs@scaleset

\jsScale 〔実数値マクロ〕 和文スケール値。
450 \def\jsScale{0.924715}

    base オプションの処理。
451 \define@key{bxjs}{base}{\bxjs@setbasefontsize{#1}}

    jbase オプションの処理。ここでは \jsScale の値を使用する。scale の処理との順序
    依存を消すため、jbase の処理の実行を遅延させている。
452 \@onlypreamble\bxjs@do@opt@jbase
453 \let\bxjs@do@opt@jbase\relax
454 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
455 \def\bxjs@setjbasefontsize#1{%
456     \def\bxjs@do@opt@jbase{%
457         \bxjs@setbasefontlength\@tempdima{#1}%
458         \bxjs@invscale\@tempdima\jsScale
459         \bxjs@setbasefontsize{\@tempdima}}

    scale オプションの処理。
460 \define@key{bxjs}{scale}{%
461     \bxjs@scalesettrue
462     \edef\jsScale{#1}}

    noscale オプションの処理。
463 \DeclareOption{noscale}{%
464     \bxjs@scalesettrue
465     \def\jsScale{1}}

\bxjs@param@mag mag オプションの値。
466 \let\bxjs@param@mag\relax

    mag オプションの処理。
467 \define@key{bxjs}{mag}{\edef\bxjs@param@mag{#1}}

    paper オプションの処理。
468 \define@key{bxjs}{paper}{\edef\bxjs@param@paper{#1}}

\bxjs@jadriver 和文ドライバの名前。
469 \let\bxjs@jadriver\relax
470 %\let\bxjs@jadriver@given\@undefined

    ja オプションの処理。
    ※ jadriver は 0.9 版で用いられた旧称。
    ※単なる ja という指定は無視される（Pandoc 対策）。
471 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
472 \define@key{bxjs}{ja}{\relax}{%
473     \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}

\jsJaFont 和文フォント設定の名前。
474 \let\jsJaFont\@empty

```

jafont オプションの処理。

```

475 \define@key{bxjs}{jafont}{\edef\jsJaFont{#1}}

```

\jsJaParam 和文ドライバパラメタの文字列。

```

476 \let\jsJaParam\@empty

```

japaram オプションの処理。

```

477 \define@key{bxjs}{japaram}{\edef\jsJaParam{#1}}

```

\bxjs@magstyle magstyle 設定値。(古いイマイチな名前。)

```

478 \let\bxjs@magstyle@mag=m
479 \let\bxjs@magstyle@real=r
480 \let\bxjs@magstyle@xreal=x

```

(新しい素敵な名前。)

※ただし制御綴としては、\*付の名前は扱い難いので、\bxjs@magstyle@xreal の方を優先させる。

```

481 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
482 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
483 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal

```

\bxjs@magstyle@default は既定の値を表す。

```

484 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
485 \ifx l\jsEngine \ifnum\luatexversion>86
486   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
487 \fi\fi
488 \ifjsWithpTeXng
489   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
490 \fi
491 \let\bxjs@magstyle\bxjs@magstyle@default

```

magstyle オプションの処理。

```

492 \define@key{bxjs}{magstyle}{%
493   \expandafter\let\expandafter\bxjs@magstyle\csname
494     bxjs@magstyle@#1\endcsname
495   \ifx\bxjs@magstyle\relax
496     \ClassError{bxjs}{clsname
497       {Invalid value '#1' for option magstyle}}{\@ehc
498     \let\bxjs@magstyle\bxjs@magstyle@default
499   \fi}

```

\bxjs@geometry geometry オプションの値。

```

500 \let\bxjs@geometry@class=c
501 \let\bxjs@geometry@user=u
502 \let\bxjs@geometry\bxjs@geometry@class

```

geometry オプションの処理。

```

503 \define@key{bxjs}{geometry}{%
504   \expandafter\let\expandafter\bxjs@geometry\csname

```

```

505     bxjs@geometry@#1\endcsname
506     \ifx\bxjs@geometry\relax
507         \ClassError\bxjs@clsname
508         {Invalid value '#1' for option geometry}\@ehc
509     \let\bxjs@geometry\bxjs@geometry@class
510     \fi}

```

\ifbxjs@fancyhdr [スイッチ] fancyhdr パッケージに対する調整を行うか。

```

511 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

    fancyhdr オプションの処理。
512 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
513 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
514 \define@key{bxjs}{fancyhdr}[true]{%
515     \bxjs@set@keyval{fancyhdr}{#1}{}}

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

516 \newif\ifbxjs@dvi@opt

    DVI モードのドライバとドライバ種別との対応。
517 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
518 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
519 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
520 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
521 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

    dvi オプションの処理。
522 \define@key{bxjs}{dvi}{%
523     \expandafter\let\expandafter\bxjs@tmpa\csname
524     bxjs@dvidriver@@#1\endcsname
525     \ifx\bxjs@tmpa\relax
526         \ClassError\bxjs@clsname
527         {Invalid value '#1' for option dvi}\@ehc
528     \else

        \bxjs@driver@given を未定義にしていることに注意。
529         \def\bxjs@driver@opt{#1}%
530         \let\bxjs@driver@given\undefined
531         \bxjs@dvi@opttrue
532     \fi}

```

\ifbxjs@layout@buggyhmargin [スイッチ] bxjsbook の左右マージンがアレか。

※ layout が v1 の場合はアレになる。

```

533 \newif\ifbxjs@layout@buggyhmargin

```

\ifbxjs@force@chapterabstract [スイッチ] abstract 環境を chapterabstract にするか。

※ bxjsbook では常に真。bxjsreport では layout が v1 の場合に真になる。

```

534 \newif\ifbxjs@force@chapterabstract
535 %<book>\bxjs@force@chapterabstracttrue

```



layout オプションの処理。

```
536 \@namedef{bxjs@kv@layout@v1}{%
537 %<book>\bxjs@layout@buggyhmargintrue
538 %<report>\bxjs@force@chapterabstracttrue
539 }
540 \@namedef{bxjs@kv@layout@v2}{%
541 %<book>\bxjs@layout@buggyhmarginfalse
542 %<report>\bxjs@force@chapterabstractfalse
543 }
544 \define@key{bxjs}{layout}{%
545   \bxjs@set@keyval{layout}{#1}{}}
```

`\bxjs@textwidth@limit` `textwidth-limit` の指定値。`\textwidth` の上限。

```
546 %\let\bxjs@textwidth@limit\undefined
547 \define@key{bxjs}{textwidth-limit}{%
548   \edef\bxjs@textwidth@limit{#1}}
```

`\bxjs@paragraph@mark` `paragraph-mark` の指定値。パラグラフのマーク。

```
549 %\let\bxjs@paragraph@mark\undefined
550 \define@key{bxjs}{paragraph-mark}{%
551   \edef\bxjs@paragraph@mark{#1}}
```

`\ifbxjs@whole@zw@lines` [スイッチ] `whole-zw-lines` の指定値。

```
552 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
553 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
554 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
555 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}
```

`\ifbxjs@xkanjiskip@cmd` [スイッチ] `xkanjiskip-cmd` の指定値。

```
556 \newif\ifbxjs@xkanjiskip@cmd \bxjs@xkanjiskip@cmdtrue
557 \let\bxjs@kv@xkanjiskipcmd@true\bxjs@xkanjiskip@cmdtrue
558 \let\bxjs@kv@xkanjiskipcmd@false\bxjs@xkanjiskip@cmdfalse
559 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{xkanjiskipcmd}{#1}{}}
```

`\ifbxjs@hyperref@enc` [スイッチ] `hyperref-enc` の指定値。

```
560 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
561 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
562 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
563 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}
```

---

## ■オプションの実行

---

L<sup>A</sup>T<sub>E</sub>X の実装では、クラスやパッケージのオプションのトークン列の中に { } が含まれると正常に処理ができない。これに対処する為 `\@removeelement` の実装に少し手を加える(仕様は変わらない)。

※クラスに `\DeclareOption*` がある場合は `\@unusedoptions` は常に空のままであることを利用している。

```
564 \let\bxjs@ltx@removeelement\@removeelement
565 \def\@removeelement#1#2#3{%
566   \def\reserved@a{#2}%
567   \ifx\reserved@a\@empty \let#3\@empty
568   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
569   \fi}
```

---

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。

```
570 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
571 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
572 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
573 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
574 \ProcessOptions\relax
575 \bxjs@composite@proc
```

後処理

```
576 \if@slide
577   \def\maybeblue{\@ifundefined{ver@color.sty}{}{\color{blue}}}
578 \fi
579 \if@landscape
580   \setlength\@tempdima {\paperheight}
581   \setlength\paperheight{\paperwidth}
582   \setlength\paperwidth {\@tempdima}
583 \fi
```

---

## ■ グローバルオプションの整理

グローバルオプションのトークン列に `{ }` が含まれていると、やはり後のパッケージの読込処理で不具合を起こすようである (`\ProcessOptions*` がエラーになる)。従って、このようなオプションは除外することにする。

```
584 \@onlypreamble\bxjs@purge@brace@elts
585 \def\bxjs@purge@brace@elts{%
586   \def\bxjs@tmpa{\@gobble}%
587   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
588   \let\@classoptionslist\bxjs@tmpa}
589 \@onlypreamble\bxjs@purge@be@a
590 \def\bxjs@purge@be@a#1,{%
591   \ifx\@nil#1\relax\else
592     \bxjs@purge@be@b#1{}\@nil
593     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
594     \expandafter\bxjs@purge@be@a
595   \fi}
596 \@onlypreamble\bxjs@purge@be@b
```

```

597 \def\bxjs@purge@be@b#1#\bxjs@purge@be@c}
598 \@onlypreamble\bxjs@purge@be@c
599 \def\bxjs@purge@be@c#1\@nil{%
600   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
601 \bxjs@purge@brace@elts

```

`papersize`、`10pt`、`noscale` の各オプションは他のパッケージと衝突を起こす可能性があるため、グローバルオプションから外す。

```

602 \@expandtwoargs\@removeelement
603   {papersize}\@classoptionslist\@classoptionslist
604 \@expandtwoargs\@removeelement
605   {10pt}\@classoptionslist\@classoptionslist
606 \@expandtwoargs\@removeelement
607   {noscale}\@classoptionslist\@classoptionslist

```

---

■使用エンジンの検査・自動判定 ユーザが `uplatex` オプションの有無により指定したエンジンが、実際に使われているものと一致しているかを検査し、一致しない場合はエラーメッセージを表示します。

[2016-11-09] `pLATEX` / `upLATEX` を自動判別するオプション `autodetect-engine` を新設しました。`upLATEX` の場合は、グローバルオプションに `uplatex` を追加することで、自動判定に応じて `otf` パッケージにも `uplatex` オプションが渡るようにします。

[2016-11-11] `pLATEX` の場合は、オプション `uplatex` が指定されていれば必ずエラーを出します。`autodetect-engine` が有効になっていてもエラーを出しますが、これは `otf` パッケージに `uplatex` オプションが渡ってしまうのを防ぐためです。

---

現在の（正規化前の）和文ドライバの値を `\bxjs@jadriver@given` に保存する。

```

608 \ifx\bxjs@jadriver\relax\else
609   \let\bxjs@jadriver@given\bxjs@jadriver
610 \fi

```

エンジン明示指定のオプションが与えられた場合は、それが実際のエンジンと一致するかを検査する。

```

611 \let\bxjs@tmpb\jsEngine
612 \ifx j\bxjs@tmpb\ifjsWithpTeXng
613   \let\bxjs@tmpb=g
614 \fi\fi
615 \ifx j\bxjs@tmpb\ifjsWithupTeX
616   \let\bxjs@tmpb=u
617 \fi\fi
618 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
619   \let\bxjs@tmpb=n
620 \fi\fi

```

（この時点で `\bxjs@tmpb` は `\bxjs@engine@given` と同じ規則で分類したコードをもって  
いる。）

```

621 \ifx *\bxjs@engine@given
622   \let\bxjs@engine@given\bxjs@tmpb

```

エンジン指定が `autodetect-engine` であり、かつ実際のエンジンが (u)pL<sup>A</sup>T<sub>E</sub>X だった場合は、本来のエンジンオプションをグローバルオプションに加える。

```

623 \ifx j\bxjs@engine@given
624   \g@addto@macro\@classoptionslist{,latex}
625 \else\ifx u\bxjs@engine@given
626   \g@addto@macro\@classoptionslist{,uplatex}
627 \fi\fi
628 \fi
629 \ifx\bxjs@engine@given\@undefined\else
630   \ifx\bxjs@engine@given\bxjs@tmpb\else
631     \ClassError\bxjs@clsname
632       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
633   \fi
634 \fi


```

エンジンが pT<sub>E</sub>X-ng の場合、グローバルオプションに `uplatex` を追加する。

```

635 \ifjsWithpTeXng
636   \g@addto@macro\@classoptionslist{,uplatex}
637 \fi

```

■ **ドライバ指定**  ドライバ指定のオプションが与えられた場合は、それがエンジンと整合するかを検査する。

```

638 \@tempswatrue
639 \ifx \bxjs@driver@given\@undefined\else
640   \ifjsInPdfMode
641     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
642       \@tempswafalse
643     \fi
644   \else\ifx x\jsEngine
645     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
646       \@tempswafalse
647     \fi
648   \else
649     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
650       \@tempswafalse
651     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
652       \@tempswafalse
653     \fi\fi
654   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
655     \@tempswafalse
656   \fi\fi
657 \fi\fi
658 \fi
659 \if@tempswa\else
660   \ClassError\bxjs@clsname
661     {Option '\bxjs@driver@opt' used on wrong engine}\@ehc

```

662 \fi

DVI 出力のエンジンである場合の追加処理。

```
663 \ifjsInPdfMode \@tempswafalse
664 \else\ifx x\jsEngine \@tempswafalse
665 \else\ifjsWithpTeXng \@tempswafalse
666 \else \@tempwattrue
667 \fi\fi\fi
668 \if@tempswa
```

ドライバオプションがない場合は警告を出す。

※ただし ja 非指定の場合はスキップする (0.3 版との互換性のため)。

```
669 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
670 \ClassWarningNoLine\bxjs@clsname
671 {No driver option is given}
672 \fi\fi
```

dvi=XXX が指定されていた場合は、XXX が指定された時と同じ動作にする。(グローバルオプションに XXX を追加する。)

```
673 \ifbxjs@dvi@opt
674 \edef\bxjs@nxt{%
675 \let\noexpand\bxjs@driver@given
676 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
677 \noexpand\g@addto@macro\noexpand\@classoptionslist
678 {,\bxjs@driver@opt}%
679 }\bxjs@nxt
680 \fi
681 \fi
```

エンジンが pTeX-ng の場合、グローバルオプションに dvipdfmx を追加する。ただし、エンジンオプションが platex-ng\* (\*付) の場合、および既に dvipdfmx が指定されている場合を除く。

```
682 \ifjsWithpTeXng
683 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
684 \let\bxjs@platexng@nodrv\@undefined
685 \else\ifx t\bxjs@platexng@nodrv\else
686 \g@addto@macro\@classoptionslist{,dvipdfmx}
687 \fi\fi
688 \fi
```

ドライバが nodvidriver であった場合の処理。DVI ウェア依存の処理を全て無効化する。

```
689 \ifx\bxjs@driver@given\bxjs@driver@@none
690 \bxjs@papersizefalse
691 \fi
```

■その他の BXJS 特有の後処理 ☞ \bxjs@jadriver の正規化。値が未指定の場合は minimal に変える。ただしエンジンが (u)pTeX である場合は standard に変える。

```
692 \def\bxjs@@minimal{minimal}
693 \ifx\bxjs@jadriver\relax
```

```

694 \ifx j\jsEngine
695   \def\bxjs@jadriver{standard}
696 \else
697   \let\bxjs@jadriver\bxjs@@minimal
698 \fi
699 \fi

```

エンジンオプションがない場合はエラーを出す。

※ただし `ja` 非指定の場合はスキップする。

```

700 \ifx\bxjs@jadriver@given\@undefined\else
701 \ifx\bxjs@engine@given\@undefined
702   \ClassError\bxjs@clsname
703     {An engine option must be explicitly given}%
704     {When you use a Japanese-driver you must specify a correct\MessageBreak
705       engine option.\MessageBreak\@ehc}
706 \fi\fi

```

新しい Lua<sub>T</sub><sub>E</sub><sub>X</sub> (0.87 版以降) では `mag` がアレなので、`magstyle=usemag` が指定されていた場合はエラーを出す。(この場合の既定値は `nomag*` であり、エラーの場合は既定値に置き換えられる。)

```

707 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
708 \ifx\bxjs@magstyle\bxjs@magstyle@mag
709   \let\bxjs@magstyle\bxjs@magstyle@default
710   \ClassError\bxjs@clsname
711     {The engine does not support 'magstyle=usemag'}%
712     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
713       The default value 'nomag*' is used instead.\MessageBreak \@ehc}
714 \fi
715 \fi

```

オプション処理時に遅延させていた `jbase` の処理をここで実行する。

```

716 \bxjs@do@opt@jbase

```

`\Cjascale` 和文クラス共通仕様 (※ただし ZR 氏提唱) における、和文スケール値の変数。

```

717 \let\Cjascale\jsScale

```

---



---

8bit 欧文  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  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。)

```

718 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
719   \@tempcnta="80 \loop \ifnum \@tempcnta<"100
720     \catcode\@tempcnta\active
721     \advance\@tempcnta\@ne
722   \repeat
723 \fi

```

`js` オプション指定時は、`jsarticle` (または `jsbook`) クラスを読込済のように振舞う。

※「2つのクラスを読み込んだ状態」は `\LoadClass` を使用した場合に出現するので、別に異常ではない。

```
724 \ifbxjs@disguise@js
725 %<book|report>\def\bxjs@js@clsname{jsbook}
726 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
727 \@namedef{ver@bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
728 \fi
```

`color/graphics` パッケージが持つ出力用紙サイズ設定の機能は、BXJS クラスでは余計なので無効にしておく。このため、グローバルで `nosetpagesize` を設定しておく。

```
729 \g@addto@macro\@classoptionslist{,nosetpagesize}
```

`oldfontcommands` オプション指定時は `\allowoldfontcommands` 命令を実行する。

```
730 \ifbxjs@oldfontcommands
731 \AtEndOfClass{\allowoldfontcommands}
732 \fi
```

---

■papersize スペシャルの出力 dvips の papersize スペシャルを出力します。これで dvips や新しい dviout で出力領域が自動設定できます。トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> はトンボ出力幅を両側に 1 インチとていますので、dvips 使用時に

```
-0 -0.5in,-0.5in
```

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

[2003-05-17] トンボをプレビューに使うことを考えて 1 インチを 2 インチにしました。

[2016-07-11] memoir クラスのマニュアルによると、トンボを含めた用紙の寸法は `\stockwidth`, `\stockheight` と呼ぶようですので、これを使うことにしました。

[2017-01-11] トンボオプションが指定されているとき「だけ」`\stockwidth`, `\stockheight` を定義するようにしました。

---

BXJS では出力用紙サイズ記録は `geometry` パッケージが行う。

また、JS クラスと異なり、`\stockwidth`, `\stockheight` は常に定義される。

---

```
733 \newdimen\stockwidth \newdimen\stockheight
734 \begingroup\expandafter\expandafter\expandafter\endgroup
735 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
736 % \newdimen\stockwidth \newdimen\stockheight
737 \setlength{\stockwidth}{\paperwidth}
738 \setlength{\stockheight}{\paperheight}
739 \advance \stockwidth 2in
740 \advance \stockheight 2in
741 \fi
```

## ■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。

```
742 %<slide>\def\n@baseline{13}%
743 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
744 %<!slide>\else \def\n@baseline{16}\fi
```

## ■拡大率の設定

---

`\bxjs@magstyle` の値に応じてスイッチ `jsc@mag` と `jsc@mag@xreal` を設定する。

```
745 \ifx\bxjs@magstyle\bxjs@magstyle@mag
746   \jsc@magtrue
747 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
748   \jsc@mag@xrealtrue
749 \fi\fi
```

---

サイズの変更は  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  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に若干縮めました。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 `truedimen` 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。

```
750 \ifx\bxjs@param@mag\relax
751   \@tempdima=\bxjs@param@basefontsize
752   \advance\@tempdima.001pt \multiply\@tempdima25
753   \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
754   \edef\bxjs@param@mag{\the\@tempcmta}
755 \else
756 % mag 値が直接指定された場合
757   \let\c@bxjs@cmta\@tempcmta
758   \setcounter{bxjs@cmta}{\bxjs@param@mag}
759   \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
760 % 有効な mag 値の範囲は 1--32768
761   \edef\bxjs@param@mag{\the\@tempcmta}
762   \advance\@tempcmta100000
763   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
764   \expandafter\bxjs@tmpa\the\@tempcmta\@nil
765   \edef\bxjs@param@basefontsize{\the\@tempdima}
766 \fi
```



```

767 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
768 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
769 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
770 \edef\jsc@magscale{\strip@pt\@tempdima}
771 \let\jsBaseFontSize\bxjs@param@basefontsize
772 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
773 %\typeout{\string\jsEngine: \meaning\jsEngine}
774 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
775 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
776 %\typeout{\string\jsc@magscale: \jsc@magscale}
777 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
778 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}

```

---

[2016-07-08] \jsc@mpt および \jsc@mmm に、それぞれ 1pt および 1mm を拡大させた値を格納します。以降のレイアウト指定ではこちらを使います。

---

\mag する場合（現状はこれが既定）にコードの変更を低減するために、以下では必要に応じて、\jsc@mpt を \p@? と書く。その上で、\mag する場合は ? を無視して \p@ と解釈させ、\mag しなない場合は ? を英字扱いにして \p@? という制御綴を \jsc@mpt と同値にする。※（多分 2.0 版あたりで）JS クラスに合わせるため \p@? 表記を止める予定。

```

779 \newdimen\jsc@mpt
780 \newdimen\jsc@mmm
781 \ifjsc@mag
782   \jsc@mpt=1\p@
783   \jsc@mmm=1mm
784   \catcode\?=9 % \p@? read as \p@
785 \else
786   \jsc@mpt=\jsc@magscale\p@
787   \jsc@mmm=\jsc@magscale mm
788   \catcode\?=11 \let\p@?\jsc@mpt
789 \fi
790 \chardef\bxjs@qmcc=\catcode\?\relax

```

ここで pTeX 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 \jsScale × (指定フォントサイズ) に等しい。

noz 非指定時は \zw を \jsZw と同義にする。

```

791 \newdimen\jsZw
792 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
793 \ifbxjs@usezw
794   \providecommand*\zw{\jsZw}
795 \fi

```

\zwspace 全角幅の水平空き。

```

796 \def\zwspace{\hskip\jsZw\relax}

```

そして、magstyle が nomag\* の場合は、NFSS にパッチを当てる。

```

797 \ifjsc@mag@xreal
798 \RequirePackage{type1cm}
799 \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

---

```

800 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
801 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
802 \let\jsc@get@external@font\get@external@font
803 \def\get@external@font{%
804   \jsc@preadjust@extract@font
805   \jsc@get@external@font}
806 \def\jsc@fstrunc#1{%
807   \edef\jsc@tmpa{\strip@pt#1}%
808   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
809 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
810   \if#5*\else
811     \edef\jsc@tmpa{#1%
812       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
813   \fi}
814 \def\jsc@preadjust@extract@font{%
815   \let\jsc@req@size\f@size
816   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
817   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
818   \let\jsc@ref@size\jsc@tmpa
819   \let\f@size\jsc@ref@size}
820 \def\execute@size@function#1{%
821   \let\jsc@cref@size\f@size
822   \let\f@size\jsc@req@size
823   \csname s@fct@#1\endcsname}
824 \let\jsc@DeclareErrorFont\DeclareErrorFont
825 \def\DeclareErrorFont#1#2#3#4#5{%
826   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
827   \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
828   \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
829 \def\gen@sfcnt{%
830   \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
831   \empty@sfcnt}
832 \def\genb@sfcnt{%
833   \edef\mandatory@arg{%
834     \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@{}}%
835   \empty@sfcnt}
836 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
837 \fi

```

[2016-11-16] latex.ltx (ltspace.dtx) で定義されている `\smallskip` の、単位 `pt` を `\jsc@mpt` に置き換えた `\jsc@smallskip` を定義します。これは `\maketitle` で用いられます。`\jsc@medskip` と `\jsc@bigskip` は必要ないのでコメントアウトしています。

```

\jsc@smallskip
\jsc@medskip 838 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 839 %\def\jsc@medskip{\vspace\jsc@medskipamount}
840 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount 841 \newskip\jsc@smallskipamount
842 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
\jsc@bigskipamount 843 %\newskip\jsc@medskipamount
844 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
845 %\newskip\jsc@bigskipamount
846 %\jsc@bigskipamoun =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

[2016-07-11] 新しく追加した`\stockwidth`, `\stockheight` も`\mag`にあわせてスケールします。

[2017-01-11] トンボオブションが指定されているとき「だけ」`\stockwidth`, `\stockheight` が定義されています。

■`pagesize` スペシャルの出力 [2003-05-17] `dvipdfm(x)` の `pagesize` スペシャルを出力します。

[2004-08-08] 今の `dvipdfmx` は `dvips` 用スペシャルを理解するようなので外しました。

```

847 % \ifpapersize
848 %   \setlength{\@tempdima}{\paperwidth}
849 %   \setlength{\@tempdimb}{\paperheight}
850 %   \iftombow
851 %     \advance \@tempdima 2truein
852 %     \advance \@tempdimb 2truein
853 %   \fi
854 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima space height \the\@tempdimb}}
855 % \fi

```

### 3 和文フォントの変更

---

和文フォントの設定は和文ドライバの管轄。

---

`\@` 欧文といえば,  $\text{\LaTeX}$  の `\def\@{\spacefactor\@m}` という定義 (`\@m` は 1000) では `I watch TV\@.` と書くと `V` とピリオドのベ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。

[2016-07-14] 2015-01-01 の  $\text{\LaTeX}$  で, auxiliary files に書き出されたときにスペースが食われないようにする修正が入りました。これに合わせて `{}` を補いました。

```

856 \chardef\bxjs@periodchar=`\
857 \bxjs@protected\def\bxjs@SE{\spacefactor\sffcode\bxjs@periodchar}
858 \def\@{\bxjs@SE{}}

```

## 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 L<sup>A</sup>T<sub>E</sub>X 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、L<sup>A</sup>T<sub>E</sub>X 本体で定義されています。

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

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 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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` `\fontsize` 命令 (`\large` 等でなく) でフォントサイズ変更した場合にもフックが実行されるように、`\setfontsize` ではなく `\set@fontsize` に対してパッチを当てるように変更。

```
859 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
860 \expandafter\bxjs@tmpa\expandafter{%
861   \set@fontsize{#1}{#2}{#3}%
862 % 末尾にコードを追加
863   \expandafter\def\expandafter\size@update\expandafter{%
864     \size@update
865     \jsFontSizeChanged}%
866 }
```

`\jsFontSizeChanged` フォントサイズ変更時に呼ばれるフック。`\jsZw` を再設定している。その後でユーザ定義用のフック `\jsResetDimen` を実行する。

```
867 \newcommand*\jsFontSizeChanged{%
868   \jsZw=\f@size\p@
869   \jsZw=\jsScale \jsZw
870   \ifdim\parindent>\z@
871     \if@english \parindent=1em
872     \else       \parindent=1\jsZw
873   \fi
874   \fi\relax
875   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
876 \newcommand*\jsResetDimen{}
```

---

`\jsc@setfontsize` クラスファイルの内部では、拡大率も考慮した `\jsc@setfontsize` を `\@setfontsize` の代わりに用いることにします。

```
877 \ifjsc@mag
878   \let\jsc@setfontsize\@setfontsize
879 \else
880   \def\jsc@setfontsize#1#2#3{%
881     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
882 \fi
```

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。

---

```
883 \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 `\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、しば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。

TODO: Hasumi さん [qa:54539] のご指摘は考慮中です。

---

別行立て数式に入るときに `\narrowbaselines` が呼ばれるが、このコードでは「数式中で `\normalsize` などのサイズ命令 (`\@currsize` の実体) が呼ばれた」ことになり警告が出る。JS クラスでは、`\@setfontsize` 中の `\@nomath` 実行を消して「そもそもサイズ命令で警告が出ない」ようにしている。警告が常に出ないのも望ましくないので、BXJS クラスの実装では、`\narrowbaselines` の時だけ警告が出ないようにする。

---

```

884 \newif\ifnarrowbaselines
885 \if@english
886   \narrowbaselinestrue
887 \fi
888 \def\narrowbaselines{%
889   \narrowbaselinestrue
890   \skip0=\abovedisplayskip
891   \skip2=\abovedisplayshortskip
892   \skip4=\belowdisplayskip
893   \skip6=\belowdisplayshortskip
894 % 一時的に警告を無効化する
895   \let\bxjs@ltx@nomath\@nomath
896   \let\@nomath\@gobble
897   \@currsize\selectfont
898   \let\@nomath\bxjs@ltx@nomath
899   \abovedisplayskip=\skip0
900   \abovedisplayshortskip=\skip2
901   \belowdisplayskip=\skip4
902   \belowdisplayshortskip=\skip6\relax}
903 \def\widebaselines{\narrowbaselinesfalse\@currsize\selectfont}

```

---

`microtype` パッケージを読み込んだ場合、`\normalsize` 等のフォントサイズ変更命令の定義の中に `if` 文が使われていると、不可解なエラーが発生する。これは `microtype` が邪悪なトリックを使用しているせいなのだが、一応こちら側で対策をとることにする。

```

904 \def\bxjs@if@narrowbaselines{%
905   \ifnarrowbaselines\expandafter\@firstoftwo
906   \else \expandafter\@secondoftwo
907   \fi
908 }

```

---

`\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。

---

`microtype` 対策のため `if` 文を避ける。

---

```

909 \renewcommand{\normalsize}{%
910   \bxjs@if@narrowbaselines{%
911     \jsc@setfontsize\normalsize\@xpt\@xipt

```

```

912 }{%else
913   \jsc@setfontsize\normalsize\xpt{\n@baseline}%
914 }%

```

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。

[2009-08-26] T<sub>E</sub>X Q & A 52569 から始まる議論について逡巡していましたが, 結局, 微調節してみることにしました。

```

915 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
916 \abovedisplayshortskip \z@ \@plus3\p@?
917 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
918 \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。

```

919 \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。

```

920 \normalsize

```

`\Cht` 基準となる長さの設定をします。pL<sub>A</sub>T<sub>E</sub>X 2<sub>ε</sub> 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` [2017-08-31] 基準とする文字を「全角空白」(EUC コード 0xA1A1) から「漢」(JIS コード 0x3441) へ変更しました。

`\Cvs`

`\Chs`

---

`\Cwd` 等の変数は pT<sub>E</sub>X 系以外では未定義なのでここで定義する。

```

921 \ifx\Cht\@undefined \newdimen\Cht \fi
922 \ifx\Cdp\@undefined \newdimen\Cdp \fi
923 \ifx\Cwd\@undefined \newdimen\Cwd \fi
924 \ifx\Cvs\@undefined \newdimen\Cvs \fi
925 \ifx\Chs\@undefined \newdimen\Chs \fi

```

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

---

```

926 \setlength\Cht{0.88\jsZw}
927 \setlength\Cdp{0.12\jsZw}
928 \setlength\Cwd{1\jsZw}
929 \setlength\Cvs{\baselineskip}
930 \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  $4 \pm 2$ ,  $2 \pm 1$  ポイントでしたが, ここではゼロ (`\z@`) にしました。

---

microtype 対策のため if 文を避ける。後の \footnotesize も同様。

---

```

931 \newcommand{\small}{%
932   \bxjs@if@narrowbaselines{%
933     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
934     %<kiyou>    \jsc@setfontsize\small{8.888}{11}%
935   }{%else
936     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
937     %<kiyou>    \jsc@setfontsize\small{8.888}{13.2418}%
938   }%
939   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
940   \abovedisplayshortskip \z@ \@plus3\p@?
941   \belowdisplayskip \abovedisplayskip
942   \belowdisplayshortskip \belowdisplayskip
943   \def\@listi{\leftmargin\leftmargini
944             \topsep \z@
945             \parsep \z@
946             \itemsep \parsep}}

```

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

```

947 \newcommand{\footnotesize}{%
948   \bxjs@if@narrowbaselines{%
949     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
950     %<kiyou>    \jsc@setfontsize\footnotesize{8.888}{11}%
951   }{%else
952     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
953     %<kiyou>    \jsc@setfontsize\footnotesize{8.888}{13.2418}%
954   }%
955   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
956   \abovedisplayshortskip \z@ \@plus2\p@?
957   \belowdisplayskip \abovedisplayskip
958   \belowdisplayshortskip \belowdisplayskip
959   \def\@listi{\leftmargin\leftmargini
960             \topsep \z@
961             \parsep \z@
962             \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間で行が揃うようにします。

\LARGE [2004-11-03] \HUGE を追加。

```

\huge 963 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
964 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 965 \if@twocolumn
\HUGE 966 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
```



```

967 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
968 \else
969 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large\xi\pt{17}}
970 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
971 \fi
972 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\xiv\pt{21}}
973 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
974 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\xv\pt{25}}
975 \newcommand{\huge}{\jsc@setfontsize\huge\xx\pt{28}}
976 \newcommand{\Huge}{\jsc@setfontsize\Huge\xxv\pt{33}}
977 \newcommand{\HUGE}{\jsc@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り合いに大きくなるためです。

本文中の数式の中では `\narrowbaselines` 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。

```

978 \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 2ε` 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。

```

979 % \newcommand{\headfont}{\bfseries}
980 \newcommand{\headfont}{\sffamily}
981 % \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。

```

982 %<kiyou>\setlength\columnsep{2\Cwd}
983 %<kiyou>\setlength\columnsep{28truebp}
984 \setlength\columnseprule{0\jsc@empt}

```

### ■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。

```

\lineskiplimit

```

```

\normallineskiplimit

```

```

985 \setlength\lineskip{1\jsc@empt}
986 \setlength\normallineskip{1\jsc@empt}
987 \setlength\lineskiplimit{1\jsc@empt}
988 \setlength\normallineskiplimit{1\jsc@empt}

```

`\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。

```
989 \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。

```

990 \setlength\parskip{0\jsc@empt}
991 \if@slide
992   \setlength\parindent{0\p@}
993 \else
994   \setlength\parindent{1\Cwd}
995 \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になっています。ここはオリジナル通りです。

```

\@highpenalty 996 \@lowpenalty 51
997 \@medpenalty 151
998 \@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。

```
999 % \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。

```
1000 % \brokenpenalty 100
```

## 5.1 ページレイアウト

---

BXJS ではページレイアウトの処理は `geometry` パッケージが担当している。

---

### ■準備

---

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
1001 \ifjsc@mag
```

```

1002 \mag=\bxjs@param@mag
1003 \fi
1004 \setlength{\topskip}{10\p@?}

    \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。

1005 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
1006 \def\bxjs@read@b{%
1007   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
1008   \else \expandafter\bxjs@read@d \fi}
1009 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
1010 \def\bxjs@read@d#1\@nil{}
1011 \expandafter\bxjs@read@a\bxjs@param@paper\@nil

```

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

```

1012 \edef\bxjs@layout@paper{%
1013   \ifjsc@mag truedimen,\fi
1014   \if@landscape landscape,\fi
1015   \bxjs@param@paper}

```

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

```

1016 %<*article|report>
1017 \def\bxjs@layout{%
1018   headheight=\topskip,footskip=0.03367\paperheight,%
1019   headsep=\footskip-\topskip,includeheadfoot,%
1020   hscale=0.76,hmarginratio=1:1,%
1021   vscale=0.83,vmarginratio=1:1,%
1022 }
1023 %</article|report>
1024 %<*book>
1025 \ifbxjs@layout@buggyhmargin %---
1026 % アレ
1027 \def\bxjs@layout{%
1028   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1029   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1030   vscale=0.83,vmarginratio=1:1,%
1031 }
1032 \else %---
1033 % 非アレ
1034 \def\bxjs@layout{%
1035   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1036   hmargin=18\jsc@mmm,%
1037   vscale=0.83,vmarginratio=1:1,%
1038 }
1039 \fi %---
1040 %</book>
1041 %<*slide>
1042 \def\bxjs@layout{%
1043   noheadfoot,%

```

```

1044 hscale=0.9,hmarginratio=1:1,%
1045 vscale=0.944,vmarginratio=1:1,%
1046 }
1047 %</slide>

```

`\fullwidth` 〔寸法レジスタ〕ヘッダ・フッタ領域の横幅。

```

1048 \newdimen\fullwidth

```

`\jsTextWidthLimit` 〔実数値マクロ〕`bxjsbook` における、`\textwidth` の上限の全角単位での値。

```

1049 %<*book>
1050 \newcommand\jsTextWidthLimit{40}
1051 \ifx\bxjs@textwidth@limit\undefined\else
1052   \let\c@bxjs@cmta\@tempcmta
1053   \setcounter{bxjs@cmta}{\bxjs@textwidth@limit}
1054   \long\edef\jsTextWidthLimit{\the\@tempcmta}
1055 \fi
1056 %</book>

```

`\bxjs@postproc@layout` `geometry` の後処理。

```

1057 \def\bxjs@postproc@layout{%
1058 % ドライバ再設定
1059   \ifx\bxjs@geometry@driver\relax\else
1060     \let\Gm@driver\bxjs@geometry@driver
1061   \fi
1062 % textwidth 調整
1063   \ifbxjs@whole@zw@lines
1064     \@tempdimb=\textwidth
1065     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1066     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1067     \advance\@tempdimb-\textwidth
1068     \advance\oddsidemargin 0.5\@tempdimb
1069     \advance\evensidemargin 0.5\@tempdimb
1070   \fi
1071   \fullwidth=\textwidth
1072 %<*book>
1073   \ifdim\textwidth>\jsTextWidthLimit\Cwd
1074     \textwidth=\jsTextWidthLimit\Cwd
1075     \addtolength\evensidemargin{\fullwidth-\textwidth}
1076   \fi
1077 %</book>
1078 % textheight 調整
1079   \@tempdimb=\textheight
1080   \advance\textheight-\topskip
1081   \divide\textheight\baselineskip \multiply\textheight\baselineskip
1082   \advance\textheight\topskip
1083   \advance\@tempdimb-\textheight
1084   \advance\topmargin0.5\@tempdimb
1085 % headheight 調整
1086   \@tempdima=\topskip

```

```

1087 \advance\headheight\@tempdima
1088 \advance\topmargin-\@tempdima
1089 % marginpar 設定
1090 \setlength\marginparsep{\columnsep}
1091 \setlength\marginparpush{\baselineskip}
1092 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1093     -\textwidth-10\jsc@mmm-\marginparsep}
1094 \ifbxjs@whole@zw@lines
1095     \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1096 \fi
1097 % 連動する変数
1098 \maxdepth=.5\topskip
1099 \stockwidth=\paperwidth
1100 \stockheight=\paperheight
1101 }

```

`\jsGeometryOptions` geometry パッケージに渡すオプションのリスト。

※ `geometry=user` 指定時にユーザが利用することを想定している。

```

1102 \edef\jsGeometryOptions{%
1103     \bxjs@layout@paper,\bxjs@layout}

```

---

## ■ geometry パッケージ読込

---

geometry オプションの値に応じて分岐する。

まずは `geometry=class` の場合。

```

1104 \ifx\bxjs@geometry\bxjs@geometry@class

```

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。

```

1105 \@onlypreamble\bxjs@revert
1106 \let\bxjs@revert\@empty
1107 \edef\bxjs@tmpa{\the\ht\strutbox}
1108 \ht\strutbox=10\p@?
1109 \g@addto@macro\bxjs@revert{\ht\strutbox=\bxjs@tmpa\relax}

```

geometry のドライバオプション指定。 `nopapersize` 指定時は、special 命令出力を抑止するためにドライバを `none` にする。そうでない場合は、クラスで指定したドライバオプションが引き継がれるので何もしなくてよいが、例外として、ドライバが `dvipdfmx` の時は、現状の geometry は `dvipdfm` を指定する必要がある。

```

1110 \ifbxjs@papersize
1111     \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1112         \PassOptionsToPackage{dvipdfm}{geometry}
1113     \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1114         \PassOptionsToPackage{dvipdfm}{geometry}
1115     \fi\fi

```

```

1116 \let\bxPapersizeSpecialDone=t
1117 \else
1118 \PassOptionsToPackage{driver=none}{geometry}
1119 \fi

```

ここで `geometry` を読み込む。

※ `geometry` の `begin-document` フックにおいて、`LuaTeX` の旧版互換を有効にする。

```

1120 \edef\bxjs@nxt{%
1121 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1122 \AtBeginDocument{\bxjs@pre@geometry@hook}
1123 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1124 \bxjs@nxt \bxjs@revert
1125 \AtBeginDocument{\RevokeOldLuaTeXBehavior}

```

`\bxjs@geometry@driver` `geometry` が用いるドライバの名前。

※この値は一度決めた後は変わってほしくないので、`\bxjs@postproc@layout` において書き戻す処理を入れている。

```

1126 \let\bxjs@geometry@driver\Gm@driver
1127 \bxjs@postproc@layout

```

`\bxjs@pre@geometry@hook` 1.2 版より、`geometry` の 4.x 版の使用は非推奨とする。

※将来サポートを廃止する予定。

```

1128 \@onlypreamble\bxjs@pre@geometry@hook
1129 \def\bxjs@pre@geometry@hook{%
1130 \ifpackageloaded{geometry}{%
1131 \ifpackagelater{geometry}{2010/02/12}{%}%else
1132 \PackageWarningNoLine\bxjs@clsname
1133 {The 'geometry' package installed\MessageBreak
1134 is too old (< v5.0)}%
1135 \if x\jsEngine \ifnum\mag=\@m\else
1136 \def\bxjs@Gm@driver{pdftex}
1137 \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1138 \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1139 \fi\fi
1140 }%
1141 \ifjsWithpTeXng
1142 \ifx\Gm@driver\@empty
1143 \def\Gm@driver{pdftex}%
1144 \fi
1145 \fi
1146 }{}}

```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

- `\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。

```

1147 \def\setpagelayout{\@ifstar
1148   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1149 \def\bxjs@modify@layout#1{%
1150   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1151   \bxjs@nxt\bxjs@postproc@layout}
1152 \def\bxjs@reset@layout#1{%
1153   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1154   \bxjs@nxt\bxjs@postproc@layout}

```

geometry=class の場合の処理はここで終わり。

次に、geometry=user の場合の処理。

```

1155 \else\ifx\bxjs@geometry\bxjs@geometry@user

```

この場合はユーザが何らかの方法（例えば geometry を読み込む）でページレイアウトを設定する必要がある。もし、本体開始時に \textwidth がカーネル設定の値（.5\maxdimen）のままになっている場合はエラーを出す。

※\jsUseMinimalPageLayout は動作テスト用。

```

1156 \AtBeginDocument{\bxjs@check@page@layout}
1157 \onlypreamble\bxjs@check@page@layout
1158 \def\bxjs@check@page@layout{%
1159   \ifdim\textwidth=.5\maxdimen
1160     \ClassError\bxjs@clsname
1161       {Page layout is not properly set}%
1162       {\@ehd}
1163   \fi}
1164 \def\jsUseMinimalPageLayout{%
1165   \setlength{\textwidth}{6.5in}%
1166   \setlength{\textheight}{8in}}

```

\setpagelayout はとりあえず無効にしておく。

```

1167 \let\bxjs@geometry@driver\relax
1168 \def\setpagelayout{\@ifstar
1169   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1170 \def\bxjs@pagelayout@a#1{%
1171   \ClassError\bxjs@clsname
1172     {Command '\string\setpagelayout' is not supported,\MessageBreak
1173     because 'geometry' value is not 'class'}\@eha}

```

geometry=user の場合の処理はここで終わり。

```

1174 \fi\fi

```

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。

---



---

※ diff が崩壊するのを避けるためオリジナルのコードを無効化した状態で挿入しておく。

---

---

<\*jsclasses>

## ■縦方向のスペース

`\headheight` `\topskip` は本文領域上端と本文 1 行目のベースラインとの距離です。あまりぎりぎりの値にすると、本文中に  $\int$  のような高い文字が入ったときに 1 行目のベースラインが他のページより下がってしまいます。ここでは本文の公称フォントサイズ (10pt) にします。

[2003-06-26] `\headheight` はヘッダの高さで、元は 12pt でしたが、新ドキュメントクラスでは `\topskip` と等しくしていました。ところが、`fancyhdr` パッケージで `\headheight` が小さいとおかしいことになるようですので、2 倍に増やしました。代わりに、版面の上下揃えの計算では `\headheight` ではなく `\topskip` を使うことにしました。

[2016-08-17] 圏点やルビが一行目に来た場合に下がるのを防ぐため、`\topskip` を 10pt から 1.38zw に増やしました。`\headheight` は従来と同じ 20pt のままとします。

```
1175 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1176 \if@slide
1177   \setlength\headheight{0\jsc@empt}
1178 \else
1179   \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1180 \fi
```

`\footskip` `\footskip` は本文領域下端とフッタ下端との距離です。標準クラスファイルでは、book で 0.35in (約 8.89mm)、book 以外で 30pt (約 10.54mm) となっていました。ここでは A4 判のときちょうど 1cm となるように、`\paperheight` の 0.03367 倍 (最小 `\baselineskip`) としました。書籍については、フッタは使わないことにして、ゼロにしました。

```
1181 %<*article|kiyou>
1182 \if@slide
1183   \setlength\footskip{0pt}
1184 \else
1185   \setlength\footskip{0.03367\paperheight}
1186   \ifdim\footskip<\baselineskip
1187     \setlength\footskip{\baselineskip}
1188   \fi
1189 \fi
1190 %</article|kiyou>
1191 %<jspf>\setlength\footskip{9\jsc@mmm}
1192 %<*book>
1193 \if@report
1194   \setlength\footskip{0.03367\paperheight}
1195   \ifdim\footskip<\baselineskip
1196     \setlength\footskip{\baselineskip}
1197   \fi
1198 \else
1199   \setlength\footskip{0pt}
1200 \fi
```



```

1201 %</book>
1202 %<*report>
1203 \setlength\footskip{0.03367\paperheight}
1204 \ifdim\footskip<\baselineskip
1205   \setlength\footskip{\baselineskip}
1206 \fi
1207 %</report>

```

`\headsep` `\headsep` はヘッダ下端と本文領域上端との距離です。元は book で 18pt (約 6.33mm), それ以外で 25pt (約 8.79mm) になっていました。ここでは article は `\footskip - \topskip` としました。

[2016-10-08] article の `slide` のとき, および book の非 `report` と `kiyou` のときに `\headsep` を減らしそこねていたのを修正しました (2016-08-17 での修正漏れ)。

```

1208 %<*article>
1209 \if@slide
1210   \setlength\headsep{0\jsc@empt}
1211   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1212   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1213 \else
1214   \setlength\headsep{\footskip}
1215   \addtolength\headsep{-\topskip}
1216 \fi
1217 %</article>
1218 %<*book>
1219 \if@report
1220   \setlength\headsep{\footskip}
1221   \addtolength\headsep{-\topskip}
1222 \else
1223   \setlength\headsep{6\jsc@mmm}
1224   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1225   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1226 \fi
1227 %</book>
1228 %<*report>
1229 \setlength\headsep{\footskip}
1230 \addtolength\headsep{-\topskip}
1231 %</report>
1232 %<*jspf>
1233 \setlength\headsep{9\jsc@mmm}
1234 \addtolength\headsep{-\topskip}
1235 %</jspf>
1236 %<*kiyou>
1237 \setlength\headheight{0\jsc@empt}
1238 \setlength\headsep{0\jsc@empt}
1239 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1240 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1241 %</kiyou>

```

`\maxdepth` `\maxdepth` は本文最下行の最大の深さで, plain  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  や  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  2.09 では 4pt に固定でした。 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}2\mathrm{e}$  では `\maxdepth + \topskip` を本文フォントサイズの 1.5 倍にしたいのですが, `\topskip` は本文フォントサイズ (ここでは 10pt) に等しいので, 結局 `\maxdepth` は `\topskip` の半分の値 (具体的には 5pt) にします。

```
1242 \setlength\maxdepth{.5\topskip}
```

## ■本文の幅と高さ

`\fullwidth` 本文の幅が全角 40 文字を超えると読みにくなります。そこで, 書籍の場合に限って, 紙の幅が広いときは外側のマージンを余分にとって全角 40 文字に押え, ヘッダやフッタは本文領域より広く取ることにします。このときヘッダやフッタの幅を表す `\fullwidth` という長さを定義します。

```
1243 \newdimen\fullwidth
```

この `\fullwidth` は `article` では紙幅 `\paperwidth` の 0.76 倍を超えない全角幅の整数倍 (二段組では全角幅の偶数倍) にします。0.76 倍という数値は A4 縦置きの場合に紙幅から約 2 インチを引いた値になるように選びました。`book` では紙幅から 36 ミリを引いた値にしました。

`\textwidth` 書籍以外では本文領域の幅 `\textwidth` は `\fullwidth` と等しくします。`article` では A4 縦置きで 49 文字となります。某学会誌スタイルでは 50zw (25 文字 × 2 段) + 段間 8mm とします。

```
1244 %<*article>
1245 \if@slide
1246   \setlength\fullwidth{0.9\paperwidth}
1247 \else
1248   \setlength\fullwidth{0.76\paperwidth}
1249 \fi
1250 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1251 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1252 \setlength\textwidth{\fullwidth}
1253 %</article>
1254 %<*book>
1255 \if@report
1256   \setlength\fullwidth{0.76\paperwidth}
1257 \else
1258   \setlength\fullwidth{\paperwidth}
1259   \addtolength\fullwidth{-36\jsc@mmm}
1260 \fi
1261 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1262 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1263 \setlength\textwidth{\fullwidth}
1264 \if@report \else
1265   \if@twocolumn \else
1266     \ifdim \fullwidth>40zw
1267       \setlength\textwidth{40zw}
```

```

1268 \fi
1269 \fi
1270 \fi
1271 %</book>
1272 %<*report>
1273 \setlength\fullwidth{0.76\paperwidth}
1274 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1275 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1276 \setlength\textwidth{\fullwidth}
1277 %</report>
1278 %<*jspf>
1279 \setlength\fullwidth{50zw}
1280 \addtolength\fullwidth{8\jsc@mmm}
1281 \setlength\textwidth{\fullwidth}
1282 %</jspf>
1283 %<*kiyou>
1284 \setlength\fullwidth{48zw}
1285 \addtolength\fullwidth{\columnsep}
1286 \setlength\textwidth{\fullwidth}
1287 %</kiyou>

```

`\textheight` 紙の高さ `\paperheight` は、1 インチと `\topmargin` と `\headheight` と `\headsep` と `\textheight` と `\footskip` とページ下部の余白を加えたものです。

本文部分の高さ `\textheight` は、紙の高さ `\paperheight` の 0.83 倍から、ヘッダの高さ、ヘッダと本文の距離、本文とフッタ下端の距離、`\topskip` を引き、それを `\baselineskip` の倍数に切り捨て、最後に `\topskip` を加えます。念のため 0.1 ポイント余分に加えておきます。0.83 倍という数値は、A4 縦置きの場合に紙の高さから上下マージン各約 1 インチを引いた値になるように選びました。

某学会誌スタイルでは 44 行にします。

[2003-06-26] `\headheight` を `\topskip` に直しました。以前はこの二つは値が同じであったので、変化はないはずです。

[2016-08-26] `\topskip` を 10pt から 1.38zw に増やしましたので、その分 `\textheight` を増やします (2016-08-17 での修正漏れ)。

[2016-10-08] article の slide のときに `\headheight` はゼロなので、さらに修正しました (2016-08-17 での修正漏れ)。

```

1288 %<*article|book|report>
1289 \if@slide
1290 \setlength{\textheight}{0.95\paperheight}
1291 \else
1292 \setlength{\textheight}{0.83\paperheight}
1293 \fi
1294 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1295 \addtolength{\textheight}{-\headsep}
1296 \addtolength{\textheight}{-\footskip}
1297 \addtolength{\textheight}{-\topskip}

```

```

1298 \divide\textheight\baselineskip
1299 \multiply\textheight\baselineskip
1300 %</article|book|report>
1301 %<jspf>\setlength{\textheight}{51\baselineskip}
1302 %<kiyou>\setlength{\textheight}{47\baselineskip}
1303 \addtolength{\textheight}{\topskip}
1304 \addtolength{\textheight}{0.1\jsc@mpt}
1305 %<jspf>\setlength{\mathindent}{10\jsc@mmm}

```

`\flushbottom` [2016-07-18] `\textheight` に念のため 0.1 ポイント余裕を持たせているのと同様に、`\flushbottom` にも余裕を持たせます。元の  $\text{\LaTeX 2}_\epsilon$  での完全な `\flushbottom` の定義は

```

\def\flushbottom{%
  \let\@textbottom\relax \let\@texttop\relax}

```

ですが、次のようにします。

```

1306 \def\flushbottom{%
1307   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1308   \let\@texttop\relax}

```

`\marginparsep` `\marginparsep` は欄外の書き込みと本文との間隔です。`\marginparpush` は欄外の書き込みどうしの最小の間隔です。

```

1309 \setlength\marginparsep{\columnsep}
1310 \setlength\marginparpush{\baselineskip}

```

`\oddsidemargin` それぞれ奇数ページ、偶数ページの左マージンから 1 インチ引いた値です。片面印刷では `\evensidemargin` が使われます。 $\text{\TeX}$  は上・左マージンに `1truein` を挿入しますが、トンボ関係のオプションが指定されると  $\text{\pLaTeX 2}_\epsilon$  (`plcore.ltx`) はトンボの内側に `1in` のスペース (`1truein` ではなく) を挿入するので、場合分けしています。

```

1311 \setlength{\oddsidemargin}{\paperwidth}
1312 \addtolength{\oddsidemargin}{-\fullwidth}
1313 \setlength{\oddsidemargin}{.5\oddsidemargin}
1314 \iftombow
1315   \addtolength{\oddsidemargin}{-1in}
1316 \else
1317   \addtolength{\oddsidemargin}{-\inv@mag in}
1318 \fi
1319 \setlength{\evensidemargin}{\oddsidemargin}
1320 \if@mparswitch
1321   \addtolength{\evensidemargin}{\fullwidth}
1322   \addtolength{\evensidemargin}{-\textwidth}
1323 \fi

```

`\marginparwidth` `\marginparwidth` は欄外の書き込みの横幅です。外側マージンの幅 (`\evensidemargin` + 1 インチ) から 1 センチを引き、さらに `\marginparsep` (欄外の書き込みと本文のアキ) を引いた値にしました。最後に `1zw` の整数倍に切り捨てます。

```

1324 \setlength\marginparwidth{\paperwidth}

```

```

1325 \addtolength\marginparwidth{-\oddsidemargin}
1326 \addtolength\marginparwidth{-\inv@mag in}
1327 \addtolength\marginparwidth{-\textwidth}
1328 \addtolength\marginparwidth{-10\jsc@mmm}
1329 \addtolength\marginparwidth{-\marginparsep}
1330 \@tempdima=1zw
1331 \divide\marginparwidth\@tempdima
1332 \multiply\marginparwidth\@tempdima

```

`\topmargin` 上マージン（紙の上端とヘッダ上端の距離）から 1 インチ引いた値です。

[2003-06-26] `\headheight` を `\topskip` に直しました。以前はこの二つは値が同じであったので、変化はないはずです。

[2016-08-17] `\topskip` を 10pt から 1.38zw に直しましたが、`\topmargin` は従来の値から変わらないように調節しました。…のつもりでしたが、`\textheight` を増やし忘れていたので変わってしまっていました（2016-08-26 修正済み）。

```

1333 \setlength\topmargin{\paperheight}
1334 \addtolength\topmargin{-\textheight}
1335 \if@slide
1336   \addtolength\topmargin{-\headheight}
1337 \else
1338   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1339 \fi
1340 \addtolength\topmargin{-\headsep}
1341 \addtolength\topmargin{-\footskip}
1342 \setlength\topmargin{0.5\topmargin}
1343 %<kiyou>\setlength\topmargin{81truebp}
1344 \iftombow
1345   \addtolength\topmargin{-1in}
1346 \else
1347   \addtolength\topmargin{-\inv@mag in}
1348 \fi

```

</jsclasses>

## ■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。

---

ここは元々は

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

としていたが、そもそも `\global\setlength~` は calc 使用時には有意義な動作をしない。`\global\footnotesep` だと所望の値が得られるが、同時に `\footnotesize` のフォントを固定させてしまうという副作用をもつ。なので、実際の設定値を直接使うことにする。

---

1349 `\footnotesep=11\p@? \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。

1350 `\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）とフロートだけのページで設定が異なります。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。

`\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。

1351 `\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。

1352 `\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。

1353 `\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。

1354 `\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。

1355 `\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。

1356 `\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。

1357 `\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。

1358 `\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。

1359 `\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。

```
1360 \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。

```
1361 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1362 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1363 \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 1364 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1365 \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 1366 \setlength\@fptop{0\p@? \@plus 1fil}
```

```
1367 \setlength\@fpsep{8\p@? \@plus 2fil}
```

```
1368 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpptop` 段抜きフロートについての値です。

```
\@dblfpsep 1369 \setlength\@dblfpptop{0\p@? \@plus 1fil}
```

```
\@dblfpbot 1370 \setlength\@dblfpsep{8\p@? \@plus 2fil}
```

```
1371 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

## 6 改ページ（日本語 T<sub>E</sub>X 開発コミュニティ版のみ）

`\pltx@cleartorightpage` [2017-02-24] コミュニティ版 p<sub>L</sub>T<sub>E</sub>X の標準クラス 2017/02/15 に合わせて, 同じ命令を追  
`\pltx@cleartoleftpage` 加しました。

- `\pltx@cleartooddpage` 1. `\pltx@cleartorightpage`: 右ページになるまでページを繰る命令
- `\pltx@cleartoevenpage` 2. `\pltx@cleartoleftpage`: 左ページになるまでページを繰る命令
3. `\pltx@cleartooddpage`: 奇数ページになるまでページを繰る命令
4. `\pltx@cleartoevenpage`: 偶数ページになるまでページを繰る命令

となっています。

```
1372 %\def\pltx@cleartorightpage{\clearpage\if@twoside
```

```
1373 % \ifodd\c@page
```

```
1374 % \iftdir
```

```
1375 % \hbox{}\thispagestyle{empty}\newpage
```

```
1376 % \if@twocolumn\hbox{}\newpage\fi
```

```
1377 % \fi
```

```
1378 % \else
```

```
1379 % \ifydir
```

```
1380 % \hbox{}\thispagestyle{empty}\newpage
```

```

1381 %      \if@twocolumn\hbox{}\newpage\fi
1382 %      \fi
1383 %    \fi\fi}
1384 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1385 %  \ifodd\c@page
1386 %    \ifydir
1387 %      \hbox{}\thispagestyle{empty}\newpage
1388 %      \if@twocolumn\hbox{}\newpage\fi
1389 %    \fi
1390 %  \else
1391 %    \iftdir
1392 %      \hbox{}\thispagestyle{empty}\newpage
1393 %      \if@twocolumn\hbox{}\newpage\fi
1394 %    \fi
1395 %  \fi\fi}
1396 %\def\pltx@cleartooddpage{\clearpage\if@twoside
1397 %  \ifodd\c@page\else
1398 %    \hbox{}\thispagestyle{empty}\newpage
1399 %    \if@twocolumn\hbox{}\newpage\fi
1400 %  \fi\fi}
1401 %\def\pltx@cleartoevenpage{\clearpage\if@twoside
1402 %  \ifodd\c@page
1403 %    \hbox{}\thispagestyle{empty}\newpage
1404 %    \if@twocolumn\hbox{}\newpage\fi
1405 %  \fi\fi}

```

---

BXJS クラスでは `\iftdir` 等が使えないので、横組を仮定した定義を用いる。

```

1406 \let\pltx@cleartorightpage\pltx@cleartooddpage
1407 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

---

`\cleardoublepage` [2017-02-24] コミュニティ版 p $\text{\LaTeX}$  の標準クラス 2017/02/15 に合わせて、`report` と `book` クラスの場合に `\cleardoublepage` を再定義します。

```

1408 %<*book|report>
1409 \if@openleft
1410   \let\cleardoublepage\pltx@cleartoleftpage
1411 \else\if@openright
1412   \let\cleardoublepage\pltx@cleartorightpage
1413 \fi\fi
1414 %</book|report>

```

## 7 ページスタイル

ページスタイルとして、 $\text{\LaTeX}$  2 $\epsilon$  (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  $\text{\LaTeX}$  2 $\epsilon$  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、おかしい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<sup>A</sup>T<sub>E</sub>X 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。

```
1415 % \def\ps@empty{%
1416 %   \let\@mkboth\@gobbletwo
1417 %   \let\@oddhead\@empty
1418 %   \let\@oddfoot\@empty
1419 %   \let\@evenhead\@empty
1420 %   \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。

```
1421 \def\ps@plainfoot{%
1422   \let\@mkboth\@gobbletwo
1423   \let\@oddhead\@empty
1424   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1425   \let\@evenhead\@empty
1426   \let\@evenfoot\@oddfoot}
1427 \def\ps@plainhead{%
1428   \let\@mkboth\@gobbletwo
1429   \let\@oddfoot\@empty
1430   \let\@evenfoot\@empty
1431   \def\@evenhead{%
1432     \if@mparswitch \hss \fi
```

```

1433 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1434 \if@mparswitch\else \hss \fi}%
1435 \def\@oddhead{%
1436 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1437 %<book>\let\ps@plain\ps@plainhead
1438 %<!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。

```

1439 %<*article|slide>
1440 \if@twoside
1441 \def\ps@headings{%
1442 \let\@oddfoot\@empty
1443 \let\@evenfoot\@empty
1444 \def\@evenhead{\if@mparswitch \hss \fi
1445 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1446 \if@mparswitch\else \hss \fi}%
1447 \def\@oddhead{%
1448 \underline{%
1449 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1450 \let\@mkboth\markboth
1451 \def\sectionmark##1{\markboth{%
1452 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1453 ##1}{}}%
1454 \def\subsectionmark##1{\markright{%
1455 \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
1456 ##1}}%
1457 }
1458 \else % if not twoside
1459 \def\ps@headings{%
1460 \let\@oddfoot\@empty
1461 \def\@oddhead{%
1462 \underline{%
1463 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1464 \let\@mkboth\markboth
1465 \def\sectionmark##1{\markright{%
1466 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1467 ##1}}%
1468 \fi
1469 %</article|slide>

```

次は `book` および `report` の場合です。[2011-05-10] しっぽ愛好家さん [qa:6370] のパッチを取り込ませていただきました（北見さん [qa:55896] のご指摘ありがとうございます）。

---

`\autoxspacing` は未定義の可能性があるので、「`\autoxspacing` が定義済なら実行する」マクロ `\bxjs@maybe@autoxspacing` を代わりに用いる。

---

```

1470 %<*book|report>
1471 \def\bxjs@maybe@autoxspacing{%
1472   \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1473 \newif\if@omit@number
1474 \def\ps@headings{%
1475   \let\@oddfoot\@empty
1476   \let\@evenfoot\@empty
1477   \def\@evenhead{%
1478     \if@mparswitch \hss \fi
1479     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1480       \textbf{\thepage}\hfil\leftmark}}}%
1481     \if@mparswitch\else \hss \fi}%
1482   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1483     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1484   \let\@mkboth\markboth
1485   \def\chaptermark##1{\markboth{%
1486     \ifnum \c@secnumdepth >\m@ne
1487       \if@mainmatter
1488         \if@omit@number\else
1489           \@chapapp\thechapter\@chappos\hskip1\jsZw
1490         \fi
1491       \fi
1492     \fi
1493     ##1}{}}}%
1494   \def\sectionmark##1{\markright{%
1495     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1496     ##1}}}%
1497 %</book|report>

```

最後は学会誌の場合です。

```

1498 %<*jspf>
1499 \def\ps@headings{%
1500   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1501   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1502   \def\@oddhead{\normalfont\hfil \@title \hfil}
1503   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1504 %</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。

```

1505 \def\ps@myheadings{%
1506   \let\@oddfoot\@empty\let\@evenfoot\@empty
1507   \def\@evenhead{%
1508     \if@mparswitch \hss \fi%
1509     \hbox to \fullwidth{\thepage\hfil\leftmark}}%
1510     \if@mparswitch\else \hss \fi}%
1511   \def\@oddhead{%
1512     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%

```

```

1513 \let\mkboth\@gobbletwo
1514 %<book|report> \let\chaptermark\@gobble
1515 \let\sectionmark\@gobble
1516 %<!book&!report> \let\subsectionmark\@gobble
1517 }

```

## 8 文書のマークアップ

### 8.1 表題

`\title` これらは L<sup>A</sup>T<sub>E</sub>X 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。

```

\date 1518 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1519 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1520 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1521 % \date{\today}

```

---

`\subtitle` 副題を設定する。

`\jsSubtitle` ※プレアンプルにおいて `\newcommand*{\subtitle}{...}` が行われることへの対策として、`\subtitle` の定義を `\title` の実行まで遅延させることにする。もしどうしても主題より前に副題を設定したい場合は、`\jsSubtitle` 命令を直接用いればよい。

本体を `\jsSubtitle` として定義する。

```

1522 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1523 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1524 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1525 \AtBeginDocument{\bxjs@decl@subtitle}
1526 \def\bxjs@decl@subtitle{%
1527   \global\let\bxjs@decl@subtitle\relax
1528   \ifx\subtitle\@undefined
1529     \global\let\subtitle\jsSubtitle
1530   \fi}

```

`\bxjs@annihilate@subtitle` `\subtitle` 命令を無効化する。

※独自の `\subtitle` が使われている場合は無効化しない。

```

1531 \def\bxjs@annihilate@subtitle{%
1532   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1533   \global\let\jsSubtitle\relax}

```

---

`\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。

```

\author 1534 %<*jspf>
\keywords 1535 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}

```

```

1536 \newcommand*\eauthor}[1]{\gdef\@eauthor{#1}}
1537 \newcommand*\keywords}[1]{\gdef\@keywords{#1}}
1538 \newcommand*\email}[1]{\gdef\authors@mail{#1}}
1539 \newcommand*\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1540 %</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なります。

```

1541 \def\plainifnotempty{%
1542   \ifx \@oddhead \@empty
1543     \ifx \@oddfoot \@empty
1544     \else
1545       \thispagestyle{plainfoot}%
1546     \fi
1547   \else
1548     \thispagestyle{plainhead}%
1549   \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。

[2016-11-16] 新設された `nomag` および `nomag*` オプションの場合をデフォルト (`usemag` 相当) に合わせるため、`\smallskip` を `\jsc@smallskip` に置き換えました。`\smallskip` のままでは `nomag(*)` の場合にスケールしなくなり、レイアウトが変わってしまいます。

```

1550 %<*article|book|report|slide>
1551 \if@titlepage
1552   \newcommand{\maketitle}{%
1553     \begin{titlepage}%
1554       \let\footnotesize\small
1555       \let\footnoterule\relax
1556       \let\footnote\thanks
1557       \null\vfil
1558       \if@slide
1559         {\footnotesize \@date}%
1560       \begin{center}
1561         \mbox{} \[\[1\jsZw]
1562         \large
1563         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1564         \jsc@smallskip
1565         \@title
1566         \ifx\bxjs@subtitle\@undefined\else
1567           \par\vskip\z@
1568           {\small \bxjs@subtitle\par}
1569         \fi
1570       \end{center}

```

```

1571         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1572         \vfill
1573         {\small \@author}%
1574     \end{center}
1575 \else
1576 \vskip 60\p@?
1577 \begin{center}%
1578     {\LARGE \@title \par}%
1579     \ifx\bxjs@subtitle\@undefined\else
1580         \vskip5\p@?
1581         {\normalsize \bxjs@subtitle\par}
1582     \fi
1583     \vskip 3em%
1584     {\large
1585         \lineskip .75em
1586         \begin{tabular}[t]{c}%
1587             \@author
1588         \end{tabular}\par}%
1589     \vskip 1.5em
1590     {\large \@date \par}%
1591 \end{center}%
1592 \fi
1593 \par
1594 \@thanks\vfil\null
1595 \end{titlepage}%
1596 \setcounter{footnote}{0}%
1597 \global\let\thanks\relax
1598 \global\let\maketitle\relax
1599 \global\let\@thanks\@empty
1600 \global\let\@author\@empty
1601 \global\let\@date\@empty
1602 \global\let\@title\@empty
1603 \global\let\title\relax
1604 \global\let\author\relax
1605 \global\let\date\relax
1606 \global\let\and\relax
1607 \bxjs@annihilate@subtitle
1608 }%
1609 \else
1610 \newcommand{\maketitle}{\par
1611     \begin{group}
1612         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1613         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1614         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1615             \parindent 1\jsZw\noindent
1616             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1617         \if@twocolumn
1618             \ifnum \col@number=\@ne
1619                 \@maketitle

```

```

1620     \else
1621         \twocolumn[\@maketitle]%
1622     \fi
1623     \else
1624         \newpage
1625         \global\@topnum\z@ % Prevents figures from going at top of page.
1626         \@maketitle
1627     \fi
1628     \plainifnotempty
1629     \@thanks
1630 \endgroup
1631 \setcounter{footnote}{0}%
1632 \global\let\thanks\relax
1633 \global\let\maketitle\relax
1634 \global\let\@thanks\@empty
1635 \global\let\@author\@empty
1636 \global\let\@date\@empty
1637 \global\let\@title\@empty
1638 \global\let\title\relax
1639 \global\let\author\relax
1640 \global\let\date\relax
1641 \global\let\and\relax
1642 \bxjs@annihilate@subtitle
1643 }

```

`\@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。

```

1644 \def\@maketitle{%
1645     \newpage\null
1646     \vskip 2em
1647     \begin{center}%
1648         \let\footnote\thanks
1649         {\LARGE \@title \par}%
1650         \ifx\bxjs@subtitle\@undefined\else
1651             \vskip3\p@?
1652             {\normalsize \bxjs@subtitle\par}
1653         \fi
1654         \vskip 1.5em
1655         {\large
1656             \lineskip .5em
1657             \begin{tabular}[t]{c}%
1658                 \@author
1659             \end{tabular}\par}%
1660         \vskip 1em
1661         {\large \@date}%
1662     \end{center}%
1663     \par\vskip 1.5em
1664 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1665 }
1666 \fi

```

```

1667 %</article|book|report|slide>
1668 %<*jspf>
1669 \newcommand{\maketitle}{\par
1670   \begin{group}
1671     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1672     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1673     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1674       \parindent 1\jsZw\noindent
1675       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1676     \twocolumn[\@maketitle]%
1677     \plainifnotempty
1678     \@thanks
1679   \end{group}
1680   \setcounter{footnote}{0}%
1681   \global\let\thanks\relax
1682   \global\let\maketitle\relax
1683   \global\let\@thanks\@empty
1684   \global\let\@author\@empty
1685   \global\let\@date\@empty
1686 % \global\let\@title\@empty % \@title は柱に使う
1687   \global\let\title\relax
1688   \global\let\author\relax
1689   \global\let\date\relax
1690   \global\let\and\relax
1691   \ifx\authors@mail\@undefined\else{%
1692     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1693     \footnotetext[0]{\itshape\authors@mail}%
1694   }\fi
1695   \global\let\authors@mail\@undefined}
1696 \def\@maketitle{%
1697   \newpage\null
1698   \vskip 6em % used to be 2em
1699   \begin{center}
1700     \let\footnote\thanks
1701     \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1702     \lineskip .5em
1703     \ifx\@author\@undefined\else
1704       \vskip 1em
1705       \begin{tabular}[t]{c}%
1706         \@author
1707       \end{tabular}\par
1708     \fi
1709     \ifx\@etitle\@undefined\else
1710       \vskip 1em
1711       {\large \@etitle \par}%
1712     \fi
1713     \ifx\@eauthor\@undefined\else
1714       \vskip 1em
1715       \begin{tabular}[t]{c}%

```



```

1716      \@eauthor
1717      \end{tabular}\par
1718      \fi
1719      \vskip 1em
1720      \@date
1721      \end{center}
1722      \vskip 1.5em
1723      \centerline{\box\@abstractbox}
1724      \ifx\@keywords\@undefined\else
1725      \vskip 1.5em
1726      \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1727      \fi
1728      \vskip 1.5em}
1729 %</jspf>

```

## 8.2 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、見出しの下側の空きです。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。

```

1730 \def\startsection#1#2#3#4#5#6{%
1731   \if@noskipsec \leavevmode \fi
1732   \par

```

```

1733 % 見出し上の空きを \@tempskipa にセットする
1734 \@tempskipa #4\relax
1735 % \@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
1736 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1737 % 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
1738 \ifdim \@tempskipa <\z@
1739     \@tempskipa -\@tempskipa \@afterindentfalse
1740 \fi
1741 \if@nobreak
1742 %     \everypar{\everyparhook}% これは間違い
1743     \everypar{}%
1744 \else
1745     \addpenalty\@secpenalty
1746 % 次の行は削除
1747 %     \addvspace\@tempskipa
1748 % 次の \noindent まで追加
1749     \ifdim \@tempskipa >\z@
1750         \if@slide\else
1751             \null
1752             \vspace*{-\baselineskip}%
1753         \fi
1754         \vskip\@tempskipa
1755     \fi
1756 \fi
1757 \noindent
1758 % 追加終わり
1759 \@ifstar
1760     {\@ssect{#3}{#4}{#5}{#6}}%
1761     {\@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。

```

1762 \def\@sect#1#2#3#4#5#6[#7]#8{%
1763     \ifnum #2>\c@secnumdepth
1764         \let\@svsec\@empty
1765     \else
1766         \refstepcounter{#1}%
1767         \protected@edef\@svsec{\@secntformat{#1}\relax}%
1768     \fi
1769 % 見出し後の空きを \@tempskipa にセット
1770 \@tempskipa #5\relax
1771 % 条件判断の順序を入れ替えました
1772 \ifdim \@tempskipa<\z@
1773     \def\@svsechd{%
1774         #6{\hskip #3\relax
1775             \@svsec #8}%
1776         \csname #1mark\endcsname{#7}%
1777         \addcontentsline{toc}{#1}{%
1778             \ifnum #2>\c@secnumdepth \else

```

```

1779         \protect\numberline{\csname the#1\endcsname}%
1780     \fi
1781     #7}}% 目次にフルネームを載せるなら #8
1782 \else
1783     \begingroup
1784     \interlinepenalty \@M % 下から移動
1785     #6{%
1786         \@hangfrom{\hskip #3\relax\@svsec}%
1787 %     \interlinepenalty \@M % 上に移動
1788     #8\@@par}%
1789 \endgroup
1790 \csname #1mark\endcsname{#7}%
1791 \addcontentsline{toc}{#1}{%
1792     \ifnum #2>\c@secnumdepth \else
1793         \protect\numberline{\csname the#1\endcsname}%
1794     \fi
1795     #7}}% 目次にフルネームを載せるならここは #8
1796 \fi
1797 \@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。

[2016-07-28] `slide` オプションと `twocolumn` オプションを同時に指定した場合の罫線の位置を微調整しました。

```

1798 \def\@xsect#1{%
1799 % 見出しの後ろの空きを \@tempskipa にセット
1800 \@tempskipa #1\relax
1801 % 条件判断の順序を変えました
1802 \ifdim \@tempskipa<\z@
1803     \@nobreakfalse
1804     \global\@noskipsectrue
1805     \everypar{%
1806         \if@noskipsec
1807             \global\@noskipsecfalse
1808             {\setbox\z@\lastbox}%
1809             \clubpenalty\@M
1810             \begingroup \@svsechd \endgroup
1811             \unskip
1812             \@tempskipa #1\relax
1813             \hskip -\@tempskipa
1814             \bxjs<ltj@inhibitglue
1815         \else
1816             \clubpenalty \@clubpenalty
1817             \everypar{\everyparhook}%
1818         \fi\everyparhook}%
1819 \else
1820     \par \nobreak
1821     \vskip \@tempskipa

```

```

1822 \afterheading
1823 \fi
1824 \if@slide
1825 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1826 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1827 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1828 \fi
1829 \par % 2000-12-18
1830 \ignorespaces}
1831 \def\@ssect#1#2#3#4#5{%
1832 \@tempskipa #3\relax
1833 \ifdim \@tempskipa<\z@
1834 \def\@svsechd{#4{\hskip #1\relax #5}}%
1835 \else
1836 \begingroup
1837 #4{%
1838 \@hangfrom{\hskip #1}%
1839 \interlinepenalty \@M #5\@@par}%
1840 \endgroup
1841 \fi
1842 \@xsect{#3}}

```

---

上記の定義中の `\bxjs@ltj@inhibitglue` は Lua<sub>T</sub><sub>E</sub>X-j<sub>a</sub> で用いられるフック。

```

1843 \let\bxjs@ltj@inhibitglue\@empty

```

---

## ■柱関係の命令

`\chaptermark` `\...mark` の形の命令を初期化します (第 7 節参照)。`\chaptermark` 以外は L<sup>A</sup>T<sub>E</sub>X 本体で定義済みです。

```

\subsectionmark 1844 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1845 % \newcommand*\sectionmark[1]{}
1846 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1847 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1848 % \newcommand*\paragraphmark[1]{}
1849 % \newcommand*\subparagraphmark[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決めるカウンタです。

```

1850 %<!book&!report>\setcounter{secnumdepth}{3}
1851 %<book|report>\setcounter{secnumdepth}{2}

```

`\c@chapter` 見出し番号のカウンタです。`\newcounter` の第 1 引数が新たに作るカウンタです。これは

`\c@section` 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。

```

\c@subsection 1852 \newcounter{part}
\c@subsubsection 1853 %<book|report>\newcounter{chapter}

\c@paragraph
\c@subparagraph

```

```

1854 %<book|report>\newcounter{section}[chapter]
1855 %<!book&!report>\newcounter{section}
1856 \newcounter{subsection}[section]
1857 \newcounter{subsubsection}[subsection]
1858 \newcounter{paragraph}[subsubsection]
1859 \newcounter{subparagraph}[paragraph]

```

`\thepart` カウンタの値を出力する命令 `\the` 何々 を定義します。

`\thechapter` 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。

```

1860 \renewcommand{\thepart}{\@Roman\c@part}
1861 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1862 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1863 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1864 %<*book|report>
1865 \renewcommand{\thechapter}{\@arabic\c@chapter}
1866 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1867 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1868 %</book|report>
1869 \renewcommand{\thesubsubsection}{%
1870   \thesubsection.\@arabic\c@subsubsection}
1871 \renewcommand{\theparagraph}{%
1872   \thesubsubsection.\@arabic\c@paragraph}
1873 \renewcommand{\thesubparagraph}{%
1874   \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。

```

1875 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1876 %<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。

[2017-03-05] `\frontmatter` と `\mainmatter` の 2 つの命令は, 改丁または改ページした  
後で `\pagenumbering{...}` でノンブルを 1 にリセットします。長い間 `\frontmatter` は

openany のときに単なる改ページとしていましたが、これではノンブルをリセットする際に偶奇逆転が起こる場合があります。openany かどうかに関らず奇数ページまで繰るように修正することで、問題を解消しました。実は、L<sup>A</sup>T<sub>E</sub>X の標準クラスでは 1998 年に修正されていた問題です（コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の標準クラス 2017/03/05 も参照）。

```
1877 %<*book|report>
1878 \newcommand\frontmatter{%
1879   \pltx@cleartooddpage
1880   \@mainmatterfalse
1881   \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。

```
1882 \newcommand\mainmatter{%
1883   \pltx@cleartooddpage
1884   \@mainmattertrue
1885   \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。

```
1886 \newcommand\backmatter{%
1887   \if@openleft
1888     \cleardoublepage
1889   \else\if@openright
1890     \cleardoublepage
1891   \else
1892     \clearpage
1893   \fi\fi
1894   \@mainmatterfalse}
1895 %</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 と report のクラス以外です。

```
1896 %<*&!book&!report>
1897 \newcommand\part{%
1898   \if@noskipsec \leavevmode \fi
```

```

1899 \par
1900 \addvspace{4ex}%
1901 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1902 \secdef\@part\@spart}
1903 %<*/!book&!report>

```

book および report クラスの場合は、少し複雑です。

```

1904 %<*book|report>
1905 \newcommand\part{%
1906   \if@openleft
1907     \cleardoublepage
1908   \else\if@openright
1909     \cleardoublepage
1910   \else
1911     \clearpage
1912   \fi\fi
1913   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1914   \if@twocolumn
1915     \onecolumn
1916     \@restonecoltrue
1917   \else
1918     \@restonecolfalse
1919   \fi
1920   \null\vfil
1921   \secdef\@part\@spart}
1922 %</book|report>

```

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

book および report 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。

```

1923 %<*!book&!report>
1924 \def\@part[#1]#2{%
1925   \ifnum \c@secnumdepth >\m@ne
1926     \refstepcounter{part}%
1927     \addcontentsline{toc}{part}{%
1928       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1929   \else
1930     \addcontentsline{toc}{part}{#1}%
1931   \fi
1932   \markboth{}{}%
1933   {\parindent\z@
1934     \raggedright
1935     \interlinepenalty \@M
1936     \normalfont
1937     \ifnum \c@secnumdepth >\m@ne
1938       \Large\headfont\prepartname\thepart\postpartname
1939       \par\nobreak
1940     \fi
1941     \huge \headfont #2%

```

```

1942 \markboth{}{}\par}%
1943 \nobreak
1944 \vskip 3ex
1945 \@afterheading}
1946 %</!book&!report>

```

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

```

1947 %<*book|report>
1948 \def\@part[#1]#2{%
1949 \ifnum \c@secnumdepth >-2\relax
1950 \refstepcounter{part}%
1951 \addcontentsline{toc}{part}{%
1952 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1953 \else
1954 \addcontentsline{toc}{part}{#1}%
1955 \fi
1956 \markboth{}{}\%
1957 {\centering
1958 \interlinepenalty \@M
1959 \normalfont
1960 \ifnum \c@secnumdepth >-2\relax
1961 \huge\headfont \prepartname\thepart\postpartname
1962 \par\vskip20\p@?
1963 \fi
1964 \Huge \headfont #2\par}%
1965 \@endpart}
1966 %</book|report>

```

\@spart 番号を付けない部です。

```

1967 %<*!book&!report>
1968 \def\@spart#1{%
1969 \parindent \z@ \raggedright
1970 \interlinepenalty \@M
1971 \normalfont
1972 \huge \headfont #1\par}%
1973 \nobreak
1974 \vskip 3ex
1975 \@afterheading}
1976 %</!book&!report>
1977 %<*book|report>
1978 \def\@spart#1{%
1979 \centering
1980 \interlinepenalty \@M
1981 \normalfont
1982 \Huge \headfont #1\par}%
1983 \@endpart}
1984 %</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。

[2016-12-13] `openany` のときには白ページが追加されるのは変なので、その場合は追加しないようにしました。このバグは L<sup>A</sup>T<sub>E</sub>X では `classes.dtx` v1.4b (2000/05/19) で修正されています。

```
1985 %<*book|report>
1986 \def\@endpart{\vfil\newpage
1987   \if@twoside
1988     \if@openleft %% added (2017/02/24)
1989       \null\thispagestyle{empty}\newpage
1990     \else\if@openright %% added (2016/12/13)
1991       \null\thispagestyle{empty}\newpage
1992     \fi\fi %% added (2016/12/13, 2017/02/24)
1993   \fi
1994   \if@restonecol
1995     \twocolumn
1996   \fi}
1997 %</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。

```
1998 %<*book|report>
1999 \newcommand{\chapter}{%
2000   \if@openleft\cleardoublepage\else
2001   \if@openright\cleardoublepage\else\clearpage\fi\fi
2002   \plainifnotempty % 元: \thispagestyle{plain}
2003   \global\@topnum\z@
2004   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2005   \secdef
2006     {\@omit@numberfalse\@chapter}%
2007     {\@omit@numbertrue\@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。

```
2008 \def\@chapter[#1]#2{%
2009   \ifnum \c@secnumdepth >\m@ne
2010     \if@mainmatter
2011       \refstepcounter{chapter}%
2012       \typeout{\@chapapp\thechapter\@chappos}%
2013       \addcontentsline{toc}{chapter}%
2014         {\protect\numberline
2015 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2016         {\@chapapp\thechapter\@chappos}%
2017         #1}%
2018     \else\addcontentsline{toc}{chapter}{#1}\fi
2019   \else
```

```

2020 \addcontentsline{toc}{chapter}{#1}%
2021 \fi
2022 \chaptermark{#1}%
2023 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
2024 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
2025 \if@twocolumn
2026 \topnewpage[\@makechapterhead{#2}]%
2027 \else
2028 \@makechapterhead{#2}%
2029 \afterheading
2030 \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。

```

2031 \def\@makechapterhead#1{%
2032 \vspace*{2\Cvs}% 欧文は 50pt
2033 {\parindent \z@ \raggedright \normalfont
2034 \ifnum \c@secnumdepth >\m@ne
2035 \if@mainmatter
2036 \huge\headfont \@chapapp\thechapter\@chappos
2037 \par\nobreak
2038 \vskip \Cvs % 欧文は 20pt
2039 \fi
2040 \fi
2041 \interlinepenalty\@M
2042 \Huge \headfont #1\par\nobreak
2043 \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。

```

2044 \def\@schapter#1{%
2045 \chaptermark{#1}%
2046 \if@twocolumn
2047 \topnewpage[\@makeschapterhead{#1}]%
2048 \else
2049 \@makeschapterhead{#1}\@afterheading
2050 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2051 \def\@makeschapterhead#1{%
2052 \vspace*{2\Cvs}% 欧文は 50pt
2053 {\parindent \z@ \raggedright
2054 \normalfont
2055 \interlinepenalty\@M
2056 \Huge \headfont #1\par\nobreak
2057 \vskip 3\Cvs}} % 欧文は 40pt
2058 %</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。

```
2059 \if@twocolumn
2060   \newcommand{\section}{%
2061     %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2062     \@startsection{section}{1}{\z@}%
2063     %<kiyou>    {0.6\Cvs}{0.4\Cvs}%
2064     %<kiyou>    {\Cvs}{0.5\Cvs}%
2065     %    {\normalfont\large\headfont\@secapp}}
2066     {\normalfont\large\headfont\raggedright}}
2067 \else
2068   \newcommand{\section}{%
2069     \if@slide\clearpage\fi
2070     \@startsection{section}{1}{\z@}%
2071     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2072     {.5\Cvs \@plus.3\Cdp}% 後アキ
2073     %    {\normalfont\Large\headfont\@secapp}}
2074     {\normalfont\Large\headfont\raggedright}}
2075 \fi
```

`\subsection` 同上です。

```
2076 \if@twocolumn
2077   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2078     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2079     {\normalfont\normalsize\headfont}}
2080 \else
2081   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2082     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2083     {.5\Cvs \@plus.3\Cdp}% 後アキ
2084     {\normalfont\large\headfont}}
2085 \fi
```

`\subsubsection` [2016-07-22] slide オプション指定時に `\subsubsection` の文字列と罫線が重なる問題に対処しました (forum:1982)。

```
2086 \if@twocolumn
2087   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2088     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2089     {\normalfont\normalsize\headfont}}
2090 \else
2091   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2092     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2093     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2094     {\normalfont\normalsize\headfont}}
2095 \fi
```

`\paragraph` 見出しの後ろで改行されません。

[2016-11-16] 従来は `\paragraph` の最初に出るマークを「■」に固定していましたが、このマークを変更可能にするため `\jsParagraphMark` というマクロに切り出しました。これ

で、たとえば

```
\renewcommand{\jsParagraphMark}{★}
```

とすれば「★」に変更できますし、マークを空にすることも容易です。なお、某学会クラスでは従来どおりマークは付きません。

---

※ BXJS クラスでは、1.1 版 [2016-02-14] から `\jsParagraphMark` をサポートしている。

段落のマーク (■) が必ず和文フォントで出力されるようにする。

`\jsJaChar` は standard 和文ドライバが読み込まれた場合は `\jachar` と同義になるが、それ以外は何もしない。

---

```
2096 \newcommand\jsParagraphMark{\jsJaChar{■}}
2097 \ifx\bxjs@paragraph@mark\empty
2098   \let\jsParagraphMark\empty
2099 \else\ifx\bxjs@paragraph@mark\undefined\else
2100   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2101 \fi\fi
2102 \let\jsJaChar\empty
2103 \if@twocolumn
2104   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2105     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2106     %<jspf> {\normalfont\normalsize\headfont}}
2107     %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2108   \else
2109     \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2110       {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2111       {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2112       %<jspf> {\normalfont\normalsize\headfont}}
2113       %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2114   \fi
```

`\subparagraph` 見出しの後ろで改行されません。

```
2115 \if@twocolumn
2116   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2117     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2118     {\normalfont\normalsize\headfont}}
2119   \else
2120     \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2121       {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2122       {\normalfont\normalsize\headfont}}
2123   \fi
```

### 8.3 リスト環境

第  $k$  レベルのリストの初期化をするのが `\@listk` です ( $k = i, ii, iii, iv$ )。 `\@listk` は `\leftmargin` を `\leftmargink` に設定します。

`\leftmargini` 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。

```
2124 \if@slide
2125   \setlength\leftmargini{1\jsZw}
2126 \else
2127   \if@twocolumn
2128     \setlength\leftmargini{2\jsZw}
2129   \else
2130     \setlength\leftmargini{3\jsZw}
2131   \fi
2132 \fi
```

`\leftmarginii` ii, iii, iv は `\labelsep` とそれぞれ ‘(m)’, ‘vii.’, ‘M.’ の幅との和より大きくすることになっています。ここでは全角幅の整数倍に丸めました。

```
\leftmarginiv 2133 \if@slide
\leftmarginiv 2134   \setlength\leftmarginii {1\jsZw}
\leftmarginiv 2135   \setlength\leftmarginiii {1\jsZw}
\leftmarginiv 2136   \setlength\leftmarginiv {1\jsZw}
\leftmarginiv 2137   \setlength\leftmarginv {1\jsZw}
\leftmarginiv 2138   \setlength\leftmarginvi {1\jsZw}
\leftmarginiv 2139 \else
\leftmarginiv 2140   \setlength\leftmarginii {2\jsZw}
\leftmarginiv 2141   \setlength\leftmarginiii {2\jsZw}
\leftmarginiv 2142   \setlength\leftmarginiv {2\jsZw}
\leftmarginiv 2143   \setlength\leftmarginv {1\jsZw}
\leftmarginiv 2144   \setlength\leftmarginvi {1\jsZw}
\leftmarginiv 2145 \fi
```

`\labelsep` `\labelsep` はラベルと本文の間の距離です。`\labelwidth` はラベルの幅です。これは二分 `\labelwidth` に変えました。

```
2146 \setlength \labelsep {0.5\jsZw} % .5em
2147 \setlength \labelwidth{\leftmargini}
2148 \addtolength\labelwidth{-\labelsep}
```

`\partopsep` リスト環境の前に空行がある場合, `\parskip` と `\topsep` に `\partopsep` 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。

```
2149 \setlength\partopsep{\z@} % {2\p@ \@plus 1\p@ \@minus 1\p@}
```

`\@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 2150 \@beginparpenalty -\@lowpenalty
\@itempenalty 2151 \@endparpenalty -\@lowpenalty
2152 \@itempenalty -\@lowpenalty
```

`\@listi` `\@listi` は `\leftmargin`, `\parsep`, `\topsep`, `\itemsep` などのトップレベルの定義を `\@listI` 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  $\pm_{0.1}^{0.2}$  `\baselineskip` を思い切って外しました。

```
2153 \def\@listi{\leftmargin\leftmarginI
2154   \parsep \z@
2155   \topsep 0.5\baselineskip
2156   \itemsep \z@ \relax}
2157 \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）。

```
2158 \@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 2159 \def\@listii{\leftmargin\leftmarginii
\@listiv 2160   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2161   \topsep \z@
\@listv 2162   \parsep \z@
\@listvi 2163   \itemsep\parsep}
2164 \def\@listiii{\leftmargin\leftmarginiii
2165   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2166   \topsep \z@
2167   \parsep \z@
2168   \itemsep\parsep}
2169 \def\@listiv {\leftmargin\leftmarginiv
2170   \labelwidth\leftmarginiv
2171   \advance\labelwidth-\labelsep}
2172 \def\@listv {\leftmargin\leftmarginv
2173   \labelwidth\leftmarginv
2174   \advance\labelwidth-\labelsep}
2175 \def\@listvi {\leftmargin\leftmarginvi
2176   \labelwidth\leftmarginvi
2177   \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 L<sup>A</sup>T<sub>E</sub>X 本体 (`ltlists.dtx` 参照) で定義済みですが、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。

```
2178 \renewcommand{\theenumi}{\@arabic\c@enumi}
2179 \renewcommand{\theenumii}{\@alph\c@enumii}
2180 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2181 \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  
`\labelenumii` 換え、その両側に入る余分なグルーを `\inhibitglue` で取り除いています。

`\labelenumiv`

---

和文の括弧で囲むための補助命令 `\jsInJaParen` を定義して `\labelenumii` でそれを用いている。

※現状の `zxjatype` の `\inhibitglue` の実装には「前後のグルーを消してしまう」という不備があって、そのため `enumii` の出力が異常になるという不具合があった。`zxjatype` を修正するまでの回避策として、サイズがゼロの罫 (`\bxjs@dust`) でガードしておく。

---

```
2182 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2183 \newcommand*\jsInJaParen}[1]{%
2184   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2185 \newcommand{\labelenumi}{\theenumi.}
2186 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
2187 \newcommand{\labelenumiii}{\theenumiii.}
2188 \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  
`\p@enumiii` の書式です。これも第 2 レベルは和文用かっこにしました。

```
\p@enumiv 2189 \renewcommand{\p@enumii}{\theenumi}
2190 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
2191 \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。

```
\labelitemii 2192 \newcommand\labelitemi{\textbullet}
2193 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 2194 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2195 \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。

```
2196 \newenvironment{description}{%
2197   \list{}{%
2198     \labelwidth=\leftmargin
2199     \labelsep=1\jsZw
2200     \advance \labelwidth by -\labelsep
2201     \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  
 (たとえば `\hspace{1\jsZw}`) を入れるのもいいと思います。

```
2202 \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。

---

**bxjsreport** クラスの **abstract** 環境は：

- **layout=v1** の場合は **jsbook + report** の動作を継承する。つまり **jsbook** と同じになる。
- **layout=v2** の場合は新設の **jsreport** の動作を継承する。つまり **jsarticle (+titlepage)** と同じになる。

**chapterabstract** **jsbook** の **abstract** 環境（「各章の初めにちょっとしたことを書く」ためのもの）を **chapterabstract** と呼ぶことにする。

```
2203 %<*book|report>
2204 \newenvironment{chapterabstract}{%
2205   \begin{list}{}{%
2206     \listparindent=1\jsZw
2207     \itemindent=\listparindent
2208     \rightmargin=\z@
2209     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2210 %</book|report>
```

“普通の” **abstract** 環境の定義。

```
2211 %<*article|report|slide>
2212 \newbox\@abstractbox
2213 \if@titlepage
2214   \newenvironment{abstract}{%
2215     \titlepage
2216     \null\vfil
2217     \@beginparpenalty\@lowpenalty
2218     \begin{center}%
2219       \headfont \abstractname
2220       \@endparpenalty\@M
2221     \end{center}%
```

BXJS クラスでは、概要の最初の段落に段落下げが入るようにする。

```
2222   \par}%
2223   {\par\vfil\null\endtitlepage}
2224 \else
2225   \newenvironment{abstract}{%
2226     \if@twocolumn
2227       \ifx\maketitle\relax
2228         \section*{\abstractname}%
```



```

2229 \else
2230 \global\setbox\@abstractbox\hbox\bgroup
2231 \begin{minipage}[b]{\textwidth}
2232 \small\parindent1\jsZw
2233 \begin{center}%
2234 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2235 \end{center}%
2236 \list{}{%
2237 \listparindent\parindent
2238 \itemindent \listparindent
2239 \rightmargin \leftmargin}%
2240 \item\relax
2241 \fi
2242 \else
2243 \small
2244 \begin{center}%
2245 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2246 \end{center}%
2247 \list{}{%
2248 \listparindent\parindent
2249 \itemindent \listparindent
2250 \rightmargin \leftmargin}%
2251 \item\relax
2252 \fi}{\if@twocolumn
2253 \ifx\maketitle\relax
2254 \else
2255 \endlist\end{minipage}\egroup
2256 \fi
2257 \else
2258 \endlist
2259 \fi}
2260 \fi
2261 %</article|report|slide>
2262 %<*jspf>
2263 \newbox\@abstractbox
2264 \newenvironment{abstract}{%
2265 \global\setbox\@abstractbox\hbox\bgroup
2266 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2267 \small
2268 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2269 {\end{minipage}\egroup}
2270 %</jspf>

```

bxjs@force@chapterabstract が真の場合は、abstract 環境を chapterabstract 環境と等価にする。

```

2271 %<*book|report>
2272 \ifbxjs@force@chapterabstract
2273 \let\abstract\chapterabstract
2274 \let\endabstract\endchapterabstract

```

```
2275 \fi
2276 %</book|report>
```

---

## ■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。

```
2277 %<*jspf>
2278 %\newbox\@keywordsbox
2279 %\newenvironment{keywords}{%
2280 % \global\setbox\@keywordsbox\hbox\bgroup
2281 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2282 % \small\parindent0\jsZw}%
2283 % {\end{minipage}\egroup}
2284 %</jspf>
```

## ■verse 環境

**verse** 詩のための `verse` 環境です。

```
2285 \newenvironment{verse}{%
2286 \let \=\@centercr
2287 \list{}{%
2288 \itemsep \z@
2289 \itemindent -2\jsZw % 元: -1.5em
2290 \listparindent\itemindent
2291 \rightmargin \z@
2292 \advance\leftmargin 2\jsZw}% 元: 1.5em
2293 \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。

```
2294 \newenvironment{quotation}{%
2295 \list{}{%
2296 \listparindent\parindent
2297 \itemindent\listparindent
2298 \rightmargin \z@}%
2299 \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。

```
2300 \newenvironment{quote}%
2301 {\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。

[2009-08-23] `\bfseries` を `\headfont` に直し、`\labelsep` を 1zw にし、括弧を全角にしました。

```
2302 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2303   \item[\hskip \labelsep{\headfont #1\ #2}]}
2304 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2305   \item[\hskip \labelsep{\headfont #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。

[2017-02-24] コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の標準クラス 2017/02/15 に合わせて、book クラスでタイトルを必ず奇数ページに送るようにしました。といっても、横組クラスしかありませんでしたので、従来の挙動は何も変わっていません。また、book 以外の場合のページ番号のリセットもコミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の標準クラス 2017/02/15 に合わせましたが、こちらも片面印刷あるいは独立のタイトルページを作らないクラスばかりでしたので、従来の挙動は何も変わらずに済みました。

```
2306 \newenvironment{titlepage}{%
2307 %<book> \pltx@cleartooddpage %% 2017-02-24
2308   \if@twocolumn
2309     \@restonecoltrue\onecolumn
2310   \else
2311     \@restonecolfalse\newpage
2312   \fi
2313   \thispagestyle{empty}%
2314   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2315   24
2316 }%
2317 {\if@restonecol\twocolumn \else \newpage \fi
2318   \if@twoside\else
2319     \setcounter{page}\@ne
2320   \fi}
```

## ■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2320 %<*&!book&!report>
2321 \newcommand{\appendix}{\par
2322   \setcounter{section}{0}%
2323   \setcounter{subsection}{0}%
2324   \gdef\presectionname{\appendixname}%

```

```

2325 \gdef\postsectionname{}%
2326 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2327 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2328 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2329 %<!/book&!report>
2330 %<*book|report>
2331 \newcommand{\appendix}{\par
2332 \setcounter{chapter}{0}%
2333 \setcounter{section}{0}%
2334 \gdef\@chapapp{\appendixname}%
2335 \gdef\@chappos{}}%
2336 \gdef\thechapter{\@Alph\c@chapter}}
2337 %</book|report>

```

## 8.4 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。

```
2338 \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。

```
2339 \setlength\tabcolsep{6\p@?}
```

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

```
2340 \setlength\arrayrulewidth{.4\p@}
```

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

```
2341 \setlength\doublerulesep{2\p@}
```

### ■tabbing 環境

`\tabbingsep` \ ' コマンドで入るアキです。

```
2342 \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。

```
2343 \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。

```
2344 \setlength\fboxsep{3\p@?}
```

```
2345 \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。

```
2346 %<!*book&!report>\renewcommand \theequation {\@arabic\c@equation}
2347 %<*book|report>
2348 \@addtoreset{equation}{chapter}
2349 \renewcommand\theequation
2350 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
2351 %</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。

```
2352 % \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。

```
2353 % \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。

```
2354 % \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip\@italiccorr)}}
```

## 8.5 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。

```
2355 %<!*book&!report>
2356 \newcounter{figure}
2357 \renewcommand \thefigure {\@arabic\c@figure}
2358 %</!book&!report>
2359 %<*book|report>
2360 \newcounter{figure}[chapter]
2361 \renewcommand \thefigure
```

```

2362      {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2363 %</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  
`\ftype@figure` しました。

```

\ext@figure 2364 \def\fps@figure{tbp}
2365 \def\ftype@figure{1}
\fnun@figure 2366 \def\ext@figure{lof}
2367 \def\fnun@figure{\figurename\nobreak\thefigure}

```

`figure` \* 形式は段抜きのフロートです。

```

figure* 2368 \newenvironment{figure}%
2369      {\@float{figure}}%
2370      {\end@float}
2371 \newenvironment{figure*}%
2372      {\@dblfloat{figure}}%
2373      {\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。

```

2374 %<*&!book&!report>
2375 \newcounter{table}
2376 \renewcommand\thetable{\@arabic\c@table}
2377 %<*&!book&!report>
2378 %<*&book|report>
2379 \newcounter{table}[chapter]
2380 \renewcommand \thetable
2381      {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2382 %</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  
`\ftype@table` した。

```

\ext@table 2383 \def\fps@table{tbp}
2384 \def\ftype@table{2}
\fnun@table 2385 \def\ext@table{lot}
2386 \def\fnun@table{\tablename\nobreak\thetable}

```

`table` \* は段抜きのフロートです。

```

table* 2387 \newenvironment{table}%
2388      {\@float{table}}%
2389      {\end@float}
2390 \newenvironment{table*}%
2391      {\@dblfloat{table}}%
2392      {\end@dblfloat}

```

## 8.6 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 になっていましたので、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。

```
2393 \newlength\abovecaptionskip
2394 \newlength\belowcaptionskip
2395 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2396 \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。

```
2397 %<*\jpf>
2398 % \long\def\@makecaption#1#2{{\small
2399 %   \advance\leftskip 10\jsc@mmm
2400 %   \advance\rightskip 10\jsc@mmm
2401 %   \vskip\abovecaptionskip
2402 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2403 %   \ifdim \wd\@tempboxa >\hsize
2404 %     #1\hskip1\jsZw\relax #2\par
2405 %   \else
2406 %     \global \@minipagefalse
2407 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2408 %   \fi
2409 %   \vskip\belowcaptionskip}}
2410 \long\def\@makecaption#1#2{{\small
2411   \advance\leftskip .0628\linewidth
2412   \advance\rightskip .0628\linewidth
2413   \vskip\abovecaptionskip
2414   \sbox\@tempboxa{#1\zwspace#2}%
2415   \ifdim \wd\@tempboxa <\hsize \centering \fi
2416   #1\zwspace#2\par
2417   \vskip\belowcaptionskip}}
2418 %</!\jpf>
2419 %<*\jpf>
2420 \long\def\@makecaption#1#2{%
2421   \vskip\abovecaptionskip
2422   \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2423   \ifdim \wd\@tempboxa >\hsize
2424     {\small\sffamily
2425       \list{#1}{%
2426         \renewcommand{\makelabel}[1]{##1\hfil}
2427         \itemsep \z@
```

```

2428      \itemindent \z@
2429      \labelsep \z@
2430      \labelwidth 11\jsc@mmm
2431      \listparindent\z@
2432      \leftmargin 11\jsc@mmm\item\relax #2\endlist}
2433 \else
2434   \global \@minipagefalse
2435   \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2436 \fi
2437 \vskip\belowcaptionskip}
2438 %</jspf>

```

## 9 フォントコマンド

ここでは L<sup>A</sup>T<sub>E</sub>X 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。

[2016-07-15] KOMA-Script 中の `\scr@DeclareOldFontCommand` に倣い、これらの命令を使うときには警告を発することにしました。

[2016-07-16] 警告を最初の一回だけ発することにしました。また、例外的に警告を出さないようにするスイッチも付けます。

---

```

\if@jsc@warnoldfontcmd
\if@jsc@warnoldfontcmdexception
\if@jsc@warnoldfontcmd
\if@jsc@warnoldfontcmdexception

```

---

`\if@jsc@warnoldfontcmd` は BXJS クラスでは不使用。

`\if@jsc@warnoldfontcmdexception` は `\allow/disallowoldfontcommands` の状態を表す。

---

```

2439 \newif\if@jsc@warnoldfontcmd
2440 \@jsc@warnoldfontcmdtrue
2441 \newif\if@jsc@warnoldfontcmdexception
2442 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand

2443 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2444   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2445   \DeclareOldFontCommand{#1}{%
2446     \bxjs@oldfontcmd{#1}#2%
2447   }{%
2448     \bxjs@oldfontcmd{#1}#3%
2449   }%
2450 }
2451 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2452   \ClassInfo\bxjs@clsname
2453   {Old font command '\string#1' is used!!\MessageBreak
2454     The first occurrence is}%
2455 }

```



---

`\allowoldfontcommands` “二文字フォント命令”の使用を許可する（警告しない）。

`\disallowoldfontcommands` “二文字フォント命令”の使用に対して警告を出す。

```
2456 \newcommand*{\allowoldfontcommands}{%
2457   \@jsc@warnoldfontcmdexceptiontrue}
2458 \newcommand*{\disallowoldfontcommands}{%
2459   \@jsc@warnoldfontcmdexceptionfalse}
```

※ 1.x 版では Warning ではなく Info に留めておく。

```
2460 \let\bxjs@oldfontcmd@list\@empty
2461 \def\bxjs@oldfontcmd#1{%
2462   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2463 \def\bxjs@oldfontcmd@a#1#2{%
2464   \if@jsc@warnoldfontcmdexception\else
2465     \global\@jsc@warnoldfontcmdfalse
2466     \ifx#1\relax
2467       \global\let#1=t%
2468       \jsc@warnoldfontcmd{#2}%
2469     \fi
2470   \fi}
2471 \def\bxjs@warnoldfontcmd@final{%
2472 % \par
2473   \global\let\bxjs@warnoldfontcmd@final\@empty
2474   \let\@tempa\@empty
2475   \def\do##1{%
2476     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2477       \edef\@tempa{\@tempa \space\string##1}\fi}
2478   \bxjs@oldfontcmd@list
2479   \ifx\@tempa\@empty\else
2480     \ClassWarningNoLine\bxjs@clsname
2481       {Some old font commands were used in text\MessageBreak
2482        (see the log file for detail)}%
2483     \ClassInfo\bxjs@clsname
2484       {Some old font commands were used in text:\MessageBreak
2485        \space\@tempa\MessageBreak
2486        You should note, that since 1994 LaTeX2e provides a\MessageBreak
2487        new font selection scheme called NFSS2 with several\MessageBreak
2488        new, combinable font commands. The
2489        class provides\MessageBreak
2490        the old font commands
2491        only for compatibility%
2492     \@gobble}%
2493   \fi}
```

単純に `\AtEndDocument` のフックの中で `\bxjs@warnoldfontcmd@final` を実行した場合、最終ページのヘッダ・フッタの中にある二文字フォント命令はそれより後に実行されるため捕捉できない。これに対処するため、`\end{document}` 中に実行される `\clearpage` の

処理の直後に `\bxjs...final` が呼ばれるようにする。

```
2494 \def\bxjs@warnoldfontcmd@kick@final{%
2495   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2496 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}
```

---

`\mc` フォントファミリーを変更します。

```
\gt 2497 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2498 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 2499 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\tt 2500 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2501 \jsc@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すコマンドは `\mdseries` です。

```
2502 \jsc@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ップスは数式中では何もしま  
`\sl` 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すコマンドは `\upshape`  
`\sc` です。

```
2503 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2504 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2505 \jsc@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 2506 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2507 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

## 10 相互参照

### 10.1 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。

```
2508 \newcommand\@pnumwidth{1.55em}
2509 \newcommand\@tocrmarg{2.55em}
2510 \newcommand\@dotsep{4.5}
2511 %<!book&!report>\setcounter{tocdepth}{2}
2512 %<book|report>\setcounter{tocdepth}{1}
```

## ■目次

`\tableofcontents` 目次を生成します。

`\jsc@tocl@width` [2013-12-30] `\prechaptername` などから見積もった目次のラベルの長さです。(by ts)

```
2513 \newdimen\jsc@tocl@width
2514 \newcommand{\tableofcontents}{%
2515 %<*book|report>
2516   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2517   \settowidth\@tempdima{\headfont\appendixname}%
2518   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2519   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2520   \if@twocolumn
2521     \@restonecoltrue\onecolumn
2522   \else
2523     \@restonecolfalse
2524   \fi
2525   \chapter*{\contentsname}%
2526   \@mkboth{\contentsname}{}%
2527 %</book|report>
2528 %<*&!book&!report>
```

```

2529 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2530 \settowidth\@tempdima{\headfont\appendixname}%
2531 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2532 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2533 \section*{\contentsname}%
2534 \@mkboth{\contentsname}{\contentsname}%
2535 %<!/book&!report>
2536 \@starttoc{toc}%
2537 %<book|report> \if@restonecol\twocolumn\fi
2538 }

```

\l@part 部の目次です。

```

2539 \newcommand*{\l@part}[2]{%
2540 \ifnum \c@tocdepth >-2\relax
2541 %<!book&!report> \addpenalty\@secpenalty
2542 %<book|report> \addpenalty{-\@highpenalty}%
2543 \addvspace{2.25em \@plus\p@}%
2544 \begingroup
2545 \parindent \z@
2546 % \@pnumwidth should be \@tocrmarg
2547 % \rightskip \@pnumwidth
2548 \rightskip \@tocrmarg
2549 \parfillskip -\rightskip
2550 {\leavevmode
2551 \large \headfont
2552 \setlength\@lnumwidth{4\jsZw}%
2553 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2554 \nobreak
2555 %<book|report> \global\@nobreaktrue
2556 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2557 \endgroup
2558 \fi}

```

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

[2013-12-30] \@lnumwidth を \jsc@tocl@width から決めるようにしてみました。(by ts)

```

2559 %<*book|report>
2560 \newcommand*{\l@chapter}[2]{%
2561 \ifnum \c@tocdepth >\m@ne
2562 \addpenalty{-\@highpenalty}%
2563 \addvspace{1.0em \@plus\p@}
2564 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2565 \begingroup
2566 \parindent\z@
2567 % \rightskip\@pnumwidth
2568 \rightskip\@tocrmarg
2569 \parfillskip-\rightskip
2570 \leavevmode\headfont
2571 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi

```

```

2572 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2573 \advance\leftskip\@lnumwidth \hskip-\leftskip
2574 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2575 \penalty\@highpenalty
2576 \endgroup
2577 \fi}
2578 %</book|report>

```

\l@section 節の目次です。

```

2579 %<*&book&!report>
2580 \newcommand*{\l@section}[2]{%
2581 \ifnum \c@tocdepth >\z@
2582 \addpenalty{\@secpenalty}%
2583 \addvspace{1.0em \@plus\p@}%
2584 \begingroup
2585 \parindent\z@
2586 % \rightskip\@pnumwidth
2587 \rightskip\@tocrmarg
2588 \parfillskip-\rightskip
2589 \leavevmode\headfont
2590 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2591 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2592 \advance\leftskip\@lnumwidth \hskip-\leftskip
2593 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2594 \endgroup
2595 \fi}
2596 %</!book&!report>

```

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

```

2597 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

\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 [2013-12-30] こども \jsc@tocl@width から決めるようにしてみました。(by ts)

```

\l@subparagraph 2598 %<*&book&!report>
2599 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2600 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2601 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2602 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2603 %
2604 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2605 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2606 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2607 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2608 %
2609 \newcommand*{\l@subsection}{%
2610 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw

```

```

2611 \dottedtocline{2}{\@tempdima}{3\jsZw}}
2612 \newcommand*\l@subsubsection}{%
2613 \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2614 \dottedtocline{3}{\@tempdima}{4\jsZw}}
2615 \newcommand*\l@paragraph}{%
2616 \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2617 \dottedtocline{4}{\@tempdima}{5\jsZw}}
2618 \newcommand*\l@subparagraph}{%
2619 \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2620 \dottedtocline{5}{\@tempdima}{6\jsZw}}
2621 %</!book&!report>
2622 %<*book|report>
2623 % \newcommand*\l@section}{\@dottedtocline{2}{3.8em}{3.2em}}
2624 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2625 % \newcommand*\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
2626 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
2627 \newcommand*\l@section}{%
2628 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2629 \dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2630 \newcommand*\l@subsubsection}{%
2631 \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2632 \dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2633 \newcommand*\l@subsubsection}{%
2634 \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2635 \dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2636 \newcommand*\l@paragraph}{%
2637 \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2638 \dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2639 \newcommand*\l@subparagraph}{%
2640 \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2641 \dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2642 %</book|report>

```

\numberline 欧文版 L<sup>A</sup>T<sub>E</sub>X では \numberline{...} は幅 \@tempdima の箱に左詰めで出力する命令で  
 \@lnumwidth 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。

```

2643 \newdimen\@lnumwidth
2644 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

\dottedtocline L<sup>A</sup>T<sub>E</sub>X 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。

```

2645 \def\dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2646 \vskip \z@ \@plus.2\p@?
2647 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2648 \parindent #2\relax\@afterindenttrue
2649 \interlinepenalty\@M
2650 \leavevmode

```

```

2651 \lnumwidth #3\relax
2652 \advance\leftskip \lnumwidth \null\nobreak\hskip -\leftskip
2653 {#4}\nobreak
2654 \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2655 mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2656 \hfil\normalfont \normalcolor #5}\par}\fi}

```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2657 \newcommand{\listoffigures}{%
2658 %<*book|report>
2659 \if@twocolumn\@restonecoltrue\onecolumn
2660 \else\@restonecolfalse\fi
2661 \chapter*{\listfigurename}%
2662 \@mkboth{\listfigurename}{}%
2663 %</book|report>
2664 %<*&!book&!report>
2665 \section*{\listfigurename}%
2666 \@mkboth{\listfigurename}{\listfigurename}%
2667 %</&!book&!report>
2668 \@starttoc{lof}%
2669 %<book|report> \if@restonecol\twocolumn\fi
2670 }

```

`\l@figure` 図目次の項目を出力します。

```

2671 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsw}{3.683\jsw}}

```

`\listoftables` 表目次を出力します。

```

2672 \newcommand{\listoftables}{%
2673 %<*book|report>
2674 \if@twocolumn\@restonecoltrue\onecolumn
2675 \else\@restonecolfalse\fi
2676 \chapter*{\listtablename}%
2677 \@mkboth{\listtablename}{}%
2678 %</book|report>
2679 %<*&!book&!report>
2680 \section*{\listtablename}%
2681 \@mkboth{\listtablename}{\listtablename}%
2682 %</&!book&!report>
2683 \@starttoc{lot}%
2684 %<book|report> \if@restonecol\twocolumn\fi
2685 }

```

`\l@table` 表目次は図目次と同じです。

```

2686 \let\l@table\l@figure

```

## 10.2 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。

```
2687 \newdimen\bibindent
2688 \setlength\bibindent{2\jsZw}
```

`thebibliography` 参考文献リストを出力します。

[2016-07-16] L<sup>A</sup>T<sub>E</sub>X 2.09 で使われていたフォントコマンドの警告を、文献スタイル (.bst) ではよく `\bf` がいまだに用いられることが多いため、`thebibliography` 環境内では例外的に出さないようにしました。

```
2689 \newenvironment{thebibliography}[1]{%
2690   \@jsc@warnoldfontcmdexceptiontrue
2691   \global\let\presectionname\relax
2692   \global\let\postsectionname\relax
2693   %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2694   %<*kiyou>
2695   \vspace{1.5\baselineskip}
2696   \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2697   \vspace{0.5\baselineskip}
2698   %</kiyou>
2699   %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2700   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2701   \list{\@biblabel{\@arabic\c@enumiv}}%
2702     {\settowidth\labelwidth{\@biblabel{#1}}%
2703     \leftmargin\labelwidth
2704     \advance\leftmargin\labelsep
2705     \@openbib@code
2706     \usecounter{enumiv}%
2707     \let\p@enumiv\@empty
2708     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2709   %<kiyou> \small
2710   \sloppy
2711   \clubpenalty4000
2712   \@clubpenalty\clubpenalty
2713   \widowpenalty4000%
2714   \sfcode`\.\@m}
2715   {\def\@noitemerr
2716     {\@latex@warning{Empty `thebibliography' environment}}}%
2717   \endlist}
```

`\newblock` `\newblock` はデフォルトでは小さなスペースを生成します。

```
2718 \newcommand{\newblock}{\hspace .11em\@plus.33em\@minus.07em}
```

`\@openbib@code` `\@openbib@code` はデフォルトでは何もありません。この定義は `openbib` オプションによって変更されます。

```
2719 \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。

```
2720 % \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。かっこの前後に入るグルーを `\jsInhibitGlue` で取っ  
ていきますので、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。

```
2721 % \def\@citex[#1]#2{\leavevmode
2722 %   \let\@citea\@empty
2723 %   \@cite{\@for\@citeb:=#2\do
2724 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m }%
2725 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2726 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2727 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2728 %       \G@refundefinedtrue
2729 %       \@latex@warning
2730 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2731 %       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2732 % \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。

```
2733 % \DeclareRobustCommand\cite{\unskip
2734 %   \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}]
2735 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2736 %   , \jsInhibitGlue\ #2\fi}})}$}
```

## 10.3 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)。

```
2737 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2738   \if@twocolumn
2739     \onecolumn\@restonecolfalse
2740   \else
2741     \clearpage\@restonecoltrue
2742   \fi
2743   \columnseprule.4pt \columnsep 2\jsZw
2744   \ifx\multicols\@undefined
2745 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}]%
2746 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2747 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2748 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2749   \else
```

```

2750      \ifdim\textwidth<\fullwidth
2751      \setlength{\evensidemargin}{\oddsidemargin}
2752      \setlength{\textwidth}{\fullwidth}
2753      \setlength{\linewidth}{\fullwidth}
2754 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}]%
2755 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2756 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2757 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2758      \else
2759 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}]%
2760 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2761 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2762 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2763      \fi
2764      \fi
2765 %<book|report>      \@mkboth{\indexname}{}%
2766 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2767      \plainifnotempty % \thispagestyle{plain}
2768      \parindent\z@
2769      \parskip\z@ \@plus .3\p@?\relax
2770      \let\item\@idxitem
2771      \raggedright
2772      \footnotesize\narrowbaselines
2773    }{
2774      \ifx\multicols\@undefined
2775      \if@restonecol\onecolumn\fi
2776      \else
2777      \end{multicols}
2778      \fi
2779      \clearpage
2780    }

```

`\@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 2781 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2782 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2783 \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。

```

2784 \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*, *see also*

`\alsoname` 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。

```

2785 \newcommand\seename{\if@english see\else →\fi}
2786 \newcommand\alsoname{\if@english see also\else →\fi}

```

## 10.4 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` `\inhibitglue` を入れることにします。pL<sup>A</sup>T<sub>E</sub>X の日付が 2016/09/03 より新しい場合は、このパッチが不要なのであてません。

---

パッチの必要性は「`\pltx@foot@penalty` が未定義か」で行う。`\inhibitglue` の代わりに `\jsInhibitGlue` を使う。

---

```
2787 \ifx\pltx@foot@penalty\@undefined
2788   \let\footnotes@ve=\footnote
2789   \def\footnote{\jsInhibitGlue\footnotes@ve}
2790   \let\footnotemarks@ve=\footnotemark
2791   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2792 \fi
```

`\@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。

[2013-04-23] 新しい pT<sub>E</sub>X では脚注番号のまわりにスペースが入りすぎることを防ぐため、北川さんのパッチ [qa:57090] を取り込みました。

[2013-05-14] plcore.ltx に倣った形に書き直しました (Thanks: 北川さん)。

[2016-07-11] コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の変更に追随しました (Thanks: 角藤さん)。pL<sup>A</sup>T<sub>E</sub>X の日付が 2016/04/17 より新しい場合は、このパッチが不要なのであてません。

---

pT<sub>E</sub>X 依存のコードなので、minimal 和文ドライバ実装に移動。

---

`\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。

[2016-10-08] TODO: 脚注番号が `newtxtext` や `newpxtext` の使用時におかしくなってしまう。これらのパッケージは内部で `\thefootnote` を再定義していますので、気になる場合はパッケージを読み込むときに `defaultsup` オプションを付けてください (qa:57284, qa:57287)。

```
2793 \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。

```
2794 % \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。

```

2795 \renewcommand{\footnoterule}{%
2796   \kern-2.6\p@? \kern-.4\p@
2797   \hrule width .4\columnwidth
2798   \kern 2.6\p@?}

```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2799 %<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, *T<sub>E</sub>X and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9)

[2016-08-25] コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の「閉じ括弧類の直後に `\footnotetext` が続く場合に改行が起きることがある問題に対処」と同等のコードを追加しました。

[2016-09-08] コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X のバグ修正に追随しました。

[2016-11-29] 古い pL<sup>A</sup>T<sub>E</sub>X で使用された場合を考慮してコードを改良。

```

2800 \long\def\@footnotetext{%
2801   \insert\footins\bgroup
2802     \normalfont\footnotesize
2803     \interlinepenalty\interfootnotelinepenalty
2804     \splittopskip\footnotesep
2805     \splitmaxdepth \dp\strutbox \floatingpenalty \MM
2806     \hsize\columnwidth \@parboxrestore
2807     \protected@edef\@currentlabel{%
2808       \csname p@footnote\endcsname\@thefnmark
2809     }%
2810     \color@begingroup
2811     \@makefnmark
2812     \rule{z@footnotesep}{\ignorespaces}%
2813     \futurelet\next\fo@t}
2814 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
2815           \else \let\next\fo@t\fi \next}
2816 \def\fo@t{\bgroup\aftergroup\@foot\let\next}
2817 \def\fo@t#1{#1\@foot}
2818 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2819   \ifx\pltx@foot@penalty\undefined\else
2820     \ifhmode\null\fi
2821     \ifnum\pltx@foot@penalty=z@\else
2822       \penalty\pltx@foot@penalty
2823       \pltx@foot@penaltyz@
2824     \fi
2825     \fi}

```

`\@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。

```

2826 \newcommand\@makefnmark[1]{%
2827   \advance\leftskip 3\jsZw
2828   \parindent 1\jsZw
2829   \noindent
2830   \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。

```
2831 % \def\@xfootnotenext[#1]{%
2832 %   \begingroup
2833 %     \ifnum#1>\z@
2834 %       \csname c@\@mpfn\endcsname #1\relax
2835 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2836 %     \else
2837 %       \unrestored@protected@xdef\@thefnmark{}%
2838 %     \fi
2839 %   \endgroup
2840 %   \@footnotetext}
```

---

ここまでのコードは JS クラスを踏襲する。

---

## 11 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。

`\jsInhibitGlueAtParTop` 「段落頭の括弧の空き補正」の処理を `\jsInhibitGlueAtParTop` という命令にして、これを再定義可能にした。

```
2841 \let\jsInhibitGlueAtParTop\@empty
```

`\everyparhook` 全ての段落の冒頭で実行されるフック。これの初期値を先述の `\jsInhibitGlueAtParTop` とする。

```
2842 \def\everyparhook{\jsInhibitGlueAtParTop}
2843 \AtBeginDocument{\everypar{\everyparhook}}
```

---

[2016-07-18] `\inhibitglue` の発行対象を `\inhibitxspcode` が 2 に設定されているものすべてに拡大しました。

[2016-12-01] すぐ上の変更で `\@tempa` を使っていたのがよくなかったので、プレフィックスを付けて `\jsc@tempa` にしました (forum:2085)。

[2017-02-13] `\jsc@tempa` は実はテンポラリーではなく「この処理専用のユニーク制御綴」である必要があります。間違って別の箇所です使う危険性が高いので、専用の命令 `\jsc@ig@temp` に置き換えました (Issue #54)。

---

`\@inhibitglue` JS クラスでの `\jsInhibitGlueAtParTop` の実装。(これは (u)pT<sub>E</sub>X 専用である。)

```
2844 \def\@inhibitglue{%
2845   \futurelet\@let@token\@@inhibitglue}
2846 \begingroup
2847 \let\GDEF=\gdef
2848 \let\CATCODE=\catcode
2849 \let\ENDGROUP=\endgroup
2850 \CATCODE`k=12
2851 \CATCODE`a=12
2852 \CATCODE`n=12
2853 \CATCODE`j=12
2854 \CATCODE`i=12
2855 \CATCODE`c=12
2856 \CATCODE`h=12
2857 \CATCODE`r=12
2858 \CATCODE`t=12
2859 \CATCODE`e=12
2860 \GDEF\KANJI@CHARACTER{kanji character }
2861 \ENDGROUP
2862 \def\@@inhibitglue{%
2863   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
2864   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
2865   \def\jsc@ig@temp{#1}%
2866   \ifx\jsc@ig@temp\@empty
2867     \ifnum\the\inhibitxspcode`#2=2\relax
2868       \inhibitglue
2869     \fi
2870   \fi}
```

---

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。

[2016-11-19] `ltlists.dtx` 2015/05/10 v1.0t の変更に従って `\clubpenalty` のリセットを追加しました。

```
2871 \def\@doendpe{%
2872   \@endpetrue
2873   \def\par{%
2874     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2875   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] `minipage` 環境にも対策します。

```
2876 \def\@setminipage{%
2877   \@minipagetrue
2878   \everypar{\@minipagefalse\everypar{\everyparhook}}%
2879 }
```

`\item` 命令の直後です。

```

2880 \def\@item[#1]{%
2881   \if@noparitem
2882     \@donoparitem
2883   \else
2884     \if@inlabel
2885       \indent \par
2886     \fi
2887     \ifhmode
2888       \unskip\unskip \par
2889     \fi
2890     \if@newlist
2891       \if@nobreak
2892         \@nbitem
2893       \else
2894         \addpenalty\@beginparpenalty
2895         \addvspace\@topsep
2896         \addvspace{-\parskip}%
2897       \fi
2898     \else
2899       \addpenalty\@itempenalty
2900       \addvspace\itemsep
2901     \fi
2902     \global\@inlabeltrue
2903   \fi
2904   \everypar{%
2905     \@minipagefalse
2906     \global\@newlistfalse
2907     \if@inlabel
2908       \global\@inlabelfalse
2909       {\setbox\z@\lastbox
2910        \ifvoid\z@
2911          \kern-\itemindent
2912        \fi}%
2913     \box\@labels
2914     \penalty\z@
2915   \fi
2916   \if@nobreak
2917     \@nobreakfalse
2918     \clubpenalty \@M
2919   \else
2920     \clubpenalty \@clubpenalty
2921     \everypar{\everyparhook}%
2922   \fi
2923   \bxjs@ltj@inhibitglue
2924   \everyparhook}%
2925 \if@noitemarg
2926   \@noitemargfalse
2927   \if@nmbrlist
2928     \refstepcounter\@listctr

```

```

2929 \fi
2930 \fi
2931 \sbox\@tempboxa{\makelabel{#1}}%
2932 \global\setbox\@labels\hbox{%
2933 \unhbox\@labels
2934 \hskip \itemindent
2935 \hskip -\labelwidth
2936 \hskip -\labelsep
2937 \ifdim \wd\@tempboxa >\labelwidth
2938 \box\@tempboxa
2939 \else
2940 \hbox to\labelwidth {\unhbox\@tempboxa}%
2941 \fi
2942 \hskip \labelsep}%
2943 \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。

```

2944 \def\@afterheading{%
2945 \@nobreake>true
2946 \everypar{%
2947 \if@nobreake
2948 \@nobreake>false
2949 \clubpenalty \@M
2950 \if@afterindent \else
2951 {\setbox\z@\lastbox}%
2952 \fi
2953 \else
2954 \clubpenalty \@clubpenalty
2955 \everypar{\everyparhook}%
2956 \fi\everyparhook}}

```

`\@gnewline` についてはちょっと複雑な心境です。もともとの  $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X } 2_{\epsilon}$  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。

```

2957 \def\@gnewline #1{%
2958 \ifvmode
2959 \@nolnerr
2960 \else
2961 \unskip \reserved@a {\reserved@a#1}\nobreake \hfil \break \null
2962 \jsInhibitGlue \ignorespaces
2963 \fi}

```



## 12 いろいろなロゴ

L<sup>A</sup>T<sub>E</sub>X 関連のロゴを作り直します。

[2016-07-14] ロゴの定義は `jslogo` パッケージに移転しました。後方互換のため、`jsclasses` ではデフォルトでこれを読み込みます。`nojslogo` オプションが指定されている場合は読み込みません。

---

BXJS クラスでも `jslogo` オプション指定の場合に `jslogo` パッケージを読み込むようにした。ただし JS クラスと異なり、既定では読み込まない。

※`\小`、`\上小` の制御綴は定義しない。

---

```
2964 \if@jslogo
2965   \IfFileExists{jslogo.sty}{%
2966     \RequirePackage{jslogo}%
2967   }{%
2968     \ClassWarningNoLine{bxjs@clsname}
2969       {The package 'jslogo' is not installed.\MessageBreak
2970        It is included in the recent release of\MessageBreak
2971        the 'jsclasses' bundle}
2972   }
2973 \fi
```

## 13 amsmath との衝突の回避

`\ltx@ifnextchar` `amsmath` パッケージでは行列中で `\@ifnextchar` を再定義していますが、これが L<sup>A</sup>T<sub>E</sub>X の `\ProvidesFile` で悪さをする例が F<sub>T</sub>E<sub>X</sub> で報告されています。これを避けるための tDB さんのフィックスを挿入しておきます。副作用がありましたらお知らせください。

この現象については私の TeX 掲示板 4273～, 16058～ で議論がありました。なお、AMS 関係のパッケージを読み込む際に `psamsfonts` オプションを与えても回避できます (Thanks: しっぱ愛好家さん)。

[2016-11-19] 本家の `ltclass.dtx` 2004/01/28 v1.1g で修正されているのでコメントアウトしました。

```
2974 %\let\ltx@ifnextchar\@ifnextchar
2975 %\def\ProvidesFile#1{%
2976 %  \begingroup
2977 %    \catcode`\ 10 %
2978 %    \ifnum \endlinechar<256 %
2979 %        \ifnum \endlinechar>\m@ne
2980 %            \catcode\endlinechar 10 %
2981 %        \fi
2982 %    \fi
2983 %    \@makeother\/%
```

```

2984 %      \@makeother\&%
2985 %      \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 2986 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2987 \newcommand{\postpartname}{\if@english\else 部\fi}
                2988 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2989 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2990 \newcommand{\presectionname}{}% 第
\postsectionname 2991 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 2992 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2993 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
                2994 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 2995 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2996 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
            2997 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 2998 %<!\jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
            2999 %<jspf>\newcommand{\figurename}{Fig.~}
            3000 %<!\jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
            3001 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3002 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
            3003 \newcommand{\appendixname}{\if@english \else 付録\fi}
            3004 %<!\book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

■今日の日付 L<sup>A</sup>T<sub>E</sub>X 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。

---

環境変数 `SOURCE_DATE_EPOCH_TEX_PRIMITIVES` が設定されている場合は“今日”が過去の日付になる可能性があるが、その場合、和暦表記は平成 2 年（1990 年）以降でのみサポートする。

※ “新元号” への対応？

---

```
\today
```

```

3005 \@tempswafalse
3006 \if p\jsEngine \@tempswatrue \fi
3007 \if n\jsEngine \@tempswatrue \fi
3008 \if@tempswa \expandafter\@firstoftwo
3009 \else \expandafter\@secondoftwo
3010 \fi
3011 {%
3012 % 欧文 8bitTeX の場合
3013 \newif\ifjsSeireki \jsSeirekitrue
3014 \def\bxjs@decl@Seireki@cmds{%
3015   \def\西暦{\jsSeirekitrue}%
3016   \def\和暦{\jsSeirekifalse}}
3017 \def\Seireki{\jsSeirekitrue}
3018 \def\Wareki{\jsSeirekifalse}
3019 \def\bxjs@if@use@seireki{%
3020   \ifjsSeireki \expandafter\@firstoftwo
3021   \else \expandafter\@secondoftwo \fi}
3022 }{%
3023 \newif\if 西暦 \西暦 true
3024 \def\bxjs@decl@Seireki@cmds{%
3025   \def\西暦{\西暦 true}%
3026   \def\和暦{\西暦 false}}
3027 \def\Seireki{\西暦 true}
3028 \def\Wareki{\西暦 false}
3029 \def\bxjs@if@use@seireki{%
3030   \if 西暦 \expandafter\@firstoftwo
3031   \else \expandafter\@secondoftwo \fi}
3032 }
3033 \bxjs@decl@Seireki@cmds
3034 % \bxjs@unxp
3035 \let\bxjs@unxp\@firstofone
3036 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
3037 % \bxjs@iai
3038 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3039   \def\bxjs@iai{\noexpand~}
3040 \else \def\bxjs@iai{}
3041 \fi
3042 % \heisei
3043 \newcount\heisei \heisei\year \advance\heisei-1988\relax
3044 % \today
3045 \edef\bxjs@today{%
3046   \if@english
3047     \ifcase\month\or
3048       January\or February\or March\or April\or May\or June\or
3049       July\or August\or September\or October\or November\or December\fi
3050     \space\number\day, \number\year
3051   \else
3052     \ifnum\heisei>\@ne
3053       \expandafter\noexpand\expandafter\bxjs@if@use@seireki

```

```

3054 \else \expandafter\@firstoftwo
3055 \fi {%
3056 \number\year\bxjs@iai\bxjs@unxp{年}%
3057 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3058 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3059 }{%
3060 \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
3061 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3062 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3063 }%
3064 \fi}
3065 \let\today\bxjs@today

```

---

texjporg 版の日本語用 Babel 定義ファイル (japanese.ldf) が読み込まれた場合に影響を受けないようにする。

```

3066 \AtBeginDocument{%
3067 \ifx\bb1@jpn@Seirekitrue\undefined\else
3068 \bxjs@decl@Seireki@cmds
3069 \g@addto@macro\datejapanese{%
3070 \let\today\bxjs@today}%
3071 \fi}

```

---

■ハイフネーション例外 TeX 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)

```

3072 \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。

```

3073 %<slide>\pagestyle{empty}%
3074 %<article|report>\pagestyle{plain}%
3075 %<book>\pagestyle{headings}%
3076 \pagenumbering{arabic}
3077 \if@twocolumn
3078 \twocolumn
3079 \sloppy
3080 \flushbottom
3081 \else
3082 \onecolumn
3083 \raggedbottom
3084 \fi
3085 %<*slide>
3086 \renewcommand\familydefault{\sfdefault}
3087 \raggedright
3088 %</slide>

```

■BXJS 独自の追加処理 

---

和文ドライバのファイルを読み込む。

```
3089 \catcode`\?=12
3090 \ifx\bxjs@jadriver\relax\else
3091 \input{bxjsja-\bxjs@jadriver.def}
3092 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
3093 \bxjs@restore@jltrcc
3094 %</cls>
```

---

以上です。

## 付録 A 和文ドライバの仕様

次の命令が BXJS クラス本体と和文ドライバの連携のために用意されている。このうち、★印を付けたものは“書込”が許されるものである。

- `\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)。
  - `\jsArticle` `bxjsarticle` クラス
  - `\jsBook` `bxjsbook` クラス
  - `\jsReport` `bxjsreport` クラス
  - `\jsSlide` `bxjsslide` クラス
- `\jsEngine` [文字トークンの `let`] 使用されているエンジンの種別。(`\if` で判定可能)。
  - `p` `pdfTeX` (DVI モードも含む)
  - `l` `LuaTeX` (＃)
  - `x` `XYTeX`
  - `j` `pTeX` または `upTeX`
  - `n` 以上の何れでもない
- `\ifjsWithupTeX` [スイッチ] 使用されているエンジンが `upTeX` であるか。
- `\ifjsWitheTeX` [スイッチ] 使用されているエンジンが  $\epsilon$ -`TeX` 拡張であるか。
- `\ifjsInPdfMode` [スイッチ] 使用されているエンジンが (`pdfTeX`・`LuaTeX` の) PDF モードであるか。
- `\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。)
- `\jsJaFont` [マクロ] 和文フォント設定を表す文字列。クラスオプション `jafont` で指定された値。
- `\jsJaParam` [マクロ] 和文モジュールに渡すパラメタを表す文字列。この値が何を表すかは決まっておらず、各々の和文モジュールが独自に解釈する。クラスオプション `japaram` で指定された値。
- `\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。
- `\jsInhibitGlueAtParTop` [マクロ] ★ 段落先頭におけるカギ括弧の位置調整を行うマクロ。全ての段落先頭で呼び出される。
- `\jsZw` [内部寸法値] 「現在の全角幅」を表す変数。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。

- `\jsFontSizeChanged` [マクロ] フォントサイズが変更された時に必ず呼び出されるマクロ。
- `\jsResetDimen` [マクロ] ★ 上記 `\jsFontSizeChanged` の中で呼び出される、ユーザ (和文モジュール) 用のフック。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。

以下で標準で用意されている和文ドライバの実装を示す。

```
3095 %<*drv>
```

## 付録 B 和文ドライバ : minimal

`jadriver` の指定が無い場合に適用されるドライバ。また、`standard` ドライバはまずこのドライバファイルを読み込んでいる。

このドライバでは、各エンジンについての必要最低限の処理だけを行っている。日本語処理のためのパッケージ (`xeCJK` や `LuaTeX-ja` 等) を自分で読み込んで適切な設定を行うという使用状況を想定している。

ただし、`(u)pTeX` エンジンについては例外で、和文処理機構の選択の余地がないため、このドライバにおいて、「JS クラスと同等の指定」を完成させるためのコードを記述する。

### B.1 補助マクロ

```
3096 %<*minimal>
```

```
3097 %% このファイルは日本語文字を含みます
```

```
\DeclareJaTextFontCommand 和文書体のための、「余計なこと」をしない \DeclareTextFontCommand.
```

```
3098 \def\DeclareJaTextFontCommand#1#2{%
```

```
3099   \DeclareRobustCommand#1[1]{%
```

```
3100     \relax
```

```
3101     \ifmmode \expandafter\nfss@text \fi
```

```
3102     {#2##1}}%
```

```
3103 }
```

```
\DeclareJaMathFontCommand 和文数式フォントが無効な場合に、それをエミュレートするもの。
```

```
3104 \def\DeclareJaMathFontCommand#1#2{%
```

```
3105   \DeclareRobustCommand#1[1]{%
```

```
3106     \relax
```

```
3107     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3108     \nfss@text{\fontfamily\familydefault
```

```
3109         \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3110     #2##1}}%
```

```

3111 }%
3112 }

```

`\bxjs@if@sf@default` `\familydefault` の定義が “`\sfdefault`” である場合に引数のコードを実行する。

```

3113 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3114 \@onlypreamble\bxjs@if@sf@default
3115 \def\bxjs@if@sf@default#1{%
3116   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3117   \AtBeginDocument{%
3118     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3119 }

```

`\jsInverseScale` `\jsScale` の逆数。

※`\CS=\jsInverseScale\CS` は `\bxjs@invscale\CS\jsScale` よりも精度が劣るが処理が軽い。

```

3120 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3121 \edef\jsInverseScale{\strip@pt\@tempdima}

```

`\jsLetHeadChar` `\jsLetHeadChar\CS{〈トークン列〉}` : トークン列の先頭の文字を抽出し、`\CS` をその文字トークン (に展開されるマクロ) として定義する。

※先頭にあるのが制御綴やグループである場合は `\CS` は `\relax` に等置される。

※文字トークンは “`\the-文字列`” のカテゴリコードをもつ。

※非 Unicode エンジンの場合は文字列が UTF-8 で符号化されていると見なし、先頭が高位バイトの場合は 1 文字分のバイト列 (のトークン列) を抽出する。この場合は元のカテゴリコードが保持される。

```

3122 \def\jsLetHeadChar#1#2{%
3123   \begingroup
3124   \escapechar=`\ %
3125   \let\bxjs@tmpa={% brace-match-hack
3126   \bxjs@let@hchar@exp#2}%
3127   \endgroup
3128   \let#1\bxjs@g@tmpa}
3129 \def\bxjs@let@hchar@exp{%
3130   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3131 \def\bxjs@let@hchar@exp@a{%
3132   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3133   \bxjs@let@hchar@out\let\relax
3134   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3135   \bxjs@let@hchar@out\let\space%
3136   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3137   \bxjs@let@hchar@out\let\@backslashchar
3138   }{\bxjs@let@hchar@exp@b}}}%
3139 \def\bxjs@let@hchar@exp@b#1{%
3140   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3141 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3142   %\message{<#1#2>}%
3143   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```



```

3144 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3145 \bxjs@let@hchar@out\let\relax
3146 }{%else
3147 \expandafter\bxjs@let@hchar@exp
3148 }%
3149 }{%else
3150 \bxjs@let@hchar@chr#1%
3151 }}
3152 \def\bxjs@let@hchar@chr#1{%
3153 \bxjs@let@hchar@out\def{{#1}}}}
3154 \def\bxjs@let@hchar@out#1#2{%
3155 \global#1\bxjs@g@tmpa#2\relax
3156 \toks@ \bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3157 \chardef\bxjs@let@hchar@csta=128
3158 \chardef\bxjs@let@hchar@cstb=192
3159 \chardef\bxjs@let@hchar@cstc=224
3160 \chardef\bxjs@let@hchar@cstd=240
3161 \chardef\bxjs@let@hchar@cste=248
3162 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3163 \def\bxjs@let@hchar@chr@ue#1{%
3164 \@tempcnta=`#1\relax
3165 %\message{\the\@tempcnta}%
3166 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3167 \bxjs@let@hchar@chr@ue@a#1%
3168 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3169 \bxjs@let@hchar@out\let\relax
3170 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3171 \bxjs@let@hchar@chr@ue@b
3172 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3173 \bxjs@let@hchar@chr@ue@c
3174 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3175 \bxjs@let@hchar@chr@ue@d
3176 }{%else
3177 \bxjs@let@hchar@out\let\relax
3178 }}}}
3179 \def\bxjs@let@hchar@chr@ue@a#1{%
3180 \bxjs@let@hchar@out\def{{#1}}}}
3181 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3182 \bxjs@let@hchar@out\def{{#1#2}}}}
3183 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3184 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3185 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3186 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

## B.2 (u)pTeX 用の設定

```
3187 \ifx j\jsEngine
```

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。

■補助マクロ `\jsLetHeadChar` を和文文字トークンに対応させる。

```
3188 \def\bxjs@let@hchar@chr@pp#1{%
3189   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
3190 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
3191   \%message{(#1)}%
3192   \bxjs@cond\if#1t\fi{%
3193     \bxjs@let@hchar@chr@ue#3%
3194   }{%else
3195     \bxjs@let@hchar@out\def{#3}}%
3196   }}
3197 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@pp
```

■エンジン依存の定義 最初にエンジン (pTeX か upTeX か) に依存する定義を行う。`\ifjsWithupTeX` 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。

```
3198 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3199 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3200 \edef\jsc@pfx@{\ifjsWithupTeX 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。

```
3201 \@onlypreamble\bxjs@declarefontshape
3202 \ifjsWithupTeX
3203 \def\bxjs@declarefontshape{%
3204   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3205   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3206   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3207   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3208 }
3209 \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。

```
3210 \else
3211 \def\bxjs@declarefontshape{%
3212   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3213   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3214   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3215 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3216 }
3217 \def\bxjs@sizereference{jis}
3218 \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。

```

3219 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
3220   \def\bxjs@y{#5}}
3221 \ifjsWithpTeXng \def\bxjs@y{10}%
3222 \else
3223 \expandafter\expandafter\expandafter\bxjs@tmpa
3224 \expandafter\string\the\jfont\relax
3225 \fi
3226 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3227               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3228   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
3229    \expandafter\let\csname\bxjs@x/\bxjs@y\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。

```

3230 \begingroup
3231 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3232 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3233 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3234 % 幅が丁度 10pt なら補正は不要
3235 \ifdim\wd\z@=10pt
3236   \global\let\bxjs@scale\jsScale
3237 \else
3238 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3239   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3240   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3241   \bxjs@invscale\@tempdima\bxjs@tmpa
3242   \xdef\bxjs@scale{\strip@pt\@tempdima}
3243 \fi
3244 \endgroup
3245 %\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。

```

3246 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3247 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3248 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3249 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3250 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3251 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3252 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3253 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3254 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3255 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3256 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3257 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3258 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3259 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3260 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}

```

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。

```

3261 \DeclareRobustCommand\rmfamily
3262   {\not@math@alphabet\rmfamily\mathrm
3263     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3264 \DeclareRobustCommand\sffamily
3265   {\not@math@alphabet\sffamily\mathsf
3266     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3267 \DeclareRobustCommand\ttfamily
3268   {\not@math@alphabet\ttfamily\mathtt
3269     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3270 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3271 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3272 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3273 \fi
3274 \bxjs@if@sf@default{%
3275   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3276 \selectfont

```

## ■パラメタの設定

```

3277 \prebreakpenalty\jis"2147=10000
3278 \postbreakpenalty\jis"2148=10000
3279 \prebreakpenalty\jis"2149=10000
3280 \inhibitxspcode`!=1
3281 \inhibitxspcode`〒=2
3282 \xspcode`+=3
3283 \xspcode`%=3

```

"80～"FF の範囲の \xspcode を 3 に変更。

```

3284 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3285   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

\jsInhibitGlueAtParTop の定義。「JS クラスでの定義」を利用する。

```

3286 \let\jsInhibitGlueAtParTop\@inhibitglue

```

\jsResetDimen は空のままでよい。

■組方向依存の処理 組方向判定の if-トークン (`\if?dir`) は pTeX 以外では未定義であるため、そのまま if 文に入れることができない。これを回避するため部分的に `!` をエスケープ文字に使う。

```
3287 \begingroup
3288 \catcode`\!=0
```

`\bxjs@ptex@dir` 現在の組方向: t=縦、y=横、?=その他。

```
3289 \gdef\bxjs@ptex@dir{%
3290   !iftdir t%
3291   !else!ifydir y%
3292   !else ?%
3293   !fi!fi}
```

新版の pTeX で脚注番号の周囲の空きが過大になる現象への対処。

※現在の pLaTeX カーネルでは対処が既に行われている。ここでは、`\@makefnmark` の定義が古いものであった場合に、新しいものに置き換える。

```
3294 % 古い \@makefnmark の定義
3295 \long\def\bxjs@tmpa{\hbox{%
3296   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3297   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3298 \ifx\@makefnmark\bxjs@tmpa
3299 \long\gdef\@makefnmark{%
3300   !ifydir \hbox{ }\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{ }%
3301   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3302 \fi
3303 \endgroup
```

### B.3 pdfTeX 用の処理

```
3304 \else\ifx p\jsEngine
3305 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3306 \@onlypreamble\bxjs@cjk@loaded
3307 \def\bxjs@cjk@loaded{%
3308   \def\@footnotemark{%
3309     \leavevmode
3310     \ifhmode
3311       \edef\@x@sf{\the\spacefactor}%
3312       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3313         \unkern\unkern
3314         \ifdim\lastskip>\z@ \unskip \fi
3315       \fi\fi
3316       \nobreak
3317     \fi
3318     \@makefnmark
3319     \ifhmode \spacefactor\@x@sf \fi
3320     \relax}%
```

```

3321 \let\bxjs@CJK@loaded\relax
3322 }
3323 \AtBeginDocument{%
3324 \ifpackageloaded{CJK}{%
3325 \bxjs@CJK@loaded
3326 }{}%
3327 }

```

## B.4 Xe<sub>La</sub>TeX 用の処理

```

3328 \else\ifx x\jsEngine

```

\bxjs@let@hchar@chr について、「BMP 外の文字の文字トークンに対して \string を適用するとサロゲートペアに分解される」という問題に対する応急措置を施す。

```

3329 \def\bxjs@let@hchar@chr#1{%
3330 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3331 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3332 \bxjs@let@hchar@chr@xe
3333 }{\bxjs@let@hchar@out\def{{#1}}}}
3334 \def\bxjs@let@hchar@chr@xe#1{%
3335 \lccode`0=`#1\relax
3336 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

\bxjs@do@precisetext precisetext オプションの処理。

```

3337 \ifx\XeTeXgenerateactualtext\@undefined\else
3338 \def\bxjs@do@precisetext{%
3339 \XeTeXgenerateactualtext=\@ne}
3340 \fi

```

\bxjs@do@simplejasetup simplejasetup オプションの処理。

```

3341 \@onlypreamble\bxjs@do@simplejasetup
3342 \def\bxjs@do@simplejasetup{%
3343 \ifnum\XeTeXinterchartokenstate>\z@
3344 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3345 \jsSimpleJaSetup
3346 \ClassInfo\bxjs@clsname
3347 {\string\jsSimpleJaSetup' is applied\@gobble}%
3348 \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

3349 \newcommand*{\jsSimpleJaSetup}{%
3350 \XeTeXlinebreaklocale "ja"\relax
3351 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3352 \XeTeXlinebreakpenalty=0\relax}

```

## B.5 後処理（エンジン共通）

```

3353 \fi\fi\fi

```

simplejasetup オプションの処理。

```

3354 \ifx\bxjs@do@simplejasetup\@undefined\else
3355   \AtBeginDocument{%
3356     \ifbxjs@simplejasetup
3357       \bxjs@do@simplejasetup
3358     \fi}
3359 \fi

```

`precisetext` オプションの処理。

```

3360 \ifbxjs@precisetext
3361   \ifx\bxjs@do@precisetext\@undefined
3362     \ClassWarning\bxjs@clsname
3363       {The current engine does not support the\MessageBreak
3364         'precisetext' option\@gobble}
3365   \else
3366     \bxjs@do@precisetext
3367   \fi
3368 \fi

```

■`fancyhdr` 対策 `fancyhdr` オプションの値が `true` であり、かつ `fancyhdr` が使用された場合に以下の対策を行う。

- デフォルトの書式設定に含まれる“二文字フォント命令”を除去する。
- `bxjsbook` において、ヘッダ・フッタの横幅を `\fullwidth` に変える。

```

3369 \ifbxjs@fancyhdr

```

`\bxjs@adjust@fancyhdr` `fancyhdr` の初期設定に関する変更の処理。`fancyhdr` 読み完了と `\pagestyle{fancy}` 実行の間で実行されるべき。

```

3370 \@onlypreamble\bxjs@adjust@fancyhdr
3371 \def\bxjs@adjust@fancyhdr{%

```

ヘッダ・フッタの要素の書式について、それが既定のままであれば、“二文字フォント命令”を除去したものに置き換える。

※和文なので `\sl` は無い方がよいはず。

```

3372   \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3373   \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3374   \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3375   \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3376   \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3377   \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3378   \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3379   \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3380   \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3381   \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3382   \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3383   \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3384   \def\bxjs@tmpa{\rm\thepage\strut}%
3385   \def\bxjs@tmpb{\thepage\strut}%
3386   \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3387   \ifx\f@ncyocf\bxjs@tmpa \global\let\f@ncyocf\bxjs@tmpb \fi

```

`\fullwidth` が (定義済で) `\textwidth` よりも大きい場合、ヘッダ・フッタの横幅を `\fullwidth` に合わせる。

```
3388 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3389   \setlength{\@tempdima}{\fullwidth-\textwidth}%
3390   \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3391   }\bxjs@tmpa
3392 \fi\fi
3393 \PackageInfo\bxjs@clsname
3394 {Patch to fancyhdr is applied\@gobble}}
```

`\bxjs@pagestyle@hook` `\pagestyle` へのフックの本体。

```
3395 \def\bxjs@pagestyle@hook{%
3396   \@ifpackageloaded{fancyhdr}{%
3397     \bxjs@adjust@fancyhdr
3398     \global\let\bxjs@adjust@fancyhdr\relax
3399   }{}}
```

`\pagestyle` にフックを入れ込む。

```
3400 \let\bxjs@org@pagestyle\pagestyle
3401 \def\pagestyle{%
3402   \bxjs@pagestyle@hook \bxjs@org@pagestyle}
```

`begin-document` フック。

※これ以降に `fancyhdr` が読み込まれることはあり得ない。

```
3403 \AtBeginDocument{%
3404   \bxjs@pagestyle@hook
3405   \global\let\bxjs@pagestyle@hook\relax}
```

終わり。

```
3406 \fi
```

以上で終わり。

```
3407 %</minimal>
```

## 付録 C 和文ドライバ : standard

標準のドライバ。

- `\rmfamily/\sffamily/\ttfamily` での和文ファミリ連動
- `\mcfamily/\gtfamily`
- `\textmc/\textgt`
- `\zw`
- `\jQ/\jH`
- `\trueQ/\trueH/\ascQ`
- `\setkanjiskip/\getkanjiskip`
- `\setxkanjiskip/\getxkanjiskip`
- `\autospacing/\noautospacing`



- `\autoxspacing/\noautoxspacing`

■和文フォント指定の扱い `standard` 和文ドライバでは `\jsJaFont` の値を和文フォントの“プリセット”の指定として用いる。プリセットの値は、`TeX Live` の `kanji-config-updmap` コマンドで使う“ファミリ”と同じにすることを想定する。特別な値として、`auto` は `kanji-config-updmap` で現在指定されているファミリを表す。

## C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
3408 %<*standard>
3409 %% このファイルは日本語文字を含みます
3410 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3411 \bxjs@simplejasetupfalse
```

■`japaram` オプションの処理 `japaram` の値を key-value リストとして解釈する。keyval のファミリは `bxjsStd` とする。

`\bxjs@jp@jismmiv` 2004JIS 字形を優先させるか。

```
3412 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3413 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3414 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3415 \define@key{bxjsStd}{jis2004}[true]{%
3416   \bxjs@set@keyval{jis2004}{#1}{}}
```

`\bxjs@jp@font` フォントパッケージの追加オプション。

```
3417 \let\bxjs@jp@font\@empty

jis2004 オプションの処理。
3418 \define@key{bxjsStd}{font}{%
3419   \edef\bxjs@jp@font{#1}}
```

実際の `japaram` の値を適用する。

```
3420 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3421 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

■`jis2004` パラメタ `jis2004` パラメタが有効の場合は、グローバルオプションに `jis2004` を追加する。

※ `otf` や `luatexja-preset` 等のパッケージがこのオプションを利用する。

```
3422 \ifbxjs@jp@jismmiv
3423   \g@addto@macro\@classoptionslist{,jis2004}
3424 % \@ifpackagewith 判定への対策
3425   \PassOptionsToPackage{jis2004}{otf}
3426 \fi
```

■共通命令の実装 `\jQ` 等の「単位」系の共通命令を実装する。まず  $\varepsilon$ -TeX 拡張が使えるか  
 検査する。

```
3427 \ifjsWitheTeX
```

使える場合は、「`\dimexpr` 外部寸法表記`\relax`」の形式（これは内部値なので単位として  
 使える）で各命令定義する。

`\jQ` `\jQ` と `\jH` はともに 0.25 mm に等しい。

```
\jH 3428 \@tempdima=0.25mm
      3429 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
      3430 \let\jH\jQ
```

`\trueQ` `\trueQ` と `\trueH` はともに 0.25 true mm に等しい。

```
\trueH 3431 \ifjsc@mag
      3432 \@tempdimb=\jsBaseFontSize\relax
      3433 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
      3434 \@tempdima=2.5mm
      3435 \bxjs@invscale\@tempdima\bxjs@tmpa
      3436 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
      3437 \@tempdima=10pt
      3438 \bxjs@invscale\@tempdima\bxjs@tmpa
      3439 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
      3440 \else \let\trueQ\jQ \let\bxjs@truept\p@
      3441 \fi
      3442 \let\trueH\trueQ
```

`\ascQ` `\ascQ` は `\trueQ` を和文スケール値で割った値。例えば、`\fontsize{12\ascQ}{16\trueH}`  
`\ascpt` とすると、和文が 12Q になる。

同様に、`\ascpt` は `truept` を和文スケールで割った値。

```
3443 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3444 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3445 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3446 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3447 \fi
```

`\jsSetQHLength` `\jsSetQHLength\CS{⟨長さ式⟩}` : `\setlength` の変種で、通常の `calc` の長さ式の代わ  
 りに、`⟨実数⟩Q · ⟨実数⟩H · ⟨実数⟩trueQ · ⟨実数⟩trueH` が記述できる。

```
3448 \def\jsSetQHLength#1#2{%
3449 \begingroup
3450 \bxjs@parse@qh{#2}%
3451 \ifx\bxjs@tmpb\relax \setlength\@tempdima{#2}%
3452 \else \@tempdima\bxjs@tmpb\relax
3453 \fi
3454 \xdef\bxjs@g@tmpa{\the\@tempdima}%
3455 \endgroup
3456 #1=\bxjs@g@tmpa\relax}
```

`\bxjs@parse@qh` `jsSetQHLlength` の下請け。#1 が Q/H/trueQ/trueH で終わる場合、`\bxjs@tmpb` にそれに等しい寸法の表現を返す。それ以外では、`\bxjs@tmpb` は `\relax` になる。

```
3457 \if j\jsEngine
```

(u)pL<sup>A</sup>T<sub>E</sub>X ではこの処理は不要なので無効化する。

```
3458 \def\bxjs@parse@qh#1{%
3459   \let\bxjs@tmpb\relax}
3460 \else
3461 \def\bxjs@parse@qh#1{%
3462   \def\bxjs@tmpa{#1}\let\bxjs@tmpb\relax
3463   \bxjs@parse@qh@a{trueQ}\trueQ
3464   \bxjs@parse@qh@a{trueH}\trueH
3465   \bxjs@parse@qh@a{Q}\jQ
3466   \bxjs@parse@qh@a{H}\jH}
3467 \def\bxjs@parse@qh@a#1#2{%
3468   \ifx\bxjs@tmpb\relax
3469     \def\bxjs@next##1#1\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}}%
3470     \expandafter\bxjs@next\bxjs@tmpa\@nil#1\@nil\@nnil{#2}%
3471   \fi}
3472 \def\bxjs@parse@qh@b#1#2#3{%
3473   \ifx\@nnil#2\@nnil\else \def\bxjs@tmpb{#1#3}\fi}
3474 \fi
```

`\jafontsize` `\jafontsize{〈フォントサイズ〉}{〈行送り〉}` : 和文フォント規準で、すなわち、1zw が〈フォントサイズ〉に等しくなるようにフォントサイズを指定する。この命令の引数では、Q/H の単位が使用できる。

```
3475 \def\jafontsize#1#2{%
3476   \begingroup
3477     \bxjs@jafontsize@a{#1}%
3478     \@tempdimb\jsInverseScale\@tempdima
3479     \bxjs@jafontsize@a{#2}%
3480     \xdef\bxjs@g@tmpa{%
3481       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3482   \endgroup\bxjs@g@tmpa}
3483 \def\bxjs@jafontsize@a#1{%
3484   \bxjs@parse@qh{#1}%
3485   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3486   \@defaultunits\@tempdima\bxjs@tmpb pt\relax\@nnil}
```

続いて、和文間空白・和欧文間空白関連の命令を実装する。(エンジン依存のコード。)

`\bxjs@kanjiskip` 和文間空白の量を表すテキスト。

```
3487 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3488 \newcommand*\setkanjiskip[1]{%
3489   \edef\bxjs@kanjiskip{#1}%
3490   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3491 \newcommand*\getkanjiskip{%  
3492   \bxjs@kanjiskip}
```

`\ifbxjs@kanjiskip@enabled` 和文間空白の挿入が有効か。ただし pTeX では自身の `\(no)autospacing` での制御を用いるのでこの変数は常に真とする。

```
3493 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

`\bxjs@enable@kanjiskip` 和文間空白の挿入を有効／無効にする。(pTeX 以外)

```
\bxjs@disable@kanjiskip 3494 \bxjs@robust@def\bxjs@enable@kanjiskip{%  
3495   \bxjs@kanjiskip@enabledtrue  
3496   \bxjs@reset@kanjiskip}  
3497 \bxjs@robust@def\bxjs@disable@kanjiskip{%  
3498   \bxjs@kanjiskip@enabledfalse  
3499   \bxjs@reset@kanjiskip}
```

`\bxjs@reset@kanjiskip` 現在の和文間空白の設定を実際にエンジンに反映させる。

```
3500 \bxjs@robust@def\bxjs@reset@kanjiskip{%  
3501   \ifbxjs@kanjiskip@enabled  
3502     \setlength{\@tempskipa}{\bxjs@kanjiskip}%  
3503   \else \@tempskipa\z@  
3504   \fi  
3505   \bxjs@apply@kanjiskip}
```

`\bxjs@xkanjiskip` 和欧文間空白について同様のものを用意する。

```
\setxkanjiskip 3506 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3507 \newcommand*\setxkanjiskip[1]{%
```

```
3508   \edef\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3509   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3510 \newcommand*\getxkanjiskip{%
```

```
3511   \bxjs@xkanjiskip}
```

```
\bxjs@disable@xkanjiskip 3512 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3513 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3514   \bxjs@xkanjiskip@enabledtrue
```

```
3515   \bxjs@reset@xkanjiskip}
```

```
3516 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3517   \bxjs@xkanjiskip@enabledfalse
```

```
3518   \bxjs@reset@xkanjiskip}
```

```
3519 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```
3520   \ifbxjs@xkanjiskip@enabled
```

```
3521     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
```

```
3522   \else \@tempskipa\z@
```

```
3523   \fi
```

```
3524   \bxjs@apply@xkanjiskip}
```

`\jsResetDimen` を用いて、フォントサイズが変更された時に空白の量が追従するようにする。

```
3525 \g@addto@macro\jsResetDimen{%
```

```

3526 \bxjs@reset@kanjiskip
3527 \bxjs@reset@xkanjiskip}
3528 \let\bxjs@apply@kanjiskip\relax
3529 \let\bxjs@apply@xkanjiskip\relax

```

#### ■和文フォント指定の扱い

\bxjs@adjust@jafont \jsJaFont に入っている和文フォント設定の値を“調整”して、その結果を \bxjs@tmpa に返す。#1 が f の場合は“非埋込 (noEmbed)”の設定が禁止される。この禁止の場合も含め、何か異常がある場合は \bxjs@tmpa は空になる。

```

3530 \@onlypreamble\bxjs@adjust@jafont
3531 \def\bxjs@adjust@jafont#1{%
3532   \ifx\jsJaFont\bxjs@@auto
3533     \bxjs@get@kanjiEmbed
3534     \ifx\bxjs@kanjiEmbed\relax
3535       \let\bxjs@tmpa\@empty
3536     \else
3537       \let\bxjs@tmpa\bxjs@kanjiEmbed
3538     \fi
3539   \else
3540     \let\bxjs@tmpa\jsJaFont
3541   \fi
3542   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3543     \ClassWarningNoLine\bxjs@clsname
3544     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3545      not available on the current situation}%
3546     \let\bxjs@tmpa\@empty
3547   \fi\fi
3548 }
3549 \def\bxjs@@auto{auto}
3550 \def\bxjs@@noEmbed{noEmbed}

```

\bxjs@kanjiEmbed 現在の updmap の kanjiEmbed パラメタの値。 \bxjs@get@kanjiEmbed により実際の設定値が取得されてここに設定される。

```

3551 \let\bxjs@kanjiEmbed\relax

```

\bxjs@get@kanjiEmbed 現在の updmap の kanjiEmbed パラメタの値を取得する。

```

3552 \@onlypreamble\bxjs@get@kanjiEmbed
3553 \def\bxjs@get@kanjiEmbed{%
3554   \begingroup\setbox\z@=\hbox{%
3555     \global\let\bxjs@g@tmpa\relax
3556     \endlinechar\m@ne
3557     \let\do\@makeother\dospecials
3558     \catcode32=10 \catcode12=10 %form-feed
3559     \let\bxjs@tmpa\@empty
3560     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3561     \ifeof\@inputcheck\else
3562       \read\@inputcheck to\bxjs@tmpa

```

```

3563     \closein\@inputcheck
3564     \fi
3565     \ifx\bxjs@tmpa\@empty\else
3566         \openin\@inputcheck="\bxjs@tmpa"\relax
3567         \@tempwattrue
3568         \loop\if@tempwa
3569             \read\@inputcheck to\bxjs@tmpa
3570             \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3571             \ifx\bxjs@tmpb\relax\else
3572                 \global\let\bxjs@g@tmpa\bxjs@tmpb
3573                 \@tempwafalse
3574             \fi
3575             \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3576             \ifx\bxjs@tmpb\relax\else
3577                 \global\let\bxjs@g@tmpa\bxjs@tmpb
3578                 \@tempwafalse
3579             \fi
3580             \ifeof\@inputcheck \@tempwafalse \fi
3581         \repeat
3582     \fi
3583 } \endgroup
3584 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3585 }
3586 \@onlypreamble\bxjs@get@ke@a
3587 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3588     \ifx$#1$\def\bxjs@tmpb{#2}%
3589     \else \let\bxjs@tmpb\relax
3590     \fi}
3591 \@onlypreamble\bxjs@get@ke@b
3592 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3593     \ifx$#1$\def\bxjs@tmpb{#2}%
3594     \else \let\bxjs@tmpb\relax
3595     \fi}

```

\jachar \jachar{< 文字 >} : 和文文字として出力する。

```

3596 \newcommand*\jachar[1]{%
3597     \begingroup
3598         \jsLetHeadChar\bxjs@tmpa{#1}%
3599         \ifx\bxjs@tmpa\relax
3600             \ClassWarningNoLine\bxjs@clsname
3601             {Illegal argument given to \string\jachar}%
3602         \else
3603             \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3604         \fi
3605     \endgroup}

```

\jsJaChar を \jachar と等価にする。

```

3606 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
3607 \let\bxjs@jachar\@firstofone
```

■hyperref 対策 出力ページサイズに館する処理は `geometry` パッケージが行うので、`hyperref` 側の処理は無効にしておく。

```
3608 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

`\bxjs@fix@hyperref@unicode` `hyperref` の `unicode` オプションの値を固定する。

```
3609 \@onlypreamble\bxjs@fix@hyperref@unicode
3610 \def\bxjs@fix@hyperref@unicode#1{%
3611   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3612   \@namedef{KV@Hyp@bxjs/hook}##1{%
3613     \KV@Hyp@unicode{##1}%
3614     \def\KV@Hyp@unicode####1{%
3615       \expandafter\ifx\csname if##1\expandafter\endcsname
3616       \csname if####1\endcsname\else
3617       \ClassWarningNoLine\bxjs@clsname
3618       {Blcoked hyperref option 'unicode=####1'}%
3619       \fi
3620     }%
3621   }%
3622 }
```

`\bxjs@urgent@special` DVI のなるべく早い位置に `special` を出力する。

```
3623 \@onlypreamble\bxjs@urgent@special
3624 \def\bxjs@urgent@special#1{%
3625   \AtBeginDvi{\special{#1}}%
3626   \AtBeginDocument{%
3627     \@ifpackageloaded{atbegshi}{%
3628       \begingroup
3629       \toks\z@{\special{#1}}%
3630       \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3631       \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3632     \endgroup
3633   }{}%
3634 }%
3635 }
```

## C.2 pTeX 用設定

```
3636 \if j\jsEngine
```

■共通命令の実装

```
3637 \def\bxjs@apply@kanjiskip{%
3638   \kanjiskip\@tempskipa}
3639 \def\bxjs@apply@xkanjiskip{%
3640   \xkanjiskip\@tempskipa}
```

`\jaJaChar` のサブマクロ。

```
3641 \def\bxjs@jachar#1{%
3642   \bxjs@jachar@a#1...\@nil}
3643 \def\bxjs@jachar@a#1#2#3#4#5\@nil{%
```

引数が単一トークンなら和文文字トークンが得られたと見なし、それをそのまま出力する。

```
3644   \ifx.#2#1%
```

引数が複数トークンの場合は、UTF-8 のバイト列であるに見なし、そのスカラー値を `\@tempcnta` に代入する。

```
3645   \else\ifx.#3%
3646     \@tempcnta`#1 \multiply\@tempcnta64
3647     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3648     \bxjs@jachar@b
3649   \else\ifx.#4%
3650     \@tempcnta`#1 \multiply\@tempcnta64
3651     \advance\@tempcnta`#2 \multiply\@tempcnta64
3652     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3653     \bxjs@jachar@b
3654   \else
3655     \@tempcnta`#1 \multiply\@tempcnta64
3656     \advance\@tempcnta`#2 \multiply\@tempcnta64
3657     \advance\@tempcnta`#3 \multiply\@tempcnta64
3658     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3659     \bxjs@jachar@b
3660   \fi\fi\fi}
```

符号値が `\@tempcnta` の和文文字を出力する処理。

```
3661 \ifjsWithupTeX
3662   \def\bxjs@jachar@b{\kchar\@tempcnta}
3663 \else
3664   \def\bxjs@jachar@b{%
3665     \ifx\bxUInt\@undefined\else
3666       \bxUInt{\@tempcnta}%
3667     \fi}
3668 \fi
```

和欧文間空白の命令 `\>` の実装。

```
3669 \ifbxjs@xkanjiskip@cmd
3670   \def\bxjs@put@xkanjiskip{%
3671     \relax\ifmmode \mskip\medmuskip
3672     \else \hskip\xkanjiskip
3673     \fi}
3674   \ifjsWithTeX \protected\def\>{\bxjs@put@xkanjiskip}
3675   \else \def\>{\protect\bxjs@put@xkanjiskip}
3676   \fi
3677 \fi
```



■jis2004 パラメタ pxchfon と pxbabel では 2004JIS を指定するオプションの名が prefer2004jis である。

```
3678 \ifbxjs@jp@jismmiv
3679   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3680   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3681 \fi
```

■和文フォント指定の扱い pTeX は既定で kanji-config-updmap の設定に従うため、\jsJaFont が auto の場合は何もする必要がない。無指定でも auto でもない場合は、\jsJaFont をオプションにして pxchfon パッケージを読み込む。ここで、和文ドライバパラメタ font が指定されている場合は、その値を pxchfon のオプションに追加する。

```
3682 \let\bxjs@tmpa\jsJaFont
3683 \ifx\bxjs@tmpa\bxjs@@auto
3684   \let\bxjs@tmpa@empty
3685 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3686   \def\bxjs@tmpa{noembed}
3687 \fi\fi
3688 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3689 \ifx\bxjs@tmpa@empty\else
3690   \edef\bxjs@next{%
3691     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3692   }\bxjs@next
3693 \fi
```

■otf パッケージ対策 インストールされている otf パッケージが scale オプションに対応している場合は scale=(\jsScale の値) を事前に otf に渡す。

※ otf.sty の中に「\RequirePackage{keyval}」の行が存在するかにより判定している。(もっといい方法はないのか……。)

```
3694 \beginingroup
3695   \global\let\@gtempa\relax
3696   \catcode`\|=0 \catcode`\|=12
3697   |def|bxjs@check#1|@nil{%
3698     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3699   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3700     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3701   |catcode`\|=0 \catcode`\|=12
3702   \def\bxjs@check@b#1keyval#2@nnil{%
3703     \ifx$#2$\else
3704       \xdef\@gtempa{%
3705         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3706       \fi}
3707 \@firstofone{%
3708   \catcode10=12 \endlinechar\m@ne
3709   \let\do\@makeother \dospecials \catcode32=10
3710   \openin\@inputcheck=otf.sty\relax
3711   \@tempwattrue
```

```

3712 \loop\if@tempswa
3713 \ifeof\@inputcheck \@tempswafalse \fi
3714 \if@tempswa
3715 \read\@inputcheck to\bxjs@line
3716 \expandafter\bxjs@check\bxjs@line\@nil
3717 \fi
3718 \repeat
3719 \closein\@inputcheck
3720 \endgroup}
3721 \@gtempa

```

■hyperref 対策 unicode にはいけない。

```

3722 \ifbxjs@hyperref@enc
3723 \bxjs@fix@hyperref@unicode{false}
3724 \fi

```

tounicode special 命令を出力する。

```

3725 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3726 \else\ifjsWithpTeXng T\else F\fi\fi T%
3727 \ifnum\jis"2121="A1A1 %euc
3728 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3729 \else\ifnum\jis"2121="8140 %sjis
3730 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3731 \else\ifnum\jis"2121="3000 %uptex
3732 \ifbxjs@bigcode
3733 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3734 \PassOptionsToPackage{bigcode}{pxjahyper}
3735 \else
3736 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3737 \fi
3738 \fi\fi\fi
3739 \let\bxToUnicodeSpecialDone=t
3740 \fi

```

■和文数式ファミリ 和文数式ファミリは既定で有効とする。すなわち enablejfam=false 以外の場合は @enablejfam を真にする。

```

3741 \ifx f\bxjs@enablejfam\else
3742 \@enablejfamtrue
3743 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

3744 \if@enablejfam
3745 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3746 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3747 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3748 \jfam\symmincho
3749 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3750 \AtBeginDocument{%
3751 \ifx\reDeclareMathAlphabet\undefined\else

```

```

3752 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3753 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3754 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3755 \fi}
3756 \fi

```

### C.3 pdfTeX 用設定 : CJK + bxcjkatype

```

3757 \else\if p\jsEngine

```

■bxcjkatype パッケージの読み込 \jsJaFont が指定されている場合は、その値を bxcjkatype のオプション（プリセット指定）に渡す。（auto ならば \bxjs@get@kanjiEmbed を実行する。）スケール値（\jsScale）の反映は bxcjkatype の側で行われる。

```

3758 \bxjs@adjust@jafont{f}
3759 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3760 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole,autotilde}}
3761 \edef\bxjs@next{%
3762 \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
3763 }\bxjs@next
3764 \bxjs@cjk@loaded

```

■hyperref 対策 bxcjkatype 使用時は unicode にするべき。

※取りあえず固定はしない。

```

3765 \ifbxjs@hyperref@enc
3766 \PassOptionsToPackage{unicode}{hyperref}
3767 \fi

```

\hypersetup 命令で（CJK\* 環境に入れなくても）日本語文字を含む文書情報を設定できるようにするための細工。

※ bxcjkatype を whole 付きで使っていることが前提。

※パッケージオプションでの指定に対応するのは、「アクティブな高位バイトトークンがその場で展開されてしまう」ため困難である。

```

3768 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3769 \begingroup
3770 \CJK@input{UTF8.bdg}
3771 \endgroup
3772 \g@addto@macro\pdfstringdefPreHook{%
3773 \@nameuse{CJK@UTF8Binding}%
3774 }
3775 \fi

```

~ が和欧文間空白である場合は PDF 文字列中で空白文字でなく空に展開させる。

```

3776 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3777 \g@addto@macro\pdfstringdefPreHook{%
3778 \ifx~\bxjs@@CJKtilde
3779 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace

```

```

3780 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3781 \let~\@empty
3782 \fi
3783 }
3784 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3785 \def\bxjs@@tildecmd{~}
3786 \def\bxjs@LetUnexpandableSpace#1{%
3787 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
3788 \bxjs@org@LetUnexpandableSpace#1%
3789 \fi}
3790 \fi

```

### ■ 共通命令の実装

```

3791 \newskip\jsKanjiSkip
3792 \newskip\jsXKanjiSkip
3793 \ifx\CJKecglue\@undefined
3794 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3795 \fi
3796 \let\autospacing\bxjs@enable@kanjiskip
3797 \let\noautospacing\bxjs@disable@kanjiskip
3798 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3799 \def\bxjs@apply@kanjiskip{%
3800 \jsKanjiSkip\@tempskipa
3801 \let\CJKglue\bxjs@CJKglue}
3802 \let\autoxspacing\bxjs@enable@xkanjiskip
3803 \let\noautoxspacing\bxjs@disable@xkanjiskip
3804 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3805 \def\bxjs@apply@xkanjiskip{%
3806 \jsXKanjiSkip\@tempskipa
3807 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

3808 \def\bxjs@jachar#1{%
3809 \CJKforced{#1}}

```

\> は bxcjkjatype で定義されているが、保護付のマクロに変換する。

```

3810 \ifbxjs@xkanjiskip@cmd
3811 \protected\def\>{%
3812 \relax\ifmmode \mskip\medmuskip \else \CJKecglue\ignorespaces \fi}
3813 \fi

```

■ 和文数式ファミリ CJK パッケージは（恐らく）数式文字として CJK 文字をサポートしていない。従って @enablejfam は常に偽になる。

```

3814 \ifx t\bxjs@enablejfam
3815 \ClassWarningNoLine\bxjs@clsname
3816 {You cannot use 'enablejfam=true', since the\MessageBreak
3817 CJK package does not support Japanese math}
3818 \fi

```

## C.4 X<sub>Ǝ</sub>T<sub>E</sub>X 用設定：xeCJK + zxjatype

```
3819 \else\if x\jsEngine
```

■zxjatype パッケージの読み込 スケール値 (\jsScale) の反映は zxjatype の側で行われる。

```
3820 \RequirePackage{zxjatype}
3821 \PassOptionsToPackage{no-math}{fontspec}%!
3822 \PassOptionsToPackage{xetex}{graphicx}%!
3823 \PassOptionsToPackage{xetex}{graphics}%!
3824 \ifx\zxJaFamilyName\@undefined
3825   \ClassError\bxjs@clsname
3826   {xeCJK or zxjatype is too old}\@ehc
3827 \fi
```

■和文フォント定義 \jsJaFont が指定された場合は、その値をオプションとして zxjafont を読み込む。非指定の場合は IPAex フォントを使用する。

```
3828 \bxjs@adjust@jafont{f}
3829 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3830 \ifx\bxjs@tmpa\@empty
3831   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3832   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3833 \else
3834   \edef\bxjs@next{%
3835     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
3836   }\bxjs@next
3837 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X<sub>Ǝ</sub>T<sub>E</sub>X の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、X<sub>Ǝ</sub>T<sub>E</sub>X の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「X<sub>Ǝ</sub>T<sub>E</sub>X のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```
3838 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3839   \ifbxjs@hyperref@enc
3840     \PassOptionsToPackage{unicode}{hyperref}
3841   \fi
3842 \fi
```

■段落頭でのグルー挿入禁止 どうやら、zxjatype の \inhibitglue の実装が極めて杜撰なため、1.0 版での実装では全く期待通りの動作をしていないし、そもそも（少なくとも現

状の) `\xeCJK` では、段落頭での `\inhibitglue` は実行しないほうが JS クラスの出力に近いものが得られるらしい。

従って、`\jsInhibitGlueAtParTop` は結局何もしないことにする。

```
3843 \let\jsInhibitGlueAtParTop\@empty
```

#### ■ 共通命令の実装

```
3844 \newskip\jsKanjiSkip
3845 \newskip\jsXKanjiSkip
3846 \ifx\CJKecglue\@undefined
3847   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3848 \fi
3849 \let\autospacing\bxjs@enable@kanjiskip
3850 \let\noautospacing\bxjs@disable@kanjiskip
3851 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3852 \def\bxjs@apply@kanjiskip{%
3853   \jsKanjiSkip\@tempskipa
3854   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3855 \let\autoxspacing\bxjs@enable@xkanjiskip
3856 \let\noautoxspacing\bxjs@disable@xkanjiskip
3857 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3858 \def\bxjs@apply@xkanjiskip{%
3859   \jsXKanjiSkip\@tempskipa
3860   \xeCJKsetup{CJKecglue={\bxjs@CJKecglue}}}
```

`\mcfamily`、`\gtfamily` は本来は `zxjatype` の方で定義すべきであろうが、現状は暫定的にここで定義する。

```
3861 \ifx\mcfamily\@undefined
3862   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3863   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3864 \fi
```

`\jachar` のサブマクロの実装。

```
3865 \def\bxjs@jachar#1{%
3866   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3867   #1}
```

`\>` は `zxjatype` で定義されているが、`standard` の仕様に合わせて変更する。

※元は `\ignorespaces` でなく `\relax` (`\scan_stop:`) だった。

```
3868 \ifbxjs@xkanjiskip@cmd
3869   \protected\def\>{%
3870     \relax\ifmmode \mskip\medmuskip \else \CJKecglue\ignorespaces \fi}
3871 \fi
```

■ 和文数式ファミリ 和文数式ファミリは既定で無効とする。すなわち `enablejfam=true` の場合にのみ `@enablejfam` を真にする。

```
3872 \ifx t\bxjs@enablejfam
3873   \@enablejfamtrue
3874 \fi
```

実際に和文用の数式ファミリの設定を行う。

※ FIXME: 要検討。

```
3875 \if@enablejfam
3876   \xeCJKsetup{CJKmath=true}
3877 \fi
```

## C.5 LuaTeX 用設定 : LuaTeX-ja

```
3878 \else\if 1\jsEngine
```

■ LuaTeX-ja パッケージの読込 `luatexja` とともに `luatexja-fontspec` パッケージを読み込む。

`luatexja` は自前の `\zw` (これは実際の現在和文フォントに基づく値を返す) を定義するので、`\zw` の定義を消しておく。なお、レイアウト定義の「全角幅」は「規定」に基づく `\jsZw` であることに注意が必要。

※ 1.0b 版から「graphics パッケージに `pdftex` オプションを渡す」処理を行っていたが、1.4 版で廃止された。

```
3879 \let\zw\@undefined
3880 \RequirePackage{luatexja}
3881 \RequirePackage{luatexja-fontspec}
```

■ 和文フォント定義 `luatexja-fontspec` で使用する和文スケール値を `\jsScale` と合致させたいのだが……もっと良い方法はないのか？

```
3882 \ExplSyntaxOn
3883 \fp_gset:Nn \g_ljtj_fontspec_scale_fp { \jsScale }
3884 \ExplSyntaxOff
```

`\jsJaFont` が指定された場合は、その値をオプションとして `luatexja-preset` を読み込む。非指定の場合は、`luatexja-preset` パッケージの `ipaex` オプション (IPAex フォント使用) と等価な設定を用いる (`luatexja-preset` は読み込まない)。

```
3885 \bxjs@adjust@jafont{t}
3886 \ifx\bxjs@tmpa\bxjs@noEmbed
3887   \def\bxjs@tmpa{noembed}
3888 \fi
3889 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3890 \ifx\bxjs@tmpa\@empty
3891   \defaultjfontfeatures{ Kerning=Off }
3892   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3893   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3894 \else
3895   \edef\bxjs@next{%
3896     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
3897   }\bxjs@next
3898 \fi
```

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。

```

3899 \DeclareRobustCommand\rmfamily
3900   {\not@math@alphabet\rmfamily\mathrm
3901     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3902 \DeclareRobustCommand\sffamily
3903   {\not@math@alphabet\sffamily\mathsf
3904     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3905 \DeclareRobustCommand\ttfamily
3906   {\not@math@alphabet\ttfamily\mathtt
3907     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3908 \AtBeginDocument{%
3909   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3910   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
3911   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
3912 \bxjs@if@sf@default{%
3913   \renewcommand\kanjifamilydefault{\gtdefault}}

```

### ■和文パラメタの設定

```

3914 % 次の3つは既定値の通り
3915 \ltjsetparameter{prebreakpenalty}={`',10000}}
3916 \ltjsetparameter{postbreakpenalty}={`",10000}}
3917 \ltjsetparameter{prebreakpenalty}={`",10000}}
3918 \ltjsetparameter{jaxspmode}={`!',1}}
3919 \ltjsetparameter{jaxspmode}={`〒,2}}
3920 \ltjsetparameter{alxspmode}={`+,3}}
3921 \ltjsetparameter{alxspmode}={`%,3}}

```

### ■段落頭でのグルー挿入禁止

```

3922 \protected\def\@inhibitglue{%
3923   \directlua{%
3924     luatexja.jfmglue.create_beginpar_node()}}
3925 \let\bxjs@ltj@inhibitglue\@inhibitglue
3926 \let\@inhibitglue\@undefined

```

### ■hyperref 対策 unicode にするべき。

```

3927 \ifbxjs@hyperref@enc
3928   \bxjs@fix@hyperref@unicode{true}
3929 \fi

```

### ■共通命令の実装

```

3930 \protected\def\autospacing{%
3931   \ltjsetparameter{autospacing=true}}
3932 \protected\def\noautospacing{%
3933   \ltjsetparameter{autospacing=false}}
3934 \protected\def\autoxspacing{%
3935   \ltjsetparameter{autoxspacing=true}}
3936 \protected\def\noautoxspacing{%
3937   \ltjsetparameter{autoxspacing=false}}
3938 \def\bxjs@apply@kanjiskip{%

```



```

3939 \ltjsetparameter{kanjiskip={\@tempskipa}}
3940 \def\bxjs@apply@xkanjiskip{%
3941 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

\jachar のサブマクロの実装。

3942 \def\bxjs@jachar#1{%
3943 \ltjjachar`#1\relax}

\> の実装。

3944 \ifbxjs@xkanjiskip@cmd
3945 \protected\def\>{%
3946 \relax\ifmmode \mskip\medmuskip
3947 \else \hskip\ltjgetparameter{xkanjiskip}\ignorespaces
3948 \fi}
3949 \fi

```

■和文数式ファミリ Lua<sub>T</sub><sub>E</sub>X-j<sub>a</sub> では和文数式ファミリは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablejfam は常に真になる。

```

3950 \ifx f\bxjs@enablejfam
3951 \ClassWarningNoLine\bxjs@clsname
3952 {You cannot use 'enablejfam=false', since the\MessageBreak
3953 LuaTeX-ja always provides Japanese math families}
3954 \fi

```

## C.6 共通処理 (2)

```

3955 \fi\fi\fi\fi

```

### ■共通命令の実装

\textmc minimal ドライバ実装中で定義した \DeclareJaTextFontCommand を利用する。

```

\textgt 3956 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3957 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3958 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3959 \fi

```

\mathmc この時点で未定義である場合に限り、\DeclareJaMathFontCommand を利用したフォール

\mathgt バックの定義を行う。

```

3960 \ifx\mathmc\@undefined
3961 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
3962 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
3963 \fi

```

### ■和文・和欧文間空白の初期値

```

3964 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3965 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3966 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3967 \fi

```

以上で終わり。

```
3968 %</standard>
```

## 付録 D 和文ドライバ : modern

モダンな設定。

standard ドライバの設定を引き継ぐ。

```
3969 %<*modern>
```

```
3970 \input{bxjsja-standard.def}
```

### D.1 フォント設定

T1 エンコーディングに変更する。

※以下のコードは `\usepackage[T1]{fontenc}` と同等。

```
3971 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
```

```
3972 \def\encodingdefault{T1}%
```

```
3973 \input{t1enc.def}%
```

```
3974 \fontencoding\encodingdefault\selectfont
```

```
3975 \fi
```

基本フォントを Latin Modern フォントファミリーに変更する。

※以下は `\usepackage[noamth]{lmodern}` と同じ。ユーザは後で `lmodern` を好きなオプションを付けて読み込むことができる。

```
3976 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
```

```
3977 \renewcommand{\rmdefault}{lmr}
```

```
3978 \renewcommand{\sfdefault}{lms}
```

```
3979 \renewcommand{\ttdefault}{lmtt}
```

```
3980 \fi
```

大型演算子用の数式フォントの設定。

※ `amsfonts` パッケージと同等にする。

```
3981 \DeclareFontShape{OMX}{cmex}{m}{n}{%
```

```
3982 <-7.5>cmex7<7.5-8.5>cmex8%
```

```
3983 <8.5-9.5>cmex9<9.5->cmex10}{}%
```

```
3984 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
```

`amsmath` 読込時に上書きされるのを防ぐ。

```
3985 \def\cmex@opt{10}
```

### D.2 fixltx2e 読込

※ `fixltx2e` 廃止前の L<sup>A</sup>T<sub>E</sub>X カーネルの場合。

```
3986 \ifx\@IncludeInRelease\@undefined
```

```
3987 \RequirePackage{fixltx2e}
```

```
3988 \fi
```

### D.3 和文カテゴリコード

和文カテゴリコード設定のための補助パッケージを読みこむ。

```
3989 \RequirePackage{bxjscjkat}
```

### D.4 完了

おしまい。

```
3990 %</modern>
```

## 付録 E 和文ドライバ : pandoc

Pandoc 用の何か。

standard ドライバの設定を引き継ぐ。

```
3991 %<*pandoc>
```

```
3992 \input{bxjsja-standard.def}
```

### E.1 dupload システム

パッケージが重複して読み込まれたときに “option clash” の検査をスキップする。この時に何らかのコードを実行させることができる。

`\bxjs@set@dupload@proc` `\bxjs@set@dupload@proc{〈ファイル名〉}{〈定義本体〉}` 特定のファイルの読込が `\@filewithoptions` で指示されて、しかもそのファイルが読込済である場合に、オプション重複検査をスキップして、代わりに 〈定義本体〉 のコードを実行する。このコード中で #1 は渡されたオプション列のテキストに置換される。

```
3993 \@onlypreamble\bxjs@set@dupload@proc
```

```
3994 \def\bxjs@set@dupload@proc#1{%
```

```
3995   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
3996 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
3997 \def\bxjs@set@dupload@proc@a#1{%
```

```
3998   \@onlypreamble#1\def#1##1}
```

`\@if@options` `\@if@options` の再定義。

```
3999 \@onlypreamble\bxjs@org@if@options
```

```
4000 \let\bxjs@org@if@options\@if@options
```

```
4001 \newif\ifbxjs@dlp
```

```
4002 \def\@if@options#1#2#3{%
```

```
4003   \bxjs@dlpfalse
```

```
4004   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentt}%
```

```
4005   \ifx\bxjs@tmpa\bxjs@tmpb
```

```
4006     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
```

```
4007       \bxjs@dlptrue \fi
```

```
4008   \fi
```

```

4009 \ifbxjs@dlp \expandafter\bxjs@do@upload@proc
4010 \else \expandafter\bxjs@org@if@ptions
4011 \fi {#1}{#2}{#3}}
4012 \AtBeginDocument{%
4013 \let\@if@ptions\bxjs@org@if@ptions}
4014 \@onlypreamble\bxjs@do@upload@proc
4015 \def\bxjs@do@upload@proc#1#2#3{%
4016 \csname bxjs@dlp/#2.#1\endcsname{#3}%
4017 \@firstoftwo}

```

\bxjs@mark@as@loaded \bxjs@mark@as@loaded{(ファイル名)} : 特定のファイルに対して、  
(\@filewithoptions の処理に関して) 読込済であるとマークする。

```

4018 \def\bxjs@mark@as@loaded#1{%
4019 \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
4020 \def\bxjs@mal@a#1#2{%
4021 \ifx#1\relax
4022 \def#1{2001/01/01}%
4023 \ClassInfo\bxjs@clsname
4024 {File '#2' marked as loaded\@gobble}%
4025 \fi}

```

## E.2 lang 変数

lang=ja という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐため、とりあえず両パッケージを無効化しておく。

```

4026 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0

```

Polyglossia について。

```

4027 \bxjs@mark@as@loaded{polyglossia.sty}
4028 \bxjs@set@upload@proc{polyglossia.sty}{%
4029 \ClassWarning\bxjs@clsname
4030 {Loading of polyglossia is blocked}}
4031 \ifx\setmainlanguage\@undefined
4032 \newcommand*\setmainlanguage[2][]{%
4033 \newcommand*\setotherlanguage[2][]{%
4034 \ifcat_#2_\else
4035 \expandafter\let\csname #2\endcsname\@empty
4036 \expandafter\let\csname end#2\endcsname\@empty
4037 \expandafter\let\csname text#2\endcsname\@firstofone
4038 \fi}
4039 \newcommand*\setotherlanguages[2][]{%
4040 \@for\bxjs@tmpa:={#2}\do{%
4041 \setotherlangauge{\bxjs@tmpa}}}
4042 \fi
4043 \else

```

Babel について。

```

4044 \bxjs@mark@as@loaded{babel.sty}
4045 \bxjs@set@dupload@proc{babel.sty}{%
4046   \ClassWarning\bxjs@clsname
4047     {Loading of babel is blocked}}
4048 \let\foreignlanguage\@secondoftwo
4049 \let\otherlanguage\@gobble
4050 \let\endotherlanguage\@empty
4051 \fi

```

### E.3 geometry 変数

geometry を “再度読み込んだ” 場合に、そのパラメタで `\setpagelayout*` が呼ばれるようにする。

```

4052 \bxjs@set@dupload@proc{geometry.sty}{%
4053   \setpagelayout*{#1}}

```

### E.4 CJKmainfont 変数

LuaTeX (+ LuaTeX-ja) の場合に CJKmainfont 変数が指定された場合は `\setmainfont` の指定にまわす。

```

4054 \if 1\jsEngine
4055   \bxjs@mark@as@loaded{xeCJK.sty}
4056   \providecommand*\setCJKmainfont{\setmainfont}
4057 \fi

```

### E.5 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近 (2015 年版以降) の L<sup>A</sup>T<sub>E</sub>X ではこれで警告が出る。これを抑止する。

L<sup>A</sup>T<sub>E</sub>X カーネルが新しい場合は `fixltx2e` を読込済にする。

```

4058 \ifx\@IncludeInRelease\@undefined\else
4059   \bxjs@mark@as@loaded{fixltx2e.sty}
4060 \fi

```

### E.6 cmap パッケージ

エンジンが (u)pL<sup>A</sup>T<sub>E</sub>X のときに `cmap` パッケージが読み込まれるのを阻止する。(実際は警告が出るだけで無害であるが。)

```

4061 \if j\jsEngine
4062   \bxjs@mark@as@loaded{cmap.sty}
4063 \fi

```

## E.7 microtype パッケージ

警告が多すぎなので消す。

```
4064 \if j\jsEngine \else
4065   \PassOptionsToPackage{verbose=silent}{microtype}
4066 \fi
```

エンジンが (u)pL<sup>A</sup>T<sub>E</sub>X のときに microtype パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は standard ドライバでこの処理を行っていたが、元来は Pandoc 用の処理なので、1.5 版で pandoc に移動。

```
4067 \if j\jsEngine
4068   \bxjs@mark@as@loaded{microtype.sty}
4069   \newcommand*\UseMicrotypeSet[2][]{\}
4070 \fi
```

## E.8 完了

おしまい。

```
4071 %</pandoc>
```

和文ドライバ実装はここまで。

```
4072 %</drv>
```

## 付録 F 補助パッケージ一覧

BXJS クラスの機能を実現するために用意されたものだが、他のクラスの文書で読み込んで利用することもできる。

- bxjscjkat : modern ドライバ用の和文カテゴリを適用する。

```
4073 %<*anc>
```

## 付録 G 補助パッケージ : bxjscompat

ムニャムニャムニャ……。

### G.1 準備

```
4074 %<*compat>
4075 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4076 \let\bxac@engine=n
4077 \def\bxac@do#1#2{%
```

```

4078 \edef\bxac@tmpa{\string#1}%
4079 \edef\bxac@tmpb{\meaning#1}%
4080 \ifx\bxac@tmpa\bxac@tmpb #2\fi}
4081 \bxac@do\XeTeXversion{\let\bxac@engine=x}
4082 \bxac@do\luatexversion{\let\bxac@engine=l}

```

`\bxac@delayed@if@bxjs` もし BXJS クラスの読込中でこのパッケージが読み込まれているならば、BXJS のクラスの  
 終わりまで実行を遅延する。

```

4083 \ifx\jsAtEndOfClass\@undefined
4084 \let\bxac@delayed@if@bxjs\@firstofone
4085 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
4086 \fi

```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```

\RevokeOldLuaTeXBehavior 4087 \newif\ifbxac@in@old@behavior
4088 \let\ImposeOldLuaTeXBehavior\relax
4089 \let\RevokeOldLuaTeXBehavior\relax

```

## G.2 XeTeX 部分

```

4090 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。
4091 \@onlypreamble\bxac@adjust@charclass
4092 \bxac@delayed@if@bxjs{%
4093 \ifpackageloaded{xCJK}{-}{%else
4094 \ifx\Xe@alloc@intercharclass\@undefined\else
4095 \ifnum\Xe@alloc@intercharclass=\z@
4096 \PackageInfo\bxac@pkgname
4097 {Setting up interchar class for CJK...\@gobble}%
4098 \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4099 \Xe@alloc@intercharclass=3
4100 }{%else
4101 \PackageWarning\bxac@pkgname
4102 {Cannot find file 'load-unicode-xetex-classes.tex'%
4103 \@gobble}%
4104 }%
4105 \fi\fi
4106 \ifnum\XeTeXcharclass"3041=\z@
4107 \PackageInfo\bxac@pkgname
4108 {Adjusting interchar class for CJK...\@gobble}%
4109 \@for\bxac@x:={%
4110 3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4111 3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4112 30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4113 31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4114 31FF%
4115 }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4116 \fi

```

```

4117 }%
4118 }

```

以上。

```

4119 \fi

```

### G.3 LuaTeX 部分

```

4120 \ifx l\bxac@engine

```

ムニヤムニヤ。

```

4121 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4122 \chardef\pdftexversion=200
4123 \def\pdftexrevision{0}
4124 \let\pdftexbanner\luatexbanner
4125 \fi\fi

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 4126 \begingroup\expandafter\expandafter\expandafter\endgroup
4127 \expandafter\ifx\csname outputmode\endcsname\relax\else
4128 \def\bxac@ob@list{%
4129 \do{\let}\pdfoutput{\outputmode}%
4130 \do{\let}\pdfpagewidth{\pagewidth}%
4131 \do{\let}\pdfpageheight{\pageheight}%
4132 \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4133 \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}%
4134 \def\bxac@ob@do#1#2{\begingroup
4135 \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4136 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4137 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4138 \else \let#3#1\relax \let#1\@undefined
4139 \fi}
4140 \protected\def\ImposeOldLuaTeXBehavior{%
4141 \unless\ifbxac@in@old@behavior
4142 \bxac@in@old@behaviortrue
4143 \let\do\bxac@ob@do \bxac@ob@list
4144 \fi}
4145 \protected\def\RevokeOldLuaTeXBehavior{%
4146 \ifbxac@in@old@behavior
4147 \bxac@in@old@behaviorfalse
4148 \let\do\bxac@ob@do \bxac@ob@list
4149 \fi}
4150 \fi

```

漢字および完成形ハングルのカテゴリコードのムニヤムニヤ。

```

4151 \ifnum\luatexversion>64 \directlua{
4152 local function range(cs, ce, cc, ff)
4153 if ff or not tex.getcatcode(cs) == cc then
4154 local setcc = tex.setcatcode
4155 for c = cs, ce do setcc(c, cc) end
4156 end

```



```

4157     end
4158     range(0x3400, 0x4DB5, 11, false)
4159     range(0x4DB5, 0x4DBF, 11, true)
4160     range(0x4E00, 0x9FCC, 11, false)
4161     range(0x9FCD, 0x9FFF, 11, true)
4162     range(0xAC00, 0xD7A3, 11, false)
4163     range(0x20000, 0x2A6D6, 11, false)
4164     range(0x2A6D7, 0x2A6FF, 11, true)
4165     range(0x2A700, 0x2B734, 11, false)
4166     range(0x2B735, 0x2B73F, 11, true)
4167     range(0x2B740, 0x2B81D, 11, false)
4168     range(0x2B81E, 0x2B81F, 11, true)
4169     range(0x2B820, 0x2CEA1, 11, false)
4170     range(0x2CEA2, 0x2FFFD, 11, true)
4171 } \fi

```

以上。

```

4172 \fi

```

## G.4 完了

おしまい。

```

4173 %</compat>

```

## 付録 H 補助パッケージ : bxjscjkat

modern ドライバ用の和文カテゴリを適用する。

### H.1 準備

```

4174 %<*cjkat>
4175 \def\bxjx@pkgname{bxjscjkat}
4176 \newcount\bxjx@canta
\bxjx@engine エンジンの種別。
4177 \let\bxjx@engine=n
4178 \def\bxjx@do#1#2{%
4179   \edef\bxjx@tmpa{\string#1}%
4180   \edef\bxjx@tmpb{\meaning#1}%
4181   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4182 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
4183 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
4184 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
4185 \bxjx@do\pdfTeXversion{\let\bxjx@engine=p}
4186 \bxjx@do\luatexversion{\let\bxjx@engine=l}

```

それぞれのエンジンで、前提となる日本語処理パッケージが実際に読み込まれているかを  
 検査する。

```

4187 \def\bxjx@do#1#2{%
4188   \if#1\bxjx@engine
4189     \@ifpackageloaded{#2}{-}{%else
4190       \PackageError\bxjx@pkgname
4191         {Package '#2' must be loaded}%
4192         {Package loading is aborted.\MessageBreak\@ehc}%
4193     \endinput}
4194   \fi}
4195 \bxjx@do{p}{\bxcjkatype}
4196 \bxjx@do{x}{\xeCJK}
4197 \bxjx@do{l}{\luatexja}

```

古い L<sup>A</sup>T<sub>E</sub>X の場合、`\TextOrMath` は `fixltx2e` パッケージで提供される。

```

4198 \ifx\TextOrMath\undefined
4199   \RequirePackage{fixltx2e}
4200 \fi

```

## H.2 和文カテゴリコードの設定

upL<sup>A</sup>T<sub>E</sub>X の場合、和文カテゴリコードの設定を LuaT<sub>E</sub>X-ja と（ほぼ）等価なものに変更する。

※ LuaT<sub>E</sub>X-ja との相違点：A830、A960、1B000。

```

4201 \if u\bxjx@engine
4202 \@for\bxjx@x:={%
4203 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4204 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
4205 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
4206 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
4207 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
4208 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
4209 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
4210 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
4211 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
4212 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
4213 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
4214 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4215 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4216 10480,10500,10530,10600,10800,10840,10860,10880,%
4217 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
4218 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
4219 11000,11080,110D0,11100,11150,11180,111E0,11200,%
4220 11280,112B0,11300,11480,11580,11600,11680,11700,%
4221 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
4222 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4223 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
4224 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
4225 1F700,1F780,1F800,1F900,E0000,F0000,100000%
4226 }\do{\kcatcode"\bxjx@x=15 }

```

### H.3 ギリシャ・キリル文字の扱い

※ここで「ギリシャ・キリル文字」は Unicode と JIS X 0213 に共通して含まれるもののみを指すことにする。

\bxjx@grkcyr@list 対象のギリシャ・キリル文字に関するデータ。

4228	\def\bxjx@grkcyr@list{%	
4229	\do{0391}{LGR}{\textAlpha}{A}%	% GR. C. L. ALPHA
4230	\do{0392}{LGR}{\textBeta}{B}%	% GR. C. L. BETA
4231	\do{0393}{LGR}{\textGamma}{\Gamma}%	% GR. C. L. GAMMA
4232	\do{0394}{LGR}{\textDelta}{\Delta}%	% GR. C. L. DELTA
4233	\do{0395}{LGR}{\textEpsilon}{E}%	% GR. C. L. EPSILON
4234	\do{0396}{LGR}{\textZeta}{Z}%	% GR. C. L. ZETA
4235	\do{0397}{LGR}{\textEta}{H}%	% GR. C. L. ETA
4236	\do{0398}{LGR}{\textTheta}{\Theta}%	% GR. C. L. THETA
4237	\do{0399}{LGR}{\textIota}{I}%	% GR. C. L. IOTA
4238	\do{039A}{LGR}{\textKappa}{K}%	% GR. C. L. KAPPA
4239	\do{039B}{LGR}{\textLambda}{\Lambda}%	% GR. C. L. LAMDA
4240	\do{039C}{LGR}{\textMu}{M}%	% GR. C. L. MU
4241	\do{039D}{LGR}{\textNu}{N}%	% GR. C. L. NU
4242	\do{039E}{LGR}{\textXi}{\Xi}%	% GR. C. L. XI
4243	\do{039F}{LGR}{\textOmicron}{O}%	% GR. C. L. OMICRON
4244	\do{03A0}{LGR}{\textPi}{\Pi}%	% GR. C. L. PI
4245	\do{03A1}{LGR}{\textRho}{P}%	% GR. C. L. RHO
4246	\do{03A3}{LGR}{\textSigma}{\Sigma}%	% GR. C. L. SIGMA
4247	\do{03A4}{LGR}{\textTau}{T}%	% GR. C. L. TAU
4248	\do{03A5}{LGR}{\textUpsilon}{\Upsilon}%	% GR. C. L. UPSILON
4249	\do{03A6}{LGR}{\textPhi}{\Phi}%	% GR. C. L. PHI
4250	\do{03A7}{LGR}{\textChi}{X}%	% GR. C. L. CHI
4251	\do{03A8}{LGR}{\textPsi}{\Psi}%	% GR. C. L. PSI
4252	\do{03A9}{LGR}{\textOmega}{\Omega}%	% GR. C. L. OMEGA
4253	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
4254	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
4255	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
4256	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4257	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4258	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4259	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4260	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4261	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4262	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4263	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4264	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4265	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4266	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4267	\do{03BF}{LGR}{\textomicron}{o}%	% GR. S. L. OMICRON

4268	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4269	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4270	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4271	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4272	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4273	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4274	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4275	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4276	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4277	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4278	\do{0401}{T2A}{\CYRYO}{\O}%	% CY. C. L. IO
4279	\do{0410}{T2A}{\CYRA}{\A}%	% CY. C. L. A
4280	\do{0411}{T2A}{\CYRB}{\B}%	% CY. C. L. BE
4281	\do{0412}{T2A}{\CYRV}{\V}%	% CY. C. L. VE
4282	\do{0413}{T2A}{\CYRG}{\G}%	% CY. C. L. GHE
4283	\do{0414}{T2A}{\CYRD}{\D}%	% CY. C. L. DE
4284	\do{0415}{T2A}{\CYRE}{\E}%	% CY. C. L. IE
4285	\do{0416}{T2A}{\CYRZH}{\ZH}%	% CY. C. L. ZHE
4286	\do{0417}{T2A}{\CYRZ}{\Z}%	% CY. C. L. ZE
4287	\do{0418}{T2A}{\CYRI}{\I}%	% CY. C. L. I
4288	\do{0419}{T2A}{\CYRISHRT}{\I}%	% CY. C. L. SHORT I
4289	\do{041A}{T2A}{\CYRK}{\K}%	% CY. C. L. KA
4290	\do{041B}{T2A}{\CYRL}{\L}%	% CY. C. L. EL
4291	\do{041C}{T2A}{\CYRM}{\M}%	% CY. C. L. EM
4292	\do{041D}{T2A}{\CYRN}{\N}%	% CY. C. L. EN
4293	\do{041E}{T2A}{\CYRO}{\O}%	% CY. C. L. O
4294	\do{041F}{T2A}{\CYRP}{\P}%	% CY. C. L. PE
4295	\do{0420}{T2A}{\CYRR}{\R}%	% CY. C. L. ER
4296	\do{0421}{T2A}{\CYRS}{\S}%	% CY. C. L. ES
4297	\do{0422}{T2A}{\CYRT}{\T}%	% CY. C. L. TE
4298	\do{0423}{T2A}{\CYRU}{\U}%	% CY. C. L. U
4299	\do{0424}{T2A}{\CYRF}{\F}%	% CY. C. L. EF
4300	\do{0425}{T2A}{\CYRH}{\H}%	% CY. C. L. HA
4301	\do{0426}{T2A}{\CYRC}{\C}%	% CY. C. L. TSE
4302	\do{0427}{T2A}{\CYRCH}{\CH}%	% CY. C. L. CHE
4303	\do{0428}{T2A}{\CYRSH}{\SH}%	% CY. C. L. SHA
4304	\do{0429}{T2A}{\CYRSHCH}{\CH}%	% CY. C. L. SHCHA
4305	\do{042A}{T2A}{\CYRHRDSN}{\SH}%	% CY. C. L. HARD SIGN
4306	\do{042B}{T2A}{\CYRERY}{\Y}%	% CY. C. L. YERU
4307	\do{042C}{T2A}{\CYRSFTSN}{\SH}%	% CY. C. L. SOFT SIGN
4308	\do{042D}{T2A}{\CYREREV}{\E}%	% CY. C. L. E
4309	\do{042E}{T2A}{\CYRYU}{\YU}%	% CY. C. L. YU
4310	\do{042F}{T2A}{\CYRYA}{\YA}%	% CY. C. L. YA
4311	\do{0430}{T2A}{\cyra}{\A}%	% CY. S. L. A
4312	\do{0431}{T2A}{\cyrb}{\B}%	% CY. S. L. BE
4313	\do{0432}{T2A}{\cyrv}{\V}%	% CY. S. L. VE
4314	\do{0433}{T2A}{\cyrg}{\G}%	% CY. S. L. GHE
4315	\do{0434}{T2A}{\cyrd}{\D}%	% CY. S. L. DE
4316	\do{0435}{T2A}{\cyre}{\E}%	% CY. S. L. IE

```

4317 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE
4318 \do{0437}{T2A}{\cyrz}{}% % CY. S. L. ZE
4319 \do{0438}{T2A}{\cyri}{}% % CY. S. L. I
4320 \do{0439}{T2A}{\cyrishrt}{}% % CY. S. L. SHORT I
4321 \do{043A}{T2A}{\cyrk}{}% % CY. S. L. KA
4322 \do{043B}{T2A}{\cyr1}{}% % CY. S. L. EL
4323 \do{043C}{T2A}{\cyrM}{}% % CY. S. L. EM
4324 \do{043D}{T2A}{\cyrn}{}% % CY. S. L. EN
4325 \do{043E}{T2A}{\cyro}{}% % CY. S. L. O
4326 \do{043F}{T2A}{\cyrp}{}% % CY. S. L. PE
4327 \do{0440}{T2A}{\cyr r}{}% % CY. S. L. ER
4328 \do{0441}{T2A}{\cyr s}{}% % CY. S. L. ES
4329 \do{0442}{T2A}{\cyr t}{}% % CY. S. L. TE
4330 \do{0443}{T2A}{\cyr u}{}% % CY. S. L. U
4331 \do{0444}{T2A}{\cyr f}{}% % CY. S. L. EF
4332 \do{0445}{T2A}{\cyr h}{}% % CY. S. L. HA
4333 \do{0446}{T2A}{\cyr c}{}% % CY. S. L. TSE
4334 \do{0447}{T2A}{\cyr ch}{}% % CY. S. L. CHE
4335 \do{0448}{T2A}{\cyr sh}{}% % CY. S. L. SHA
4336 \do{0449}{T2A}{\cyrshch}{}% % CY. S. L. SHCHA
4337 \do{044A}{T2A}{\cyrhrdsn}{}% % CY. S. L. HARD SIGN
4338 \do{044B}{T2A}{\cyr rery}{}% % CY. S. L. YERU
4339 \do{044C}{T2A}{\cyr sftsn}{}% % CY. S. L. SOFT SIGN
4340 \do{044D}{T2A}{\cyr rerev}{}% % CY. S. L. E
4341 \do{044E}{T2A}{\cyr yu}{}% % CY. S. L. YU
4342 \do{044F}{T2A}{\cyr ya}{}% % CY. S. L. YA
4343 \do{0451}{T2A}{\cyr yo}{}% % CY. S. L. IO
4344 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4345 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4346 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4347 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN
4348 \do{00B4}{TS1}{\textasciicute}{}% % ACUTE ACCENT
4349 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4350 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4351 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN
4352 }

4353 \providecommand*{\mathdegree}{\circ}

\ifbxjx@gcc@cjk [スイッチ] ギリシャ・キリル文字を和文扱いにするか。
4354 \newif\ifbxjx@gcc@cjk

\greekasCJK ギリシャ・キリル文字を和文扱いにする。

\nogreekasCJK ギリシャ・キリル文字を欧文扱いにする。
4355 \newcommand*\greekasCJK{%
4356 \bxjx@gcc@cjktrue}
4357 \newcommand*\nogreekasCJK{%
4358 \bxjx@gcc@cjkfalse}

```

`\bx@fake@grk` `\bx@fake@grk{(出力文字)}{(基準文字)}` :

```

4359 \def\bxjx@do#1\relax{%
4360   \def\bxjx@fake@grk##1##2{%
4361     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4362   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4363     \ifx\\##1\\%
4364       \bxjx@cmta##4\divide\bxjx@cmta\@cclvi
4365       \multiply\bxjx@cmta\@cclvi \advance\bxjx@cmta`##3\relax
4366       \mathchar\bxjx@cmta
4367     \else ##3\fi}
4368 } \expandafter\bxjx@do\string\mathchar\relax

```

### ■pdfLaTeX・upLaTeX の場合

```

4369 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0

```

まず `inputenc` を読み込んで入力エンコーディングを `utf8` に変更する。

```

4370 \@ifpackageloaded{inputenc}{\fi}{%else
4371   \RequirePackage[utf8]{inputenc}}
4372 \def\bxjx@tmpa{utf8}
4373 \ifx\bxjx@tmpa\inputencdoingname
4374   \PackageWarningNoLine\bxjx@pkgname
4375     {Input encoding changed to utf8}%
4376   \inputencoding{utf8}%
4377 \fi

```

`upLaTeX` の場合は当該の文字を含むブロックをの和文カテゴリコードを変更する。

```

4378 \if u\bxjx@engine
4379 \kcatcode"0370=15
4380 \kcatcode"0400=15
4381 \kcatcode"0500=15
4382 \fi

```

各文字について `\DeclareUnicodeCharacter` を実行する。

```

4383 \def\do#1{%
4384   \@tempcnta="#1\relax
4385   \@tempcntb=\@tempcnta \divide\@tempcntb256
4386   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4387   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4388 \def\bxjx@do@a#1#2#3#4#5{%
4389   \ifx\\#5\\%
4390     \def\bxjx@tmpa{\@inmathwarn#4}%
4391   \else\ifcat A\noexpand#5%
4392     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4393       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4394   \else \def\bxjx@tmpa{#5}%
4395   \fi\fi
4396   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4397   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
4398 \if u\bxjx@engine

```

```

4399 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
4400 \def\bxjx@do@b#1#2#3#4#5{%
4401   \kchardef#1=\@tempcnta
4402   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4403   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4404 \else\if p\bxjx@engine
4405 \def\bxjx@do@b#1#2#3#4#5{%
4406   \mathchardef#1=\@tempcnta
4407   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4408   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4409 \fi\fi
4410 \bxjx@grkcyr@list
4411 \let\bxjx@do@a\undefined
4412 \let\bxjx@do@b\undefined

```

`\bxjx@DeclareUnicodeCharacter` `\bxjx@DeclareUnicodeCharacter` を改変して、ギリシャ・キリル文字の場合に再定義を抑制したもの。

```

4413 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4414 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4415 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4416 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4417   \count@=#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
4418   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
4419     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4420   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4421     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4422   \else
4423     \wlog{ \space\space skipped defining Unicode char U+#1}%
4424   \fi\fi}

```

`\bxjx@ja@or@not` `\bxjx@ja@or@not`

```

4425 \def\bxjx@ja@or@not#1#2#3{%
  \greekasCJK の場合は、無条件に和文用コードを実行する。

```

```

4426 \ifbxjx@gcc@CJK #1%

```

`\nogreekasCJK` の場合は、エンコーディングを固定して欧文用のコードを実行するが、そのエンコーディングが未定義の場合は（フォールバックとして）和文用コードを使う。

```

4427 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4428 \else \UseTextSymbol{#2}{#3}%
4429 \fi\fi}

```

`\DeclareFontEncoding@` `\DeclareFontEncoding@` にパッチを当てて、`\DeclareFontEncoding` の実行中だけ改変後の `\DeclareUnicodeCharacter` が使われるようにする。

```

4430 \begingroup
4431 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4432 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4433   \noexpand\bxjx@swap@DUC@cmd
4434   \the\toks@

```

```

4435 \noexpand\bxjx@swap@DUC@cmd}}
4436 \endgroup\next
4437 \def\bxjx@swap@DUC@cmd{%
4438 \let\bxjx@tmpa\DeclareUnicodeCharacter
4439 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4440 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

### ■ Xe<sub>La</sub>TeX・Lua<sub>La</sub>TeX の場合

```

4441 \else\ifnum0\if x\bxjx@engine1\fi\if 1\bxjx@engine1\fi>0

```

各文字について、math active を設定する。

```

4442 \def\do#1{%
4443 \bxjx@cmta="#1\relax
4444 \begingroup
4445 \lccode`~=\bxjx@cmta
4446 \lowercase{\endgroup
4447 \bxjx@do@a{~}}{#1}}
4448 \def\bxjx@do@a#1#2#3#4#5{%
4449 \ifx\#5\\\let\bxjx@tmpa\relax
4450 \else\ifcat A\noexpand#5%
4451 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4452 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4453 \else \def\bxjx@tmpa{#5}%
4454 \fi\fi
4455 \ifx\bxjx@tmpa\relax\else
4456 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4457 \fi}

```

「Unicode な数式」の設定が行われているかを（簡易的に）検査して、そうでない場合にのみ、こちらの設定を有効にする。

```

4458 \mathchardef\bxjx@tmpa="119
4459 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4460 \let\bxjx@do@a\undefined

```

Lua<sub>La</sub>TeX における \(\text{no}\)greekasCJK の定義。jacharrange の設定を変更する。

```

4461 \if 1\bxjx@engine
4462 \protected\def\greekasCJK{%
4463 \bxjx@gcc@cjktrue
4464 \ltjsetparameter{jacharrange={+2, +8}}
4465 \protected\def\nogreekasCJK{%
4466 \bxjx@gcc@cjkfalse
4467 \ltjsetparameter{jacharrange={-2, -8}}
4468 \fi

```

Xe<sub>La</sub>TeX における \(\text{no}\)greekasCJK の定義。

```

4469 \if x\bxjx@engine
4470 \protected\def\greekasCJK{%
4471 \bxjx@gcc@cjktrue

```



```

4472 \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4473 \bxjx@grkcyr@list}
4474 \protected\def\nogreekasCJK{%
4475 \bxjx@gcc@cjkbfalse
4476 \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4477 \bxjx@grkcyr@list}
4478 \fi

```

以上。

```

4479 \fi\fi

```

## H.4 初期設定

ギリシャ・キリル文字を欧文扱いにする。

```

4480 \nogreekasCJK

```

## H.5 完了

おしまい。

```

4481 %</cjkat>

```

補助パッケージ実装はここまで。

```

4482 %</anc>

```