

# 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”)

v1.2a [2016/08/16]

---

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、ユーザマニュアル `bxjscls-manual.pdf` を参照してください。

---

## 目次

1	はじめに	1
2	オプション	5
3	和文フォントの変更	29
4	フォントサイズ	29
5	レイアウト	35
5.1	ページレイアウト	36
6	ページスタイル	43
7	文書のマークアップ	46
7.1	表題	46
7.2	章・節	51
7.3	リスト環境	62
7.4	パラメータの設定	69
7.5	フロート	70
7.6	キャプション	71
8	フォントコマンド	73
9	相互参照	75

9.1	目次の類 . . . . .	75
9.2	参考文献 . . . . .	80
9.3	索引 . . . . .	82
9.4	脚注 . . . . .	83
10	段落の頭へのグルー挿入禁止 . . . . .	85
11	いろいろなロゴ . . . . .	88
12	amsmath との衝突の回避 . . . . .	88
13	初期設定 . . . . .	88
付録 A	和文ドライバの仕様 	92
付録 B	和文ドライバ: minimal 	93
B.1	補助マクロ . . . . .	93
B.2	(u)pTeX 用の設定 . . . . .	95
B.3	pdfTeX 用の処理 . . . . .	99
B.4	X <sub>Y</sub> TeX 用の処理 . . . . .	99
B.5	後処理 (エンジン共通) . . . . .	100
付録 C	和文ドライバ: standard 	102
C.1	共通処理 (1) . . . . .	102
C.2	pTeX 用設定 . . . . .	107
C.3	pdfTeX 用設定: CJK + bxcjkatype . . . . .	110
C.4	X <sub>Y</sub> TeX 用設定: xeCJK + zxjatype . . . . .	111
C.5	LuaTeX 用設定: LuaTeX-ja . . . . .	113
C.6	共通処理 (2) . . . . .	115
付録 D	和文ドライバ: modern 	115
D.1	フォント設定 . . . . .	115
D.2	fixltx2e 読込 . . . . .	116
D.3	和文カテゴリコード . . . . .	116
D.4	完了 . . . . .	116
付録 E	和文ドライバ: pandoc 	116
E.1	dupload システム . . . . .	116
E.2	lang 変数 . . . . .	117
E.3	geometry 変数 . . . . .	118
E.4	CJKmainfont 変数 . . . . .	118
E.5	fixltx2e パッケージ . . . . .	119
E.6	cmap パッケージ . . . . .	119

E.7	microtype パッケージ . . . . .	119
E.8	完了 . . . . .	119
付録 F	補助パッケージ一覧 罫	119
付録 G	補助パッケージ: bxjscompat 罫	119
G.1	準備 . . . . .	120
G.2	X <sub>q</sub> TeX 部分 . . . . .	120
G.3	LuaTeX 部分 . . . . .	121
G.4	完了 . . . . .	122
付録 H	補助パッケージ: bxjscjcat 罫	122
H.1	準備 . . . . .	122
H.2	和文カテゴリコードの設定 . . . . .	123
H.3	ギリシャ・キリル文字の扱い . . . . .	124
H.4	初期設定 . . . . .	130
H.5	完了 . . . . .	130

## 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ものである。

```

<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「pL<sup>A</sup>T<sub>E</sub>X 2<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 jclassses.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 upL<sup>A</sup>T<sub>E</sub>X 対応パッチを取り込みました。

以下では実際のコードに即して説明します。

### ■ BXJS クラス特有の設定 罫

---

```
1 %<*<cls>
```

2 %% このファイルは日本語文字を含みます。

長さ値の指定で式を利用可能にするため calc を読み込む。

```
3 \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。

```
4 \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。

```
5 \IfFileExists{bxjscompat.sty}{%
```

```
6 \let\jsAtEndOfClass@gobble
```

```
7 \RequirePackage{bxjscompat}%
```

```
8 }{}
```

`\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`。

```
9 \let\jsArticle=a
```

```
10 \let\jsBook=b
```

```
11 \let\jsReport=r
```

```
12 \let\jsSlide=s
```

```
13 %<article>\let\jsDocClass\jsArticle
```

```
14 %<article>\def\bxjs@clsname{bxjsarticle}
```

```
15 %<book>\let\jsDocClass\jsBook
```

```
16 %<book>\def\bxjs@clsname{bxjsbook}
```

```
17 %<report>\let\jsDocClass\jsReport
```

```
18 %<report>\def\bxjs@clsname{bxjsreport}
```

```
19 %<slide>\let\jsDocClass\jsSlide
```

```
20 %<slide>\def\bxjs@clsname{bxjsslide}
```

`\jsEngine` [暗黙文字トークン] エンジン ( $\text{T}_{\text{E}}\text{X}$  の種類) の種別:  $j = \text{pT}_{\text{E}}\text{X}$  系、 $x = \text{X}_{\text{g}}\text{T}_{\text{E}}\text{X}$ 、 $p = \text{pdfT}_{\text{E}}\text{X}$  (含 DVI モード)、 $l = \text{LuaT}_{\text{E}}\text{X}$ 、 $J = \text{NTT } j\text{T}_{\text{E}}\text{X}$ 、 $0 = \text{Omega}$  系、 $n =$  以上の何れでもない。

```
21 \let\jsEngine=n
```

```
22 \def\bxjs@test@engine#1#2{%
```

```
23 \edef\bxjs@tmpa{\string#1}%
```

```
24 \edef\bxjs@tmpb{\meaning#1}%
```

```
25 \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
```

```
26 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
```

```
27 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
```

```
28 \bxjs@test@engine\Omegaversion{\let\jsEngine=0}
```

```
29 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
```

```
30 \bxjs@test@engine\pdftexversion{\let\jsEngine=p}
```

```
31 \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)  $\text{upT}_{\text{E}}\text{X}$  であるか。

```
32 \newif\ifjsWithupTeX
```

```
33 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
```

```
34 \jsWithupTeXtrue
```

```

35 \fi\fi
36 \let\if@jsc@uplatex\ifjsWithupTeX

\ifjsWithpTeXng 〔スイッチ〕 エンジンが pTeX-ng であるか。
37 \newif\ifjsWithpTeXng
38 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}

\ifjsWitheTeX 〔スイッチ〕 エンジンが  $\epsilon$ -TeX 拡張をもつか。
39 \newif\ifjsWitheTeX
40 \bxjs@test@engine\TeXversion{\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。
41 \let\bxjs@tmpa\relax
42 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
43 \ifx O\jsEngine \def\bxjs@tmpa{Omega}\fi
44 \ifx\bxjs@tmpa\relax \expandafter\@gobble
45 \else
46 \ClassError\@currname
47 {The engine in use (\bxjs@tmpa) is not supported}
48 {It's a fatal error. I'll quit right now.}
49 \expandafter\@firstofone
50 \fi{\endinput\@@end}

\bxjs@protected  $\epsilon$ -TeX 拡張が有効な場合にのみ \protected の効果をもつ。
51 \ifjsWitheTeX \let\bxjs@protected\protected
52 \else \let\bxjs@protected\@empty
53 \fi

\bxjs@robust@def 無引数の頑強な命令を定義する。
54 \ifjsWitheTeX
55 \def\bxjs@robust@def{\protected\def}
56 \else
57 \def\bxjs@robust@def{\DeclareRobustCommand*}
58 \fi

\ifjsInPdfMode 〔スイッチ〕 pdfTeX / LuaTeX が PDF モードで動作しているか。
※ LuaTeX 0.8x 版でのプリミティブ名変更に対応。
59 \newif\ifjsInPdfMode
60 \@nameuse{ImposeOldLuaTeXBehavior}
61 \let\bxjs@tmpa\PackageWarningNoLine
62 \let\PackageWarningNoLine\PackageInfo % suppress warning
63 \RequirePackage{ifpdf}
64 \let\PackageWarningNoLine\bxjs@tmpa
65 \@nameuse{RevokeOldLuaTeXBehavior}
66 \let\ifjsInPdfMode\ifpdf

\bxjs@cond \bxjs@cond\ifXXX……\fi{(真)}{(偽)}

```

TeX 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。

```
67 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
68   #1\expandafter\@firstoftwo
69   \else\expandafter\@secondoftwo
70   \fi}
```

`\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。)

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

LuaTeX の場合、原版のコード中のコントロールワード中に現れる日本語文字のカテゴリコードを一時的に 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 LuaTeX 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。

```
73 \@onlypreamble\bxjs@restore@jltrcc
74 \let\bxjs@restore@jltrcc\@empty
75 \if 1\jsEngine
76 \def\bxjs@change@jltrcc#1{%
77   \xdef\bxjs@restore@jltrcc{%
78     \bxjs@restore@jltrcc
79     \catcode`#1=\the\catcode`#1\relax}%
80   \catcode`#1=11\relax}
81 \@tfor\bxjs@x:=西曆\do
82   {\expandafter\bxjs@change@jltrcc\bxjs@x}
83 \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。

```
84 \bxjs@robust@def\jsInhibitGlue{%
85   \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。

```
86 \if@compatibility
87   \ClassError\bxjs@clsname
88   {Something went chaotic!\MessageBreak
89   I cannot go a single step further...}
90   {If the chant of \string\documentstyle\space was just a blunder
91   of yours,\MessageBreak then there'll still be hope....}
92   \expandafter\@firstofone
93 \else \expandafter\@gobble
94 \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。
95 \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。
96 \newif\if@titlepage

\if@openright \chapter, \part を奇数ページ起こしにするかどうかです。書籍では真が標準です。
97 %<book|report>\newif\if@openright

\if@mainmatter 真なら本文，偽なら前付け・後付けです。偽なら \chapter で章番号が出ません。
98 %<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。
99 \newif\if@enablejfam \@enablejfamtrue
```

以下で各オプションを宣言します。

■用紙サイズ JIS や ISO の A0 判は面積  $1\text{m}^2$ ，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， $\text{\LaTeX} 2_{\epsilon}$  の `b5paper` は  $250\text{mm} \times 176\text{mm}$  ですが， $\text{p}\text{\LaTeX} 2_{\epsilon}$  の `b5paper` は  $257\text{mm} \times 182\text{mm}$  になっています。ここでは  $\text{p}\text{\LaTeX} 2_{\epsilon}$  にならって JIS に従いました。

デフォルトは `a4paper` です。

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

---

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

```
100 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
101 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
102 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
103 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
104 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
105 \DeclareOption{b4paper}{\bxjs@setpaper{{257truemm}{364truemm}}}
106 \DeclareOption{b5paper}{\bxjs@setpaper{{182truemm}{257truemm}}}
```

```

107 \DeclareOption{b6paper}{\bxjs@setpaper{{128truemm}{182truemm}}}
108 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
109 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
110 \DeclareOption{b4j}{\bxjs@setpaper{{257truemm}{364truemm}}}
111 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
112 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
113 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
114 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
115 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
116 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

---

■横置き 用紙の縦と横の長さを入れ換えます。

```

117 \newif\if@landscape
118 \@landscapefalse
119 \DeclareOption{landscape}{\@landscapetrue}

```

■slide オプション slide を新設しました。

```

120 \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。

```

121 %<!slide>\@slidefalse
122 %<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。

---

`\@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。

```

123 \newcommand{\@ptsize}{0}
124 \def\bxjs@param@basefontsize{10pt}
125 \def\jsUnusualPtSize{-20}

```

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

```
126 \def\bxjs@setbasefontsize#1{%
127   \bxjs@setbasefontlength\@tempdima{#1}%
128   \edef\bxjs@param@basefontsize{\the\@tempdima}%
129   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
130   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
131   \else\ifdim\@tempdima=12pt  \long\def\@ptsize{2}%
132   \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。

```
133 \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。

```
134   \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。

```
135   \ifx j\jsEngine \setlength#1{#2}%
136   \else
137     \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
138     \ifx\bxjs@tmpa\relax \setlength#1{#2}%
139     \else \tempdimc0.25mm #1=\bxjs@tmpa\@tempdimc
140     \fi
141   \fi}
142 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
143   \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
144   \else \def\bxjs@tmpa{#1}%
145   \fi}
146 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
147   \ifx\@nnil#2\@nnil\else
148     \ClassWarningNoLine\bxjs@clsname
149     {You should not use 'true' lengths here}%
150   \fi}
```

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

```
151 \newif\ifjsc@mag
152 %\let\jsc@magscale\undefined
153 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
154 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
155 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
156 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
157 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
158 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
159 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
160 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
```

```

161 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
162 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
163 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
164 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
165 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
166 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
167 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
168 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
169 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
170 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
171 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

172 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
173 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
174 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}

```

## ■ トンボオプション

欧文 L<sup>A</sup>T<sub>E</sub>X のカーネルではサポートされないため削除。  
ただし JS クラスとの互換のため以下の変数を用意する。

```
175 \newdimen\stockwidth \newdimen\stockheight
```

## ■ 面付け

欧文 L<sup>A</sup>T<sub>E</sub>X のカーネルではサポートされないため削除。

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

```

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

```

■ 二段組 `twocolumn` で二段組になります。

```

179 \DeclareOption{onecolumn}{\@twocolumnfalse}
180 \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。

```

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

```

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

```
183 %<book|report>\DeclareOption{openright}{\@openrighttrue}
184 %<book|report>\DeclareOption{openany}{\@openrightfalse}
```

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

eqnarray L<sup>A</sup>T<sub>E</sub>X 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。

```
185 \def\eqnarray{%
186   \stepcounter{equation}%
187   \def\@currentlabel{\p@equation\theequation}%
188   \global\@eqnswtrue
189   \m@th
190   \global\@eqcnt\z@
191   \tabskip\@centering
192   \let\\\@eqnocr
193   $$\everycr{}\halign to\displaywidth\bgroup
194     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
195     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
196     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
197     &\global\@eqcnt\thr@@ \hbxt@\z@\bgroup\hss##\egroup
198     \tabskip\z@skip
199   \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 force されます。森本さんにしたがって訂正しました。

```
200 \DeclareOption{leqno}{\input{leqno.clo}}
201 \DeclareOption{fleqn}{\input{fleqn.clo}}
202 % fleqn 用の eqnarray 環境の再定義
203 \def\eqnarray{%
204   \stepcounter{equation}%
205   \def\@currentlabel{\p@equation\theequation}%
206   \global\@eqnswtrue\m@th
207   \global\@eqcnt\z@
208   \tabskip\mathindent
209   \let\=\@eqnocr
210   \setlength\abovedisplayskip{\topsep}%
211   \ifvmode
212     \addtolength\abovedisplayskip{\partopsep}%
213   \fi
214   \addtolength\abovedisplayskip{\parskip}%
215   \setlength\belowdisplayskip{\abovedisplayskip}%
216   \setlength\belowdisplayshortskip{\abovedisplayskip}%
217   \setlength\abovedisplayshortskip{\abovedisplayskip}%
218   $$\everycr{}\halign to\linewidth% $$
219   \bgroup
220     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
221     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
222     &\global\@eqcnt\tw@
```

```

223     $\displaystyle{##}$\hfil \tabskip\@centering
224     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
225     \tabskip\z@skip\cr
226     }}

```

■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。

```

227 % \DeclareOption{openbib}{%
228 %   \AtEndOfPackage{%
229 %     \renewcommand\@openbib@code{%
230 %       \advance\leftmargin\bibindent
231 %       \itemindent -\bibindent
232 %       \listparindent \itemindent
233 %       \parsep \z@}%
234 %     \renewcommand\newblock{\par}}}

```

■数式フォントとして和文フォントを登録しないオプション

---

ここは和文ドライバの管轄。

---

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

---

`\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` は 2016/07/13 版で廃止された。よって `\ifdraft` は 2.0 版で廃止を予定する。

```

235 \let\ifjsDraft\iffalse
236 \@onlypreamble\bxjs@draft
237 \def\bxjs@draft#1{%
238   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
239 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
240 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
241 \AtBeginDocument{%
242   \expandafter\ifx\csname ifdraft\endcsname\relax
243     \expandafter\let\csname ifdraft\expandafter\endcsname
244     \csname ifjsDraft\endcsname
245   \fi}

```

■和文フォントメトリックの選択

---

ここは和文ドライバの管轄。

---

## ■papersize スペシャルの利用

---

geometry パッケージが行う。

`\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。

```
246 \newif\ifbxjs@papersize
247 \bxjs@papersizetrue
248 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
249 \DeclareOption{papersize}{\bxjs@papersizetrue}
```

---

## ■英語化 オプション `english` を新設しました。

```
250 \newif\if@english
251 \@englishfalse
252 \DeclareOption{english}{\@englishtrue}
```

## ■jsreport 相当 オプション `report` を新設しました。

---

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

---

## ■BXJS 特有のオプションの一覧

---

- 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` :  $X_{\text{q}}\text{T}_{\text{E}}\text{X}$  の “generateactualtext” を有効 / 無効にする。
- `simplejasetup / nosimplejasetup` :  $X_{\text{q}}\text{T}_{\text{E}}\text{X}$  の “linebreaklocale” を有効 / 無効にする。
- `bigcode / nobigcode` :  $\text{upT}_{\text{E}}\text{X}$  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=\langle dimen \rangle` : 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 10pt である。)

- `jbaser=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` の上限の全角単位の値。
- `paragraph-mark=char` : パラグラフのマーク。
- `whole-zw-lines` : (予定)
- `dummy-microtype` : (予定)
- `hyperref-enc` : (予定)

`\bxjs@invscale` `\bxjs@invscale` は  $\TeX$  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、これだと数詞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。

```

253 \mathchardef\bxjs@csta=259
254 \def\bxjs@invscale#1#2{%
255   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
256     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
257     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
258     \@tempcntb\p@ \divide\@tempcntb\@tempdimb
259     \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
260     \@tempdimb\@tempcnta\@ne
261     \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
262     \advance\@tempcnta\bxjs@csta \@tempdimc\@tempcnta\@ne
263     \@whiledim\@tempdimb<\@tempdimc\do{%
264       \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
265       \advance\@tempcntb\@ne \divide\@tempcntb\tw@
266       \ifdim #2\@tempcntb>\@tempdima
267         \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne

```

```

268     \else \@tempdimb=\@tempcntb\@ne \fi}%
269     \xdef\bxjs@gtmpa{\the\@tempdimb}%
270     \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性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。

```
271 \let\bxjs@composite@proc\relax
```

`pandoc` オプションは、Pandoc で  $\text{\LaTeX}$  用の既定テンプレートを用いて他形式から  $\text{\LaTeX}$  (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。

```
272 \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。

```

273   \def\bxjs@composite@proc{%
274     \bxjs@oldfontcommandstrue
275     \setkeys{bxjs}{ja=pandoc}%
276     \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…。

```

277   \def\bxjs@driver@opt{dvipdfmx}%
278   \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。

```
279 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
280 %\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  $\text{\LaTeX}$  の慣習に従って「ドライバの指定」とみなすべきだから。

```

281 \DeclareOption{autodetect-engine}{%
282   \let\bxjs@engine@given=*}

```

```

283 \DeclareOption{latex}{%
284   \def\bxjs@engine@opt{latex}%
285   \let\bxjs@engine@given=n}
286 \DeclareOption{platex}{%
287   \def\bxjs@engine@opt{platex}%
288   \let\bxjs@engine@given=j}
289 \DeclareOption{uplatex}{%
290   \def\bxjs@engine@opt{uplatex}%
291   \let\bxjs@engine@given=u}
292 \DeclareOption{xelatex}{%
293   \def\bxjs@engine@opt{xelatex}%
294   \let\bxjs@engine@given=x}
295 \DeclareOption{pdflatex}{%
296   \def\bxjs@engine@opt{pdflatex}%
297   \let\bxjs@engine@given=p}
298 \DeclareOption{lualatex}{%
299   \def\bxjs@engine@opt{lualatex}%
300   \let\bxjs@engine@given=l}
301 \DeclareOption{platex-ng}{%
302   \def\bxjs@engine@opt{platex-ng}%
303   \let\bxjs@engine@given=g}
304 \DeclareOption{platex-ng*}{%
305   \def\bxjs@engine@opt{platex-ng*}%
306   \let\bxjs@platexng@nodrv=t%
307   \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。

```

308 %\let\bxjs@driver@given\undefined
309 \let\bxjs@driver@@dvimode=0
310 \let\bxjs@driver@@dvipdfmx=1
311 \let\bxjs@driver@@pdfmode=2
312 \let\bxjs@driver@@xetex=3

```

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

```

313 %\let\bxjs@driver@opt\undefined
314 \DeclareOption{dvips}{%
315   \def\bxjs@driver@opt{dvips}%
316   \let\bxjs@driver@given\bxjs@driver@@dvimode}
317 \DeclareOption{dviout}{%
318   \def\bxjs@driver@opt{dviout}%
319   \let\bxjs@driver@given\bxjs@driver@@dvimode}
320 \DeclareOption{xdvi}{%
321   \def\bxjs@driver@opt{xdvi}%
322   \let\bxjs@driver@given\bxjs@driver@@dvimode}
323 \DeclareOption{dvipdfmx}{%
324   \def\bxjs@driver@opt{dvipdfmx}%
325   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
326 \DeclareOption{pdftex}{%

```

```

327 \def\bxjs@driver@opt{pdftex}%
328 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
329 \DeclareOption{luatex}{%
330 \def\bxjs@driver@opt{luatex}%
331 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
332 \DeclareOption{xetex}{%
333 \def\bxjs@driver@opt{xetex}%
334 \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。

```

335 \DeclareOption{dvipdfmx-if-dvi}{%
336 \setkeys{bxjs}{dvi=dvipdfmx}}

```

---

### ■その他の BXJS 独自オプション 密

---

`\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。

```
337 \newif\ifbxjs@usezw \bxjs@usezwtrue
```

`zw`、`noz` オプションの定義。

```

338 \DeclareOption{noz}{%
339 \bxjs@usezwfalse}
340 \DeclareOption{zw}{%
341 \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。

```
342 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
```

`nojs`、`js` オプションの定義。

```

343 \DeclareOption{nojs}{%
344 \bxjs@disguise@jsfalse}
345 \DeclareOption{js}{%
346 \bxjs@disguise@jstrue}

```

`\ifbxjs@precisetext` X<sub>Y</sub>TEX の “generateactualtext” を有効にするか。既定は偽。

```
347 \newif\ifbxjs@precisetext
```

`noprecisetext` / `precisetext` オプションの定義。

```

348 \DeclareOption{noprecisetext}{%
349 \bxjs@precisetextfalse}
350 \DeclareOption{precisetext}{%
351 \bxjs@precisetexttrue}

```

`\ifbxjs@simplejasetup` X<sub>Y</sub>TEX 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）。

```
352 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
```

nosimplejasetup / simplejasetup オプションの定義。

```
353 \DeclareOption{nosimplejasetup}{%
354   \bxjs@simplejasetupfalse}
355 \DeclareOption{simplejasetup}{%
356   \bxjs@simplejasetuptrue}
```

`\ifbxjs@bigcode` up $\TeX$  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。

```
357 \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。

```
358 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
359 \def\bxjs@tmpb#1 #2#3\@nil{%
360   \ifx1#2\bxjs@bigcodetrue \fi}
361 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil
```

nobigcode / bigcode オプションの定義。

```
362 \DeclareOption{nobigcode}{%
363   \bxjs@bigcodefalse}
364 \DeclareOption{bigcode}{%
365   \bxjs@bigcodetrue}
```

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

```
366 \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。

```
367 \DeclareOption{nooldfontcommands}{%
368   \bxjs@oldfontcommandsfalse}
369 \DeclareOption{oldfontcommands}{%
370   \bxjs@oldfontcommandstrue}
```

---

## ■ keyval 型のオプション

---

```
371 \def\bxjs@setkey{%
372   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
373 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
374 \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`。

```
375 \def\bxjs@safe@setkeys#1#2{%
376   \let\bxjs@KV@errx\KV@errx
```

```

377 \let\KV@errx\bxjs@safe@setkeys@a
378 \setkeys{#1}{#2}%
379 \let\KV@errx\bxjs@KV@errx}
380 \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。
381 \def\bxjs@set@keyval#1#2#3{%
382 \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
383 \ifx\bxjs@next\relax
384 \bxjs@error@keyval{#1}{#2}%
385 #3%
386 \else \bxjs@next
387 \fi}
388 \@onlypreamble\bxjs@error@keyval
389 \def\bxjs@error@keyval#1#2{%
390 \ClassError\bxjs@clsname
391 {Invalid value '#2' for option #1}\@ehc}

\ifbxjs@scaleset 和文スケール値が指定されたか。
392 \newif\ifbxjs@scaleset

\jsScale [実数値マクロ] 和文スケール値。
393 \def\jsScale{0.924715}

    base オプションの処理。
394 \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。
395 \@onlypreamble\bxjs@do@opt@jbase
396 \let\bxjs@do@opt@jbase\relax
397 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
398 \def\bxjs@setjbasefontsize#1{%
399 \def\bxjs@do@opt@jbase{%
400 \bxjs@setbasefontlength\@tempdima{#1}%
401 \bxjs@invscale\@tempdima\jsScale
402 \bxjs@setbasefontsize{\@tempdima}}}

    scale オプションの処理。
403 \define@key{bxjs}{scale}{%
404 \bxjs@scalesettrue
405 \edef\jsScale{#1}}

    noscale オプションの処理。
406 \DeclareOption{noscale}{%
407 \bxjs@scalesettrue
408 \def\jsScale{1}}

    CJK オプションは廃止された。

```

※ 1.3 版で宣言そのものを削除する。

```
409 \DeclareOption{CJK}{%
410   \ClassWarning{bxjs@clsname
411     {Option `CJK' was abolished}}
```

`\bxjs@param@mag` mag オプションの値。

```
412 \let\bxjs@param@mag\relax

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

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

`\bxjs@jadriver` 和文ドライバの名前。

```
415 \let\bxjs@jadriver\relax
416 %\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)。
417 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
418 \define@key{bxjs}{ja}[\relax]{%
419   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}
```

`\jsJaFont` 和文フォント設定の名前。

```
420 \let\jsJaFont\@empty

jafont オプションの処理。
421 \define@key{bxjs}{jafont}{\edef\jsJaFont{#1}}
```

`\jsJaParam` 和文ドライバパラメタの文字列。

```
422 \let\jsJaParam\@empty

japaram オプションの処理。
423 \define@key{bxjs}{japaram}{\edef\jsJaParam{#1}}
```

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

```
424 \let\bxjs@magstyle@mag=m
425 \let\bxjs@magstyle@real=r
426 \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。

```
427 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
428 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
429 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

`\bxjs@magstyle@default` は既定の値を表す。

```
430 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
431 \ifx l\jsEngine \ifnum\luatexversion>86
432 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
433 \fi\fi
434 \ifjsWithpTeXng
435 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
436 \fi
437 \let\bxjs@magstyle\bxjs@magstyle@default
```

`magstyle` オプションの処理。

```
438 \define@key{bxjs}{magstyle}{%
439 \expandafter\let\expandafter\bxjs@magstyle\csname
440 bxjs@magstyle@#1\endcsname
441 \ifx\bxjs@magstyle\relax
442 \ClassError\bxjs@clsname
443 {Invalid magstyle value `#1'}\@ehc
444 \let\bxjs@magstyle\bxjs@magstyle@default
445 \fi}
```

`\bxjs@geometry geometry` オプションの値。

```
446 \let\bxjs@geometry@class=c
447 \let\bxjs@geometry@user=u
448 \let\bxjs@geometry\bxjs@geometry@class
```

`geometry` オプションの処理。

```
449 \define@key{bxjs}{geometry}{%
450 \expandafter\let\expandafter\bxjs@geometry\csname
451 bxjs@geometry@#1\endcsname
452 \ifx\bxjs@geometry\relax
453 \ClassError\bxjs@clsname
454 {Invalid geometry value `#1'}\@ehc
455 \let\bxjs@geometry\bxjs@geometry@class
456 \fi}
```

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

```
457 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue
```

`fancyhdr` オプションの処理。

```
458 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
459 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
460 \define@key{bxjs}{fancyhdr}{%
461 \bxjs@set@keyval{fancyhdr}{#1}{}}
```

`\ifbxjs@dvi@opt` `dvi` オプションが指定されたか。

```
462 \newif\ifbxjs@dvi@opt
```

DVI モードのドライバとドライバ種別との対応。

```
463 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
```

```

464 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvimode
465 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
466 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode

```

dvi オプションの処理。

```

467 \define@key{bxjs}{dvi}{%
468   \expandafter\let\expandafter\bxjs@tmpa\csname
469     bxjs@dvidriver@@#1\endcsname
470   \ifx\bxjs@tmpa\relax
471     \ClassError\bxjs@clsname
472     {Invalid dvi value `#1'}\@ehc
473   \else

```

\bxjs@driver@given を未定義にしていることに注意。

```

474     \def\bxjs@driver@opt{#1}%
475     \let\bxjs@driver@given\undefined
476     \bxjs@dvi@opttrue
477   \fi}

```

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

※既定はアレだが 1.3 版で非アレになる予定。

```

478 \newif\ifbxjs@layout@buggyhmargin \bxjs@layout@buggyhmargintrue

```

layout オプションの処理。

```

479 \@namedef{bxjs@kv@layout@v1}{%
480   \bxjs@layout@buggyhmargintrue}
481 \@namedef{bxjs@kv@layout@v2}{%
482   \bxjs@layout@buggyhmarginfalse}
483 \define@key{bxjs}{layout}{%
484   \bxjs@set@keyval{layout}{#1}{}}

```

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

```

485 %\let\bxjs@textwidth@limit\undefined
486 \define@key{bxjs}{textwidth-limit}{%
487   \edef\bxjs@textwidth@limit{#1}}

```

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

```

488 %\let\bxjs@paragraph@mark\undefined
489 \define@key{bxjs}{paragraph-mark}{%
490   \edef\bxjs@paragraph@mark{#1}}

```

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

```

491 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
492 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
493 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
494 \define@key{bxjs}{whole-zw-lines}{\bxjs@set@keyval{wholezwlines}{#1}{}}

```

\ifbxjs@dummy@microtype [スイッチ] dummy-microtype の指定値。

```

495 \newif\ifbxjs@dummy@microtype \bxjs@dummy@microtypetrue

```

```

496 \let\bxjs@kv@dummymicrotype@true\bxjs@dummy@microtypetrue
497 \let\bxjs@kv@dummymicrotype@false\bxjs@dummy@microtypefalse
498 \define@key{bxjs}{dummy-microtype}{\bxjs@set@keyval{dummymicrotype}{#1}{}}

```

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

```

499 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
500 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
501 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
502 \define@key{bxjs}{hyperref-enc}{\bxjs@set@keyval{hyperrefenc}{#1}{}}

```

## ■ オプションの実行

LaTeX 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。

```

503 \let\bxjs@ltx@removeelement\@removeelement
504 \def\@removeelement#1#2#3{%
505   \def\reserved@a{#2}%
506   \ifx\reserved@a\@empty \let#3\@empty
507   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
508   \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。

```

509 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
510 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
511 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
512 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
513 \ProcessOptions\relax
514 \bxjs@composite@proc

```

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。

```

515 \@onlypreamble\bxjs@purge@brace@elts
516 \def\bxjs@purge@brace@elts{%
517   \def\bxjs@tmpa{\@gobble}%
518   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
519   \let\@classoptionslist\bxjs@tmpa}
520 \@onlypreamble\bxjs@purge@be@a

```

```

521 \def\bxjs@purge@be@a#1,{%
522   \ifx\@nil#1\relax\else
523     \bxjs@purge@be@b#1{\@nil
524     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
525     \expandafter\bxjs@purge@be@a
526   \fi}
527 \@onlypreamble\bxjs@purge@be@b
528 \def\bxjs@purge@be@b#1#\bxjs@purge@be@c}
529 \@onlypreamble\bxjs@purge@be@c
530 \def\bxjs@purge@be@c#1\@nil{%
531   \ifx\@nil#1\@nil \@tempwatrue \else \@tempwafalse \fi}
532 \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。

```

533 \@expandtwoargs\@removeelement
534 {papersize}\@classoptionslist\@classoptionslist
535 \@expandtwoargs\@removeelement
536 {10pt}\@classoptionslist\@classoptionslist
537 \@expandtwoargs\@removeelement
538 {noscale}\@classoptionslist\@classoptionslist

```

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。

```

539 \ifx\bxjs@jadriver\relax\else
540   \let\bxjs@jadriver@given\bxjs@jadriver
541 \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。

```

542 \let\bxjs@tmpb\jsEngine
543 \ifx j\bxjs@tmpb\ifjsWithpTeXng
544   \let\bxjs@tmpb=g
545 \fi\fi
546 \ifx j\bxjs@tmpb\ifjsWithupTeX
547   \let\bxjs@tmpb=u
548 \fi\fi
549 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
550   \let\bxjs@tmpb=n
551 \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。）

```

552 \ifx *\bxjs@engine@given
553   \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。

```

554 \ifx j\bxjs@engine@given
555   \g@addto@macro\@classoptionslist{,platex}
556 \else\ifx u\bxjs@engine@given

```

```

557 \g@addto@macro\@classoptionslist{,uplatex}
558 \fi\fi
559 \fi
560 \ifx\bxjs@engine@given\@undefined\else
561 \ifx\bxjs@engine@given\bxjs@tmpb\else
562 \ClassError\bxjs@clsname%
563 {Option `\

```

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。

```

566 \ifjsWithpTeXng
567 \g@addto@macro\@classoptionslist{,uplatex}
568 \fi

```

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。

```

569 \@tempswatru
570 \ifx \bxjs@driver@given\@undefined\else
571 \ifjsInPdfMode
572 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
573 \@tempswafalse
574 \fi
575 \else\ifx x\jsEngine
576 \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
577 \@tempswafalse
578 \fi
579 \else
580 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
581 \@tempswafalse
582 \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
583 \@tempswafalse
584 \fi\fi
585 \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
586 \@tempswafalse
587 \fi\fi
588 \fi\fi
589 \fi
590 \if@tempswa\else
591 \ClassError\bxjs@clsname%
592 {Option `\

```

DVI 出力のエンジンである場合の追加処理。

```

594 \ifjsInPdfMode \@tempswafalse
595 \else\ifx x\jsEngine \@tempswafalse
596 \else\ifjsWithpTeXng \@tempswafalse
597 \else \@tempswatru
598 \fi\fi\fi
599 \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)。

```
600 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
601   \ClassWarningNoLine\bxjs@clsname
602   {No driver option is given}
603 \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。)

```
604 \ifbxjs@dvi@opt
605   \edef\bxjs@nxt{%
606     \let\noexpand\bxjs@driver@given
607     \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
608     \noexpand\g@addto@macro\noexpand\@classoptionslist
609     {,\bxjs@driver@opt}%
610   }\bxjs@nxt
611 \fi
612 \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。

```
613 \ifjsWithpTeXng
614   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
615     \let\bxjs@platexng@nodrv\@undefined
616   \else\ifx t\bxjs@platexng@nodrv\else
617     \g@addto@macro\@classoptionslist{,dvipdfmx}
618   \fi\fi
619 \fi
```

\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。

```
620 \def\bxjs@@minimal{minimal}
621 \ifx\bxjs@jadriver\relax
622   \ifx j\jsEngine
623     \def\bxjs@jadriver{standard}
624   \else
625     \let\bxjs@jadriver\bxjs@@minimal
626   \fi
627 \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。

```
628 \ifx\bxjs@jadriver@given\@undefined\else
629   \ifx\bxjs@engine@given\@undefined
630     \ClassError\bxjs@clsname%
631     {An engine option must be explicitly given}%
632     {When you employ a Japanese-driver you must specify a
633     correct\MessageBreak engine option.\MessageBreak\@ehc}
634 \fi\fi
```

新しい Lua<sub>T</sub><sub>E</sub>X (0.87 版以降) では mag がアレなので、magstyle=mag 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 xreal であり、エラーの場合は既定値に置き換えられる。)

```
635 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
636   \ifx\bxjs@magstyle\bxjs@magstyle@mag
637     \let\bxjs@magstyle\bxjs@magstyle@default
638     \ClassError\bxjs@clsname%
639     {The engine does not support 'magstyle=mag'}%
640     {LuaTeX v0.87 or later no longer supports "mag" feature.\MessageBreak
641     The default value is used instead.\MessageBreak \@ehc}
642   \fi
643 \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。

```
644 \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。

```
645 \let\Cjascale\jsScale
```

---

後処理

```
646 \if@slide
647   \def\maybeblue{\@ifundefined{ver@color.sty}{{\color{blue}}}
648 \fi
649 \if@landscape
650   \setlength\@tempdima {\paperheight}
651   \setlength\paperheight{\paperwidth}
652   \setlength\paperwidth {\@tempdima}
653 \fi
```

---

8bit 欧文 T<sub>E</sub>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。)

```
654 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
655   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
656     \catcode\@tempcnta\active
657     \advance\@tempcnta\@ne
658   \repeat
659 \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。

```
660 \ifbxjs@disguise@js
661 %<book|report>\def\bxjs@js@clsname{jsbook}
662 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
663 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
664 \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。

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

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

```
666 \ifbxjs@oldfontcommands
667 \AtEndOfClass{\allowoldfontcommands}
668 \fi
```

---

## ■papersize スペシャルの出力

---

`geometry` パッケージが行う。

---

## ■基準となる行送り

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

```
669 %<slide>\def\n@baseline{13}%
670 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
671 %<!slide>\else \def\n@baseline{16}\fi
```

## ■拡大率の設定

---

`\ifbxjs@nfss@patch` NFSS パッチするか。

```
672 \newif\ifbxjs@nfss@patch
673 \ifx\bxjs@magstyle\bxjs@magstyle@mag
674 \jsc@magtrue
675 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
676 \bxjs@nfss@patchtrue
677 \fi\fi
```

---

サイズの変更は T<sub>E</sub>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も若干縮めました。サイズについては全面的に見直しました。

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

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

---

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

```

678 \ifx\bxjs@param@mag\relax
679   \@tempdima=\bxjs@param@basefontsize
680   \advance\@tempdima.001pt \multiply\@tempdima25
681   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
682   \edef\bxjs@param@mag{\the\@tempcnta}
683 \else
684   \let\c@bxjs@cnta\@tempcnta
685   \setcounter{bxjs@cnta}{\bxjs@param@mag}
686   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
687 % 有効な mag 値の範囲は 1--32768
688   \edef\bxjs@param@mag{\the\@tempcnta}
689   \advance\@tempcnta100000
690   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
691   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
692   \edef\bxjs@param@basefontsize{\the\@tempdima}
693 \fi
694 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
695 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
696 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
697 \edef\jsc@magscale{\strip@pt\@tempdima}
698 \let\jsBaseFontSize\bxjs@param@basefontsize
699 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
700 %\typeout{\string\jsEngine: \meaning\jsEngine}
701 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
702 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
703 %\typeout{\string\jsc@magscale: \jsc@magscale}
704 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
705 %\typeout{\string\ifbxjs@nfss@patch: \meaning\ifbxjs@nfss@patch}

```

---

[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。

```

706 \ifjsc@mag
707   \let\jsc@mpt\p@
708   \newdimen\jsc@mmm \jsc@mmm=1mm
709   \catcode\?=9 % \p@? read as \p@
710 \else
711   \newdimen\jsc@mpt \jsc@mpt=\jsc@magscale \p@
712   \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
713   \catcode\?=11 \let\p@?\jsc@mpt
714 \fi
715 \chardef\bxjs@qmcc=\catcode\?\relax

```

ここで p<sub>T</sub>E<sub>X</sub> 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<sub>w</sub> 非指定時は \zw を \jsZw と同義にする。

```
716 \newdimen\jsZw
717 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
718 \ifbxjs@usezw
719 \providecommand*\zw{\jsZw}
720 \fi
```

---

## ■NFSS パッチ

---

ムニヤムニヤムニヤ……。

```
721 \ifbxjs@nfss@patch
722 \RequirePackage{type1cm}
723 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
724 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
725 \let\bxjs@get@external@font\get@external@font
726 \def\get@external@font{%
727   \bxjs@preadjust@extract@font
728   \bxjs@get@external@font}
729 \def\bxjs@fstrunc#1{%
730   \edef\bxjs@tmpa{\strip@pt#1}%
731   \expandafter\bxjs@fstrunc@a\bxjs@tmpa.***\@nil}
732 \def\bxjs@fstrunc@a#1.#2#3#4#5#6\@nil{%
733   \if#5*\else
734     \edef\bxjs@tmpa{#1%
735       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
736   \fi}
737 \def\bxjs@preadjust@extract@font{%
738   \let\bxjs@req@size\font@size
739   \dimen@f@size\p@ \bxjs@invscale\dimen@\jsc@magscale
740   \advance\dimen@.005pt\relax \bxjs@fstrunc\dimen@
741   \let\bxjs@ref@size\bxjs@tmpa
742   \let\font@size\bxjs@ref@size}
743 \def\execute@size@function#1{%
744   \let\bxjs@cref@size\font@size
745   \let\font@size\bxjs@req@size
746   \csname s@fct@#1\endcsname}
747 \let\bxjs@DeclareErrorFont\DeclareErrorFont
748 \def\DeclareErrorFont#1#2#3#4#5{%
749   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
750   \edef\bxjs@tmpa{#{1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
751   \expandafter\bxjs@DeclareErrorFont\bxjs@tmpa}
752 \def\gen@sfcnt{%
753   \edef\mandatory@arg{\mandatory@arg\bxjs@cref@size}%
754   \empty@sfcnt}
```

```

755 \def\genb@sfcnt{%
756   \edef\mandatory@arg{%
757     \mandatory@arg\expandafter\genb@x\bxjs@ceref@size..\@@}%
758   \empty@sfcnt}
759 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
760 \fi

```

---

### ■pagesize スペシャルの出力

---

削除。

---

## 3 和文フォントの変更

---

和文フォントの設定は和文ドライバの管轄。

---

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

```

761 \chardef\bxjs@periodchar=`\
762 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcode\bxjs@periodchar}
763 \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>\@viipt</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  $\text{pLATEX}_2\epsilon$  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。

---

`\set@fontsize` `\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。

```
764 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
765 \expandafter\bxjs@tmpa\expandafter{%
766   \set@fontsize{#1}{#2}{#3}%
767 % 末尾にコードを追加
768   \expandafter\def\expandafter\size@update\expandafter{%
769     \size@update
770     \jsFontSizeChanged}%
771 }
```

`\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。

```
772 \newcommand*\jsFontSizeChanged{%
773   \jsZw=\f@size\p@
774   \jsZw=\jsScale \jsZw
775   \ifdim\parindent>\z@
776     \if@english \parindent=1em
777     \else      \parindent=1\jsZw
778   \fi
779   \fi\relax
780   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
781 \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。

```
782 \ifjsc@mag
783   \let\jsc@setfontsize\@setfontsize
```

```

784 \else
785   \def\jsc@setfontsize#1#2#3{%
786     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
787 \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。

---

```
788 \emergencystretch 3\jsZw
```

```

\ifnarrowbaselines  欧文用に行間を狭くする論理変数と、それを真・偽にするためのコマンドです。
\narrowbaselines    [2003-06-30] 数式に入るところで \narrowbaselines を実行しているので
\widebaselines      \abovedisplayskip 等が初期化されてしまうという shintok さんのご指摘に対し
                    て、しっぽ愛好家さんが次の修正を教えてくださいました。
                    [2008-02-18] english オプションで最初の段落のインデントをしないようにしました。
                    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。

---

```

789 \newif\ifnarrowbaselines
790 \if@english
791   \narrowbaselinestrue
792 \fi
793 \def\narrowbaselines{%
794   \narrowbaselinestrue
795   \skip0=\abovedisplayskip
796   \skip2=\abovedisplayshortskip
797   \skip4=\belowdisplayskip
798   \skip6=\belowdisplayshortskip
799 % 一時的に警告を無効化する
800   \let\bxjs@ltx@nomath\@nomath
801   \let\@nomath@gobble
802   \@currsize\selectfont
803   \let\@nomath\bxjs@ltx@nomath
804   \abovedisplayskip=\skip0
805   \abovedisplayshortskip=\skip2
806   \belowdisplayskip=\skip4
807   \belowdisplayshortskip=\skip6\relax}

```

```
808 \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。

```
809 \def\bxjs@if@narrowbaselines{%
810   \ifnarrowbaselines\expandafter\@firstoftwo
811   \else \expandafter\@secondoftwo
812   \fi
813 }
```

---

\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 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。

---

```
814 \renewcommand{\normalsize}{%
815   \bxjs@if@narrowbaselines{%
816     \jsc@setfontsize\normalsize\@xpt\@xiipt
817   }{%else
818     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
819   }%
```

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。

```
820 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
821 \abovedisplayshortskip \z@ \@plus3\p@?
822 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
823 \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。

```
824 \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。

```
825 \normalsize
```

`\Cht` 基準となる長さの設定をします。pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> カーネル (`plfonts.dtx`) で宣言されているパ  
`\Cdp` ラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (`1zw`) です。

---

`\Cwd`

`\Cvs` まず `\Cwd` 等の変数を定義する。

```
\Chs 826 \ifx\Cht\@undefined \newdimen\Cht \fi
      827 \ifx\Cdp\@undefined \newdimen\Cdp \fi
      828 \ifx\Cwd\@undefined \newdimen\Cwd \fi
      829 \ifx\Cvs\@undefined \newdimen\Cvs \fi
      830 \ifx\Chs\@undefined \newdimen\Chs \fi
```

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

---

```
831 \setlength\Cht{0.88\jsZw}
832 \setlength\Cdp{0.12\jsZw}
833 \setlength\Cwd{1\jsZw}
834 \setlength\Cvs{\baselineskip}
835 \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。

---

```
836 \newcommand{\small}{%
837   \bxjs@if@narrowbaselines{%
838     <!kiyou> \jsc@setfontsize\small\@ixpt{11}%
839     <kiyou> \jsc@setfontsize\small{8.8888}{11}%
840   }{%else
841     <!kiyou> \jsc@setfontsize\small\@ixpt{13}%
842     <kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
843   }%
844   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
845   \abovedisplayshortskip \z@ \@plus3\p@?
846   \belowdisplayskip \abovedisplayskip
847   \belowdisplayshortskip \belowdisplayskip
848   \def\@listi{\leftmargin\leftmargini
849             \topsep \z@
850             \parsep \z@
851             \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。

```
852 \newcommand{\footnotesize}{%
```

```

853 \bxjs@if@narrowbaselines{%
854 %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
855 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
856 }{%else
857 %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
858 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
859 }%
860 \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
861 \abovedisplayshortskip \z@ \@plus2\p@?
862 \belowdisplayskip \abovedisplayskip
863 \belowdisplayshortskip \belowdisplayskip
864 \def\@listi{\leftmargin\leftmarginI
865 \topsep \z@
866 \parsep \z@
867 \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 868 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
869 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 870 \if@twocolumn
\HUGE 871 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
872 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
873 \else
874 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
875 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
876 \fi
877 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
878 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
879 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
880 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
881 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
882 \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合いに大きくなるためです。

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

```
883 \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。『*L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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。

```
884 % \newcommand{\headfont}{\bfseries}
885 \newcommand{\headfont}{\sffamily}
886 % \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。

```
887 %<kiyou>\setlength\columnsep{2\Cwd}
888 %<kiyou>\setlength\columnsep{28truebp}
889 \setlength\columnseprule{0\p@}
```

### ■段落

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

```
\lineskiplimit 890 \setlength\lineskip{1\p@?}
\normallineskip 891 \setlength\normallineskip{1\p@?}
\normallineskiplimit 892 \setlength\lineskiplimit{1\p@?}
893 \setlength\normallineskiplimit{1\p@?}
```

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

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

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

```
894 \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  
`\parindent` ゼロにしました。`\parindent` は段落の先頭の字下げ幅です。

```
895 \setlength\parskip{0\p@}
896 \if@slide
897 \setlength\parindent{0\p@}
898 \else
899 \setlength\parindent{1\Cwd}
900 \fi
```

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

```
\@highpenalty 901 \@lowpenalty 51
               902 \@medpenalty 151
               903 \@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。

```
904 % \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。

```
905 % \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 密

---

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
906 \ifjsc@mag
907 \mag=\bxjs@param@mag
908 \fi
909 \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。

```
910 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
911 \def\bxjs@read@b{%
912   \if\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
913   \else \expandafter\bxjs@read@d \fi}
914 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
915 \def\bxjs@read@d#1\@nil{}
916 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
917 \edef\bxjs@layout@paper{%
918   \ifjsc@mag truedimen,\fi
919   \if@landscape landscape,\fi
920   \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。

```
921 %<*article|report>
922 \def\bxjs@layout{%
```

```

923 headheight=\topskip,footskip=0.03367\paperheight,%
924 headsep=\footskip-\topskip,includeheadfoot,%
925 hscale=0.76,hmarginratio=1:1,%
926 vscale=0.83,vmarginratio=1:1,%
927 }
928 %</article|report>
929 %<*book>
930 \ifbxjs@layout@buggyhmargin %---
931 % アレ
932 \def\bxjs@layout{%
933 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
934 hmargin=36\jsc@mmm,hmarginratio=1:1,%
935 vscale=0.83,vmarginratio=1:1,%
936 }
937 \else %---
938 % 非アレ
939 \def\bxjs@layout{%
940 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
941 hmargin=18\jsc@mmm,%
942 vscale=0.83,vmarginratio=1:1,%
943 }
944 \fi %---
945 %</book>
946 %<*slide>
947 \def\bxjs@layout{%
948 noheadfoot,%
949 hscale=0.9,hmarginratio=1:1,%
950 vscale=0.944,vmarginratio=1:1,%
951 }
952 %</slide>

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
953 \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。

```

954 %<*book>
955 \newcommand\jsTextWidthLimit{40}
956 \ifx\bxjs@textwidth@limit\undefined\else
957 \let\c@bxjs@canta\tempcanta
958 \setcounter{bxjs@canta}{\bxjs@textwidth@limit}
959 \long\edef\jsTextWidthLimit{\the\tempcanta}
960 \fi
961 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

962 \def\bxjs@postproc@layout{%
963 % ドライバ再設定
964 \ifx\bxjs@geometry@driver\relax\else
965 \let\Gm@driver\bxjs@geometry@driver

```

```

966 \fi
967 % textwidth 調整
968 \@tempdimb=\textwidth
969 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
970 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
971 \advance\@tempdimb-\textwidth
972 \advance\oddsidemargin 0.5\@tempdimb
973 \advance\evensidemargin 0.5\@tempdimb
974 \fullwidth=\textwidth
975 %<*book>
976 \ifdim\textwidth>\jsTextWidthLimit\Cwd
977 \textwidth=\jsTextWidthLimit\Cwd
978 \addtolength\evensidemargin{\fullwidth-\textwidth}
979 \fi
980 %</book>
981 % textheight 調整
982 \@tempdimb=\textheight
983 \advance\textheight-\topskip
984 \divide\textheight\baselineskip \multiply\textheight\baselineskip
985 \advance\textheight\topskip
986 \advance\@tempdimb-\textheight
987 \advance\topmargin0.5\@tempdimb
988 % headheight 調整
989 \@tempdima=\topskip
990 \advance\headheight\@tempdima
991 \advance\topmargin-\@tempdima
992 % marginpar 設定
993 \setlength\marginparsep{\columnsep}
994 \setlength\marginparpush{\baselineskip}
995 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
996 -\textwidth-10\jsc@mmm-\marginparsep}
997 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
998 % 連動する変数
999 \maxdepth=.5\topskip
1000 \stockwidth=\paperwidth
1001 \stockheight=\paperheight
1002 }

```

`\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。

```

1003 \edef\jsGeometryOptions{%
1004 \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。

```
1005 \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。

```
1006 \@onlypreamble\bxjs@revert
```

```
1007 \let\bxjs@revert\@empty
```

```
1008 \edef\bxjs@tmpa{\the\ht\strutbox}
```

```
1009 \ht\strutbox=10\p@?
```

```
1010 \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。

```
1011 \ifbxjs@papersize
```

```
1012 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
```

```
1013 \PassOptionsToPackage{dvipdfm}{geometry}
```

```
1014 \fi
```

```
1015 \let\bxPapersizeSpecialDone=t
```

```
1016 \else
```

```
1017 \PassOptionsToPackage{driver=none}{geometry}
```

```
1018 \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。

```
1019 \edef\bxjs@nxt{%
```

```
1020 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
```

```
1021 \AtBeginDocument{\bxjs@pre@geometry@hook}
```

```
1022 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
```

```
1023 \bxjs@nxt \bxjs@revert
```

```
1024 \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。

```
1025 \let\bxjs@geometry@driver\Gm@driver
```

```
1026 \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。

```
1027 \@onlypreamble\bxjs@pre@geometry@hook
```

```
1028 \def\bxjs@pre@geometry@hook{%
```

```
1029 \@ifpackageloaded{geometry}{%
```

```
1030 \@ifpackagelater{geometry}{2010/02/12}{\}%else
```

```
1031 \PackageWarningNoLine\bxjs@clsname
```

```
1032 {The 'geometry' package installed\MessageBreak
```

```

1033     is too old (< v5.0)}%
1034     \if x\jsEngine \ifnum\mag=\@m\else
1035         \def\bxjs@Gm@driver{pdftex}
1036         \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1037         \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1038         \fi\fi
1039     }%
1040     \ifjsWithpTeXng
1041         \ifx\Gm@driver\@empty
1042             \def\Gm@driver{pdftex}%
1043         \fi
1044     \fi
1045 }{}}

```

`\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。

```

1046 \def\setpagelayout{\@ifstar
1047   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1048 \def\bxjs@modify@layout#1{%
1049   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1050   \bxjs@nxt\bxjs@postproc@layout}
1051 \def\bxjs@reset@layout#1{%
1052   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1053   \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。

```

1054 \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`) になっている場合はエラーを出す。

※`\jsUseMinimalPageLayout` は動作テスト用。

```

1055 \AtBeginDocument{\bxjs@check@page@layout}
1056 \@onlypreamble\bxjs@check@page@layout
1057 \def\bxjs@check@page@layout{%
1058   \ifdim\textwidth=.5\maxdimen
1059     \ClassError\bxjs@clsname
1060     {Page layout is not properly set}
1061     \@ehd
1062   \fi}
1063 \def\jsUseMinimalPageLayout{%
1064   \setlength{\textwidth}{6.5in}%
1065   \setlength{\textheight}{8in}}

```

`\setpagelayout` はとりあえず無効にしておく。

```

1066 \let\bxjs@geometry@driver\relax
1067 \def\setpagelayout{\@ifstar
1068   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1069 \def\bxjs@pagelayout@a#1{%
1070   \ClassError\bxjs@clsname
1071   {Command \string\setpagelayout\space is not supported,\MessageBreak
1072     because 'geometry' value is not 'class'}\@eha}

```

`geometry=user` の場合の処理はここで終わり。

```
1073 \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。

## ■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。

```
1074 \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。

```
1075 \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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  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。

```
1076 \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。

```
1077 \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。

```
1078 \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。

```
1079 \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。

```
1080 \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。

```
1081 \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。

```
1082 \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。

```
1083 \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。

```
1084 \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。

```
1085 \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・

`\textfloatsep` 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。

```
1086 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1087 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1088 \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 1089 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1090 \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`

```

1091 \setlength\@fptop{0\p@? \@plus 1fil}
1092 \setlength\@fpsep{8\p@? \@plus 2fil}
1093 \setlength\@fpbot{0\p@? \@plus 1fil}

```

`\@dblftop` 段抜きフロートについての値です。

```

\@dblfpsep 1094 \setlength\@dblftop{0\p@? \@plus 1fil}
\@dblfpbot 1095 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1096 \setlength\@dblfpbot{0\p@? \@plus 1fil}

```

## 6 ページスタイル

ページスタイルとして、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  (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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  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,
\@oddhead 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。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  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。

```

1097 % \def\ps@empty{%
1098 %   \let\@mkboth\@gobbletwo
1099 %   \let\@oddhead\@empty
1100 %   \let\@oddfoot\@empty
1101 %   \let\@evenhead\@empty
1102 %   \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。

```
1103 \def\ps@plainfoot{%
1104   \let\@mkboth\@gobbletwo
1105   \let\@oddhead\@empty
1106   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1107   \let\@evenhead\@empty
1108   \let\@evenfoot\@oddfoot}
1109 \def\ps@plainhead{%
1110   \let\@mkboth\@gobbletwo
1111   \let\@oddfoot\@empty
1112   \let\@evenfoot\@empty
1113   \def\@evenhead{%
1114     \if@mparswitch \hss \fi
1115     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1116     \if@mparswitch\else \hss \fi}%
1117   \def\@oddhead{%
1118     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1119 %<book>\let\ps@plain\ps@plainhead
1120 %<!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。

```
1121 %<*article|slide>
1122 \if@twoside
1123   \def\ps@headings{%
1124     \let\@oddfoot\@empty
1125     \let\@evenfoot\@empty
1126     \def\@evenhead{\if@mparswitch \hss \fi
1127       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1128     \if@mparswitch\else \hss \fi}%
1129     \def\@oddhead{%
1130       \underline{%
1131         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}\hss}%
1132     \let\@mkboth\markboth
1133     \def\sectionmark##1{\markboth{%
1134       \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
1135       ##1}{}}%
1136     \def\subsectionmark##1{\markright{%
1137       \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsw\fi
1138       ##1}}%
1139   }
1140 \else % if not twoside
1141   \def\ps@headings{%
1142     \let\@oddfoot\@empty
1143     \def\@oddhead{%
```

```

1144     \underline{%
1145     \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}\hss}%
1146     \let\@mkboth\markboth
1147     \def\sectionmark##1{\markright{%
1148     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1149     ##1}}
1150 \fi
1151 %</article|slide>

```

次は book 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）。

```

1152 %<*book|report>
1153 \newif\if@omit@number
1154 \def\ps@headings{%
1155     \let\@oddfont\@empty
1156     \let\@evenfont\@empty
1157     \def\@evenhead{%
1158     \if@mparswitch \hss \fi
1159     \underline{\hbox to \fullwidth{\autoxspacing
1160     \textbf{\thepage}\hfil\leftmark}}}%
1161     \if@mparswitch\else \hss \fi}%
1162     \def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
1163     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1164     \let\@mkboth\markboth
1165     \def\chaptermark##1{\markboth{%
1166     \ifnum \c@secnumdepth >\m@ne
1167     \if@mainmatter
1168     \if@omit@number\else
1169     \@chapapp\thechapter\@chappos\hskip1\jsZw
1170     \fi
1171     \fi
1172     \fi
1173     ##1}{}}%
1174     \def\sectionmark##1{\markright{%
1175     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1176     ##1}}}%
1177 %</book|report>

```

最後は学会誌の場合です。

```

1178 %<*jspf>
1179 \def\ps@headings{%
1180     \def\@oddfont{\normalfont\hfil\thepage\hfil}
1181     \def\@evenfont{\normalfont\hfil\thepage\hfil}
1182     \def\@oddhead{\normalfont\hfil \@title \hfil}
1183     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1184 %</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。

```
1185 \def\ps@myheadings{%
1186   \let\@oddfont\@empty\let\@evenfont\@empty
1187   \def\@evenhead{%
1188     \if@mparswitch \hss \fi%
1189     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1190     \if@mparswitch\else \hss \fi}%
1191   \def\@oddhead{%
1192     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1193   \let\@mkboth\@gobbletwo
1194 %<book|report> \let\chaptermark\@gobble
1195   \let\sectionmark\@gobble
1196 %<!book&!report> \let\subsectionmark\@gobble
1197 }
```

## 7 文書のマークアップ

### 7.1 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 1198 % \newcommand*\title}[1]{\gdef\@title{#1}}
1199 % \newcommand*\author}[1]{\gdef\@author{#1}}
1200 % \newcommand*\date}[1]{\gdef\@date{#1}}
1201 % \date{\today}
```

---

`\subtitle` 副題を設定する。

`\jsSubtitle` ※プレアンブルにおいて `\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。

```
1202 \newcommand*\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1203 %\let\bxjs@subtitle\undefined

\title にフックを入れる。
1204 \renewcommand*\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1205 \AtBeginDocument{\bxjs@decl@subtitle}
1206 \def\bxjs@decl@subtitle{%
1207   \global\let\bxjs@decl@subtitle\relax
1208   \ifx\subtitle\undefined
1209     \global\let\subtitle\jsSubtitle
1210   \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。

```

1211 \def\bxjs@annihilate@subtitle{%
1212   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1213   \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 1214 %<*jspf>
\keywords 1215 \newcommand*\etitle}[1]{\gdef\@etitle{#1}}
1216 \newcommand*\author}[1]{\gdef\@author{#1}}
1217 \newcommand*\keywords}[1]{\gdef\@keywords{#1}}
1218 \newcommand*\email}[1]{\gdef\authors@mail{#1}}
1219 \newcommand*\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1220 %</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。

```

1221 \def\plainifnotempty{%
1222   \ifx \@oddhead \@empty
1223     \ifx \@oddfoot \@empty
1224     \else
1225       \thispagestyle{plainfoot}%
1226     \fi
1227   \else
1228     \thispagestyle{plainhead}%
1229   \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。

```

1230 %<*article|book|report|slide>
1231 \if@titlepage
1232   \newcommand*\maketitle{%
1233     \begin{titlepage}%
1234       \let\footnotesize\small
1235       \let\footnoterule\relax
1236       \let\footnote\thanks
1237       \null\vfil
1238       \if@slide
1239         {\footnotesize \@date}%
1240       \begin{center}
1241         \mbox{} \\\[1\jsZw]
1242         \large
1243         {\maybeblue\hrule height0\p? depth2\p?\relax}\par
1244         \smallskip
1245         \@title

```

```

1246     \ifx\bxjs@subtitle\@undefined\else
1247     \par\vskip\z@
1248     {\small \bxjs@subtitle\par}
1249     \fi
1250     \smallskip
1251     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1252     \vfill
1253     {\small \@author}%
1254     \end{center}
1255 \else
1256 \vskip 60\p@?
1257 \begin{center}%
1258 {\LARGE \@title \par}%
1259 \ifx\bxjs@subtitle\@undefined\else
1260     \vskip5\p@?
1261     {\normalsize \bxjs@subtitle\par}
1262     \fi
1263     \vskip 3em%
1264     {\large
1265     \lineskip .75em
1266     \begin{tabular}[t]{c}%
1267         \@author
1268     \end{tabular}\par}%
1269     \vskip 1.5em
1270     {\large \@date \par}%
1271     \end{center}%
1272     \fi
1273     \par
1274     \@thanks\vfil\null
1275 \end{titlepage}%
1276 \setcounter{footnote}{0}%
1277 \global\let\thanks\relax
1278 \global\let\maketitle\relax
1279 \global\let\@thanks\@empty
1280 \global\let\@author\@empty
1281 \global\let\@date\@empty
1282 \global\let\@title\@empty
1283 \global\let\title\relax
1284 \global\let\author\relax
1285 \global\let\date\relax
1286 \global\let\and\relax
1287 \bxjs@annihilate@subtitle
1288 }%
1289 \else
1290 \newcommand{\maketitle}{\par
1291 \begin{group
1292 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1293 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1294 \long\def\@makefntext##1{\advance\leftskip 3\jsZw

```

```

1295     \parindent 1\jsZw\noindent
1296     \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1297     \if@twocolumn
1298     \ifnum \col@number=\@ne
1299         \@maketitle
1300     \else
1301         \twocolumn[\@maketitle]%
1302     \fi
1303 \else
1304     \newpage
1305     \global\@topnum\z@ % Prevents figures from going at top of page.
1306     \@maketitle
1307     \fi
1308     \plainifnotempty
1309     \@thanks
1310 \endgroup
1311 \setcounter{footnote}{0}%
1312 \global\let\thanks\relax
1313 \global\let\maketitle\relax
1314 \global\let\@thanks\@empty
1315 \global\let\@author\@empty
1316 \global\let\@date\@empty
1317 \global\let\@title\@empty
1318 \global\let\title\relax
1319 \global\let\author\relax
1320 \global\let\date\relax
1321 \global\let\and\relax
1322 \bxjs@annihilate@subtitle
1323 }

```

\@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。

```

1324 \def\@maketitle{%
1325     \newpage\null
1326     \vskip 2em
1327     \begin{center}%
1328         \let\footnote\thanks
1329         {\LARGE \@title \par}%
1330         \ifx\bxjs@subtitle\@undefined\else
1331             \vskip3\p@?
1332             {\normalsize \bxjs@subtitle\par}
1333         \fi
1334         \vskip 1.5em
1335         {\large
1336             \lineskip .5em
1337             \begin{tabular}[t]{c}%
1338                 \@author
1339             \end{tabular}\par}%
1340         \vskip 1em
1341         {\large \@date}%

```

```

1342 \end{center}%
1343 \par\vskip 1.5em
1344 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1345 }
1346 \fi
1347 %</article|book|report|slide>
1348 %<*jspf>
1349 \newcommand\maketitle{\par
1350 \begingroup
1351 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1352 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1353 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1354 \parindent 1\jsZw\noindent
1355 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1356 \twocolumn[\@maketitle]%
1357 \plainifnotempty
1358 \@thanks
1359 \endgroup
1360 \setcounter{footnote}{0}%
1361 \global\let\thanks\relax
1362 \global\let\maketitle\relax
1363 \global\let\@thanks\@empty
1364 \global\let\@author\@empty
1365 \global\let\@date\@empty
1366 % \global\let\@title\@empty % \@title は柱に使う
1367 \global\let\title\relax
1368 \global\let\author\relax
1369 \global\let\date\relax
1370 \global\let\and\relax
1371 \ifx\authors@mail\@undefined\else{%
1372 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1373 \footnotetext[0]{\itshape\authors@mail}%
1374 }\fi
1375 \global\let\authors@mail\@undefined}
1376 \def\@maketitle{%
1377 \newpage\null
1378 \vskip 6em % used to be 2em
1379 \begin{center}
1380 \let\footnote\thanks
1381 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1382 \lineskip .5em
1383 \ifx\@author\@undefined\else
1384 \vskip 1em
1385 \begin{tabular}[t]{c}%
1386 \@author
1387 \end{tabular}\par
1388 \fi
1389 \ifx\@etitle\@undefined\else
1390 \vskip 1em

```

```

1391     {\large \@etitle \par}%
1392     \fi
1393     \ifx\@eauthor\@undefined\else
1394         \vskip 1em
1395         \begin{tabular}[t]{c}%
1396             \@eauthor
1397         \end{tabular}\par
1398     \fi
1399     \vskip 1em
1400     \@date
1401 \end{center}
1402 \vskip 1.5em
1403 \centerline{\box\@abstractbox}
1404 \ifx\@keywords\@undefined\else
1405     \vskip 1.5em
1406     \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1407 \fi
1408 \vskip 1.5em}
1409 %</jspf>

```

## 7.2 章・節

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

```

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

```

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

後アキ 正の場合は、見出しの下の空きです。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。

```

1410 \def\@startsection#1#2#3#4#5#6{%
1411   \if@noskipsec \leavevmode \fi
1412   \par
1413 % 見出し上の空きを \@tempskipa にセットする
1414   \@tempskipa #4\relax
1415 % \@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
1416   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1417 % 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
1418   \ifdim \@tempskipa <\z@
1419     \@tempskipa -\@tempskipa \@afterindentfalse
1420   \fi
1421   \if@nobreak
1422 %   \everypar{\everyparhook}% これは間違い
1423     \everypar{}%
1424   \else
1425     \addpenalty\@secpenalty
1426 % 次の行は削除
1427 %   \addvspace\@tempskipa
1428 % 次の \noindent まで追加
1429     \ifdim \@tempskipa >\z@
1430       \if@slide\else
1431         \null
1432         \vspace*{-\baselineskip}%
1433       \fi
1434       \vskip\@tempskipa
1435     \fi
1436   \fi
1437   \noindent
1438 % 追加終わり
1439   \@ifstar
1440     {\@ssect{#3}{#4}{#5}{#6}}%
1441     {\@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。

```

1442 \def\@sect#1#2#3#4#5#6[#7]#8{%
1443   \ifnum #2>\c@secnumdepth
1444     \let\@svsec\@empty
1445   \else
1446     \refstepcounter{#1}%
1447     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1448   \fi
1449 % 見出し後の空きを \@tempskipa にセット
1450   \@tempskipa #5\relax
1451 % 条件判断の順序を入れ替えました
1452   \ifdim \@tempskipa <\z@

```

```

1453 \def\@svsechd{%
1454   #6{\hskip #3\relax
1455   \@svsec #8}%
1456   \csname #1mark\endcsname{#7}%
1457   \addcontentsline{toc}{#1}{%
1458     \ifnum #2>\c@secnumdepth \else
1459       \protect\numberline{\csname the#1\endcsname}%
1460     \fi
1461     #7}}% 目次にフルネームを載せるなら #8
1462 \else
1463   \begingroup
1464     \interlinepenalty \@M % 下から移動
1465     #6{%
1466       \@hangfrom{\hskip #3\relax\@svsec}%
1467 %     \interlinepenalty \@M % 上に移動
1468     #8\@@par}%
1469   \endgroup
1470   \csname #1mark\endcsname{#7}%
1471   \addcontentsline{toc}{#1}{%
1472     \ifnum #2>\c@secnumdepth \else
1473       \protect\numberline{\csname the#1\endcsname}%
1474     \fi
1475     #7}}% 目次にフルネームを載せるならここは #8
1476   \fi
1477   \@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。

```

1478 \def\@xsect#1{%
1479 % 見出しの後ろの空きを \@tempkipa にセット
1480   \@tempkipa #1\relax
1481 % 条件判断の順序を変えました
1482   \ifdim \@tempkipa<\z@
1483     \@nobreakfalse
1484     \global\@noskipsectrue
1485     \everypar{%
1486       \if@noskipsec
1487         \global\@noskipsecfalse
1488         {\setbox\z@\lastbox}%
1489         \clubpenalty\@M
1490         \begingroup \@svsechd \endgroup
1491         \unskip
1492         \@tempkipa #1\relax
1493         \hskip -\@tempkipa
1494         \bxjs<ltj@inhibitglue
1495       \else
1496         \clubpenalty \@clubpenalty
1497         \everypar{\everyparhook}%
1498       \fi\everyparhook}%

```

```

1499 \else
1500 \par \nobreak
1501 \vskip \@tempskipa
1502 \@afterheading
1503 \fi
1504 \if@slide
1505 {\vskip-6\p@?\maybeblue\hrule height0\p@? depth1\p@?\vskip7\p@?\relax}%
1506 \fi
1507 \par % 2000-12-18
1508 \ignorespaces}
1509 \def\@ssect#1#2#3#4#5{%
1510 \@tempskipa #3\relax
1511 \ifdim \@tempskipa<\z@
1512 \def\@svsechd{#4{\hskip #1\relax #5}}%
1513 \else
1514 \begingroup
1515 #4{%
1516 \@hangfrom{\hskip #1}%
1517 \interlinepenalty \@M #5\@@par}%
1518 \endgroup
1519 \fi
1520 \@xsect{#3}}

```

---

上記の定義中の `\bxjs@ltj@inhibitglue` は LuaTeX-ja で用いられるフック。

```

1521 \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 6 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 1522 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1523 % \newcommand*\sectionmark[1]{}
1524 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1525 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1526 % \newcommand*\paragraphmark[1]{}
1527 % \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定めるカウンタです。

```

1528 %<!book&!report>\setcounter{secnumdepth}{3}
1529 %<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 1530 \newcounter{part}

```

```

\c@subsubsection

```

```

\c@paragraph

```

```

\c@subparagraph

```

```

1531 %<book|report>\newcounter{chapter}
1532 %<book|report>\newcounter{section}[chapter]
1533 %<!book&!report>\newcounter{section}
1534 \newcounter{subsection}[section]
1535 \newcounter{subsubsection}[subsection]
1536 \newcounter{paragraph}[subsubsection]
1537 \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。

```

1538 \renewcommand{\thepart}{\@Roman\c@part}
1539 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1540 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1541 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1542 %<*book|report>
1543 \renewcommand{\thechapter}{\@arabic\c@chapter}
1544 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1545 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1546 %</book|report>
1547 \renewcommand{\thesubsubsection}{%
1548   \thesubsection.\@arabic\c@subsubsection}
1549 \renewcommand{\theparagraph}{%
1550   \thesubsubsection.\@arabic\c@paragraph}
1551 \renewcommand{\thesubparagraph}{%
1552   \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。

```

1553 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1554 %<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。

```

1555 %<*book|report>
1556 \newcommand\frontmatter{%

```

```

1557 \if@openright
1558   \cleardoublepage
1559 \else
1560   \clearpage
1561 \fi
1562 \@mainmatterfalse
1563 \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。

```

1564 \newcommand\mainmatter{%
1565 % \if@openright
1566 %   \cleardoublepage
1567 % \else
1568 %   \clearpage
1569 % \fi
1570 \@mainmattertrue
1571 \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。

```

1572 \newcommand\backmatter{%
1573   \if@openright
1574     \cleardoublepage
1575   \else
1576     \clearpage
1577   \fi
1578   \@mainmatterfalse}
1579 %</book|report>

```

## ■部

`\part` 新しい部を始めます。

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

```
\secdef{星なし}{星あり}
```

星なし \* のない形の定義です。

星あり \* のある形の定義です。

`\secdef` は次のようにして使います。

```

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

```

まず `book` クラス以外です。

```

1580 %<!*!book&!report>
1581 \newcommand\part{%
1582   \if@noskipsec \leavevmode \fi
1583   \par

```

```

1584 \addvspace{4ex}%
1585 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1586 \secdef\@part\@spart}
1587 %</!book&!report>

```

book スタイルの場合は、少し複雑です。

```

1588 %<*book|report>
1589 \newcommand\part{%
1590 \if@openright
1591 \cleardoublepage
1592 \else
1593 \clearpage
1594 \fi
1595 \thispagestyle{empty}% 欧文用標準スタイルでは plain
1596 \if@twocolumn
1597 \onecolumn
1598 \@restonecoltrue
1599 \else
1600 \@restonecolfalse
1601 \fi
1602 \null\vfil
1603 \secdef\@part\@spart}
1604 %</book|report>

```

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

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

```

1605 %<*!book&!report>
1606 \def\@part[#1]#2{%
1607 \ifnum \c@secnumdepth >\m@ne
1608 \refstepcounter{part}%
1609 \addcontentsline{toc}{part}{%
1610 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1611 \else
1612 \addcontentsline{toc}{part}{#1}%
1613 \fi
1614 \markboth{}{}%
1615 {\parindent\z@
1616 \raggedright
1617 \interlinepenalty \@M
1618 \normalfont
1619 \ifnum \c@secnumdepth >\m@ne
1620 \Large\headfont\prepartname\thepart\postpartname
1621 \par\nobreak
1622 \fi
1623 \huge \headfont #2%
1624 \markboth{}{}\par}%
1625 \nobreak
1626 \vskip 3ex
1627 \@afterheading}

```

```
1628 %</*!book&!report>
```

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

```
1629 %<*book|report>
```

```
1630 \def\@part[#1]#2{%
```

```
1631   \ifnum \c@secnumdepth >-2\relax
```

```
1632     \refstepcounter{part}%
```

```
1633     \addcontentsline{toc}{part}{%
```

```
1634       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
```

```
1635   \else
```

```
1636     \addcontentsline{toc}{part}{#1}%
```

```
1637   \fi
```

```
1638   \markboth{}{}%
```

```
1639   {\centering
```

```
1640     \interlinepenalty \@M
```

```
1641     \normalfont
```

```
1642     \ifnum \c@secnumdepth >-2\relax
```

```
1643       \huge\headfont \prepartname\thepart\postpartname
```

```
1644       \par\vskip20\p@?
```

```
1645     \fi
```

```
1646     \Huge \headfont #2\par}%
```

```
1647   \@endpart}
```

```
1648 %</book|report>
```

\@spart 番号を付けない部です。

```
1649 %</*!book&!report>
```

```
1650 \def\@spart#1{%
```

```
1651   \parindent \z@ \raggedright
```

```
1652   \interlinepenalty \@M
```

```
1653   \normalfont
```

```
1654   \huge \headfont #1\par}%
```

```
1655   \nobreak
```

```
1656   \vskip 3ex
```

```
1657   \@afterheading}
```

```
1658 %</*!book&!report>
```

```
1659 %<*book|report>
```

```
1660 \def\@spart#1{%
```

```
1661   \centering
```

```
1662   \interlinepenalty \@M
```

```
1663   \normalfont
```

```
1664   \Huge \headfont #1\par}%
```

```
1665   \@endpart}
```

```
1666 %</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。

```
1667 %<*book|report>
```

```
1668 \def\@endpart{\vfil\newpage
```

```
1669   \if@twoside
```

```

1670 \null
1671 \thispagestyle{empty}%
1672 \newpage
1673 \fi
1674 \if@restonecol
1675 \twocolumn
1676 \fi}
1677 %</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。

```

1678 %<*book|report>
1679 \newcommand{\chapter}{%
1680 \if@openright\cleardoublepage\else\clearpage\fi
1681 \plainifnotempty % 元: \thispagestyle{plain}
1682 \global\@topnum\z@
1683 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1684 \secdef
1685 {\@omit@numberfalse\@chapter}%
1686 {\@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。

```

1687 \def\@chapter[#1]#2{%
1688 \ifnum \c@secnumdepth >\m@ne
1689 \if@mainmatter
1690 \refstepcounter{chapter}%
1691 \typeout{\@chapapp\thechapter\@chappos}%
1692 \addcontentsline{toc}{chapter}%
1693 {\protect\numberline
1694 % %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
1695 {\@chapapp\thechapter\@chappos}%
1696 #1}%
1697 \else\addcontentsline{toc}{chapter}{#1}\fi
1698 \else
1699 \addcontentsline{toc}{chapter}{#1}%
1700 \fi
1701 \chaptermark{#1}%
1702 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
1703 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
1704 \if@twocolumn
1705 \@topnewpage[\@makechapterhead{#2}]%
1706 \else
1707 \@makechapterhead{#2}%
1708 \@afterheading
1709 \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。

```
1710 \def\@makechapterhead#1{%
1711   \vspace*{2\Cvs}% 欧文は 50pt
1712   {\parindent \z@ \raggedright \normalfont
1713     \ifnum \c@secnumdepth >\m@ne
1714       \if@mainmatter
1715         \huge\headfont \@chapapp\thechapter\@chappos
1716         \par\nobreak
1717         \vskip \Cvs % 欧文は 20pt
1718         \fi
1719         \fi
1720         \interlinepenalty\M
1721         \Huge \headfont #1\par\nobreak
1722         \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。

```
1723 \def\@schapter#1{%
1724   \chaptermark{#1}%
1725   \if@twocolumn
1726     \@topnewpage[\@makeschapterhead{#1}]%
1727   \else
1728     \@makeschapterhead{#1}\@afterheading
1729   \fi}
```

`\@makeschapterhead` 番号なしの章見出しです。

```
1730 \def\@makeschapterhead#1{%
1731   \vspace*{2\Cvs}% 欧文は 50pt
1732   {\parindent \z@ \raggedright
1733     \normalfont
1734     \interlinepenalty\M
1735     \Huge \headfont #1\par\nobreak
1736     \vskip 3\Cvs}} % 欧文は 40pt
1737 %</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。

```
1738 \if@twocolumn
1739   \newcommand{\section}{%
1740 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
1741   \@startsection{section}{1}{\z@}%
1742 %<kiyou>   {0.6\Cvs}{0.4\Cvs}%
1743 %<kiyou>   {\Cvs}{0.5\Cvs}%
1744 %   {\normalfont\large\headfont\@secapp}}
1745   {\normalfont\large\headfont\raggedright}}
1746 \else
```

```

1747 \newcommand{\section}{%
1748   \if@slide\clearpage\fi
1749   \@startsection{section}{1}{\z@}%
1750   {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1751   {.5\Cvs \@plus.3\Cdp}% 後アキ
1752 %   {\normalfont\Large\headfont\@secapp}}
1753   {\normalfont\Large\headfont\raggedright}}
1754 \fi

```

`\subsection` 同上です。

```

1755 \if@twocolumn
1756   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1757     {\z@}{\z@}%
1758     {\normalfont\normalsize\headfont}}
1759 \else
1760   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1761     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1762     {.5\Cvs \@plus.3\Cdp}% 後アキ
1763     {\normalfont\large\headfont}}
1764 \fi

```

`\subsubsection`

```

1765 \if@twocolumn
1766   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1767     {\z@}{\z@}%
1768     {\normalfont\normalsize\headfont}}
1769 \else
1770   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1771     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1772     {\z@}%
1773     {\normalfont\normalsize\headfont}}
1774 \fi

```

`\paragraph` 見出しの後ろで改行されません。

---

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。さらに、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た。

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

---

```

1775 \newcommand\jsParagraphMark{\jsJaChar{■}}
1776 \ifx\bxjs@paragraph@mark\undefined\else
1777   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
1778 \fi
1779 \let\jsJaChar\@empty
1780 \if@twocolumn
1781   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1782     {\z@}{-1\jsZw}% 改行せず 1\jsZw のアキ

```

```

1783 %<jspf>    {\normalfont\normalsize\headfont}}
1784 %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
1785 \else
1786   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1787     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1788     {-1\jsZw}% 改行せず 1\jsZw のアキ
1789 %<jspf>    {\normalfont\normalsize\headfont}}
1790 %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
1791 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

1792 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1793   {\z@}{-1\jsZw}%
1794   {\normalfont\normalsize\headfont}}

```

### 7.3 リスト環境

第  $k$  レベルのリストの初期化をするのが `\@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。

```

1795 \if@slide
1796   \setlength\leftmargini{1\jsZw}
1797 \else
1798   \if@twocolumn
1799     \setlength\leftmargini{2\jsZw}
1800   \else
1801     \setlength\leftmargini{3\jsZw}
1802   \fi
1803 \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  
`\leftmarginiii` 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 1804 \if@slide
\leftmarginv 1805   \setlength\leftmarginii {1\jsZw}
1806   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 1807   \setlength\leftmarginiv {1\jsZw}
1808   \setlength\leftmarginv  {1\jsZw}
1809   \setlength\leftmarginvi {1\jsZw}
1810 \else
1811   \setlength\leftmarginii {2\jsZw}
1812   \setlength\leftmarginiii{2\jsZw}
1813   \setlength\leftmarginiv {2\jsZw}
1814   \setlength\leftmarginv  {1\jsZw}

```

```
1815 \setlength\leftmarginvi {1\jsZw}
1816 \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。

```
1817 \setlength \labelsep {0.5\jsZw} % .5em
1818 \setlength \labelwidth{\leftmargini}
1819 \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。

```
1820 \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 1821 \@beginparpenalty -\@lowpenalty
\@itempenalty 1822 \@endparpenalty -\@lowpenalty
1823 \@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。

```
1824 \def\@listi{\leftmargin\leftmargini
1825 \parsep \z@
1826 \topsep 0.5\baselineskip
1827 \itemsep \z@ \relax}
1828 \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）。

```
1829 \@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 1830 \def\@listii{\leftmargin\leftmarginii
\@listiv 1831 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1832 \topsep \z@
\@listv 1833 \parsep \z@
\@listvi 1834 \itemsep\parsep}
1835 \def\@listiii{\leftmargin\leftmarginiii
1836 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1837 \topsep \z@
1838 \parsep \z@
1839 \itemsep\parsep}
1840 \def\@listiv {\leftmargin\leftmarginiv
```

```

1841          \labelwidth\leftmarginiv
1842          \advance\labelwidth-\labelsep}
1843 \def\@listv  {\leftmargin\leftmarginiv
1844          \labelwidth\leftmarginiv
1845          \advance\labelwidth-\labelsep}
1846 \def\@listvi {\leftmargin\leftmarginvi
1847          \labelwidth\leftmarginvi
1848          \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。

```

1849 \renewcommand{\theenumi}{\@arabic\c@enumi}
1850 \renewcommand{\theenumii}{\@alph\c@enumii}
1851 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1852 \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。

---

```

1853 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1854 \newcommand*\jsInJaParen[1]{%
1855   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1856 \newcommand{\labelenumi}{\theenumi.}
1857 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1858 \newcommand{\labelenumiii}{\theenumiii.}
1859 \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` 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 1860 \renewcommand{\p@enumii}{\theenumi}
1861 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1862 \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 1863 \newcommand\labelitemi{\textbullet}
\labelitemiii 1864 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiv 1865 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemv 1866 \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。

```
1867 \newenvironment{description}{%
1868   \list{}{%
1869     \labelwidth=\leftmargin
1870     \labelsep=1\jsZw
1871     \advance \labelwidth by -\labelsep
1872     \let \makelabel=\descriptionlabel}}{\endlist}
```

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

```
1873 \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。

```
1874 %<*book|report>
1875 \newenvironment{abstract}{%
1876   \begin{list}{}{%
1877     \listparindent=1\jsZw
1878     \itemindent=\listparindent
1879     \rightmargin=\z@
1880     \leftmargin=5\jsZw}\item[]{\end{list}\vspace{\baselineskip}}
1881 %</book|report>
1882 %<*article|slide>
1883 \newbox\@abstractbox
1884 \if@titlepage
1885   \newenvironment{abstract}{%
1886     \titlepage
1887     \null\vfil
1888     \@beginparpenalty\@lowpenalty
1889     \begin{center}%
```

```

1890     \headfont \abstractname
1891     \@endparpenalty\@M
1892     \end{center}}%
1893     {\par\vfil\null\endtitlepage}
1894 \else
1895     \newenvironment{abstract}{%
1896     \if@twocolumn
1897     \ifx\maketitle\relax
1898     \section*{\abstractname}%
1899     \else
1900     \global\setbox\@abstractbox\hbox\bgroup
1901     \begin{minipage}[b]{\textwidth}
1902     \small\parindent1\jsZw
1903     \begin{center}%
1904     {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1905     \end{center}%
1906     \list{}{%
1907     \listparindent\parindent
1908     \itemindent \listparindent
1909     \rightmargin \leftmargin}%
1910     \item\relax
1911     \fi
1912     \else
1913     \small
1914     \begin{center}%
1915     {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1916     \end{center}%
1917     \list{}{%
1918     \listparindent\parindent
1919     \itemindent \listparindent
1920     \rightmargin \leftmargin}%
1921     \item\relax
1922     \fi}{\if@twocolumn
1923     \ifx\maketitle\relax
1924     \else
1925     \endlist\end{minipage}\egroup
1926     \fi
1927     \else
1928     \endlist
1929     \fi}
1930 \fi
1931 %</article|slide>
1932 %<*.jspf>
1933 \newbox\@abstractbox
1934 \newenvironment{abstract}{%
1935     \global\setbox\@abstractbox\hbox\bgroup
1936     \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
1937     \small
1938     \if@english \parindent6mm \else \parindent1\jsZw \fi}%

```

```
1939 {\end{minipage}\egroup}
1940 %</jspf>
```

#### ■キーワード

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

```
1941 %<*jspf>
1942 %\newbox\@keywordsbox
1943 %\newenvironment{keywords}{%
1944 % \global\setbox\@keywordsbox\hbox\bgroup
1945 % \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
1946 % \small\parindent0\jsZw}%
1947 % {\end{minipage}\egroup}
1948 %</jspf>
```

#### ■verse 環境

**verse** 詩のための `verse` 環境です。

```
1949 \newenvironment{verse}{%
1950 \let \=\@centercr
1951 \list{}{%
1952 \itemsep \z@
1953 \itemindent -2\jsZw % 元: -1.5em
1954 \listparindent\itemindent
1955 \rightmargin \z@
1956 \advance\leftmargin 2\jsZw}% 元: 1.5em
1957 \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。

```
1958 \newenvironment{quotation}{%
1959 \list{}{%
1960 \listparindent\parindent
1961 \itemindent\listparindent
1962 \rightmargin \z@}%
1963 \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。

```
1964 \newenvironment{quote}%
1965 {\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。

```

1966 \def\@begintheorem#1#2{\trivlist\labelsep=1\jszw
1967   \item[\hskip \labelsep{\headfont #1\ #2}]}
1968 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jszw
1969   \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。

```

1970 \newenvironment{titlepage}{%
1971 %<book|report> \cleardoublepage
1972   \if@twocolumn
1973     \@restonecoltrue\onecolumn
1974   \else
1975     \@restonecolfalse\newpage
1976   \fi
1977   \thispagestyle{empty}%
1978   \setcounter{page}\@ne
1979 }%
1980 {\if@restonecol\twocolumn \else \newpage \fi
1981   \if@twoside\else
1982     \setcounter{page}\@ne
1983   \fi}

```

## ■付録

`\appendix` 本文と付録を分離するコマンドです。

```

1984 %<!*book&!report>
1985 \newcommand{\appendix}{\par
1986   \setcounter{section}{0}%
1987   \setcounter{subsection}{0}%
1988   \gdef\presectionname{\appendixname}%
1989   \gdef\postsectionname{}}%
1990 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
1991 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
1992 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}%
1993 %</!*book&!report>
1994 %<*book|report>
1995 \newcommand{\appendix}{\par
1996   \setcounter{chapter}{0}%
1997   \setcounter{section}{0}%
1998   \gdef\@chapapp{\appendixname}%
1999   \gdef\@chappos{}}%

```

```
2000 \gdef\thechapter{\@Alph\c@chapter}
2001 %</book|report>
```

## 7.4 パラメータの設定

### ■array と tabular 環境

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

```
2002 \setlength\arraycolsep{5\p@?}
```

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

```
2003 \setlength\tabcolsep{6\p@?}
```

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

```
2004 \setlength\arrayrulewidth{.4\p@}
```

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

```
2005 \setlength\doublerulesep{2\p@}
```

### ■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2006 \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。

```
2007 \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。

```
2008 \setlength\fboxsep{3\p@?}
```

```
2009 \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。

```
2010 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```
2011 %<*book|report>
```

```
2012 \@addtoreset{equation}{chapter}
```

```
2013 \renewcommand\theequation
```

```
2014 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

```
2015 %</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。

```
2016 % \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。

```
2017 % \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。

```
2018 % \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip\@italiccorr)}}
```

## 7.5 フロート

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

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

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

`\ext@TYPE` フロートの目次を出力するファイルの拡張子です。

`\fnum@TYPE` キャプション用の番号を生成するマクロです。

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

### ■ figure 環境

`\c@figure` 図番号のカウンタです。

`\thefigure` 図番号を出力するコマンドです。

```
2019 %<!*book&!report>
2020 \newcounter{figure}
2021 \renewcommand \thefigure {\@arabic\c@figure}
2022 %</!*book&!report>
2023 %<*book|report>
2024 \newcounter{figure}[chapter]
2025 \renewcommand \thefigure
2026     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2027 %</book|report>
```

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

```
\ext@figure 2028 \def\fps@figure{tbp}
\ftype@figure 2029 \def\ftype@figure{1}
\fnum@figure 2030 \def\ext@figure{lof}
2031 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` \* 形式は段抜きフロートです。

```

figure* 2032 \newenvironment{figure}%
2033         {\@float{figure}}%
2034         {\end@float}
2035 \newenvironment{figure*}%
2036         {\@dblfloat{figure}}%
2037         {\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。

```

2038 %<!*book&!report>
2039 \newcounter{table}
2040 \renewcommand\thetable{\@arabic\c@table}
2041 %</!*book&!report>
2042 %<*book|report>
2043 \newcounter{table}[chapter]
2044 \renewcommand \thetable
2045     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2046 %</book|report>

```

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

```

\ext@table 2047 \def\fps@table{tbp}
2048 \def\ftype@table{2}
\fnum@table 2049 \def\ext@table{lot}
2050 \def\fnum@table{\tablename\nobreak\thetable}

```

`table` \* は段抜きフロートです。

```

table* 2051 \newenvironment{table}%
2052         {\@float{table}}%
2053         {\end@float}
2054 \newenvironment{table*}%
2055         {\@dblfloat{table}}%
2056         {\end@dblfloat}

```

## 7.6 キャプション

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

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

```

2057 \newlength\abovecaptionskip

```

```

2058 \newlength\belowcaptionskip
2059 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2060 \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。

```

2061 %<!*jspf>
2062 % \long\def\@makecaption#1#2{\small
2063 % \advance\leftskip1cm
2064 % \advance\rightskip1cm
2065 % \vskip\abovecaptionskip
2066 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2067 % \ifdim \wd\@tempboxa >\hsize
2068 % #1\hskip1\jsZw\relax #2\par
2069 % \else
2070 % \global \@minipagefalse
2071 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2072 % \fi
2073 % \vskip\belowcaptionskip}}
2074 \long\def\@makecaption#1#2{\small
2075 \advance\leftskip .0628\linewidth
2076 \advance\rightskip .0628\linewidth
2077 \vskip\abovecaptionskip
2078 \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2079 \ifdim \wd\@tempboxa <\hsize \centering \fi
2080 #1\hskip1\jsZw\relax #2\par
2081 \vskip\belowcaptionskip}}
2082 %</!*jspf>
2083 %<!*jspf>
2084 \long\def\@makecaption#1#2{%
2085 \vskip\abovecaptionskip
2086 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2087 \ifdim \wd\@tempboxa >\hsize
2088 {\small\sffamily
2089 \list{#1}{%
2090 \renewcommand{\makelabel}[1]{##1\hfil}
2091 \itemsep \z@
2092 \itemindent \z@
2093 \labelsep \z@
2094 \labelwidth 11mm
2095 \listparindent\z@
2096 \leftmargin 11mm}\item\relax #2\endlist}
2097 \else
2098 \global \@minipagefalse
2099 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2100 \fi
2101 \vskip\belowcaptionskip}

```

```
2102 %</jspf>
```

## 8 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 は BXJS クラスでは不使用。
\if@jsc@warnoldfontcmdexception は \allow/disallowoldfontcommands の状態
を表す。
```

```
2103 \newif\if@jsc@warnoldfontcmd
2104 \@jsc@warnoldfontcmdtrue
2105 \newif\if@jsc@warnoldfontcmdexception
2106 \@jsc@warnoldfontcmdexceptionfalse
```

```
\jsc@DeclareOldFontCommand
```

```
2107 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2108   \@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2109   \DeclareOldFontCommand{#1}{%
2110     \bxjs@oldfontcmd{#1}#2%
2111   }{%
2112     \bxjs@oldfontcmd{#1}#3%
2113   }%
2114 }
2115 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2116   \ClassInfo\bxjs@clsname
2117   {Old font command '\string#1' is used!!\MessageBreak
2118     First occurrence is}%
2119 }
```

```
\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。
```

```
2120 \newcommand*{\allowoldfontcommands}{%
2121   \@jsc@warnoldfontcmdexceptiontrue}
2122 \newcommand*{\disallowoldfontcommands}{%
2123   \@jsc@warnoldfontcmdexceptionfalse}
```

※ 1.x 版では Warning ではなく Info に留めておく。

```
2124 \let\bxjs@oldfontcmd@list\@empty
2125 \def\bxjs@oldfontcmd#1{%
2126   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2127 \def\bxjs@oldfontcmd@a#1#2{%
2128   \if@jsc@warnoldfontcmdexception\else
2129     \global\@jsc@warnoldfontcmdfalse
2130     \ifx#1\relax
2131       \global\let#1=t%
2132       \jsc@warnoldfontcmd{#2}%
2133     \fi
2134   \fi}
2135 \def\bxjs@warnoldfontcmd@final{%
2136   \par
2137   \let\@tempa\@empty
2138   \def\do##1{%
2139     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2140       \edef\@tempa{\@tempa \space\string##1}\fi}
2141   \bxjs@oldfontcmd@list
2142   \ifx\@tempa\@empty\else
2143     \ClassWarningNoLine\bxjs@clsname
2144     {Some old font commands were used in text\MessageBreak
2145      (see the log output for detail)}%
2146     \ClassInfo\bxjs@clsname
2147     {Some old font commands were used in text:\MessageBreak
2148      \space\@tempa\MessageBreak
2149      You should note, that since 1994 LaTeX2e provides a\MessageBreak
2150      new font selection scheme called NFSS2 with several\MessageBreak
2151      new, combinable font commands. The
2152      class provides\MessageBreak
2153      the old font commands
2154      only for compatibility%
2155     \@gobble}%
2156   \fi}
2157 \AtEndDocument{\bxjs@warnoldfontcmd@final}
```

---

`\mc` フォントファミリーを変更します。

```
\gt 2158 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
2159 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\rm 2160 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2161 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2162 \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。

```
2163 \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`

`\sc`

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` です。

```
2164 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2165 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2166 \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 2167 \DeclareRobustCommand*{\cal}{\@fontswitch\relax\mathcal}
2168 \DeclareRobustCommand*{\mit}{\@fontswitch\relax\mathnormal}
```

## 9 相互参照

### 9.1 目次の類

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

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

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

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

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

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

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

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

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

という書式です。

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

**インデント** 左側の字下げ量です。

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

`\@pnumwidth` ページ番号の入る箱の幅です。

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

`\@dotsep` 点の間隔です (単位  $\mu$ )。

`\c@tocdepth` 目次ページに出力する見出しレベルです。元は `article` で 3, その他で 2 でしたが, ここでは一つずつ減らしています。

```
2169 \newcommand\@pnumwidth{1.55em}
2170 \newcommand\@tocrmarg{2.55em}
2171 \newcommand\@dotsep{4.5}
2172 %<!book&!report>\setcounter{tocdepth}{2}
2173 %<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)

```
2174 \newdimen\jsc@tocl@width
2175 \newcommand{\tableofcontents}{%
2176 %<*book|report>
2177 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2178 \settowidth\@tempdima{\headfont\appendixname}%
2179 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2180 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2181 \if@twocolumn
2182 \@restonecoltrue\onecolumn
2183 \else
2184 \@restonecolfalse
2185 \fi
2186 \chapter*{\contentsname}%
2187 \@mkboth{\contentsname}{}%
2188 %</book|report>
2189 %<*!book&!report>
2190 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2191 \settowidth\@tempdima{\headfont\appendixname}%
2192 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2193 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2194 \section*{\contentsname}%
2195 \@mkboth{\contentsname}{\contentsname}%
2196 %</!book&!report>
2197 \@starttoc{toc}%
2198 %<book|report> \if@restonecol\twocolumn\fi
2199 }
```

`\l@part` 部の目次です。

```
2200 \newcommand*{\l@part}[2]{%
2201 \ifnum \c@tocdepth >-2\relax
2202 %<!book&!report> \addpenalty\@secpenalty
2203 %<book|report> \addpenalty{-\@highpenalty}%
2204 \addvspace{2.25em \@plus\p@?}%
2205 \begingroup
2206 \parindent \z@
```

```

2207 %      \@pnumwidth should be \@tocrmarg
2208 %      \rightskip \@pnumwidth
2209      \rightskip \@tocrmarg
2210      \parfillskip -\rightskip
2211      {\leavevmode
2212       \large \headfont
2213       \setlength\@lnumwidth{4\jsZw}%
2214       #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2215      \nobreak
2216 %<book|report>      \global\@nobreaktrue
2217 %<book|report>      \everypar{\global\@nobreakfalse\everypar{}}%
2218      \endgroup
2219      \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)

```

2220 %<*book|report>
2221 \newcommand*\l@chapter}[2]{%
2222   \ifnum \c@tocdepth >\m@ne
2223     \addpenalty{-\@highpenalty}%
2224     \addvspace{1.0em \@plus\p@?}
2225 %   \vskip 1.0em \@plus\p@   % book.cls では↑がこうなっている
2226     \begingroup
2227     \parindent\z@
2228 %     \rightskip\@pnumwidth
2229     \rightskip\@tocrmarg
2230     \parfillskip-\rightskip
2231     \leavevmode\headfont
2232 %     % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2233     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2234     \advance\leftskip\@lnumwidth \hskip-\leftskip
2235     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2236     \penalty\@highpenalty
2237     \endgroup
2238     \fi}
2239 %</book|report>

```

\l@section 節の目次です。

```

2240 %<*!book&!report>
2241 \newcommand*\l@section}[2]{%
2242   \ifnum \c@tocdepth >\z@
2243     \addpenalty{\@secpenalty}%
2244     \addvspace{1.0em \@plus\p@?}%
2245     \begingroup
2246     \parindent\z@
2247 %     \rightskip\@pnumwidth
2248     \rightskip\@tocrmarg
2249     \parfillskip-\rightskip

```

```

2250     \leavevmode\headfont
2251 %     % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2252     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2253     \advance\leftskip\@lnumwidth \hskip-\leftskip
2254     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2255     \endgroup
2256     \fi}
2257 %</*!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。

```

2258 %<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@section 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@subsection しません。

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

```

\l@subsection 2259 %</*!book&!report>
2260 % \newcommand*\l@section}{\@dottedtocline{2}{1.5em}{2.3em}}
2261 % \newcommand*\l@subsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2262 % \newcommand*\l@paragraph}{\@dottedtocline{4}{7.0em}{4.1em}}
2263 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{10em}{5em}}
2264 %
2265 % \newcommand*\l@subsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2266 % \newcommand*\l@paragraph}{\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2267 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2268 %
2269 \newcommand*\l@section}{%
2270     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2271     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2272 \newcommand*\l@subsection}{%
2273     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2274     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2275 \newcommand*\l@paragraph}{%
2276     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2277     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2278 \newcommand*\l@subparagraph}{%
2279     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2280     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2281 %</*!book&!report>
2282 %<*book|report>
2283 % \newcommand*\l@section}{\@dottedtocline{2}{3.8em}{3.2em}}
2284 % \newcommand*\l@subsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2285 % \newcommand*\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
2286 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
2287 \newcommand*\l@section}{%
2288     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2289     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2290 \newcommand*\l@subsection}{%

```

```

2291         \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2292         \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2293 \newcommand*\l@subsubsection}{%
2294         \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2295         \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2296 \newcommand*\l@paragraph}{%
2297         \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2298         \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2299 \newcommand*\l@subparagraph}{%
2300         \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2301         \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2302 %</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` ですが、アスキー版では `\@tempdima` の代わりに `\@lnumwidth` という変数で幅を決めるように再定義しています。後続文字が全角か半角かでスペースが変わらないように `\hspace` を入れておきました。

```

2303 \newdimen\@lnumwidth
2304 \def\numberline#1{\hbext@\@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。

```

2305 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2306   \vskip \z@ \@plus.2\p@?
2307   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2308     \parindent #2\relax\@afterindenttrue
2309     \interlinepenalty\@M
2310     \leavevmode
2311     \@lnumwidth #3\relax
2312     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2313     {#4}\nobreak
2314     \leaders\hbox{$\m@th \mkern \dotsep mu\hbox{.}\mkern \dotsep
2315       mu$}\hfill \nobreak\hbext@\@pnumwidth{%
2316       \hfil\normalfont \normalcolor #5}\par}\fi}

```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2317 \newcommand{\listoffigures}{%
2318 %<*book|report>
2319   \if@twocolumn\@restonecoltrue\onecolumn
2320   \else\@restonecolfalse\fi
2321   \chapter*{\listfigurename}%
2322   \@mkboth{\listfigurename}{}%
2323 %</book|report>
2324 %<!*book&!report>
2325   \section*{\listfigurename}%
2326   \@mkboth{\listfigurename}{\listfigurename}%

```

```

2327 %</!book&!report>
2328 \starttoc{lof}%
2329 %<book|report> \if@restonecol\twocolumn\fi
2330 }

```

`\l@figure` 図目次の項目を出力します。

```
2331 \newcommand*\l@figure{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```

2332 \newcommand{\listoftables}{%
2333 %<*book|report>
2334 \if@twocolumn\@restonecoltrue\onecolumn
2335 \else\@restonecolfalse\fi
2336 \chapter*\listtablename}%
2337 \@mkboth{\listtablename}{}%
2338 %</book|report>
2339 %<*!book&!report>
2340 \section*\listtablename}%
2341 \@mkboth{\listtablename}{\listtablename}%
2342 %</!book&!report>
2343 \starttoc{lot}%
2344 %<book|report> \if@restonecol\twocolumn\fi
2345 }

```

`\l@table` 表目次は図目次と同じです。

```
2346 \let\l@table\l@figure
```

## 9.2 参考文献

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

```

2347 \newdimen\bibindent
2348 \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。

```

2349 \newenvironment{thebibliography}[1]{%
2350 \@jsc@warnoldfontcmdexceptiontrue
2351 \global\let\presectionname\relax
2352 \global\let\postsectionname\relax
2353 %<article|slide> \section*\refname\@mkboth{\refname}{\refname}%
2354 %<*kiyou>
2355 \vspace{1.5\baselineskip}
2356 \subsubsection*\refname\@mkboth{\refname}{\refname}%
2357 \vspace{0.5\baselineskip}
2358 %</kiyou>

```

```

2359 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2360 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2361 \list{\@biblabel{\@arabic{c@enumiv}}}%
2362     {\settowidth\labelwidth{\@biblabel{#1}}%
2363     \leftmargin\labelwidth
2364     \advance\leftmargin\labelsep
2365     \@openbib@code
2366     \usecounter{enumiv}%
2367     \let\p@enumiv\@empty
2368     \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2369 %<kiyou> \small
2370 \sloppy
2371 \clubpenalty4000
2372 \@clubpenalty\clubpenalty
2373 \widowpenalty4000%
2374 \sfcode`.\@m}
2375 {\def\@noitemerr
2376     {\@latex@warning{Empty `thebibliography' environment}}}%
2377 \endlist}

```

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

```

2378 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

```

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

```

2379 \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。

```

2380 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

`\cite` 文献の番号を出力する部分は `ltbibl.dtx` で定義されていますが、コンマとカッコを和文  
`\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必  
`\@citex` 要に応じて生かしてください。かつこの前後に入るグルーを `\jsInhibitGlue` で取っ  
ていきますので、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。

```

2381 % \def\@citex[#1]#2{%
2382 %     \let\@citea\@empty
2383 %     \@cite{\@for\@citeb:=#2\do
2384 %         {\@citea\def\@citea{, \jsInhibitGlue\penalty\@m }%
2385 %         \edef\@citeb{\expandafter\@firstofone\@citeb}%
2386 %         \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2387 %         \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2388 %         \G@refundefinedtrue
2389 %         \@latex@warning
2390 %             {Citation `@\@citeb' on page \thepage \space undefined}}%
2391 %         {\hbox{\csname b@\@citeb\endcsname}}}{#1}}
2392 % \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。

```
2393 % \DeclareRobustCommand\cite{\unskip
2394 %   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]
2395 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2396 %   , \jsInhibitGlue\ #2\fi}) }}$}
```

### 9.3 索引

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

```
2397 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2398   \if@twocolumn
2399     \onecolumn\@restonecolfalse
2400   \else
2401     \clearpage\@restonecoltrue
2402   \fi
2403   \columnseprule.4pt \columnsep 2\jsZw
2404   \ifx\multicols\@undefined
2405 %<book|report>   \twocolumn[\@makeschapterhead{\indexname}]%
2406 %<book|report>   \addcontentsline{toc}{chapter}{\indexname}]%
2407 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2408 %<!book&!report> \twocolumn[\section*{\indexname}]%
2409   \else
2410     \ifdim\textwidth<\fullwidth
2411       \setlength{\evensidemargin}{\oddsidemargin}
2412       \setlength{\textwidth}{\fullwidth}
2413       \setlength{\linewidth}{\fullwidth}
2414 %<book|report>   \begin{multicols}{3}[\chapter*{\indexname}]%
2415 %<book|report>   \addcontentsline{toc}{chapter}{\indexname}]%
2416 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2417 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]%
2418   \else
2419 %<book|report>   \begin{multicols}{2}[\chapter*{\indexname}]%
2420 %<book|report>   \addcontentsline{toc}{chapter}{\indexname}]%
2421 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2422 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]%
2423   \fi
2424   \fi
2425 %<book|report>   \@mkboth{\indexname}{}%
2426 %<!book&!report> \@mkboth{\indexname}{\indexname}%
2427   \plainifnotempty % \thispagestyle{plain}
2428   \parindent\z@
2429   \parskip\z@ \@plus .3\p@\relax
2430   \let\item\@idxitem
2431   \raggedright
2432   \footnotesize\narrowbaselines
```

```

2433 }{
2434   \ifx\multicols\undefined
2435     \if@restonecol\onecolumn\fi
2436   \else
2437     \end{multicols}
2438   \fi
2439   \clearpage
2440 }

```

`\@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 2441 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2442 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2443 \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。

```
2444 \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。

```

2445 \newcommand\seename{\if@english see\else →\fi}
2446 \newcommand\alsoname{\if@english see also\else →\fi}

```

## 9.4 脚注

`\footnote` 和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため,

`\footnotemark` `\jsInhibitGlue` を入れることにします。

```

2447 \let\footnotes@ve=\footnote
2448 \def\footnote{\jsInhibitGlue\footnotes@ve}
2449 \let\footnotemarks@ve=\footnotemark
2450 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}

```

`\@makefnmark` 脚注番号を付ける命令です。ここでは脚注番号の前に記号 \* を付けています。「注 1」の形式にするには `\textasteriskcentered` を `注\kern0.1em` にしてください。`\@xfootnotenext` と合わせて, もし脚注番号が空なら記号も出力しないようにしてあります。

[2002-04-09] インプリメントの仕方を変えたため消しました。

[2013-04-23] 新しい pTeX 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 pLaTeX の変更に追随しました (Thanks: 角藤さん)。

---

pTeX 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。

```
2451 \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。

```
2452 % \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。

```
2453 \renewcommand{\footnoterule}{%
```

```
2454 \kern-2.6\p@? \kern-.4\p@
```

```
2455 \hrule width .4\columnwidth
```

```
2456 \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2457 %<book|report>\@addtoreset{footnote}{chapter}
```

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

```
2458 \long\def\@footnotetext{%
```

```
2459 \insert\footins\bgroup
```

```
2460 \normalfont\footnotesize
```

```
2461 \interlinepenalty\interfootnotelinepenalty
```

```
2462 \splittopskip\footnotesep
```

```
2463 \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
```

```
2464 \hsize\columnwidth \@parboxrestore
```

```
2465 \protected@edef\@currentlabel{%
```

```
2466 \csname p@footnote\endcsname\@thefnmark
```

```
2467 }%
```

```
2468 \color@begingroup
```

```
2469 \@makefnmark{%
```

```
2470 \rule\z@\footnotesep\ignorespaces}%
```

```
2471 \futurelet\next\fo@t}
```

```
2472 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
```

```
2473 \else \let\next\fo@t\fi \next}
```

```
2474 \def\fo@t{\bgroup\aftergroup\@foot\let\next}
```

```
2475 \def\fo@t#1{#1\@foot}
```

```
2476 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup}
```

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

```
2477 \newcommand\@makefnmark[1]{%
```

```
2478 \advance\leftskip 3\jsZw
```

```
2479 \parindent 1\jsZw
```

```
2480 \noindent
```

```
2481 \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。

```
2482 % \def\xfootnotenext[#1]{%
2483 %   \begingroup
2484 %     \ifnum#1>\z@
2485 %       \csname c@\mpfn\endcsname #1\relax
2486 %     \unrestored@protected@xdef\@thefnmark{\thempfn}%
2487 %   \else
2488 %     \unrestored@protected@xdef\@thefnmark{}%
2489 %   \fi
2490 % \endgroup
2491 % \@footnotetext}
```

---

ここまでのコードは JS クラスを踏襲する。

---

## 10 段落の頭へのグルー挿入禁止

段落頭のかぎカッコなどを見かけ 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。

```
2492 \let\jsInhibitGlueAtParTop\@empty
```

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

```
2493 \def\everyparhook{\jsInhibitGlueAtParTop}
2494 \AtBeginDocument{\everypar{\everyparhook}}
```

`\@inhibitglue` JS クラスでの `\jsInhibitGlueAtParTop` の実装。

```
2495 \def\@inhibitglue{%
2496   \futurelet\@let@token\@@inhibitglue}
2497 \def\@@inhibitglue{%
2498   \ifx\@let@token「%
2499     \jsInhibitGlue
2500   \else
2501     \ifx\@let@token（%
2502       \jsInhibitGlue
2503     \else
2504       \ifx\@let@token『%
2505         \jsInhibitGlue
2506       \else
```

```

2507     \ifx\@let@token [%
2508     \jsInhibitGlue
2509     \fi
2510     \fi
2511     \fi
2512     \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。

```

2513 \def\@doendpe{%
2514     \@endpetrue
2515     \def\par{%
2516         \@restorepar\everypar{\everyparhook}\par\@endpefalse}%
2517     \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

`\item` 命令の直後です。

```

2518 \def\@item[#1]{%
2519     \if@noperitem
2520         \@donoperitem
2521     \else
2522         \if@inlabel
2523             \indent \par
2524         \fi
2525         \ifhmode
2526             \unskip\unskip \par
2527         \fi
2528         \if@newlist
2529             \if@nobreak
2530                 \@nbitem
2531             \else
2532                 \addpenalty\@beginparpenalty
2533                 \addvspace\@topsep
2534                 \addvspace{-\parskip}%
2535             \fi
2536         \else
2537             \addpenalty\@itempenalty
2538             \addvspace\itemsep
2539         \fi
2540         \global\@inlabeltrue
2541     \fi
2542     \everypar{%
2543         \@minipagefalse
2544         \global\@newlistfalse
2545         \if@inlabel
2546             \global\@inlabelfalse
2547         {\setbox\z@\lastbox
2548         \ifvoid\z@

```

```

2549     \kern-\itemindent
2550     \fi}%
2551     \box\@labels
2552     \penalty\z@
2553     \fi
2554     \if@nobreak
2555     \@nobreakfalse
2556     \clubpenalty \@M
2557     \else
2558     \clubpenalty \@clubpenalty
2559     \everypar{\everyparhook}%
2560     \fi
2561     \bxjs@ltj@inhibitglue
2562     \everyparhook}%
2563 \if@noitemarg
2564 \@noitemargfalse
2565 \if@nmbrrlist
2566 \refstepcounter\@listctr
2567 \fi
2568 \fi
2569 \sbox\@tempboxa{\makelabel{#1}}%
2570 \global\setbox\@labels\hbox{%
2571 \unhbox\@labels
2572 \hskip \itemindent
2573 \hskip -\labelwidth
2574 \hskip -\labelsep
2575 \ifdim \wd\@tempboxa >\labelwidth
2576 \box\@tempboxa
2577 \else
2578 \hbox to\labelwidth {\unhbox\@tempboxa}%
2579 \fi
2580 \hskip \labelsep}%
2581 \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。

```

2582 \def\@afterheading{%
2583 \@nobreaktrue
2584 \everypar{%
2585 \if@nobreak
2586 \@nobreakfalse
2587 \clubpenalty \@M
2588 \if@afterindent \else
2589 {\setbox\z@\lastbox}%
2590 \fi
2591 \else
2592 \clubpenalty \@clubpenalty
2593 \everypar{\everyparhook}%
2594 \fi\everyparhook}}

```

`\@gnewline` についてはちょっと複雑な心境です。もともとの pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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。

```

2595 \def\@gnewline #1{%
2596   \ifvmode
2597     \@nolnerr
2598   \else
2599     \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
2600     \jsInhibitGlue \ignorespaces
2601   \fi}

```

## 11 いろいろなロゴ

---

とりあえず削除。

---

## 12 amsmath との衝突の回避

---

最近の `\LaTeX` では該当の問題は対処されているので削除。

---

## 13 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 2602 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2603 \newcommand{\postpartname}{\if@english\else 部\fi}
                2604 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2605 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2606 \newcommand{\presectionname}{}% 第
\postsectionname 2607 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 2608 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename  2609 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
                2610 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 2611 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2612 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}

```

```

2613 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 2614 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
2615 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname
\abstractname 2616 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
2617 \newcommand{\appendixname}{\if@english \else 付録\fi}
2618 %<!book&!report>\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。

```

\today
2619 \@tempwafalse
2620 \if p\jsEngine \@tempwattrue \fi
2621 \if n\jsEngine \@tempwattrue \fi
2622 \if@tempswa \expandafter\@firstoftwo
2623 \else \expandafter\@secondoftwo
2624 \fi
2625 {%
2626 % 欧文 8bitTeX の場合
2627 \newif\ifjsSeireki \jsSeirekitrue
2628 \def\西暦{\jsSeirekitrue}
2629 \def\和暦{\jsSeirekifalse}
2630 \def\Seireki{\jsSeirekitrue}
2631 \def\Wareki{\jsSeirekifalse}
2632 \def\bxjs@if@use@seireki{%
2633 \ifjsSeireki \expandafter\@firstoftwo
2634 \else \expandafter\@secondoftwo \fi}
2635 }{%
2636 \newif\if西暦 \西暦 true
2637 \def\西暦{\西暦 true}
2638 \def\和暦{\西暦 false}
2639 \def\Seireki{\西暦 true}
2640 \def\Wareki{\西暦 false}
2641 \def\bxjs@if@use@seireki{%
2642 \if西暦 \expandafter\@firstoftwo
2643 \else \expandafter\@secondoftwo \fi}
2644 }
2645 % \bxjs@unxp
2646 \let\bxjs@unxp\@firstofone
2647 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2648 % \bxjs@iai
2649 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2650 \def\bxjs@iai{\noexpand~}
2651 \else \def\bxjs@iai{}

```

```

2652 \fi
2653 % \heisei
2654 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2655 % \today
2656 \edef\today{%
2657   \if@english
2658     \ifcase\month\or
2659       January\or February\or March\or April\or May\or June\or
2660       July\or August\or September\or October\or November\or December\fi
2661     \space\number\day, \number\year
2662   \else
2663     \noexpand\bxjs@if@use@seireki{%
2664       \number\year\bxjs@iai\bxjs@unxp{年}%
2665       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2666       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2667     }{%
2668       \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2669       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2670       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2671     }%
2672   \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）

```
2673 \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。

```

2674 %<slide>\pagestyle{empty}%
2675 %<article|report>\pagestyle{plain}%
2676 %<book>\pagestyle{headings}%
2677 \pagenumbering{arabic}
2678 \if@twocolumn
2679   \twocolumn
2680   \sloppy
2681   \flushbottom
2682 \else
2683   \onecolumn
2684   \raggedbottom
2685 \fi
2686 %<*slide>
2687 \renewcommand\familydefault{\sfdefault}
2688 \raggedright
2689 %</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。

```
2690 \catcode`\?=12
2691 \ifx\bxjs@jadriver\relax\else
2692 \input{bxjsja-\bxjs@jadriver.def}
2693 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
2694 \bxjs@restore@jltrcc
2695 %</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」はこの設定と辻褃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。

```
2696 %<*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

```
2697 %<*minimal>
```

```
2698 %% 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.
```

```
2699 \def\DeclareJaTextFontCommand#1#2{%
```

```
2700   \DeclareRobustCommand#1[1]{%
```

```
2701     \relax
```

```
2702     \ifmmode \expandafter\nfss@text \fi
```

```
2703     {#2##1}}%
```

```
2704 }
```

```
\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。
```

```
2705 \long\def\bxjs@@CSsfdefault{\sfdefault}%
```

```
2706 \@onlypreamble\bxjs@if@sf@default
```

```
2707 \def\bxjs@if@sf@default#1{%
```

```
2708   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
```

```
2709   \AtBeginDocument{%
```

```
2710     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
```

```
2711 }
```

`\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。

```
2712 \def\jsLetHeadChar#1#2{%
2713   \begingroup
2714     \escapechar=`\ %
2715     \let\bxjs@tmpa={% brace-match-hack
2716       \bxjs@let@hchar@exp#2}%
2717   \endgroup
2718   \let#1\bxjs@g@tmpa}
2719 \def\bxjs@let@hchar@exp{%
2720   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2721 \def\bxjs@let@hchar@exp@a{%
2722   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
2723     \bxjs@let@hchar@out\let\relax
2724   }{\bxjs@cond@ifcat\noexpand\@let@token@sptoken\fi{% 空白
2725     \bxjs@let@hchar@out\let\space%
2726   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
2727     \bxjs@let@hchar@out\let\@backslashchar
2728   }{\bxjs@let@hchar@exp@b}}}}
2729 \def\bxjs@let@hchar@exp@b#1{%
2730   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2731 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2732   %\message{<#1#2>}%
2733   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴
2734     \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2735       \bxjs@let@hchar@out\let\relax
2736     }{%else
2737       \expandafter\bxjs@let@hchar@exp
2738     }%
2739   }{%else
2740     \bxjs@let@hchar@chr#1%
2741   }}
2742 \def\bxjs@let@hchar@chr#1{%
2743   \bxjs@let@hchar@out\def{#1}}
2744 \def\bxjs@let@hchar@out#1#2{%
2745   \global#1\bxjs@g@tmpa#2\relax
2746   \toks@\bgroup}% skip to right brace

UTF-8 のバイト列を扱うコード。
2747 \chardef\bxjs@let@hchar@csta=128
2748 \chardef\bxjs@let@hchar@cstb=192
2749 \chardef\bxjs@let@hchar@cstc=224
```

```

2750 \chardef\bxjs@let@hchar@cstd=240
2751 \chardef\bxjs@let@hchar@cste=248
2752 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2753 \def\bxjs@let@hchar@chr@ue#1{%
2754   \@tempcnta=#1\relax
2755   \%message{\the\@tempcnta}%
2756   \bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2757     \bxjs@let@hchar@chr@ue@a#1%
2758   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2759     \bxjs@let@hchar@out\let\relax
2760   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2761     \bxjs@let@hchar@chr@ue@b
2762   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2763     \bxjs@let@hchar@chr@ue@c
2764   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2765     \bxjs@let@hchar@chr@ue@d
2766   }{%else
2767     \bxjs@let@hchar@out\let\relax
2768   }}}}
2769 \def\bxjs@let@hchar@chr@ue@a#1{%
2770   \bxjs@let@hchar@out\def{#1}}
2771 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2772   \bxjs@let@hchar@out\def{#1#2}}
2773 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2774   \bxjs@let@hchar@out\def{#1#2#3}}
2775 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2776   \bxjs@let@hchar@out\def{#1#2#3#4}}

```

## B.2 (u)pTeX 用の設定

```
2777 \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。

```

2778 \def\bxjs@let@hchar@chr@pp#1{%
2779   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
2780 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2781   \%message{(#1)}%
2782   \bxjs@cond@if#1t\fi{%
2783     \bxjs@let@hchar@chr@ue#3%
2784   }{%else
2785     \bxjs@let@hchar@out\def{#3}}%
2786   }}
2787 \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。

```
2788 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
```

```
2789 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
```

```
2790 \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。

```
2791 \@onlypreamble\bxjs@declarefontshape
```

```
2792 \ifjsWithupTeX
```

```
2793 \def\bxjs@declarefontshape{%
```

```
2794 \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
```

```
2795 \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
```

```
2796 \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
```

```
2797 \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
```

```
2798 }
```

```
2799 \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。

```
2800 \else
```

```
2801 \def\bxjs@declarefontshape{%
```

```
2802 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
```

```
2803 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
```

```
2804 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
```

```
2805 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
```

```
2806 }
```

```
2807 \def\bxjs@sizereference{jis}
```

```
2808 \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。

```
2809 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
```

```
2810 \def\bxjs@y{#5}}
```

```
2811 \ifjsWithpTeXng \def\bxjs@y{10}%
```

```
2812 \else
```

```
2813 \expandafter\expandafter\expandafter\bxjs@tmpa
```

```
2814 \expandafter\string\the\jfont\relax
```

```
2815 \fi
```

```
2816 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
```

```
2817 \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
```

```
2818 {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
```

```
2819 \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。

```

2820 \begingroup
2821 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2822 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2823 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2824 % 幅が丁度 10pt なら補正は不要
2825 \ifdim\wd\z@=10pt
2826 \global\let\bxjs@scale\jsScale
2827 \else
2828 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
2829 \edef\bxjs@tmpa{\strip@pt\wd\z@}
2830 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
2831 \bxjs@invscale\@tempdima\bxjs@tmpa
2832 \xdef\bxjs@scale{\strip@pt\@tempdima}
2833 \fi
2834 \endgroup
2835 %\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。

```

2836 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

2837 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2838 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
2839 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
2840 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
2841 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
2842 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
2843 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
2844 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2845 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
2846 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
2847 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
2848 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
2849 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
2850 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}

```

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

```

2851 \DeclareRobustCommand\rmfamily
2852 {\not@math@alphabet\rmfamily\mathrm}
2853 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}

```

```

2854 \DeclareRobustCommand\sffamily
2855   {\not@math@alphabet\sffamily\mathsf}
2856   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
2857 \DeclareRobustCommand\ttfamily
2858   {\not@math@alphabet\ttfamily\mathtt}
2859   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
2860 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
2861 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
2862 \bxjs@if@sf@default{%
2863   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```
2864 \selectfont
```

#### ■パラメタの設定

```

2865 \prebreakpenalty\jis"2147=10000
2866 \postbreakpenalty\jis"2148=10000
2867 \prebreakpenalty\jis"2149=10000
2868 \inhibitxspcode`!=1
2869 \inhibitxspcode`¯=2
2870 \xspcode`+=3
2871 \xspcode`%=3

```

"80～"FF の範囲の \spcode を 3 に変更。

```

2872 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
2873   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

\jsInhibitGlueAtParTop の定義。

```
2874 \let\jsInhibitGlueAtParTop\@inhibitglue
```

\jsResetDimen は空のままよい。

■組方向依存の処理 組方向判定の if-トークン (`\if?dir`) は p<sub>T</sub>E<sub>X</sub> 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。

```

2875 \begingroup
2876 \catcode`\!=0

```

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

```

2877 \gdef\bxjs@ptex@dir{%
2878   !iftdir t%
2879   !else!ifydir y%
2880   !else ?%
2881   !fi!fi}

```

新版の p<sub>T</sub>E<sub>X</sub> 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 p<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 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。

```
2882 % 古い \@makefnmark の定義
```

```

2883 \long\def\bxjs@tmpa{\hbox{%
2884 !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
2885 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
2886 \ifx\@makefnmark\bxjs@tmpa
2887 \long\gdef\@makefnmark{%
2888 !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
2889 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
2890 \fi
2891 \endgroup

```

### B.3 pdfTeX 用の処理

```

2892 \else\ifx p\jsEngine
2893 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
2894 \@onlypreamble\bxjs@cjk@loaded
2895 \def\bxjs@cjk@loaded{%
2896 \def\@footnotemark{%
2897 \leavevmode
2898 \ifhmode
2899 \edef\@x@sf{\the\spacefactor}%
2900 \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
2901 \unkern\unkern
2902 \ifdim\lastskip>\z@ \unskip \fi
2903 \fi\fi
2904 \nobreak
2905 \fi
2906 \@makefnmark
2907 \ifhmode \spacefactor\@x@sf \fi
2908 \relax}%
2909 \let\bxjs@cjk@loaded\relax
2910 }
2911 \AtBeginDocument{%
2912 \@ifpackageloaded{CJK}{%
2913 \bxjs@cjk@loaded
2914 }{}%
2915 }

```

### B.4 XeTeX 用の処理

```

2916 \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。

```

2917 \def\bxjs@let@hchar@chr#1{%
2918 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
2919 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
2920 \bxjs@let@hchar@chr@xe
2921 }{\bxjs@let@hchar@out\def{{#1}}}}
2922 \def\bxjs@let@hchar@chr@xe#1{%

```

```

2923 \lccode`0=`#1\relax
2924 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

2925 \ifx\XeTeXgenerateactualtext\@undefined\else
2926 \def\bxjs@do@precisetext{%
2927 \XeTeXgenerateactualtext=\@ne}
2928 \fi

```

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

```

2929 \@onlypreamble\bxjs@do@simplejasetup
2930 \def\bxjs@do@simplejasetup{%
2931 \ifnum\XeTeXinterchartokenstate>\z@
2932 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
2933 \jsSimpleJaSetup
2934 \ClassInfo\bxjs@clsname
2935 {\string\jsSimpleJaSetup\space is applied@gobble}%
2936 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```

2937 \newcommand*{\jsSimpleJaSetup}{%
2938 \XeTeXlinebreaklocale "ja"\relax
2939 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
2940 \XeTeXlinebreakpenalty=0\relax}

```

## B.5 後処理 (エンジン共通)

```

2941 \fi\fi\fi

```

`simplejasetup` オプションの処理。

```

2942 \ifx\bxjs@do@simplejasetup\@undefined\else
2943 \AtBeginDocument{%
2944 \ifbxjs@simplejasetup
2945 \bxjs@do@simplejasetup
2946 \fi}
2947 \fi

```

`precisetext` オプションの処理。

```

2948 \ifbxjs@precisetext
2949 \ifx\bxjs@do@precisetext\@undefined
2950 \ClassWarning\bxjs@clsname
2951 {The current engine does not support the\MessageBreak
2952 'precisetext' option@gobble}
2953 \else
2954 \bxjs@do@precisetext
2955 \fi
2956 \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。

2957 `\ifbxjs@fancyhdr`

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

2958 `\@onlypreamble\bxjs@adjust@fancyhdr`

2959 `\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 `\s1` は無い方がよいはず。

2960 `\def\bxjs@tmpa{\fancyplain{}{\s1\rightmark}\strut}%`

2961 `\def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%`

2962 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

2963 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

2964 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

2965 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

2966 `\def\bxjs@tmpa{\fancyplain{}{\s1\leftmark}\strut}%`

2967 `\def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%`

2968 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

2969 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

2970 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

2971 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

2972 `\def\bxjs@tmpa{\rm\thepage\strut}%`

2973 `\def\bxjs@tmpb{\thepage\strut}%`

2974 `\ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi`

2975 `\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。

2976 `\ifx\fullwidth@undefined\else \ifdim\textwidth<\fullwidth`

2977 `\setlength{\@tempdima}{\fullwidth-\textwidth}%`

2978 `\edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%`

2979 `}\bxjs@tmpa`

2980 `\fi\fi`

2981 `\PackageInfo\bxjs@clsname`

2982 `{Patch to fancyhdr is applied\@gobble}}`

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

2983 `\def\bxjs@pagestyle@hook{%`

2984 `\@ifpackageloaded{fancyhdr}{%`

2985 `\bxjs@adjust@fancyhdr`

2986 `\global\let\bxjs@adjust@fancyhdr\relax`

2987 `}{}}`

`\pagestyle` にフックを入れ込む。

2988 `\let\bxjs@org@pagestyle\pagestyle`

```

2989 \def\pagestyle{%
2990   \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。
2991 \AtBeginDocument{%
2992   \bxjs@pagestyle@hook
2993   \global\let\bxjs@pagestyle@hook\relax}

終わり。
2994 \fi

以上で終わり。
2995 %</minimal>

```

## 付録 C 和文ドライバ : standard

標準のドライバ。

- \rmfamily/\sffamily/\ttfamily での和文ファミリー連動
- \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“ファミリー”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ーを表す。

### C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```

2996 %<*standard>
2997 %% このファイルは日本語文字を含みます
2998 \input{bxjsja-minimal.def}

```

`simplejasetup` は `standard` では無効になる。

```

2999 \bxjs@simplejasetupfalse

```

■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  $\epsilon$ -TeX 拡張が使えるか  
検査する。

```
3000 \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。

```
\jH3001 \@tempdima=0.25mm
3002 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3003 \let\jH\jQ
```

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

```
\trueH3004 \ifjsc@mag
3005 \@tempdimb=\jsBaseFontSize\relax
3006 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3007 \@tempdima=2.5mm
3008 \bxjs@invscale\@tempdima\bxjs@tmpa
3009 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3010 \@tempdima=10pt
3011 \bxjs@invscale\@tempdima\bxjs@tmpa
3012 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3013 \else \let\trueQ\jQ \let\bxjs@truept\p@
3014 \fi
3015 \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。

```
3016 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3017 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3018 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3019 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3020 \fi
```

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。

```
3021 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3022 \newcommand*\setkanjiskip[1]{%
3023 \edef\bxjs@kanjiskip{#1}%
3024 \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3025 \newcommand*\getkanjiskip{%
3026 \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` で制御を用いるのでこの変数は常に真とする。

```
3027 \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 3028 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3029   \bxjs@kanjiskip@enabledtrue
3030   \bxjs@reset@kanjiskip}
3031 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3032   \bxjs@kanjiskip@enabledfalse
3033   \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。

```
3034 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3035   \ifbxjs@kanjiskip@enabled
3036     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3037   \else \@tempskipa\z@
3038   \fi
3039   \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 3040 \def\bxjs@xkanjiskip{Opt}
\getxkanjiskip 3041 \newcommand*\setxkanjiskip[1]{%
```

```
3042   \edef\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3043   \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3044 \newcommand*\getxkanjiskip{%
3045   \bxjs@xkanjiskip}
\bxjs@disable@xkanjiskip 3046 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3047 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3048   \bxjs@xkanjiskip@enabledtrue
3049   \bxjs@reset@xkanjiskip}
3050 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3051   \bxjs@xkanjiskip@enabledfalse
3052   \bxjs@reset@xkanjiskip}
3053 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3054   \ifbxjs@xkanjiskip@enabled
3055     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3056   \else \@tempskipa\z@
3057   \fi
3058   \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。

```
3059 \g@addto@macro\jsResetDimen{%
3060   \bxjs@reset@kanjiskip
3061   \bxjs@reset@xkanjiskip}
3062 \let\bxjs@apply@kanjiskip\relax
3063 \let\bxjs@apply@xkanjiskip\relax
```

■和文フォント指定の扱い

`\bxjs@adjust@jafont` ムニヤムニヤ…。

```

3064 \@onlypreamble\bxjs@adjust@jafont
3065 \def\bxjs@adjust@jafont#1{%
3066   \ifx\jsJaFont\bxjs@@auto
3067     \bxjs@get@kanjiEmbed
3068     \ifx\bxjs@kanjiEmbed\relax
3069       \let\bxjs@tmpa\@empty
3070     \else
3071       \let\bxjs@tmpa\bxjs@kanjiEmbed
3072     \fi
3073   \else
3074     \let\bxjs@tmpa\jsJaFont
3075   \fi
3076   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3077     \ClassWarning\bxjs@clsname
3078     {Option 'jafont=noEmbed' is ignored, because no-embed\MessageBreak
3079     is not available}%
3080     \let\bxjs@tmpa\@empty
3081   \fi\fi
3082 }
3083 \def\bxjs@@auto{auto}
3084 \def\bxjs@@noEmbed{noEmbed}

```

`\bxjs@kanjiEmbed` 現在の updmap の kanjiEmbed パラメタの値。`\bxjs@get@kanjiEmbed` により設定される。

```

3085 \let\bxjs@kanjiEmbed\relax

```

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

```

3086 \@onlypreamble\bxjs@get@kanjiEmbed
3087 \def\bxjs@get@kanjiEmbed{%
3088   \begingroup\setbox\z@=\hbox{%
3089     \global\let\bxjs@g@tmpa\relax
3090     \endlinechar\m@ne
3091     \let\do\@makeother\dospecials
3092     \catcode32=10 \catcode12=10 %form-feed
3093     \let\bxjs@tmpa\@empty
3094     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3095     \ifeof\@inputcheck\else
3096       \read\@inputcheck to\bxjs@tmpa
3097       \closein\@inputcheck
3098     \fi
3099     \ifx\bxjs@tmpa\@empty\else
3100       \openin\@inputcheck="\bxjs@tmpa"\relax
3101       \@tempwattrue
3102       \loop\if@tempswa
3103         \read\@inputcheck to\bxjs@tmpa
3104         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil

```

```

3105     \ifx\bxjs@tmpa\relax\else
3106     \global\let\bxjs@g@tmpa\bxjs@tmpa
3107     \@tempwafalse
3108     \fi
3109     \ifeof\@inputcheck \@tempwafalse \fi
3110     \repeat
3111     \fi
3112 } \endgroup
3113 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3114 }
3115 \@onlypreamble\bxjs@get@ke@a
3116 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3117   \ifx$#1$\def\bxjs@tmpa{#2}%
3118   \else \let\bxjs@tmpa\relax
3119   \fi}

```

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

```

3120 \newcommand*\jachar[1]{%
3121   \begingroup
3122     \jsLetHeadChar\bxjs@tmpa{#1}%
3123     \ifx\bxjs@tmpa\relax
3124       \ClassWarningNoLine\bxjs@clsname
3125       {Illegal argument given to \string\jachar}%
3126     \else
3127       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3128     \fi
3129   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3130 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3131 \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。

```

3132 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

`\bxjs@fix@hyperref@unicode` `hyperref` の `unicode` オプションの値を固定する。

```

3133 \@onlypreamble\bxjs@fix@hyperref@unicode
3134 \def\bxjs@fix@hyperref@unicode#1{%
3135   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3136   \@namedef{KV@Hyp@bxjs/hook}##1{%
3137     \KV@Hyp@unicode{##1}%
3138     \def\KV@Hyp@unicode####1{%
3139       \expandafter\ifx\csname if##1\expandafter\endcsname
3140       \csname if####1\endcsname\else

```

```

3141     \ClassWarningNoLine\bxjs@clsname
3142     {Blcoked hyperref option 'unicode=###1'}%
3143     \fi
3144   }%
3145 }%
3146 }

```

`\bxjs@urgent@special` DVI のなるべく早い位置に `special` を出力する。

```

3147 \onlypreamble\bxjs@urgent@special
3148 \def\bxjs@urgent@special#1{%
3149   \AtBeginDvi{\special{#1}}%
3150   \AtBeginDocument{%
3151     \ifpackageloaded{atbegshi}{%
3152       \begingroup
3153         \toks\z@\special{#1}}%
3154         \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3155         \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3156       \endgroup
3157     }{}%
3158   }%
3159 }

```

## C.2 pTeX 用設定

```
3160 \if j\jsEngine
```

### ■ 共通命令の実装

```

3161 \def\bxjs@apply@kanjiskip{%
3162   \kanjiskip\@tempkipa}
3163 \def\bxjs@apply@xkanjiskip{%
3164   \xkanjiskip\@tempkipa}

```

`\jaJaChar` のサブマクロ。

```

3165 \def\bxjs@jachar#1{%
3166   \bxjs@jachar@a#1...\@nil}
3167 \def\bxjs@jachar@a#1#2#3#4#5\@nil{%

```

引数が単一トークンなら和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。

```
3168   \ifx.#2#1%
```

引数が複数トークンの場合は、UTF-8 のバイト列であるの見なし、そのスカラー値を `\@tempcnta` に代入する。

```

3169   \else\ifx.#3%
3170     \@tempcnta`#1 \multiply\@tempcnta64
3171     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3172     \bxjs@jachar@b
3173   \else\ifx.#4%
3174     \@tempcnta`#1 \multiply\@tempcnta64
3175     \advance\@tempcnta`#2 \multiply\@tempcnta64

```

```

3176 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3177 \bxjs@jachar@b
3178 \else
3179 \@tempcnta`#1 \multiply\@tempcnta64
3180 \advance\@tempcnta`#2 \multiply\@tempcnta64
3181 \advance\@tempcnta`#3 \multiply\@tempcnta64
3182 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3183 \bxjs@jachar@b
3184 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3185 \ifjsWithupTeX
3186 \def\bxjs@jachar@b{\kchar\@tempcnta}
3187 \else
3188 \def\bxjs@jachar@b{%
3189 \ifx\bxUInt\@undefined\else
3190 \bxUInt{\@tempcnta}%
3191 \fi}
3192 \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。

```

3193 \let\bxjs@tmpa\jsJaFont
3194 \ifx\bxjs@tmpa\bxjs@@auto
3195 \let\bxjs@tmpa\@empty
3196 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3197 \def\bxjs@tmpa{noembed}
3198 \fi\fi
3199 \ifx\jsJaFont\@empty\else
3200 \edef\bxjs@nxt{%
3201 \noexpand\RequirePackage[\jsJaFont]
3202 {pxchfon}[2010/05/12]}% v0.5
3203 \bxjs@nxt
3204 \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……。)

```

3205 \begingroup
3206 \global\let\@gtempa\relax
3207 \catcode`\|=0 \catcode`\|=12
3208 |def|bxjs@check#1|@nil{%
3209 |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3210 |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3211 |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%

```

```

3212 |catcode`\|=0 \catcode`\|=12
3213 \def\bxjs@check@b#1keyval#2\@nnil{%
3214   \ifx$#2$\else
3215     \xdef\@gtempa{%
3216       \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3217     \fi}
3218 \@firstofone{%
3219   \catcode10=12 \endlinechar\m@ne
3220   \let\do\@makeother \dospecials \catcode32=10
3221   \openin\@inputcheck=otf.sty\relax
3222   \@tempswatruer
3223   \loop\if@tempswa
3224     \ifeof\@inputcheck \@tempswafalse \fi
3225     \if@tempswa
3226       \read\@inputcheck to\bxjs@line
3227       \expandafter\bxjs@check\bxjs@line\@nil
3228     \fi
3229   \repeat
3230   \closein\@inputcheck
3231 \endgroup}
3232 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3233 \bxjs@fix@hyperref@unicode{false}

```

tounicode special 命令を出力する。

```

3234 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3235   \else\ifjsWithpTeXng T\else F\fi\fi T%
3236   \ifnum\jis"2121="A1A1 %euc
3237     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3238   \else\ifnum\jis"2121="8140 %sjis
3239     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3240   \else\ifnum\jis"2121="3000 %uptex
3241     \ifbxjs@bigcode
3242       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3243       \PassOptionsToPackage{bigcode}{pxjahyper}
3244     \else
3245       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3246     \fi
3247   \fi\fi\fi
3248   \let\bxToUnicodeSpecialDone=t
3249 \fi

```

■microtype 対策

```

3250 \@namedef{ver@microtype.sty}{2000/01/01}
3251 \newcommand*\UseMicrotypeSet [2] [] {}

```

### C.3 pdfTeX 用設定 : CJK + bxcjkjatype

```
3252 \else\if p\jsEngine
```

■bxcjkjatype 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 bxcjkjatype 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 bxcjkjatype の側で行われる。

```
3253 \bxjs@adjust@jafont{f}
```

```
3254 \edef\bxjs@nxt{%
```

```
3255 \noexpand\RequirePackage[%
```

```
3256 \ifx\bxjs@tmpa\@empty\else \bxjs@tmpa,\fi
```

```
3257 whole,autotilde]{bxcjkjatype}[2013/10/15]}% v0.2c
```

```
3258 \bxjs@nxt
```

```
3259 \bxjs@cjk@loaded
```

■hyperref 対策 bxcjkjatype 使用時は unicode にするべき。

※取りあえず固定はしない。

```
3260 \PassOptionsToPackage{unicode}{hyperref}
```

\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。

※ bxcjkjatype 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。

```
3261 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
```

```
3262 \begingroup
```

```
3263 \CJK@input{UTF8.bdg}
```

```
3264 \endgroup
```

```
3265 \g@addto@macro\pdfstringdefPreHook{%
```

```
3266 \@nameuse{CJK@UTF8Binding}%
```

```
3267 }
```

```
3268 \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。

```
3269 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
```

```
3270 \g@addto@macro\pdfstringdefPreHook{%
```

```
3271 \ifx~\bxjs@@CJKtilde
```

```
3272 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
```

```
3273 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
```

```
3274 \let~\@empty
```

```
3275 \fi
```

```
3276 }
```

```
3277 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
```

```
3278 \def\bxjs@@tildecmd{~}
```

```
3279 \def\bxjs@LetUnexpandableSpace#1{%
```

```
3280 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
```

```

3281 \bxjs@org@LetUnexpandableSpace#1%
3282 \fi}
3283 \fi

```

#### ■ 共通命令の実装

```

3284 \newskip\jsKanjiSkip
3285 \newskip\jsXKanjiSkip
3286 \ifx\CJKecglue@undefined
3287 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3288 \fi
3289 \let\autospacing\bxjs@enable@kanjiskip
3290 \let\noautospacing\bxjs@disable@kanjiskip
3291 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3292 \def\bxjs@apply@kanjiskip{%
3293 \jsKanjiSkip\@tempskipa
3294 \let\CJKglue\bxjs@CJKglue}
3295 \let\autoxspacing\bxjs@enable@xkanjiskip
3296 \let\noautoxspacing\bxjs@disable@xkanjiskip
3297 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3298 \def\bxjs@apply@xkanjiskip{%
3299 \jsXKanjiSkip\@tempskipa
3300 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

3301 \def\bxjs@jachar#1{%
3302 \CJKforced{#1}}

```

## C.4 Xe<sub>La</sub>TeX 用設定 : xeCJK + zxjatype

```

3303 \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。

```

3304 \RequirePackage{zxjatype}
3305 \PassOptionsToPackage{no-math}{fontspec}%!
3306 \PassOptionsToPackage{xetex}{graphicx}%!
3307 \PassOptionsToPackage{xetex}{graphics}%!
3308 \ifx\zxJaFamilyName@undefined
3309 \ClassError\bxjs@clsname
3310 {xeCJK or zxjatype is too old}\@ehc
3311 \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。

```

3312 \bxjs@adjust@jafont{f}
3313 \ifx\bxjs@tmpa@empty
3314 \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3315 \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3316 \else

```

```

3317 \edef\bxjs@nxt{%
3318   \noexpand\RequirePackage[\bxjs@tmpa]%
3319     {zxjafont}[2013/01/28]}% v0.2a
3320 \bxjs@nxt
3321 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X<sub>Ǝ</sub>TeX 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>TeX 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>TeX 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。

```

3322 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3323   \PassOptionsToPackage{unicode}{hyperref}
3324 \fi

```

■段落頭でのグルー挿入禁止

```

3325 \let\jsInhibitGlueAtParTop\@inhibitglue

```

■共通命令の実装

```

3326 \newskip\jsKanjiSkip
3327 \newskip\jsXKanjiSkip
3328 \ifx\CJKecglue\@undefined
3329   \def\CJKtilde{\CJKglobal\def~{\CJKecglue\ignorespaces}}
3330 \fi
3331 \let\autospacing\bxjs@enable@kanjiskip
3332 \let\noautospacing\bxjs@disable@kanjiskip
3333 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3334 \def\bxjs@apply@kanjiskip{%
3335   \jsKanjiSkip\@tempskipa
3336   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3337 \let\autoxspacing\bxjs@enable@xkanjiskip
3338 \let\noautoxspacing\bxjs@disable@xkanjiskip
3339 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3340 \def\bxjs@apply@xkanjiskip{%
3341   \jsXKanjiSkip\@tempskipa
3342   \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。

```

3343 \ifx\mcfamily\@undefined
3344   \protected\def\mcfamily{\CJKfamily{\CJKrmddefault}}

```

```

3345 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3346 \fi

    \jachar のサブマクロの実装。
3347 \def\bxjs@jachar#1{%
3348 \xeCJKDeclareCharClass{CJK}{`#1}\relax
3349 #1}

```

## C.5 LuaTeX 用設定 : LuaTeX-ja

```
3350 \else\if 1\jsEngine
```

■ LuaTeX-ja パッケージの読み込 `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。

```

3351 \let\zw\undefined
3352 \RequirePackage{luatexja}
3353 \RequirePackage{luatexja-fontspec}
3354 \PassOptionsToPackage{pdftex}{graphicx}%!
3355 \PassOptionsToPackage{pdftex}{graphics}%!

```

■ 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？

```

3356 \ExplSyntaxOn
3357 \fp_gset:Nn \g_ltj_fontspec_scale_fp { \jsScale }
3358 \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)。

```

3359 \bxjs@adjust@jafont{t}
3360 \ifx\bxjs@tmpa\bxjs@noEmbed
3361 \def\bxjs@tmpa{noembed}
3362 \fi
3363 \ifx\bxjs@tmpa@empty
3364 \defaultfontfeatures{ Kerning=Off }
3365 \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3366 \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3367 \else
3368 \edef\bxjs@nxt{%
3369 \noexpand\RequirePackage[\bxjs@tmpa]
3370 {luatexja-preset}}%
3371 \bxjs@nxt
3372 \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。

```
3373 \DeclareRobustCommand\rmfamily
3374   {\not@math@alphabet\rmfamily\mathrm
3375     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3376 \DeclareRobustCommand\sffamily
3377   {\not@math@alphabet\sffamily\mathsf
3378     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3379 \DeclareRobustCommand\ttfamily
3380   {\not@math@alphabet\ttfamily\mathtt
3381     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3382 \AtBeginDocument{%
3383   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3384   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}}%
3385 \bxjs@if@sf@default{%
3386   \renewcommand\kanjifamilydefault{\gtdefault}}
```

#### ■和文パラメタの設定

```
3387 % 次の3つは既定値の通り
3388 %\ltjsetparameter{prebreakpenalty={` ,10000}}
3389 %\ltjsetparameter{postbreakpenalty={` “ ,10000}}
3390 %\ltjsetparameter{prebreakpenalty={` ” ,10000}}
3391 \ltjsetparameter{jaxspmde={` ! ,1}}
3392 \ltjsetparameter{jaxspmde={` 〒 ,2}}
3393 \ltjsetparameter{alxspmde={` + ,3}}
3394 \ltjsetparameter{alxspmde={` \%,3}}
```

#### ■段落頭でのグルー挿入禁止

```
3395 \protected\def\@inhibitglue{%
3396   \directlua{%
3397     luatexja.jfmglue.create_beginpar_node()}}
3398 \let\bxjs@ltj@inhibitglue\@inhibitglue
3399 \let\@@inhibitglue\@undefined
```

#### ■hyperref 対策 unicode にするべき。

```
3400 \bxjs@fix@hyperref@unicode{true}
```

#### ■共通命令の実装

```
3401 \protected\def\autospadding{%
3402   \ltjsetparameter{autospadding=true}}
3403 \protected\def\noautospadding{%
3404   \ltjsetparameter{autospadding=false}}
3405 \protected\def\autoxspacing{%
3406   \ltjsetparameter{autoxspacing=true}}
3407 \protected\def\noautoxspacing{%
3408   \ltjsetparameter{autoxspacing=false}}
3409 \def\bxjs@apply@kanjiskip{%
3410   \ltjsetparameter{kanjiskip={\@tempkipa}}}
```

```

3411 \def\bxjs@apply@xkanjiskip{%
3412   \ltjsetparameter{xkanjiskip={\@tempkipa}}}}

```

\jachar のサブマクロの実装。

```

3413 \def\bxjs@jachar#1{%
3414   \ltjjachar`#1\relax}

```

## C.6 共通処理 (2)

```
3415 \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 3416 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3417 \DeclareJaTextFontCommand{\textgt}{\gtfamily}

```

### ■和文・和欧文間空白の初期値

```

3418 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3419 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3420 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3421 \fi

```

以上で終わり。

```
3422 %</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。

```

3423 %<*modern>
3424 \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。

```

3425 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3426 \def\encodingdefault{T1}%
3427 \input{t1enc.def}%
3428 \fontencoding\encodingdefault\selectfont
3429 \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。

```

3430 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3431 \renewcommand{\rmdefault}{\lrm}
3432 \renewcommand{\sfdefault}{\lms}
3433 \renewcommand{\ttdefault}{\lmtt}
3434 \fi

```

大型演算子用の数式フォントの設定。

※ `amsmath` パッケージと同等にする。

```

3435 \DeclareFontShape{OMX}{cmex}{m}{n}{%
3436   <-7.5>cmex7<7.5-8.5>cmex8%
3437   <8.5-9.5>cmex9<9.5->cmex10}{}%
3438 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

```

`amsmath` 読み込時に上書きされるのを防ぐ。

```

3439 \def\cmex@opt{10}

```

## D.2 fixltx2e 読み込

※ `fixltx2e` 廃止前の  $\text{\LaTeX}$  カーネルの場合。

```

3440 \ifx\@IncludeInRelease\@undefined
3441 \RequirePackage{fixltx2e}
3442 \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。

```

3443 \RequirePackage{bxjcsjkat}

```

## D.4 完了

おしまい。

```

3444 %</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。

```

3445 %<*pandoc>
3446 \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。

```
3447 \@onlypreamble\bxjs@set@dupload@proc
3448 \def\bxjs@set@dupload@proc#1{%
3449   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
3450 \@onlypreamble\bxjs@set@dupload@proc@a
3451 \def\bxjs@set@dupload@proc@a#1{%
3452   \@onlypreamble#1\def#1##1}
```

`\@if@ptions` `\@if@ptions` の再定義。

```
3453 \@onlypreamble\bxjs@org@if@ptions
3454 \let\bxjs@org@if@ptions\@if@ptions
3455 \newif\ifbxjs@dlp
3456 \def\@if@ptions#1#2#3{%
3457   \bxjs@dlpfalse
3458   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
3459   \ifx\bxjs@tmpa\bxjs@tmpb
3460     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
3461     \bxjs@dlptrue \fi
3462   \fi
3463   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
3464   \else \expandafter\bxjs@org@if@ptions
3465   \fi {#1}{#2}{#3}}
3466 \AtBeginDocument{%
3467   \let\@if@ptions\bxjs@org@if@ptions}
3468 \@onlypreamble\bxjs@do@dupload@proc
3469 \def\bxjs@do@dupload@proc#1#2#3{%
3470   \csname bxjs@dlp/#2.#1\endcsname{#3}%
3471   \@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。

```
3472 \def\bxjs@mark@as@loaded#1{%
3473   \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3474 \def\bxjs@mal@a#1#2{%
3475   \ifx#1\relax
3476     \def#1{2001/01/01}%
3477     \ClassInfo\bxjs@clsname
3478     {File '#2' marked as loaded\@gobble}%
3479   \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。

```
3480 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0
```

Polyglossia について。

```
3481 \bxjs@mark@as@loaded{polyglossia.sty}
3482 \bxjs@set@dupload@proc{polyglossia.sty}{%
3483   \ClassWarning\bxjs@clsname
3484   {Loading of polyglossia is blocked}}
3485 \ifx\setmainlanguage\@undefined
3486 \newcommand*\setmainlanguage[2] [] {}
3487 \newcommand*\setotherlanguage[2] [] {%
3488   \ifcat_#2_ \else
3489     \expandafter\let\csname #2\endcsname\@empty
3490     \expandafter\let\csname end#2\endcsname\@empty
3491     \expandafter\let\csname text#2\endcsname\@firstofone
3492   \fi}
3493 \newcommand*\setotherlanguages[2] [] {%
3494   \@for\bxjs@tmpa:={#2}\do{%
3495     \setotherlangauge{\bxjs@tmpa}}}
3496 \fi
3497 \else
```

Babel について。

```
3498 \bxjs@mark@as@loaded{babel.sty}
3499 \bxjs@set@dupload@proc{babel.sty}{%
3500   \ClassWarning\bxjs@clsname
3501   {Loading of babel is blocked}}
3502 \let\foreignlanguage\@secondoftwo
3503 \let\otherlanguage\@gobble
3504 \let\endotherlanguage\@empty
3505 \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。

```
3506 \bxjs@set@dupload@proc{geometry.sty}{%
3507   \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 `\setmainjfont` の指定にまわす。

```
3508 \if l\jsEngine
3509   \bxjs@mark@as@loaded{xeCJK.sty}
3510   \providecommand*\setCJKmainfont{\setmainjfont}
3511 \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 `LATEX` ではこれで警告が出る。これを抑止する。

`LATEX` カーネルが新しい場合は `fixltx2e` を読込済にする。

```
3512 \ifx\@IncludeInRelease\@undefined\else
3513   \bxjs@mark@as@loaded{fixltx2e.sty}
3514 \fi
```

## E.6 cmap パッケージ

エンジンが `(u)pLATEX` 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。）

```
3515 \if j\jsEngine
3516   \bxjs@mark@as@loaded{cmap.sty}
3517 \fi
```

## E.7 microtype パッケージ

警告が多すぎなので消す。

```
3518 \PassOptionsToPackage{verbose=silent}{microtype}
```

## E.8 完了

おしまい。

```
3519 %</pandoc>
```

和文ドライバ実装はここまで。

```
3520 %</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。

```
3521 %<*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

```
3522 %<*compat>
3523 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
3524 \let\bxac@engine=n
3525 \def\bxac@do#1#2{%
3526   \edef\bxac@tmpa{\string#1}%
3527   \edef\bxac@tmpb{\meaning#1}%
3528   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
3529 \bxac@do\XeTeXversion{\let\bxac@engine=x}
3530 \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。

```
3531 \ifx\jsAtEndOfClass\undefined
3532   \let\bxac@delayed@if@bxjs\@firstofone
3533 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
3534 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 3535 \newif\ifbxac@in@old@behavior
3536 \let\ImposeOldLuaTeXBehavior\relax
3537 \let\RevokeOldLuaTeXBehavior\relax
```

## G.2 XeTeX 部分

```
3538 \ifx x\bxac@engine
```

XeTeX 文字クラスのみニャムニャ。

```
3539 \@onlypreamble\bxac@adjust@charclass
3540 \bxac@delayed@if@bxjs{%
3541   \@ifpackageloaded{xeCJK}{\}%else
3542     \ifx\xe@alloc@intercharclass\undefined\else
3543       \ifnum\xe@alloc@intercharclass=\z@
3544         \PackageInfo\bxac@pkgname
3545           {Setting up interchar class for CJK...\@gobble}%
3546         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
3547           \xe@alloc@intercharclass=3
3548         }{\%else
3549           \PackageWarning\bxac@pkgname
3550             {Cannot find file 'load-unicode-xetex-classes.tex'%
3551             \@gobble}%
3552         }%
3553   \fi\fi
3554   \ifnum\XeTeXcharclass"3041=\z@
3555     \PackageInfo\bxac@pkgname
```

```

3556     {Adjusting interchar class for CJK...\@gobble}%
3557     \@for\bxac@x:={%
3558         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3559         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3560         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3561         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3562         31FF%
3563     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
3564     \fi
3565 }%
3566 }

```

以上。

```
3567 \fi
```

### G.3 LuaTeX 部分

```
3568 \ifx l\bxac@engine
```

△ニヤムニヤ。

```

3569 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
3570   \chardef\pdftexversion=200
3571   \def\pdftexrevision{0}
3572   \let\pdftexbanner\luatexbanner
3573 \fi\fi

```

```
\ImposeOldLuaTeXBehavior △ニヤムニヤ。
```

```

\RevokeOldLuaTeXBehavior 3574 \begingroup\expandafter\expandafter\expandafter\endgroup
3575 \expandafter\ifx\csname outputmode\endcsname\relax\else
3576 \def\bxac@ob@list{%
3577   \do{\let}\pdfoutput{\outputmode}%
3578   \do{\let}\pdfpagewidth{\pagewidth}%
3579   \do{\let}\pdfpageheight{\pageheight}%
3580   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}%
3581   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}
3582 \def\bxac@ob@do#1#2{\begingroup
3583   \expandafter\bxac@ob@do@a\csname bxac@string#2\endcsname{#1}#2}
3584 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3585   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3586   \else \let#3#1\relax \let#1\@undefined
3587   \fi}
3588 \protected\def\ImposeOldLuaTeXBehavior{%
3589   \unless\ifbxac@in@old@behavior
3590     \bxac@in@old@behaviortrue
3591     \let\do\bxac@ob@do \bxac@ob@list
3592   \fi}
3593 \protected\def\RevokeOldLuaTeXBehavior{%
3594   \ifbxac@in@old@behavior
3595     \bxac@in@old@behaviorfalse
3596     \let\do\bxac@ob@do \bxac@ob@list
3597   \fi}

```

3598 \fi

漢字および完成形ハングルのカテゴリコードのムニャムニャ。

```
3599 \ifnum\luatexversion>64 \directlua{
3600   local function range(cs, ce, cc, ff)
3601     if ff or not tex.getcatcode(cs) == cc then
3602       local setcc = tex.setcatcode
3603       for c = cs, ce do setcc(c, cc) end
3604     end
3605   end
3606   range(0x3400, 0x4DB5, 11, false)
3607   range(0x4DB5, 0x4DBF, 11, true)
3608   range(0x4E00, 0x9FCC, 11, false)
3609   range(0x9FCD, 0x9FFF, 11, true)
3610   range(0xAC00, 0xD7A3, 11, false)
3611   range(0x20000, 0x2A6D6, 11, false)
3612   range(0x2A6D7, 0x2A6FF, 11, true)
3613   range(0x2A700, 0x2B734, 11, false)
3614   range(0x2B735, 0x2B73F, 11, true)
3615   range(0x2B740, 0x2B81D, 11, false)
3616   range(0x2B81E, 0x2B81F, 11, true)
3617   range(0x2B820, 0x2CEA1, 11, false)
3618   range(0x2CEA2, 0x2FFFD, 11, true)
3619 } \fi
```

以上。

3620 \fi

## G.4 完了

おしまい。

3621 %</compat>

## 付録 H 補助パッケージ : bxjcsjkat

modern ドライバ用の和文カテゴリを適用する。

### H.1 準備

```
3622 %<*cjkat>
3623 \def\bxjx@pkgname{bxjcsjkat}
3624 \newcount\bxjx@cnta
\bxjx@engine エンジンの種別。
3625 \let\bxjx@engine=n
3626 \def\bxjx@do#1#2{%
3627   \edef\bxjx@tmpa{\string#1}%
3628   \edef\bxjx@tmpb{\meaning#1}%
```

```

3629 \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3630 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3631 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3632 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3633 \bxjx@do\pdftexversion{\let\bxjx@engine=p}
3634 \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。

```

3635 \def\bxjx@do#1#2{%
3636   \if#1\bxjx@engine
3637     \ifpackageloaded{#2}{\%else
3638       \PackageError\bxjx@pkgname
3639         {Package '#2' must be loaded}%
3640         {Package loading is aborted.\MessageBreak\@ehc}%
3641     \endinput}
3642   \fi}
3643 \bxjx@do{p}{bxcjkjatype}
3644 \bxjx@do{x}{xeCJK}
3645 \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。

```

3646 \ifx\TextOrMath\@undefined
3647   \RequirePackage{fixltx2e}
3648 \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 Lua<sub>T</sub><sub>E</sub>X-ja と（ほぼ）等価なものに変更  
 する。

※ Lua<sub>T</sub><sub>E</sub>X-ja との相違点：A830、A960、1B000。

```

3649 \if u\bxjx@engine
3650 \@for\bxjx@x:={%
3651 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3652 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3653 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3654 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3655 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3656 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
3657 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3658 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
3659 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3660 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
3661 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3662 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3663 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3664 10480,10500,10530,10600,10800,10840,10860,10880,%
3665 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%

```

```

3666 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3667 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3668 11280,112B0,11300,11480,11580,11600,11680,11700,%
3669 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3670 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3671 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3672 1F030,1FOA0,1F100,1F200,1F300,1F600,1F650,1F680,%
3673 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3674 }\do{\kcatcode"\bxjx@x=15 }
3675 \fi

```

### 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。

```

3676 \def\bxjx@grkcyr@list{%
3677 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
3678 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
3679 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
3680 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
3681 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
3682 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
3683 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
3684 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
3685 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
3686 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
3687 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
3688 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
3689 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
3690 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
3691 \do{039F}{LGR}{\textOmicron}{O}%      % GR. C. L. OMICRON
3692 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
3693 \do{03A1}{LGR}{\textRho}{P}%            % GR. C. L. RHO
3694 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
3695 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
3696 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
3697 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
3698 \do{03A7}{LGR}{\textChi}{X}%           % GR. C. L. CHI
3699 \do{03A8}{LGR}{\textPsi}{\Psi}%         % GR. C. L. PSI
3700 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA
3701 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
3702 \do{03B2}{LGR}{\textbeta}{\beta}%       % GR. S. L. BETA
3703 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA
3704 \do{03B4}{LGR}{\textdelta}{\delta}%     % GR. S. L. DELTA
3705 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
3706 \do{03B6}{LGR}{\textzeta}{\zeta}%       % GR. S. L. ZETA

```

3707	$\backslash\text{do}\{03B7\}\{\text{LGR}\}\{\text{teta}\}\{\eta\}\%$	% GR. S. L. ETA
3708	$\backslash\text{do}\{03B8\}\{\text{LGR}\}\{\text{ttheta}\}\{\theta\}\%$	% GR. S. L. THETA
3709	$\backslash\text{do}\{03B9\}\{\text{LGR}\}\{\text{tiota}\}\{\iota\}\%$	% GR. S. L. IOTA
3710	$\backslash\text{do}\{03BA\}\{\text{LGR}\}\{\text{tkappa}\}\{\kappa\}\%$	% GR. S. L. KAPPA
3711	$\backslash\text{do}\{03BB\}\{\text{LGR}\}\{\text{tlambda}\}\{\lambda\}\%$	% GR. S. L. LAMDA
3712	$\backslash\text{do}\{03BC\}\{\text{LGR}\}\{\text{tmu}\}\{\mu\}\%$	% GR. S. L. MU
3713	$\backslash\text{do}\{03BD\}\{\text{LGR}\}\{\text{tnu}\}\{\nu\}\%$	% GR. S. L. NU
3714	$\backslash\text{do}\{03BE\}\{\text{LGR}\}\{\text{txi}\}\{\xi\}\%$	% GR. S. L. XI
3715	$\backslash\text{do}\{03BF\}\{\text{LGR}\}\{\text{tomicron}\}\{\omicron\}\%$	% GR. S. L. OMICRON
3716	$\backslash\text{do}\{03C0\}\{\text{LGR}\}\{\text{tppi}\}\{\pi\}\%$	% GR. S. L. PI
3717	$\backslash\text{do}\{03C1\}\{\text{LGR}\}\{\text{trho}\}\{\rho\}\%$	% GR. S. L. RHO
3718	$\backslash\text{do}\{03C2\}\{\text{LGR}\}\{\text{tvarsigma}\}\{\varsigma\}\%$	% GR. S. L. FINAL SIGMA
3719	$\backslash\text{do}\{03C3\}\{\text{LGR}\}\{\text{tsigma}\}\{\sigma\}\%$	% GR. S. L. SIGMA
3720	$\backslash\text{do}\{03C4\}\{\text{LGR}\}\{\text{ttau}\}\{\tau\}\%$	% GR. S. L. TAU
3721	$\backslash\text{do}\{03C5\}\{\text{LGR}\}\{\text{tupsilon}\}\{\upsilon\}\%$	% GR. S. L. UPSILON
3722	$\backslash\text{do}\{03C6\}\{\text{LGR}\}\{\text{tphi}\}\{\phi\}\%$	% GR. S. L. PHI
3723	$\backslash\text{do}\{03C7\}\{\text{LGR}\}\{\text{tchi}\}\{\chi\}\%$	% GR. S. L. CHI
3724	$\backslash\text{do}\{03C8\}\{\text{LGR}\}\{\text{tpsi}\}\{\psi\}\%$	% GR. S. L. PSI
3725	$\backslash\text{do}\{03C9\}\{\text{LGR}\}\{\text{tomega}\}\{\omega\}\%$	% GR. S. L. OMEGA
3726	$\backslash\text{do}\{0401\}\{\text{T2A}\}\{\text{CYRYO}\}\{\}\%$	% CY. C. L. IO
3727	$\backslash\text{do}\{0410\}\{\text{T2A}\}\{\text{CYRA}\}\{\}\%$	% CY. C. L. A
3728	$\backslash\text{do}\{0411\}\{\text{T2A}\}\{\text{CYRB}\}\{\}\%$	% CY. C. L. BE
3729	$\backslash\text{do}\{0412\}\{\text{T2A}\}\{\text{CYRV}\}\{\}\%$	% CY. C. L. VE
3730	$\backslash\text{do}\{0413\}\{\text{T2A}\}\{\text{CYRG}\}\{\}\%$	% CY. C. L. GHE
3731	$\backslash\text{do}\{0414\}\{\text{T2A}\}\{\text{CYRD}\}\{\}\%$	% CY. C. L. DE
3732	$\backslash\text{do}\{0415\}\{\text{T2A}\}\{\text{CYRE}\}\{\}\%$	% CY. C. L. IE
3733	$\backslash\text{do}\{0416\}\{\text{T2A}\}\{\text{CYRZH}\}\{\}\%$	% CY. C. L. ZHE
3734	$\backslash\text{do}\{0417\}\{\text{T2A}\}\{\text{CYRZ}\}\{\}\%$	% CY. C. L. ZE
3735	$\backslash\text{do}\{0418\}\{\text{T2A}\}\{\text{CYRI}\}\{\}\%$	% CY. C. L. I
3736	$\backslash\text{do}\{0419\}\{\text{T2A}\}\{\text{CYRISHRT}\}\{\}\%$	% CY. C. L. SHORT I
3737	$\backslash\text{do}\{041A\}\{\text{T2A}\}\{\text{CYRK}\}\{\}\%$	% CY. C. L. KA
3738	$\backslash\text{do}\{041B\}\{\text{T2A}\}\{\text{CYRL}\}\{\}\%$	% CY. C. L. EL
3739	$\backslash\text{do}\{041C\}\{\text{T2A}\}\{\text{CYRM}\}\{\}\%$	% CY. C. L. EM
3740	$\backslash\text{do}\{041D\}\{\text{T2A}\}\{\text{CYRN}\}\{\}\%$	% CY. C. L. EN
3741	$\backslash\text{do}\{041E\}\{\text{T2A}\}\{\text{CYRO}\}\{\}\%$	% CY. C. L. O
3742	$\backslash\text{do}\{041F\}\{\text{T2A}\}\{\text{CYRP}\}\{\}\%$	% CY. C. L. PE
3743	$\backslash\text{do}\{0420\}\{\text{T2A}\}\{\text{CYRR}\}\{\}\%$	% CY. C. L. ER
3744	$\backslash\text{do}\{0421\}\{\text{T2A}\}\{\text{CYRS}\}\{\}\%$	% CY. C. L. ES
3745	$\backslash\text{do}\{0422\}\{\text{T2A}\}\{\text{CYRT}\}\{\}\%$	% CY. C. L. TE
3746	$\backslash\text{do}\{0423\}\{\text{T2A}\}\{\text{CYRU}\}\{\}\%$	% CY. C. L. U
3747	$\backslash\text{do}\{0424\}\{\text{T2A}\}\{\text{CYRF}\}\{\}\%$	% CY. C. L. EF
3748	$\backslash\text{do}\{0425\}\{\text{T2A}\}\{\text{CYRH}\}\{\}\%$	% CY. C. L. HA
3749	$\backslash\text{do}\{0426\}\{\text{T2A}\}\{\text{CYRC}\}\{\}\%$	% CY. C. L. TSE
3750	$\backslash\text{do}\{0427\}\{\text{T2A}\}\{\text{CYRCH}\}\{\}\%$	% CY. C. L. CHE
3751	$\backslash\text{do}\{0428\}\{\text{T2A}\}\{\text{CYRSH}\}\{\}\%$	% CY. C. L. SHA
3752	$\backslash\text{do}\{0429\}\{\text{T2A}\}\{\text{CYRSHCH}\}\{\}\%$	% CY. C. L. SHCHA
3753	$\backslash\text{do}\{042A\}\{\text{T2A}\}\{\text{CYRHRDSN}\}\{\}\%$	% CY. C. L. HARD SIGN
3754	$\backslash\text{do}\{042B\}\{\text{T2A}\}\{\text{CYRERY}\}\{\}\%$	% CY. C. L. YERU
3755	$\backslash\text{do}\{042C\}\{\text{T2A}\}\{\text{CYRSFTSN}\}\{\}\%$	% CY. C. L. SOFT SIGN

```

3756 \do{042D}{T2A}{\CYREREV}{}%           % CY. C. L. E
3757 \do{042E}{T2A}{\CYRYU}{}%           % CY. C. L. YU
3758 \do{042F}{T2A}{\CYRYA}{}%           % CY. C. L. YA
3759 \do{0430}{T2A}{\cyra}{}%           % CY. S. L. A
3760 \do{0431}{T2A}{\cyrb}{}%           % CY. S. L. BE
3761 \do{0432}{T2A}{\cyrv}{}%           % CY. S. L. VE
3762 \do{0433}{T2A}{\cyrg}{}%           % CY. S. L. GHE
3763 \do{0434}{T2A}{\cyrd}{}%           % CY. S. L. DE
3764 \do{0435}{T2A}{\cyre}{}%           % CY. S. L. IE
3765 \do{0436}{T2A}{\cyrzh}{}%           % CY. S. L. ZHE
3766 \do{0437}{T2A}{\cyrz}{}%           % CY. S. L. ZE
3767 \do{0438}{T2A}{\cyri}{}%           % CY. S. L. I
3768 \do{0439}{T2A}{\cyrishrt}{}%       % CY. S. L. SHORT I
3769 \do{043A}{T2A}{\cyrk}{}%           % CY. S. L. KA
3770 \do{043B}{T2A}{\cyrl}{}%           % CY. S. L. EL
3771 \do{043C}{T2A}{\cyrm}{}%           % CY. S. L. EM
3772 \do{043D}{T2A}{\cyrn}{}%           % CY. S. L. EN
3773 \do{043E}{T2A}{\cyro}{}%           % CY. S. L. O
3774 \do{043F}{T2A}{\cyrp}{}%           % CY. S. L. PE
3775 \do{0440}{T2A}{\cyrr}{}%           % CY. S. L. ER
3776 \do{0441}{T2A}{\cyrs}{}%           % CY. S. L. ES
3777 \do{0442}{T2A}{\cyrt}{}%           % CY. S. L. TE
3778 \do{0443}{T2A}{\cyru}{}%           % CY. S. L. U
3779 \do{0444}{T2A}{\cyrf}{}%           % CY. S. L. EF
3780 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
3781 \do{0446}{T2A}{\cyrc}{}%           % CY. S. L. TSE
3782 \do{0447}{T2A}{\cyrch}{}%           % CY. S. L. CHE
3783 \do{0448}{T2A}{\cyrsh}{}%           % CY. S. L. SHA
3784 \do{0449}{T2A}{\cyrshch}{}%         % CY. S. L. SHCHA
3785 \do{044A}{T2A}{\cyrhrdsn}{}%        % CY. S. L. HARD SIGN
3786 \do{044B}{T2A}{\cyrery}{}%         % CY. S. L. YERU
3787 \do{044C}{T2A}{\cyrsftsn}{}%       % CY. S. L. SOFT SIGN
3788 \do{044D}{T2A}{\cyrerev}{}%        % CY. S. L. E
3789 \do{044E}{T2A}{\cyryu}{}%         % CY. S. L. YU
3790 \do{044F}{T2A}{\cyrya}{}%         % CY. S. L. YA
3791 \do{0451}{T2A}{\cyryo}{}%         % CY. S. L. IO
3792 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
3793 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
3794 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
3795 \do{00B1}{TS1}{\textpm}{\pm}%       % PLUS-MINUS SIGN
3796 \do{00B4}{TS1}{\textasciicute}{}%   % ACUTE ACCENT
3797 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
3798 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
3799 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
3800 }

3801 \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。

```

3802 \newif\ifbxjx@gcc@cjkk
\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。
3803 \newcommand*\greekasCJK{%
3804   \bxjx@gcc@cjkktrue}
3805 \newcommand*\nogreekasCJK{%
3806   \bxjx@gcc@cjkkfalse}
\bx@fake@grk  \bx@fake@grk{(出力文字)}{(基準文字)} :
3807 \def\bxjx@do#1\relax{%
3808   \def\bxjx@fake@grk##1##2{%
3809     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
3810   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
3811     \ifx\##1\%
3812       \bxjx@cnta##4\divide\bxjx@cnta\@cclvi
3813       \multiply\bxjx@cnta\@cclvi \advance\bxjx@cnta`##3\relax
3814       \mathchar\bxjx@cnta
3815     \else ##3\fi}
3816 } \expandafter\bxjx@do\string\mathchar\relax

```

#### ■ pdfLaTeX・upLaTeX の場合

```
3817 \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。

```

3818 \@ifpackageloaded{inputenc}{}{%else
3819   \RequirePackage[utf8]{inputenc}}
3820 \def\bxjx@tmpa{utf8}
3821 \ifx\bxjx@tmpa\inputencdoingname
3822   \PackageWarningNoLine\bxjx@pkname
3823     {Input encoding changed to utf8}%
3824   \inputencoding{utf8}%
3825 \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。

```

3826 \if u\bxjx@engine
3827 \kcatcode"0370=15
3828 \kcatcode"0400=15
3829 \kcatcode"0500=15
3830 \fi

```

各文字について \DeclareUnicodeCharacter を実行する。

```

3831 \def\do#1{%
3832   \@tempcnta="#1\relax
3833   \@tempcntb=\@tempcnta \divide\@tempcntb256
3834   \expandafter\let\cename bxjx@KCR/\the\@tempcntb\endcsname=t%
3835   \expandafter\bxjx@do@a\cename bxjx@KC/\the\@tempcnta\endcsname{#1}}
3836 \def\bxjx@do@a#1#2#3#4#5{%

```

```

3837 \ifx\#5\%
3838 \def\bxjx@tmpa{\@inmathwarn#4}%
3839 \else\ifcat A\noexpand#5%
3840 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3841 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
3842 \else \def\bxjx@tmpa{#5}%
3843 \fi\fi
3844 \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
3845 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
3846 \if u\bxjx@engine
3847 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
3848 \def\bxjx@do@b#1#2#3#4#5{%
3849 \kchardef#1=\@tempcnta
3850 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
3851 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
3852 \else\if p\bxjx@engine
3853 \def\bxjx@do@b#1#2#3#4#5{%
3854 \mathchardef#1=\@tempcnta
3855 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{UTF{#2}}{#3}{#4}}%
3856 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
3857 \fi\fi
3858 \bxjx@grkcyr@list
3859 \let\bxjx@do@a\undefined
3860 \let\bxjx@do@b\undefined

```

\bxjx@DeclareUnicodeCharacter \bxjx@DeclareUnicodeCharacter を変更して、ギリシャ・キリル文字の場合に再定義を抑制したもの。

```

3861 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
3862 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
3863 \@onlypreamble\bxjx@DeclareUnicodeCharacter
3864 \def\bxjx@DeclareUnicodeCharacter#1#2{%
3865 \count@=#1\relax \bxjx@cnta\count@ \divide\bxjx@cnta256
3866 \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cnta\endcsname\relax
3867 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3868 \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
3869 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3870 \else
3871 \wlog{ \space\space skipped defining Unicode char U+#1}%
3872 \fi\fi}

```

\bxjx@ja@or@not \bxjx@ja@or@not

```
3873 \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。

```
3874 \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。

```
3875 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
```

```

3876 \else \UseTextSymbol{#2}{#3}%
3877 \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。

```

3878 \begingroup
3879 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}
3880 \xdef\next{\def\noexpand\DeclareFontEncoding@###2##3{%
3881 \noexpand\bxjx@swap@DUC@cmd
3882 \the\toks@
3883 \noexpand\bxjx@swap@DUC@cmd}}
3884 \endgroup\next
3885 \def\bxjx@swap@DUC@cmd{%
3886 \let\bxjx@tmpa\DeclareUnicodeCharacter
3887 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
3888 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

#### ■ Xe<sub>La</sub>TeX・Lua<sub>La</sub>TeX の場合

```

3889 \else\ifnum0\if x\bxjx@engine1\fi\if l\bxjx@engine1\fi>0

```

各文字について、`math active` を設定する。

```

3890 \def\do#1{%
3891 \bxjx@canta="#1\relax
3892 \begingroup
3893 \lccode`~=\bxjx@canta
3894 \lowercase{\endgroup
3895 \bxjx@do@a{~}}{#1}}
3896 \def\bxjx@do@a#1#2#3#4#5{%
3897 \ifx\#5\\\let\bxjx@tmpa\relax
3898 \else\ifcat A\noexpand#5%
3899 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3900 {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
3901 \else \def\bxjx@tmpa{#5}%
3902 \fi\fi
3903 \ifx\bxjx@tmpa\relax\else
3904 \mathcode\bxjx@canta"8000 \let#1\bxjx@tmpa
3905 \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。

```

3906 \mathchardef\bxjx@tmpa="119
3907 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
3908 \let\bxjx@do@a\undefined

```

Lua<sub>La</sub>TeX における `\(no)greekasCJK` の定義。 `jacharrange` の設定を変更する。

```

3909 \if l\bxjx@engine
3910 \protected\def\greekasCJK{%
3911 \bxjx@gcc@cjctrue

```

```

3912 \ltjsetparameter{jacharrange={+2, +8}}
3913 \protected\def\nogreekasCJK{%
3914 \bxjx@gcc@cjkfalse
3915 \ltjsetparameter{jacharrange={-2, -8}}
3916 \fi

```

X<sub>Y</sub>TeX における \(\text{no}\)greekasCJK の定義。

```

3917 \if x\bxjx@engine
3918 \protected\def\greekasCJK{%
3919 \bxjx@gcc@cjktrue
3920 \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
3921 \bxjx@grkcyr@list}
3922 \protected\def\nogreekasCJK{%
3923 \bxjx@gcc@cjkfalse
3924 \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
3925 \bxjx@grkcyr@list}
3926 \fi

```

以上。

```

3927 \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。

```

3928 \nogreekasCJK

```

## H.5 完了

おしまい。

```

3929 %</cjkcat>

```

補助パッケージ実装はここまで。

```

3930 %</anc>

```