

# 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”)

v2.0a [2020/04/04]

---





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

---

## 目次

1	はじめに	3
2	オプション	10
3	和文フォントの変更	38
4	フォントサイズ	39
5	レイアウト	44
5.1	ページレイアウト . . . . .	45
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	59
7	ページスタイル	61
8	文書のマークアップ	64
8.1	表題 . . . . .	64
8.2	章・節 . . . . .	69
8.3	リスト環境 . . . . .	81
8.4	パラメータの設定 . . . . .	89
8.5	フロート . . . . .	90
8.6	キャプション . . . . .	92
9	フォントコマンド	93

10	相互参照	95
10.1	目次の類 . . . . .	95
10.2	参考文献 . . . . .	101
10.3	索引 . . . . .	102
10.4	脚注 . . . . .	104
11	段落の頭へのグルー挿入禁止	106
12	いろいろなロゴ	110
13	amsmath との衝突の回避	111
14	初期設定	111
付録 A	和文ドライバの仕様 罫	116
付録 B	和文ドライバ : minimal 罫	117
B.1	補助マクロ . . . . .	117
B.2	(u)pTeX 用の設定 . . . . .	119
B.3	pdfTeX 用の処理 . . . . .	124
B.4	X <sub>Y</sub> TeX 用の処理 . . . . .	125
B.5	後処理 (エンジン共通) . . . . .	125
付録 C	和文ドライバ : standard 罫	128
C.1	共通処理 (1) . . . . .	129
C.2	pTeX 用設定 . . . . .	136
C.3	pdfTeX 用設定 : CJK + bxcjkatype . . . . .	140
C.4	X <sub>Y</sub> TeX 用設定 : xeCJK + zxjatype . . . . .	142
C.5	LuaTeX 用設定 : LuaTeX-ja . . . . .	144
C.6	共通処理 (2) . . . . .	147
付録 D	和文ドライバ : modern 罫	148
D.1	フォント設定 . . . . .	149
D.2	fixltx2e 読込 . . . . .	149
D.3	和文カテゴリコード . . . . .	149
D.4	完了 . . . . .	149
付録 E	和文ドライバ : pandoc 罫	150
E.1	dupload システム . . . . .	150
E.2	lang 変数 . . . . .	151
E.3	geometry 変数 . . . . .	154
E.4	CJKmainfont 変数 . . . . .	154
E.5	paragraph のマーク . . . . .	154

E.6	全角空白文字 . . . . .	155
E.7	完了 . . . . .	155
付録 F	補助パッケージ一覧 	155
付録 G	補助パッケージ : bxjscompat 	156
G.1	準備 . . . . .	156
G.2	X <sub>Y</sub> TeX 部分 . . . . .	156
G.3	LuaTeX 部分 . . . . .	157
G.4	完了 . . . . .	158
付録 H	補助パッケージ : bxjscjcat 	158
H.1	準備 . . . . .	158
H.2	和文カテゴリコードの設定 . . . . .	159
H.3	ギリシャ・キリル文字の扱い . . . . .	160
H.4	初期設定 . . . . .	167
H.5	完了 . . . . .	167
付録 I	補助パッケージ : bxjspandoc 	168
I.1	準備 . . . . .	168
I.2	パッケージオプション . . . . .	168
I.3	パッケージ読込の阻止 . . . . .	168
I.4	fixltx2e パッケージ . . . . .	169
I.5	cmap パッケージ . . . . .	169
I.6	microtype パッケージ . . . . .	169
I.7	Unicode 文字変換対策 . . . . .	170
I.8	PandoLa モジュール . . . . .	171
I.9	完了 . . . . .	171

## 1 はじめに

---

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

```

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

```

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「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 `jclasses.dtx` に基づいてもともと奥村晴彦により作成されたものです。現在は日本語 T<sub>E</sub>X 開発コミュニティにより GitHub で管理されています。

<https://github.com/texjporg/jsclasses>

[2002-12-19] いろいろなものに収録していただく際にライセンスを明確にする必要が生じてきました。アスキーのものが最近では modified BSD ライセンスになっていますので、私のものもそれに準じて modified BSD とすることにします。

[2016-07-13] 日本語 T<sub>E</sub>X 開発コミュニティによる管理に移行しました。

[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。

ここでは次のドキュメントクラス（スタイルファイル）を作ります。

[2017-02-13] forum:2121 の議論を機に、jsreport クラスを新設しました。従来の jsbook の report オプションと比べると、abstract 環境の使い方および挙動がアスキーの jreport に近づきました。

<code>&lt;article&gt;</code>	<code>jsarticle.cls</code>	論文・レポート用
<code>&lt;book&gt;</code>	<code>jsbook.cls</code>	書籍用
<code>&lt;report&gt;</code>	<code>jsreport.cls</code>	レポート用
<code>&lt;jspf&gt;</code>	<code>jspf.cls</code>	某学会誌用
<code>&lt;kiyou&gt;</code>	<code>kiyou.cls</code>	某紀要用

以下では実際のコードに即して説明します。

```
1 %<*cls>
2 %% このファイルは日本語文字を含みます。
```

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

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

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

```
7 \newif\ifjsc@needsp@tch
8 \jsc@needsp@tchfalse
```

## ■ BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため calc を読み込む。

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

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

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

クラスの本体ではこの他に geometry パッケージが読み込まれる。

**TODO:** 依存パッケージの情報。

互換性のための補助パッケージを読み込む。

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

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

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

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

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

`\ifjsWithupTeX` [スイッチ] エンジンが (内部漢字コードが Unicode の)  $\mathrm{up}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$  であるか。

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

`\ifjsWithpTeXng` [スイッチ] エンジンが  $\mathrm{p}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}\mathrm{-ng}$  であるか。

```

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

\ifjsWitheTeX [スイッチ] エンジンが  $\epsilon$ -TeX 拡張をもつか。
41 \newif\ifjsWitheTeX
42 \bxjs@test@engine\eTeXversion{\jsWitheTeXtrue}

非サポートのエンジンの場合は強制終了させる。
※ NTT jTeX と Omega 系。
43 \let\bxjs@tmpa\relax
44 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
45 \ifx O\jsEngine \def\bxjs@tmpa{Omega}\fi
46 \ifx\bxjs@tmpa\relax \expandafter\@gobble
47 \else
48   \ClassError\bxjs@clsname
49   {The engine in use (\bxjs@tmpa) is not supported}
50   {It's a fatal error. I'll quit right now.}
51   \expandafter\@firstofone
52 \fi{\endinput\@@end}

LuaTeX の場合、本クラス用の Lua モジュールを用意する。
53 \ifx l\jsEngine
54   \directlua{ bxjs = {} }
55 \fi

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

\bxjs@robust@def 無引数の頑強な命令を定義する。
59 \ifjsWitheTeX
60   \def\bxjs@robust@def{\protected\def}
61 \else
62   \def\bxjs@robust@def{\DeclareRobustCommand*}
63 \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。
64 \newif\ifjsInPdfMode
65 \@nameuse{ImposeOldLuaTeXBehavior}
66 \let\bxjs@tmpa\PackageWarningNoLine
67 \let\PackageWarningNoLine\PackageInfo % suppress warning
68 \RequirePackage{ifpdf}
69 \let\PackageWarningNoLine\bxjs@tmpa
70 \@nameuse{RevokeOldLuaTeXBehavior}
71 \let\ifjsInPdfMode\ifpdf

\ifbxjs@TUenc [スイッチ] LATEX の既定のフォントエンコーディングが TU であるか。

```

※ 2017 年 1 月以降の L<sup>A</sup>T<sub>E</sub>X カーネルにおいて「Unicode を表す L<sup>A</sup>T<sub>E</sub>X 公式のフォントエンコーディング」である“TU”が導入され、これ以降の L<sup>A</sup>T<sub>E</sub>X を X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X または Lua<sup>A</sup>T<sub>E</sub>X で動かしている場合は、既定のエンコーディングが TU になる。それ以外の場合は、既定のエンコーディングは OT1 である。

```
72 \newif\ifbxjs@TUenc
73 \def\bxjs@tmpa{TU}\edef\bxjs@tmpb{\f@encoding}
74 \ifx\bxjs@tmpa\bxjs@tmpb
75   \bxjs@TUenctrue
76 \fi
```

`\bxjs@cond` `\bxjs@cond\ifXXX……\fi{⟨真⟩}{⟨偽⟩}`

T<sub>E</sub>X 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。

```
77 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
78   #1\expandafter\@firstoftwo
79   \else\expandafter\@secondoftwo
80   \fi}
```

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

```
81 \def\bxjs@cslet#1{%
82   \expandafter\let\csname#1\endcsname}
```

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

```
83 \def\bxjs@csletcs#1#2{%
84   \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}
```

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

```
85 \def\bxjs@catopt#1#2{%
86   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}
```

`\bxjs@ifplus` `\@ifstar` の + 版。

```
87 \def\bxjs@ifplus#1{\@ifnextchar+{\@firstoftwo{#1}}}
```

`\bxjs@gset@tempcnta` `calc` の整数式を用いて `\@tempcnta` の値を設定する。

```
88 \let\c@bxjs@tempcnta\@tempcnta
89 \def\bxjs@gset@tempcnta{\setcounter{bxjs@tempcnta}}
```

`\jsSetQHLength` `\jsSetQHLength\CS{⟨長さ式⟩} :` `\setlength` の変種で、通常の `calc` の長さ式の代わりに、「Q/H/trueQ/trueH/zw/zh の単位付きの実数」が記述できる（この場合は式は使えない）。

```
90 \def\jsSetQHLength#1#2{%
91   \begingroup
92     \bxjs@parse@qh{#2}%
93     \ifx\bxjs@tmpb\relax
94       \setlength\@tempdima{#2}%
95       \xdef\bxjs@g@tmpa{\the\@tempdima}%
96     \else \global\let\bxjs@g@tmpa\bxjs@tmpb
```

```

97   \fi
98   \endgroup
99   #1=\bxjs@q@tmpa\relax}

```

`\bxjs@parse@qh` #1 が Q/H/trueQ/trueH/zw/zh で終わる場合、単位用の寸法値マクロ `\bxjs@unit@XXX` が定義済なら、`\bxjs@tmpb` に #1 に等しい寸法の表現を返し、そうでないならエラーを出す。それ以外では、`\bxjs@tmpb` は `\relax` になる。

※ (u)pL<sup>A</sup>T<sub>E</sub>X の場合はこれらの和文単位はエンジンでサポートされる。しかし和文フォントの設定が完了するまでは zw/zh の値は正しくない。

```

100 \if j\jsEngine \def\bxjs@parse@qh@units{zw,zh}
101 \else \def\bxjs@parse@qh@units{trueQ,trueH,Q,H,zw,zh}
102 \fi
103 \def\bxjs@parse@qh#1{%
104   \let\bxjs@tmpb\relax
105   \@for\bxjs@tmpa:=\bxjs@parse@qh@units\do{%
106     \ifx\bxjs@tmpb\relax
107       \edef\bxjs@next{{\bxjs@tmpa}{#1}}%
108       \expandafter\bxjs@parse@qh@a\csname bxjs@unit@\bxjs@tmpa\expandafter
109         \endcsname\bxjs@next
110     \fi}}
111 \def\bxjs@parse@qh@a#1#2#3{%
112   \def\bxjs@next##1#2\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}#1}%
113   \bxjs@next#3\@nil#2\@nil\@nnil}
114 \def\bxjs@parse@qh@b#1#2#3{%
115   \ifx\@nnil#2\@nnil\else
116     \ifx#3\relax
117       \ClassError\bxjs@clsname
118       {You cannot use '\bxjs@tmpa' here}{\@ehc}%
119       \def\bxjs@tmpb{Opt}%
120     \else
121       \@tempdimb#3\relax \@tempdimb#1\@tempdimb
122       \edef\bxjs@tmpb{\the\@tempdimb}%
123     \fi
124   \fi}

```

今の段階では Q/H だけが使用可能。

```

125   \def\bxjs@unit@Q{0.25mm}\let\bxjs@unit@H\bxjs@unit@Q

```

`\bxjs@begin@document@hook` BXJS クラス用の文書本体開始時フック。

```

126 \@onlypreamble\bxjs@begin@document@hook
127 \let\bxjs@begin@document@hook\@empty
128 \AtBeginDocument{\bxjs@begin@document@hook}

```

`\bxjs@post@option@hook` `\ProcessOptions` 直後に実行されるフック。

```

129 \@onlypreamble\bxjs@post@option@hook
130 \let\bxjs@post@option@hook\@empty

```

`\bxjs@pre@jadriver@hook` 和文ドライバ読込直前に実行されるフック。



```

131 \@onlypreamble\bxjs@pre@jadriver@hook
132 \let\bxjs@pre@jadriver@hook\@empty

```

`\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。)

```

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

```

一時的な手続き用の制御綴。

```

135 \@onlypreamble\bxjs@tmpdo
136 \@onlypreamble\bxjs@tmpdo@a
137 \@onlypreamble\bxjs@tmpdo@b
138 \@onlypreamble\bxjs@tmpdo@c
139 \@onlypreamble\bxjs@tmpdo@d

```

Lua<sub>T</sub><sub>E</sub>X の場合、原版のコード中のコントロールワード中に現れる日本語文字のカテゴリコードを一時的に 11 に変更する。クラス読込終了時点で元に戻される。

※現在の Lua<sub>A</sub><sub>T</sub><sub>E</sub>X 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。

```

140 \if 1\jsEngine
141 \def\bxjs@tmpdo#1{%
142   \xdef\bxjs@pre@jadriver@hook{%
143     \bxjs@pre@jadriver@hook
144     \catcode`#1=\the\catcode`#1\relax}%
145   \catcode`#1=11\relax}
146 \@tfor\bxjs@tmpa:=和西暦\do
147   {\expandafter\bxjs@tmpdo\bxjs@tmpa}
148 \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。

```

149 \bxjs@robust@def\jsInhibitGlue{%
150   \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。

```

151 \if@compatibility
152   \ClassError\bxjs@clsname
153     {Something went chaotic!\MessageBreak
154     (How come '\string\documentstyle' is there?)\MessageBreak
155     I cannot go a single step further...}
156   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
157     then there'll still be hope....}
158   \expandafter\@firstofone
159 \else \expandafter\@gobble
160 \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。

```
161 \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。

```
162 \newif\if@titlepage
```

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

```
163 %<book|report>\newif\if@openright
```

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

```
164 %<book|report>\newif\if@openleft
```

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

---

BXJS では report 系でも定義されることに注意。

---

```
165 %<book|report>\newif\if@mainmatter \@mainmattertrue
```

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

---

JS クラスと異なり、初期値は偽とする。

---

```
166 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

B 判は JIS と ISO で定義が異なります。JIS では B0 判の面積が  $1.5\text{ m}^2$  ですが，ISO では B1 判の辺の長さが A0 判と A1 判の辺の長さの幾何平均です。したがって ISO の B0 判は  $1000\text{ mm} \times 1414\text{ mm}$  です。このため， $\text{\LaTeX 2}_\epsilon$  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。

```
167 \onlypreamble\bxjs@setpaper
168 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
169 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
170 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
171 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
172 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
173 \DeclareOption{b4paper}{\bxjs@setpaper{{257trueemm}{364trueemm}}}
174 \DeclareOption{b5paper}{\bxjs@setpaper{{182trueemm}{257trueemm}}}
175 \DeclareOption{b6paper}{\bxjs@setpaper{{128trueemm}{182trueemm}}}
176 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
177 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
178 \DeclareOption{b4j}{\bxjs@setpaper{{257trueemm}{364trueemm}}}
179 \DeclareOption{b5j}{\bxjs@setpaper{{182trueemm}{257trueemm}}}
180 \DeclareOption{a4var}{\bxjs@setpaper{{210trueemm}{283trueemm}}}
181 \DeclareOption{b5var}{\bxjs@setpaper{{182trueemm}{230trueemm}}}
```

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

```
182 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210trueemm}{283trueemm}}}
183 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182trueemm}{230trueemm}}}
184 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
185 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
186 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}
```

---

■横置き 用紙の縦と横の長さを入れ換えます。

```
187 \newif\if@landscape
188 \@landscapefalse
189 \DeclareOption{landscape}{\@landscapetrue}
```

■slide オプション `slide` を新設しました。

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

```
190 \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。

```
191 %<!slide>\@slidefalse
192 %<slide>\@slidetrue
```

---

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

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

```
193 \newcommand{\@ptsize}{0}
194 \def\bxjs@param@basefontsize{10pt}
195 \def\jsUnusualPtSize{-20}
```

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

```
196 \def\bxjs@setbasefontsize#1{%
```

Q 単位の長さ指定をサポートするため `\jsSetQHLlength` を使う。

※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、 $\text{\LaTeX}$  はクラス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。

```
197 \jsSetQHLlength\@tempdima{#1}%
198 \edef\bxjs@param@basefontsize{\the\@tempdima}%
199 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
200 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
201 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
202 \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi
```

**TODO:** 恐らく 14pt と `base=14.4pt` 等の関係も全く等価であるべき。

```
203 \def\bxjs@setjbasefontsize#1{%
204 \setkeys{bxjs}{jbase=#1}}
```

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

```
205 \newif\ifjsc@mag
206 \newif\ifjsc@mag@xreal
207 %\let\jsc@magscale\@undefined
208 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
209 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
210 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
211 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
```

```

212 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
213 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
214 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
215 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
216 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
217 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
218 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
219 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
220 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
221 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
222 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
223 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
224 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
225 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
226 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

227 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
228 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
229 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}

```

---

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

---

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

---

```

230 \if j\jsEngine
231 \hour\time \divide\hour by 60\relax
232 \@tempcnta\hour \multiply\@tempcnta 60\relax
233 \minute\time \advance\minute-\@tempcnta
234 \DeclareOption{tombow}{%
235   \tombowtrue \tombowdatetrue
236   \setlength{\@tombowwidth}{.1\p}%
237   \@bannertoken{%
238     \jobname\space(\number\year-\two@digits\month-\two@digits\day
239     \space\two@digits\hour:\two@digits\minute)}}%
240   \maketombowbox}
241 \DeclareOption{tombo}{%
242   \tombowtrue \tombowdatefalse
243   \setlength{\@tombowwidth}{.1\p}%
244   \maketombowbox}
245 \fi

```

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

```
246 \if j\jsEngine
247 \DeclareOption{mentuke}{%
248   \tombowtrue \tombowdatefalse
249   \setlength{\@tombowwidth}{\z@}%
250   \maketombowbox}
251 \fi
```

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

```
252 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
253 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
254 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
255 \DeclareOption{onecolumn}{\@twocolumnfalse}
256 \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。

```
257 \DeclareOption{titlepage}{\@titlepagetrue}
258 \DeclareOption{notitlepage}{\@titlepagefalse}
```

■右左起こし 書籍では章は通常は奇数ページ起こしになりますが、横組ではこれを `openright` と表すことにしてあります。 `openany` で偶数ページからでも始まるようになります。

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

```
259 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
260 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
261 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}
```

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

`eqnarray`  $\text{\LaTeX}$  の `eqnarray` 環境では `&` でできるアキが大きすぎるようですので、少し小さくします。また、中央の要素も `\displaystyle` にします。

```
262 \def\eqnarray{%
263   \stepcounter{equation}%
264   \def\@currentlabel{\p@equation\theequation}%
265   \global\@eqnswtrue
266   \m@th
```

```

267 \global\@eqcnt\z@
268 \tabskip\@centering
269 \let\@eqncr
270 $$\everycr{}\halign to\displaywidth\bgroup
271 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
272 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
273 &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
274 &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
275 \tabskip\z@skip
276 \cr}

```

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

```

277 \DeclareOption{leqno}{\input{leqno.clo}}
278 \DeclareOption{fleqn}{\input{fleqn.clo}}
279 % fleqn 用の eqnarray 環境の再定義
280 \def\eqnarray{%
281 \stepcounter{equation}%
282 \def\@currentlabel{\p@equation\theequation}%
283 \global\@eqnswtrue\m@th
284 \global\@eqcnt\z@
285 \tabskip\mathindent
286 \let\@eqncr
287 \setlength\abovedisplayskip{\topsep}%
288 \ifvmode
289 \addtolength\abovedisplayskip{\partopsep}%
290 \fi
291 \addtolength\abovedisplayskip{\parskip}%
292 \setlength\belowdisplayskip{\abovedisplayskip}%
293 \setlength\belowdisplayshortskip{\abovedisplayskip}%
294 \setlength\abovedisplayshortskip{\abovedisplayskip}%
295 $$\everycr{}\halign to\linewidth% $$
296 \bgroup
297 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
298 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
299 &\global\@eqcnt\tw@
300 $\displaystyle{##}$\hfil \tabskip\@centering
301 &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
302 \tabskip\z@skip\cr
303 }}

```

■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。

```

304 % \DeclareOption{openbib}{%
305 % \AtEndOfPackage{%
306 % \renewcommand\@openbib@code{%
307 % \advance\leftmargin\bibindent
308 % \itemindent -\bibindent

```

```

309 %      \listparindent \itemindent
310 %      \parsep \z@}%
311 %      \renewcommand\newblock{\par}}}
```

■数式フォントとして和文フォントを登録しないオプション 数式中では 16 通りのフォントしか使えません。AMSTeX や mathptmx パッケージを使って数式フォントをたくさん使うと “Too many math alphabets ...” というエラーが起こってしまいます。disablejfam オプションを付ければ、明朝・ゴシックを数式用フォントとして登録するのをやめますので、数式用フォントが二つ節約できます。いずれにしても \textmc や \mbox や amsmath パッケージの \text を使えば数式中で和文フォントが使えますので、この新ドキュメントクラスでは標準で和文フォントを数式用に登録しないことにしていたのですが、従来のドキュメントクラスの仕様に合わせることにしました。

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

```

312 %\let\bxjs@enablejfam\undefined

    enablejfam オプションの処理。
313 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
314 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
315 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
316 \define@key{bxjs}{enablejfam}[true]{%
317   \bxjs@set@keyval{enablejfam}{#1}{}}
```

JS クラスとの互換のため disablejfam オプションを定義する。

```

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

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

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

[2016-07-13] \ifdraft を定義するのをやめました。

\ifjsDraft draft オプションが指定されているか。

※ JS クラスの \ifdraft が廃止されたので、BXJS クラスでも \ifdraft を 2.0 版で廃止した。

```

319 \newif\ifjsDraft
320 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
321 \DeclareOption{final}{\jsDraftfalse \overfullrule=0pt }
```

■和文フォントメトリックの選択 このクラスファイルでは、和文 TFM として東京書籍印刷の小林肇さんの作られた JIS フォントメトリック (jis, jisg) を標準で使うことにしま



ですが、従来の `min10`, `goth10` などを使いたいときは `mingoth` というオプションを指定します。また、`winjis` オプションで `winjis` メトリック (OTF パッケージと同じ `psitau` さん作; ソースに書かれた Windows の機種依存文字が `dvips`, `dvipdfmx` などでも出力出来るようになる) が使えます。

[2018-02-04] `winjis` オプションはコッソリ削除しました。代替として、同等なものをパッケージ化 (`winjis.sty`) して、GitHub にはコッソリ置いておきます。

---

BXJS クラスではここは和文ドライバの管轄。

---

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

---

BXJS クラスでは `geometry` パッケージがこの処理を行う。

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

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

```
322 \newif\ifbxjs@papersize
323 \bxjs@papersizetrue
324 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
325 \DeclareOption{papersize}{\bxjs@papersizetrue}
```

---

■英語化 オプション `english` を新設しました。

---

※`\if@english` は非ユニークで衝突耐性がない。

---

```
326 \newif\if@english
327 \@englishfalse
328 \DeclareOption{english}{\@englishtrue}
```

---

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

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

---

BXJS では当初から `bxjsreport` クラスが用意されている。

---

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

---

BXJS クラスでは、nojslogo を既定とする。

---

```
329 \newif\if@jslogo \@jslogofalse
330 \DeclareOption{jslogo}{\@jslogotrue}
331 \DeclareOption{nojslogo}{\@jslogofalse}
```

#### ■複合設定オプション

**TODO:** `\bxjs@invscale` を書く場所を決める。(JS クラスと同じにはできなそう。)

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

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

※元の長さが 128 pt 以上の場合でも動作するように修正した。

```
332 \mathchardef\bxjs@isc@ll=128
333 \mathchardef\bxjs@isc@sl=259
334 \def\bxjs@isc@sl@h{65539 }
335 \def\bxjs@invscale#1#2{%
336   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
337   \ifdim\@tempdima<\bxjs@isc@ll\p@
338     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
339     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
340   \else
341     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
342     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
343   \fi
344   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
345   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
346   \@tempdimb\@tempcnta\@ne
347   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
348   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
349   \@whiledim\@tempdimb<\@tempdimc\do{%
350     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
351     \advance\@tempcntb\@ne \divide\@tempcntb\tw@
352     \ifdim #2\@tempcntb>\@tempdima
353       \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
354     \else \@tempdimb=\@tempcntb\@ne \fi}%
355   \xdef\bxjs@gtmpa{\the\@tempdimb}%
356   \endgroup #1=\bxjs@gtmpa\relax}
```

---

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

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

```
357 \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。

```
358 \g@addto@macro\bxjs@post@option@hook{%
359   \bxjs@oldfontcommandstrue
360   \setkeys{bxjs}{ja=pandoc}%
361   \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…。

```
362 \def\bxjs@driver@opt{dvipdfmx}%
363 \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。

```
364 %\let\bxjs@engine@given\undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
365 %\let\bxjs@engine@opt\undefined
```

エンジン明示指定のオプションの処理。

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

```
366 \DeclareOption{autodetect-engine}{%
367   \let\bxjs@engine@given=}%
368 \DeclareOption{latex}{%
369   \def\bxjs@engine@opt{latex}%
370   \let\bxjs@engine@given=n}
371 \DeclareOption{platex}{%
372   \def\bxjs@engine@opt{platex}%
373   \let\bxjs@engine@given=j}
374 \DeclareOption{uplatex}{%
375   \def\bxjs@engine@opt{uplatex}%
376   \let\bxjs@engine@given=u}
377 \DeclareOption{xelatex}{%
378   \def\bxjs@engine@opt{xelatex}%
379   \let\bxjs@engine@given=x}
380 \DeclareOption{pdfplatex}{%
381   \def\bxjs@engine@opt{pdfplatex}%
382   \let\bxjs@engine@given=p}
383 \DeclareOption{lualatex}{%
384   \def\bxjs@engine@opt{lualatex}%
385   \let\bxjs@engine@given=l}
```

```

386 \DeclareOption{platex-ng}{%
387   \def\bxjs@engine@opt{platex-ng}%
388   \let\bxjs@engine@given=g}
389 \DeclareOption{platex-ng*}{%
390   \def\bxjs@engine@opt{platex-ng*}%
391   \let\bxjs@platexng@nodrv=t%
392   \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。

```

393 %\let\bxjs@driver@given\@undefined
394 \let\bxjs@driver@@dvimode=0
395 \let\bxjs@driver@@dvipdfmx=1
396 \let\bxjs@driver@@pdfmode=2
397 \let\bxjs@driver@@xetex=3
398 \let\bxjs@driver@@dvips=4
399 \let\bxjs@driver@@none=5

```

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

```

400 %\let\bxjs@driver@opt\@undefined
401 \DeclareOption{dvips}{%
402   \def\bxjs@driver@opt{dvips}%
403   \let\bxjs@driver@given\bxjs@driver@@dvips}
404 \DeclareOption{dviout}{%
405   \def\bxjs@driver@opt{dviout}%
406   \let\bxjs@driver@given\bxjs@driver@@dvimode}
407 \DeclareOption{xdvi}{%
408   \def\bxjs@driver@opt{xdvi}%
409   \let\bxjs@driver@given\bxjs@driver@@dvimode}
410 \DeclareOption{dvipdfmx}{%
411   \def\bxjs@driver@opt{dvipdfmx}%
412   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
413 \DeclareOption{nodvidriver}{%
414   \def\bxjs@driver@opt{nodvidriver}%
415   \let\bxjs@driver@given\bxjs@driver@@none}
416 \DeclareOption{pdftex}{%
417   \def\bxjs@driver@opt{pdftex}%
418   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
419 \DeclareOption{luatex}{%
420   \def\bxjs@driver@opt{luatex}%
421   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
422 \DeclareOption{xetex}{%
423   \def\bxjs@driver@opt{xetex}%
424   \let\bxjs@driver@given\bxjs@driver@@xetex}

```

`dvipdfmx-if-dvi` は互換性のためのオプションで、`dvi=dvipdfmx` と同値である。

※ 2.0 版より `dvipdfmx-if-dvi` は非推奨となった。

```

425 \DeclareOption{dvipdfmx-if-dvi}{%
426   \ClassWarningNoLine\bxjs@clsname
427     {The old option 'dvipdfmx-if-dvi' is DEPRECATED}\MessageBreak

```

```

428     and may be abolished in future!\MessageBreak
429     You should write 'dvi=dvipdfmx' instead}%
430     \setkeys{bxjs}{dvi=dvipdfmx}}
431

```

---

## ■その他の BXJS 独自オプション

---

**TODO:** 互換用オプションを分離する (2.0 版で?)。

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

※ 2.0 版より、既定値を常に真とする。

```

432 \newif\ifbxjs@bigcode \bxjs@bigcodetrue

```

`nobigcode` / `bigcode` オプションの定義。

```

433 \DeclareOption{nobigcode}{%
434   \bxjs@bigcodefalse}
435 \DeclareOption{bigcode}{%
436   \bxjs@bigcodetrue}

```

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

```

437 \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。

```

438 \DeclareOption{nooldfontcommands}{%
439   \bxjs@oldfontcommandsfalse}
440 \DeclareOption{oldfontcommands}{%
441   \bxjs@oldfontcommandstrue}

```

---

## ■keyval 型のオプション

---

その他のオプションは `keyval` の機構を用いて処理する。

```

442 \DeclareOption*{%
443   \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
444   \expandafter\bxjs@next\expandafter{\CurrentOption}}

```

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

※ネスト不可。

```

445 \def\bxjs@safe@setkeys#1#2{%
446   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble

```

```

447 \setkeys{#1}{#2}%
448 \let\KV@errx\bxjs@save@KV@errx}

```

`\bxjs@declare@enum@option` `\bxjs@declare@enum@option{<オプション名>}{<enum 名>}`  
 “<オプション名>=<値>” のオプション指定に対して、`\[bxjs@<enum 名>]` を `\[bxjs@<enum 名>@@<値>]` に等値する（後者の制御綴が未定義の場合はエラー）、という動作を規定する。

```

449 \@onlypreamble\bxjs@declare@enum@option
450 \def\bxjs@declare@enum@option#1#2{%
451   \define@key{bxjs}{#1}{%
452     \expandafter\ifx\csname bxjs@#2@@#1\endcsname\relax
453     \bxjs@error@keyval{#1}{##1}%
454   \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@@#1}%
455   \fi}}

```

`\bxjs@declare@bool@option` `\bxjs@declare@bool@option{<オプション名>}{<スイッチ名>}`  
 “<オプション名>=<真偽値>” のオプション指定に対して、`\if[bxjs@<スイッチ名>]` を設定する、という動作を規定する。

```

456 \@onlypreamble\bxjs@declare@bool@option
457 \def\bxjs@declare@bool@option#1#2{%
458   \define@key{bxjs}{#1}[true]{%
459     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
460     \bxjs@error@keyval{#1}{##1}%
461   \else \@nameuse{bxjs@#2##1}%
462   \fi}}

```

`\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。

```

463 \def\bxjs@set@keyval#1#2#3{%
464   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
465   \ifx\bxjs@next\relax
466     \bxjs@error@keyval{#1}{#2}%
467     #3%
468   \else \bxjs@next
469   \fi}
470 \@onlypreamble\bxjs@error@keyval
471 \def\bxjs@error@keyval#1#2{%
472   \ClassError\bxjs@clsname
473   {Invalid value '#2' for option #1}\@ehc}

```

`\jsScale` 〔実数値マクロ〕 和文スケール値。

```

474 \def\jsScale{0.924715}

```

`\bxjs@base@opt` 明示された base オプションの値。

```

475 %\let\bxjs@base@opt\@undefined

```

base オプションの処理。

```

476 \define@key{bxjs}{base}{%
477   \edef\bxjs@base@opt{#1}%

```

```

478 \bxjs@setbasefontsize{#1}}
479 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

\bxjs@jbase@opt 明示された jbase オプションの値。
480 %\let\bxjs@jbase@opt\@undefined

jbase オプションの処理。
481 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
482 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}

\bxjs@scale@opt 明示された scale オプションの値。
483 %\let\bxjs@scale@opt\@undefined

scale オプションの処理。
484 \define@key{bxjs}{scale}{%
485 \edef\bxjs@scale@opt{#1}%
486 \let\jsScale\bxjs@scale@opt}
487 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

noscale オプションの処理。
488 \DeclareOption{noscale}{\setkeys{bxjs}{scale=1}}

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

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

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

\bxjs@jadriver 和文ドライバの名前。
492 \let\bxjs@jadriver\relax
493 %\let\bxjs@jadriver@opt\@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)。
494 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver@opt{#1}}
495 \define@key{bxjs}{ja}{\relax}{%
496 \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}

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

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

\jsJaParam 和文ドライバパラメタの文字列。
499 \let\jsJaParam\@empty

```

japaram オプションの処理。

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

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

```
501 \let\bxjs@magstyle@mag=m
```

```
502 \let\bxjs@magstyle@real=r
```

```
503 \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。

```
504 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
```

```
505 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
```

```
506 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

\bxjs@magstyle@default は既定の値を表す。

```
507 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
```

```
508 \ifx l\jsEngine \ifnum\luatexversion>86
```

```
509 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
```

```
510 \fi\fi
```

```
511 \ifjsWithpTeXng
```

```
512 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
```

```
513 \fi
```

```
514 \let\bxjs@magstyle\bxjs@magstyle@default
```

magstyle オプションの処理。

```
515 \define@key{bxjs}{magstyle}{%
```

```
516 \expandafter\let\expandafter\bxjs@magstyle\csname
```

```
517 bxjs@magstyle@#1\endcsname
```

```
518 \ifx\bxjs@magstyle\relax
```

```
519 \ClassError\bxjs@clsname
```

```
520 {Invalid value '#1' for option magstyle}\@ehc
```

```
521 \let\bxjs@magstyle\bxjs@magstyle@default
```

```
522 \fi}
```

\bxjs@geometry geometry オプションの値。

```
523 \let\bxjs@geometry@class=c
```

```
524 \let\bxjs@geometry@user=u
```

```
525 \let\bxjs@geometry\bxjs@geometry@class
```

geometry オプションの処理。

```
526 \define@key{bxjs}{geometry}{%
```

```
527 \expandafter\let\expandafter\bxjs@geometry\csname
```

```
528 bxjs@geometry@#1\endcsname
```

```
529 \ifx\bxjs@geometry\relax
```

```
530 \ClassError\bxjs@clsname
```

```
531 {Invalid value '#1' for option geometry}\@ehc
```

```
532 \let\bxjs@geometry\bxjs@geometry@class
```

```
533 \fi}
```



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

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

    fancyhdr オプションの処理。
535 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
536 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
537 \define@key{bxjs}{fancyhdr}[true]{%
538   \bxjs@set@keyval{fancyhdr}{#1}{}}
```

`\ifbxjs@dvi@opt` dvi オプションが指定されたか。

```
539 \newif\ifbxjs@dvi@opt

    DVI モードのドライバとドライバ種別との対応。
540 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
541 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
542 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
543 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
544 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

    dvi オプションの処理。
545 \define@key{bxjs}{dvi}{%
546   \expandafter\let\expandafter\bxjs@tmpa\csname
547     bxjs@dvidriver@@#1\endcsname
548   \ifx\bxjs@tmpa\relax
549     \ClassError\bxjs@clsname
550       {Invalid value '#1' for option dvi}\@ehc
551   \else
    \bxjs@driver@given を未定義にしていることに注意。
552     \def\bxjs@driver@opt{#1}%
553     \let\bxjs@driver@given\@undefined
554     \bxjs@dvi@opttrue
555   \fi}
```

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

※layout が v1 の場合はアレになる。

```
556 \newif\ifbxjs@layout@buggyhmargin
```

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

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

```
557 \newif\ifbxjs@force@chapterabstract
558 %<book>\bxjs@force@chapterabstracttrue

    layout オプションの処理。
559 \@namedef{bxjs@kv@layout@v1}{%
560 %<book>\bxjs@layout@buggyhmargintrue
561 %<report>\bxjs@force@chapterabstracttrue
562 }
563 \@namedef{bxjs@kv@layout@v2}{%
```

```

564 %<book>\bxjs@layout@buggyhmarginfalse
565 %<report>\bxjs@force@chapterabstractfalse
566 }
567 \define@key{bxjs}{layout}{%
568   \bxjs@set@keyval{layout}{#1}{}}

```

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

```

569 %\let\bxjs@textwidth@limit@opt\@undefined
570 \define@key{bxjs}{textwidth-limit}{%
571   \edef\bxjs@textwidth@limit@opt{#1}}

```

`\bxjs@textwidth@opt` `textwidth` の指定値。

```

572 %\let\bxjs@textwidth@opt\@undefined
573 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
574 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

```

`\bxjs@number@of@lines@opt` `number-of-lines` の指定値。

```

575 %\let\bxjs@number@of@lines@opt\@undefined
576 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
577 \define@key{bxjs}{number_of_lines}{\setkeys{bxjs}{number-of-lines=#1}}

```

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

```

578 %\let\bxjs@paragraph@mark\@undefined
579 \define@key{bxjs}{paragraph-mark}{%
580   \edef\bxjs@paragraph@mark{#1}}

```

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

```

581 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
582 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
583 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
584 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}

```

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

```

585 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
586 \let\bxjs@kv@jaspacecmd@true\bxjs@jaspace@cmdtrue
587 \let\bxjs@kv@jaspacecmd@false\bxjs@jaspace@cmdfalse
588 \define@key{bxjs}{jaspace-cmd}[true]{\bxjs@set@keyval{jaspacecmd}{#1}{}}
589 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{jaspacecmd}{#1}{}}

```

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

```

590 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
591 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
592 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
593 \define@key{bxjs}{fix-at-cmd}[true]{\bxjs@set@keyval{fixatcmd}{#1}{}}

```

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

```

594 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
595 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
596 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
597 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}

```

`\bxjs@everyparhook` `everyparhook` の指定値。

```
598 \chardef\bxjs@everyparhook@none=0
599 \chardef\bxjs@everyparhook@compat=1
600 \chardef\bxjs@everyparhook@modern=2
601 \if j\jsEngine
602   \let\bxjs@everyparhook\bxjs@everyparhook@compat
603 \else
604   \let\bxjs@everyparhook\bxjs@everyparhook@modern
605 \fi
606 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
607 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
608 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
609 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}
```

`\bxjs@label@section` `label-section` の指定値。

```
610 \chardef\bxjs@label@section@none=0
611 \chardef\bxjs@label@section@compat=1
612 \chardef\bxjs@label@section@modern=2
613 \let\bxjs@label@section\bxjs@label@section@compat
614 \def\bxjs@kv@labelsection@none{\let\bxjs@label@section\bxjs@label@section@none}
615 \def\bxjs@kv@labelsection@compat{\let\bxjs@label@section\bxjs@label@section@compat}
616 \def\bxjs@kv@labelsection@modern{\let\bxjs@label@section\bxjs@label@section@modern}
617 \define@key{bxjs}{label-section}{\bxjs@set@keyval{labelsection}{#1}{}}
```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```
618 \newif\ifbxjs@usezw \bxjs@usezwtrue
619 \bxjs@declare@bool@option{use-zw}{usezw}
620 \DeclareOption{nozw}{\setkeys{bxjs}{use-zw=false}}
621 \DeclareOption{zw}{\setkeys{bxjs}{use-zw=true}}
```

`\ifbxjs@disguise@js` [スイッチ] `disguise-js` の指定値。

```
622 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
623 \bxjs@declare@bool@option{disguise-js}{disguise@js}
624 \DeclareOption{nojs}{\setkeys{bxjs}{disguise-js=false}}
625 \DeclareOption{js}{\setkeys{bxjs}{disguise-js=true}}
```

`\ifbxjs@precisetext` [スイッチ] `precise-text` の指定値。

```
626 \newif\ifbxjs@precisetext
627 \bxjs@declare@bool@option{precise-text}{precisetext}
628 \DeclareOption{noprecisetext}{\setkeys{bxjs}{precise-text=false}}
629 \DeclareOption{precisetext}{\setkeys{bxjs}{precise-text=true}}
```

`\ifbxjs@simplejasetup` [スイッチ] `simple-ja-setup` の指定値。

```
630 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
631 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}
632 \DeclareOption{nosimplejasetup}{\setkeys{bxjs}{simple-ja-setup=false}}
633 \DeclareOption{simplejasetup}{\setkeys{bxjs}{simple-ja-setup=true}}
```

## ■ オプションの実行

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

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

```
634 \let\bxjs@org@removeelement\@removeelement
635 \def\@removeelement#1#2#3{%
636   \def\reserved@a{#2}%
637   \ifx\reserved@a\@empty \let#3\@empty
638   \else \bxjs@org@removeelement{#1}{#2}{#3}%
639   \fi}
```

デフォルトのオプションを実行します。multicols や url を \RequirePackage するのはやめました。

```
640 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
641 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
642 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
643 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
644 \ProcessOptions\relax
645 \bxjs@post@option@hook
```

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
```

## ■ グローバルオプションの整理

グローバルオプションのトークン列に { } が含まれていると、やはり後のパッケージの読み込み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。

```
654 \def\bxjs@tmpdo{%
655   \def\bxjs@tmpa{\gobble}%
656   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
657   \let\@classoptionslist\bxjs@tmpa}
658 \def\bxjs@tmpdo@a#1,{%
659   \ifx\@nil#1\relax\else
660     \bxjs@tmpdo@b#1{\@nil
```

```

661 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
662 \expandafter\bxjs@tmpdo@a
663 \fi}
664 \def\bxjs@tmpdo@b#1#{\bxjs@tmpdo@c}
665 \def\bxjs@tmpdo@c#1\@nil{%
666 \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
667 \bxjs@tmpdo

```

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。

```

668 \@expandtwoargs\@removeelement
669 {papersize}\@classoptionslist\@classoptionslist
670 \@expandtwoargs\@removeelement
671 {10pt}\@classoptionslist\@classoptionslist
672 \@expandtwoargs\@removeelement
673 {noscale}\@classoptionslist\@classoptionslist

```

---

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

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

---

ここのコードを削除。

---

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

---

正規化前の和文ドライバの値を `\bxjs@jadriver` に設定する。

```

674 \ifx\bxjs@jadriver@opt\@undefined\else
675 \let\bxjs@jadriver\bxjs@jadriver@opt
676 \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。

```

677 \let\bxjs@tmpb\jsEngine
678 \ifx j\bxjs@tmpb\ifjsWithpTeXng
679 \let\bxjs@tmpb=g
680 \fi\fi
681 \ifx j\bxjs@tmpb\ifjsWithupTeX
682 \let\bxjs@tmpb=u
683 \fi\fi
684 \ifx p\bxjs@tmpb\ifjsInPdfMode\else

```

```
685 \let\bxjs@tmpb=n
```

```
686 \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。)

```
687 \ifx *\bxjs@engine@given
```

```
688 \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。

```
689 \ifx j\bxjs@engine@given
```

```
690 \g@addto@macro\@classoptionslist{,platex}
```

```
691 \else\ifx u\bxjs@engine@given
```

```
692 \g@addto@macro\@classoptionslist{,uplatex}
```

```
693 \fi\fi
```

```
694 \fi
```

```
695 \ifx\bxjs@engine@given\@undefined\else
```

```
696 \ifx\bxjs@engine@given\bxjs@tmpb\else
```

```
697 \ClassError\bxjs@clsname
```

```
698 {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
```

```
699 \fi
```


```
700 \fi
```

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

```
701 \ifjsWithpTeXng
```

```
702 \g@addto@macro\@classoptionslist{,uplatex}
```

```
703 \fi
```

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

```
704 \@tempswatrue
```

```
705 \ifx \bxjs@driver@given\@undefined\else
```

```
706 \ifjsInPdfMode
```

```
707 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
```

```
708 \@tempswafalse
```

```
709 \fi
```

```
710 \else\ifx x\jsEngine
```

```
711 \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
```

```
712 \@tempswafalse
```

```
713 \fi
```

```
714 \else
```

```
715 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
```

```
716 \@tempswafalse
```

```
717 \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
```

```
718 \@tempswafalse
```

```
719 \fi\fi
```

```
720 \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
```

```
721 \@tempswafalse
```

```
722 \fi\fi
```

```

723 \fi\fi
724 \fi
725 \if@tempswa\else
726 \ClassError\bxjs@clsname
727 {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
728 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

729 \ifjsInPdfMode \@tempswafalse
730 \else\ifx x\jsEngine \@tempswafalse
731 \else\ifjsWithpTeXng \@tempswafalse
732 \else \@tempswatrue
733 \fi\fi\fi
734 \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)。

```

735 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@opt\@undefined\else
736 \ClassWarningNoLine\bxjs@clsname
737 {No driver option is given}
738 \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。)

```

739 \ifbxjs@dvi@opt
740 \edef\bxjs@next{%
741 \let\noexpand\bxjs@driver@given
742 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
743 \noexpand\g@addto@macro\noexpand\@classoptionslist
744 {,\bxjs@driver@opt}%
745 }\bxjs@next
746 \fi
747 \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。

```

748 \ifjsWithpTeXng
749 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
750 \let\bxjs@platexng@nodrv\@undefined
751 \else\ifx t\bxjs@platexng@nodrv\else
752 \g@addto@macro\@classoptionslist{,dvipdfmx}
753 \fi\fi
754 \fi

```

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

```

755 \ifx\bxjs@driver@given\bxjs@driver@@none
756 \bxjs@papersizefalse
757 \fi

```

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

※ (u)pTeX 以外で `ja` を省略するのは 2.0 版より非推奨となった。

```

758 \def\bxjs@@minimal{minimal}
759 \ifx\bxjs@jadriver\relax
760   \ifx j\jsEngine
761     \def\bxjs@jadriver{standard}
762   \else
763     \ClassWarningNoLine\bxjs@clsname
764     {The option 'ja' is MISSING!!\MessageBreak
765      So 'ja=minimal' is assumed as fallback, but\MessageBreak
766      such implicit setting is now DEPRECATED!\MessageBreak
767      You should write 'ja=minimal' explicitly,\MessageBreak
768      if it is intended}
769     \let\bxjs@jadriver\bxjs@@minimal
770   \fi
771 \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。

```

772 \ifx\bxjs@jadriver@opt\@undefined\else
773   \ifx\bxjs@engine@given\@undefined
774     \ClassError\bxjs@clsname
775     {An engine option must be explicitly given}%
776     {When you use a Japanese-driver you must specify a correct\MessageBreak
777      engine option.\MessageBreak\@ehc}
778 \fi\fi

```

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

```

779 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
780   \ifx\bxjs@magstyle\bxjs@magstyle@mag
781     \let\bxjs@magstyle\bxjs@magstyle@default
782     \ClassError\bxjs@clsname
783     {The engine does not support 'magstyle=usemag'}%
784     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
785      The default value 'nomag*' is used instead.\MessageBreak \@ehc}
786   \fi
787 \fi

```

`base`、`jbase`、`scale` の値を用いて和文スケール値を解決する。

※`\bxjs@param@basefontsize` と `\jsScale` へのオプション値の反映は既に実施されていることに注意。`jbase` 非指定の場合はこのままでよい。

```

788 \ifx\bxjs@jbase@opt\@undefined\else
789   \ifx\bxjs@base@opt\@undefined

```

`jbase` 指定済で `base` 未指定の場合は、`\jsScale` の値を採用して和文基底サイズを決定



する。

```
790 \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
791 \bxjs@invscale\@tempdima\jsScale
792 \bxjs@setbasefontsize{\@tempdima}%
793 \else
    jbase と base がともに指定済の場合は、それらの値から和文スケール値を決定する。
794 \ifx\bxjs@scale@opt\@undefined\else
795 \ClassWarningNoLine\bxjs@clsname
796 {Redundant 'scale' option is ignored}%
797 \fi
798 \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
799 \@tempdimb=\bxjs@param@basefontsize\relax
800 \edef\jsScale{\strip@pt\@tempdimb}%
801 \bxjs@invscale\@tempdima\jsScale
802 \edef\jsScale{\strip@pt\@tempdima}%
803 \fi
804 \fi
```

`\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。

```
805 \let\Cjascale\jsScale
```

---

---

8bit 欧文  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  の場合は、高位バイトをアクティブ化しておく。（和文を含むマクロ定義を  
通用させるため。）

```
806 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
807 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
808 \catcode\@tempcnta\active
809 \advance\@tempcnta\@ne
810 \repeat
811 \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。

```
812 \ifbxjs@disguise@js
813 %<book|report>\def\bxjs@js@clsname{jsbook}
814 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
815 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
816 \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。

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

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

```
818 \ifbxjs@oldfontcommands
819 \AtEndOfClass{\allowoldfontcommands}
```

---

■papersize スペシャルの出力 dvi ファイルの先頭に dvips の papersize special を書き込むことで、出力用紙サイズを設定します。これは dvipdfmx や最近の dviout にも有効です。どうやら papersize special には true 付の単位は許されず、かつ単位は常に true なものと扱われるようです。そこで、後で出てくる（☆）の部分、「\mag にあわせてスケール」よりも手前で実行しておくことになります。

トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> はトンボ出力幅を両側に 1 インチとっていますので、dvips 使用時に

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

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

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

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

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

---

BXJS では出力用紙サイズ記録は geometry パッケージが行う。

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

---

```
821 \newdimen\stockwidth \newdimen\stockheight
822 \begingroup\expandafter\expandafter\expandafter\endgroup
823 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
824 % \newdimen\stockwidth \newdimen\stockheight
825 \setlength{\stockwidth}{\paperwidth}
826 \setlength{\stockheight}{\paperheight}
827 \advance \stockwidth 2in
828 \advance \stockheight 2in
829 \fi
```

### ■基準となる行送り

\n@baseline 基準となる行送りをポイント単位で表したものです。

```
830 %<slide>\def\n@baseline{13}%
831 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
832 %<!slide>\else \def\n@baseline{16}\fi
```

### ■拡大率の設定

---

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

```

833 \ifx\bxjs@magstyle\bxjs@magstyle@mag
834   \jsc@magtrue
835 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
836   \jsc@mag@xrealtrue
837 \fi\fi

```

---

サイズの変更は  $\text{\TeX}$  のプリミティブ  $\text{\mag}$  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 /  $\text{\mag}$  に相当する  $\text{\inv@mag}$  を定義しました。truein を使っていたところを  $\text{\inv@mag in}$  に直しましたので、geometry パッケージと共存できると思います。なお、新ドキュメントクラス側で 10pt 以外にする場合の注意：

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

---

設定すべき  $\text{\mag}$  値を (基底サイズ)/(10 pt)  $\times$  1000 と算出。BXJS クラスでは、 $\text{\mag}$  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。

```

838 \ifx\bxjs@param@mag\relax
839   \@tempdima=\bxjs@param@basefontsize
840   \advance\@tempdima.001pt \multiply\@tempdima25
841   \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
842   \edef\bxjs@param@mag{\the\@tempcmta}
843 \else
844 % mag 値が直接指定された場合
845   \bxjs@gset@tempcmta{\bxjs@param@mag}
846   \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
847 % 有効な mag 値の範囲は 1--32768
848   \edef\bxjs@param@mag{\the\@tempcmta}
849   \advance\@tempcmta100000
850   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
851   \expandafter\bxjs@tmpa\the\@tempcmta\@nil
852   \edef\bxjs@param@basefontsize{\the\@tempdima}
853 \fi
854 \@tempcmta\bxjs@param@mag \advance\@tempcmta100000
855 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
856 \expandafter\bxjs@tmpa\the\@tempcmta\@nil
857 \edef\jsc@magscale{\strip@pt\@tempdima}
858 \let\jsBaseFontSize\bxjs@param@basefontsize

```

---

[2016-07-08]  $\text{\jsc@mpt}$  および  $\text{\jsc@mmm}$  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。

---

$\text{\mag}$  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、 $\text{\jsc@mpt}$  を  $\text{\p@?}$  と書く。その上で、 $\text{\mag}$  する場合は  $\text{\p@}$  を無視して  $\text{\p@}$  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。

```

859 \newdimen\jsc@mpt
860 \newdimen\jsc@mmm
861 \ifjsc@mag
862   \jsc@mpt=1\p@
863   \jsc@mmm=1mm
864   \catcode`\?=9 % \p@? read as \p@
865 \else
866   \jsc@mpt=\jsc@magscale\p@
867   \jsc@mmm=\jsc@magscale mm
868   \catcode`\?=11 \let\p@?\jsc@mpt
869 \fi
870 \chardef\bxjs@qmc=\catcode`\?\relax
871 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode`\?=12\relax}

```

ここで pTeX の `zw` に相当する単位として用いる長さ変数 `\jsZw` を作成する。約束により、これは `\jsScale` × (指定フォントサイズ) に等しい。

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

```

872 \newdimen\jsZw
873 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
874 \ifbxjs@usezw
875   \providecommand*\zw{\jsZw}
876 \fi

```

`\zwspace` 全角幅の水平空き。

```

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

```

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

```

878 \ifjsc@mag@xreal
879   \RequirePackage{type1cm}
880   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

---

```

881 \ifbxjs@TUenc
882   \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
883 \else
884   \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
885 \fi
886 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
887 \let\jsc@get@external@font\get@external@font
888 \def\get@external@font{%
889   \jsc@preadjust@extract@font
890   \jsc@get@external@font}
891 \def\jsc@fstrunc#1{%
892   \edef\jsc@tmpa{\strip@pt#1}%
893   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}

```

```

894 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
895   \if#5*\else
896     \edef\jsc@tmpa{#1%
897       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
898   \fi}
899 \def\jsc@preadjust@extract@font{%
900   \let\jsc@req@size\fontsize
901   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
902   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
903   \let\jsc@ref@size\jsc@tmpa
904   \let\fontsize\jsc@ref@size}
905 \def\execute@size@function#1{%
906   \let\jsc@cref@size\fontsize
907   \let\fontsize\jsc@req@size
908   \csname s@fct@#1\endcsname}
909 \let\jsc@DeclareErrorFont\DeclareErrorFont
910 \def\DeclareErrorFont#1#2#3#4#5{%
911   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
912   \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
913   \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
914 \def\gen@sfcnt{%
915   \def\mandatory@arg{\mandatory@arg\jsc@cref@size}%
916   \empty@sfcnt}
917 \def\genb@sfcnt{%
918   \edef\mandatory@arg{%
919     \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@{}}%
920   \empty@sfcnt}
921 \ifbxjs@TUenc\else
922   \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
923 \fi
924 \fi

```

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

```

\jsc@smallskip
\jsc@medskip 925 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 926 %\def\jsc@medskip{\vspace\jsc@medskipamount}
927 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 928 \newskip\jsc@smallskipamount
929 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
\jsc@bigskipamount 930 %\newskip\jsc@medskipamount
931 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
932 %\newskip\jsc@bigskipamount
933 %\jsc@bigskipamount =12\jsc@mpt plus 4\jsc@mpt minus 4\jsc@mpt

```

`\paperwidth`, `\paperheight` を `\mag` にあわせてスケールしておきます (☆)。

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

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

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

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

```
934 % \ifpapersize
935 %   \setlength{\@tempdima}{\paperwidth}
936 %   \setlength{\@tempdimb}{\paperheight}
937 %   \iftombow
938 %     \advance \@tempdima 2truein
939 %     \advance \@tempdimb 2truein
940 %   \fi
941 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
942 % \fi
```

### 3 和文フォントの変更

---

和文フォントの設定は和文ドライバの管轄。

---

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

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

---

BXJS クラスでの変更点:

- `fix-at-cmd` オプションが偽の場合は再定義しない。
- 固定の 3000 でなく実際のピリオドの `sfcode` 値を使う。
- 「防御的な `\@`」での不具合を防ぐため、大文字直後の `\@` は標準と同等の動作にする。

---

```
943 \chardef\bxjs@periodchar=`\
944 \bxjs@protected\def\bxjs@SE{%
945   \ifnum\spacefactor<\@m \spacefactor\@m
946   \else \spacefactor\sfcodes\bxjs@periodchar
947   \fi}
948 \ifbxjs@fix@at@cmd
949   \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 pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> で 0pt plus .4pt minus .5pt に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

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

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

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

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

```
951 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
952 \expandafter\bxjs@tmpa\expandafter{%
953   \set@fontsize{#1}{#2}{#3}%
954 % 末尾にコードを追加
955   \expandafter\def\expandafter\size@update\expandafter{%
956     \size@update
957     \jsFontSizeChanged}%
```

958 }

`\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。

```
959 \newcommand*\jsFontSizeChanged{%
960   \jsZw=\f@size\p@
961   \jsZw=\jsScale \jsZw
962   \ifdim\parindent>\z@
963     \if@english \parindent=1em
964     \else       \parindent=1\jsZw
965     \fi
966   \fi\relax
967   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
968 \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` の変わりに用いることにします。

```
969 \ifjsc@mag
970   \let\jsc@setfontsize\@setfontsize
971 \else
972   \def\jsc@setfontsize#1#2#3{%
973     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
974 % microtype 対策
975 \ifjsWitheTeX\if j\jsEngine\else
976   \def\jsc@setfontsize#1#2#3{%
977     \edef\bxjs@sfs@next{%
978       \unexpanded{\@setfontsize#1}%
979       {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
980     }\bxjs@sfs@next}
981   \fi\fi
982 \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。

---

```
983 \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。

---

```
984 \newif\ifnarrowbaselines
985 \if@english
986   \narrowbaselinestrue
987 \fi
988 \def\narrowbaselines{%
989   \narrowbaselinestrue
990   \skip0=\abovedisplayskip
991   \skip2=\abovedisplayshortskip
992   \skip4=\belowdisplayskip
993   \skip6=\belowdisplayshortskip
994 % 一時的に警告を無効化する
995   \let\bxjs@save@nomath\@nomath
996   \let\@nomath\@gobble
997   \@currsize\selectfont
998   \let\@nomath\bxjs@save@nomath
999   \abovedisplayskip=\skip0
1000  \abovedisplayshortskip=\skip2
1001  \belowdisplayskip=\skip4
1002  \belowdisplayshortskip=\skip6\relax}
1003 \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。

```
1004 \def\bxjs@ifnarrowbaselines{%
1005   \ifnarrowbaselines\expandafter\@firstoftwo
1006   \else \expandafter\@secondoftwo
1007   \fi
1008 }
```

---

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

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

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

---

microtype 対策のため if 文を避ける。

---

```
1009 \renewcommand{\normalsize}{%
1010   \bxjs@if@narrowbaselines{%
1011     \jsc@setfontsize\normalsize\@xpt\@xiipt
1012   }{%else
1013     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1014   }%
```

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。

```
1015 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1016 \abovedisplayshortskip \z@ \@plus3\p@?
1017 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1018 \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。

```
1019 \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。

```
1020 \normalsize
```

\Cht 基準となる長さの設定をします。pT<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 [2017-08-31] 基準とする文字を「全角空白」(EUC コード 0xA1A1) から「漢」(JIS コー  
\Cvs ド 0x3441) へ変更しました。

---

\Chs

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

```
1021 \ifx\Cht\@undefined \newdimen\Cht \fi
1022 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1023 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1024 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1025 \ifx\Chs\@undefined \newdimen\Chs \fi
```

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

---

```
1026 \setlength\Cht{0.88\jsZw}
```

```

1027 \setlength\Cdp{0.12\jsZw}
1028 \setlength\Cwd{1\jsZw}
1029 \setlength\Cvs{\baselineskip}
1030 \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。

---

```

1031 \newcommand{\small}{%
1032   \bxjs@if@narrowbaselines{%
1033     <!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1034     <kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1035   }{%else
1036     <!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1037     <kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1038   }%
1039   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1040   \abovedisplayshortskip \z@ \@plus3\p@?
1041   \belowdisplayskip \abovedisplayskip
1042   \belowdisplayshortskip \belowdisplayskip
1043   \def\@listi{\leftmargin\leftmargini
1044             \topsep \z@
1045             \parsep \z@
1046             \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。

```

1047 \newcommand{\footnotesize}{%
1048   \bxjs@if@narrowbaselines{%
1049     <!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1050     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1051   }{%else
1052     <!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1053     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1054   }%
1055   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1056   \abovedisplayshortskip \z@ \@plus2\p@?
1057   \belowdisplayskip \abovedisplayskip
1058   \belowdisplayshortskip \belowdisplayskip
1059   \def\@listi{\leftmargin\leftmargini
1060             \topsep \z@
1061             \parsep \z@
1062             \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、`\large` 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  
`\Large` 行が揃うようにします。

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1063 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1064 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 1065 \if@twocolumn
\LARGE 1066 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1067 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1068 \else
1069 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
1070 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1071 \fi
1072 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1073 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1074 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1075 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1076 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1077 \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` 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。

```
1078 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}
```

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

見出し用のフォントは `\bfseries` 固定ではなく、`\headfont` という命令で定めることにします。これは太ゴシックが使えるときは `\sffamily \bfseries` でいいと思いますが、通常の中ゴシックでは単に `\sffamily` だけのほうがよさそうです。『`LaTeX 2ε` 美文書作成入門

```

1079 % \newcommand{\headfont}{\bfseries}
1080 \newcommand{\headfont}{\sffamily}
1081 % \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 でしたが、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。

```
1082 %<!kiyou>\setlength\columnsep{2\Cwd}
1083 %<kiyou>\setlength\columnsep{28truebp}
1084 \setlength\columnseprule{\z@}
```

## ■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  
`\normallineskip` ます。元は 0pt でしたが 1pt に変更しました。normal... の付いた方は保存用です。

```
\lineskiplimit 1085 \setlength\lineskip{1\jsc@empt}
\normallineskip 1086 \setlength\normallineskip{1\jsc@empt}
\normallineskiplimit 1087 \setlength\lineskiplimit{1\jsc@empt}
1088 \setlength\normallineskiplimit{1\jsc@empt}
```

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

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

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

```
1089 \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。

```
1090 \setlength\parskip{\z@}
1091 \if@slide
1092 \setlength\parindent{0\p@}
1093 \else
1094 \setlength\parindent{1\Cwd}
1095 \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 1096 \@lowpenalty 51
1097 \@medpenalty 151
1098 \@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。

```
1099 % \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。

```
1100 % \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

---

`\bxjs@bd@pre@geometry@hook` `begin-document` フックのコード内で、`geometry` パッケージが挿入するコードの直前で実行されるフック。

```
1101 \onlypreamble\bxjs@bd@pre@geometry@hook
1102 \let\bxjs@bd@pre@geometry@hook\empty
```

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
1103 \ifjsc@mag
1104 \mag=\bxjs@param@mag
1105 \fi
1106 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLLength` のための和文単位の定義。

```
1107 \def\bxjs@unit@trueQ{0.25truemm}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1108 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw
```

`\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。

```
1109 \def\bxjs@tmpdo{\futurelet\bxjs@tmpa\bxjs@tmpdo@a}
1110 \def\bxjs@tmpdo@a{%
1111   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@tmpdo@b
1112   \else \expandafter\bxjs@tmpdo@c \fi}
1113 \def\bxjs@tmpdo@b#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1114 \def\bxjs@tmpdo@c#1\@nil{\bxjs@tmpdo@d#1,,\@nil}
1115 \def\bxjs@tmpdo@d#1,#2,#3\@nil{%
1116   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1117 \expandafter\bxjs@tmpdo\bxjs@param@paper\@empty\@empty\@nil
```

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

```
1118 \edef\bxjs@layout@paper{%
1119   \ifjsc@mag truedimen,\fi
1120   \if@landscape landscape,\fi
1121   \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。

```
1122 %<*article|report>
1123 \def\bxjs@layout@base{%
1124   headheight=\topskip,footskip=0.03367\paperheight,%
1125   headsep=\footskip-\topskip,includeheadfoot,%
1126 }
1127 \edef\bxjs@layout{\bxjs@layout@base
1128   hscale=0.76,hmarginratio=1:1,%
1129   vscale=0.83,vmarginratio=1:1,%
```

```

1130 }
1131 %</article|report>
1132 %<*book>
1133 \def\bxjs@layout@base{%
1134   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1135 }
1136 \ifbxjs@layout@buggyhmargin    %---
1137 % アレ
1138 \edef\bxjs@layout{\bxjs@layout@base
1139   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1140   vscale=0.83,vmarginratio=1:1,%
1141 }
1142 \else    %---
1143 % 非アレ
1144 \edef\bxjs@layout{\bxjs@layout@base
1145   hmargin=18\jsc@mmm,%
1146   vscale=0.83,vmarginratio=1:1,%
1147 }
1148 \fi    %---
1149 %</book>
1150 %<*slide>
1151 \def\bxjs@layout@base{%
1152   noheadfoot,%
1153 }
1154 \edef\bxjs@layout{\bxjs@layout@base
1155   hscale=0.9,hmarginratio=1:1,%
1156   vscale=0.944,vmarginratio=1:1,%
1157 }
1158 %</slide>

```

textwidth オプションの設定を反映する。

```

1159 %<!*book>
1160 \ifx\bxjs@textwidth@opt\@undefined\else
1161   \jsSetQHLengh\@tempdima{\bxjs@textwidth@opt}
1162   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1163 \fi
1164 %</!book>
1165 \ifx\bxjs@number@of@lines@opt\@undefined\else
1166   \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1167   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1168 \fi

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1169 \newdimen\fullwidth
```

\bxjs@textwidth@limit [寸法値マクロ] bxjsbook における、\textwidth 上限の値。

\jsTextWidthLimit [実数値マクロ] \bxjs@textwidth@limit の全角 (\Cwd) 単位での値。

```
1170 %<*book>
```

```

1171 \newcommand\jsTextWidthLimit{40}
1172 \@tempdima=\jsTextWidthLimit\Cwd
1173 \ifx\bxjs@textwidth@limit@opt\@undefined\else
1174   \bxjs@gset@tempcnta{\bxjs@textwidth@limit@opt}
1175   \@tempdima=\@tempcnta\Cwd
1176 \fi
1177 \ifx\bxjs@textwidth@opt\@undefined\else
1178   \jsSetQHLengh\@tempdima{\bxjs@textwidth@opt}
1179 \fi
1180 \edef\bxjs@textwidth@limit{\the\@tempdima}
1181 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1182   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1183   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1184 \fi
1185 %</book>

```

`\bxjs@preproc@layout` geometry の前処理。

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。

```

1186 \def\bxjs@preproc@layout{%
1187   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mpt}

```

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

```

1188 \def\bxjs@postproc@layout{%
  geometry のドライバを再設定する。
1189   \ifx\bxjs@geometry@driver\relax\else
1190     \let\Gm@driver\bxjs@geometry@driver
1191   \fi
  \ht\strutbox の値を元に戻す。
1192   \ht\strutbox=\bxjs@save@ht@strutbox\relax
  \textwidth の値を補正する。
1193   \ifbxjs@whole@zw@lines
1194     \@tempdimb=\textwidth
1195     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1196     \advance\textwidth.005pt\relax
1197     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1198     \advance\@tempdimb-\textwidth
1199     \advance\oddsidemargin 0.5\@tempdimb
1200     \advance\evensidemargin 0.5\@tempdimb
1201   \fi
1202   \fullwidth=\textwidth

```

`bxjsbook` の場合は、geometry が設定した `\textwidth` は `\fullwidth` として扱い、その値から実際の `\textwidth` を導出する。

```

1203 %<*book>

```



```

1204 \@tempdima=\bxjs@textwidth@limit\relax
1205 \ifbxjs@whole@zw@lines
1206   \advance\@tempdima.005pt\relax
1207   \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1208 \fi
1209 \ifdim\textwidth>\@tempdima
1210   \textwidth=\@tempdima
1211   \addtolength\evensidemargin{\fullwidth-\textwidth}
1212 \fi
1213 %</book>

\textheight 関連の調整。
1214 \@tempdimb=\textheight
1215 \advance\textheight-\topskip
1216 \advance\textheight.005pt\relax
1217 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1218 \advance\textheight\topskip
1219 \advance\@tempdimb-\textheight
1220 \advance\topmargin0.5\@tempdimb

\headheight 関連の調整。
1221 \@tempdima=\topskip
1222 \advance\headheight\@tempdima
1223 \advance\topmargin-\@tempdima

marginpar 関連の調整。
1224 \setlength\marginparsep{\columnsep}
1225 \setlength\marginparpush{\baselineskip}
1226 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1227   -\textwidth-10\jsc@mmm-\marginparsep}
1228 \ifbxjs@whole@zw@lines
1229   \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1230 \fi

連動する変数。
1231 \maxdepth=.5\topskip
1232 \stockwidth=\paperwidth
1233 \stockheight=\paperheight
1234 }

```

`\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。

```

1235 \edef\jsGeometryOptions{%
1236   \bxjs@layout@paper,\bxjs@layout}

```

---

## ■ geometry パッケージを読み込む

---

`geoemtry=class` の場合に、実際に `geometry` パッケージを読みこむ。

```
1237 \ifx\bxjs@geometry\bxjs@geometry@class
```

`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。

```
1238 \ifbxjs@papersize
1239   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1240     \PassOptionsToPackage{dvipdfm}{geometry}
1241   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1242     \PassOptionsToPackage{dvipdfm}{geometry}
1243   \fi\fi
1244   \let\bxPapersizeSpecialDone=t
1245 \else
1246   \PassOptionsToPackage{driver=none}{geometry}
1247 \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。

```
1248 \AtBeginDocument{\bxjs@bd@pre@geometry@hook}
1249 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1250 \bxjs@preproc@layout
1251 \edef\bxjs@next{%
1252   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1253 }\bxjs@next
1254 \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。

```
1255 \let\bxjs@geometry@driver\Gm@driver
1256 \bxjs@postproc@layout
```

`geometry` のドライバ自動判別に対する前処理。

```
1257 \g@addto@macro\bxjs@bd@pre@geometry@hook{%
```

BXJS2.0 版より、`geometry` の 4.x 版のサポートは廃止された。

```
1258   \@ifpackagelater{geometry}{2010/02/12}{\relax}{\else
1259     \PackageError\bxjs@clsname
1260       {Your 'geometry' package is too old (< v5.0)}%
1261       {\@ehc}%
1262     \let\Gm@driver\relax}%
```

エンジンが `platex-ng` の時は `geometry` のドライバを `pdftex` にする。

```
1263   \ifjsWithpTeXng
1264     \ifx\Gm@driver\@empty
1265       \def\Gm@driver{pdftex}%
1266     \fi
1267   \fi}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```
1268 \def\setpagelayout{%
1269   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1270     \@ifstar{\bxjs@setpagelayout@a@ne}{\bxjs@setpagelayout@a@z@}}
1271 \def\bxjs@setpagelayout@a#1#2{%
1272   \ifcase#1% modify
1273     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1274   \or% reset(*)
1275     \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1276   \or% semireset(+)
1277     \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1278   \fi
1279   \bxjs@preproc@layout
1280   \edef\bxjs@next{%
1281     \noexpand\geometry{\bxjs@next}%
1282   }\bxjs@next
1283   \bxjs@postproc@layout}
```

---

#### ■geometry パッケージを読み込まない

---

`geometry=user` の場合の処理。

```
1284 \else\ifx\bxjs@geometry\bxjs@geometry@user
```

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

※`\jsUseMinimalPageLayout` は動作テスト用。

```
1285 \g@addto@macro\bxjs@begin@document@hook{%
1286   \ifdim\textwidth=.5\maxdimen
1287     \ClassError\bxjs@clsname
1288       {Page layout is not properly set}%
1289     {\@ehd}%
1290   \fi}
1291 \def\jsUseMinimalPageLayout{%
1292   \setlength{\textwidth}{6.5in}%
1293   \setlength{\textheight}{8in}}
```

`\setpagelayout` はとりあえず無効にしておく。

```
1294 \let\bxjs@geometry@driver\relax
1295 \def\setpagelayout{%
1296   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1297     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1298 \def\bxjs@pagelayout@a#1{%
1299   \ClassError\bxjs@clsname
1300     {Command '\string\setpagelayout' is not supported,\MessageBreak
1301     because 'geometry' value is not 'class'}\@eha}
```

```
1302 %
1303 \fi\fi
```

---

## ■JS クラスと共通処理の開始

---

ここからのコードは以下の点を除いて JS クラスのものを踏襲する。

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

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

---

```
1304 %<*jsclasses>
```

## ■縦方向のスペース

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

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

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

```
1305 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1306 \if@slide
1307   \setlength\headheight{0\jsc@mpt}
1308 \else
1309   \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
    06-26)
1310 \fi
```

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

```
1311 %<*article|kiyou>
1312 \if@slide
1313   \setlength\footskip{0pt}
1314 \else
1315   \setlength\footskip{0.03367\paperheight}
1316   \ifdim\footskip<\baselineskip
1317     \setlength\footskip{\baselineskip}
```

```

1318 \fi
1319 \fi
1320 %</article|kiyou>
1321 %<jspf>\setlength\footskip{9\jsc@mmm}
1322 %<*book>
1323 \if@report
1324 \setlength\footskip{0.03367\paperheight}
1325 \ifdim\footskip<\baselineskip
1326 \setlength\footskip{\baselineskip}
1327 \fi
1328 \else
1329 \setlength\footskip{0pt}
1330 \fi
1331 %</book>
1332 %<*report>
1333 \setlength\footskip{0.03367\paperheight}
1334 \ifdim\footskip<\baselineskip
1335 \setlength\footskip{\baselineskip}
1336 \fi
1337 %</report>

```

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

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

```

1338 %<*article>
1339 \if@slide
1340 \setlength\headsep{0\jsc@empt}
1341 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1342 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1343 \else
1344 \setlength\headsep{\footskip}
1345 \addtolength\headsep{-\topskip}
1346 \fi
1347 %</article>
1348 %<*book>
1349 \if@report
1350 \setlength\headsep{\footskip}
1351 \addtolength\headsep{-\topskip}
1352 \else
1353 \setlength\headsep{6\jsc@mmm}
1354 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1355 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1356 \fi
1357 %</book>
1358 %<*report>
1359 \setlength\headsep{\footskip}

```

```

1360 \addtolength\headsep{-\topskip}
1361 %</report>
1362 %<*jspf>
1363 \setlength\headsep{9\jsc@mmm}
1364 \addtolength\headsep{-\topskip}
1365 %</jspf>
1366 %<*kiyou>
1367 \setlength\headheight{0\jsc@mpt}
1368 \setlength\headsep{0\jsc@mpt}
1369 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1370 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1371 %</kiyou>

```

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

```

1372 \setlength\maxdepth{.5\topskip}

```

#### ■本文の幅と高さ

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

```

1373 \newdimen\fullwidth

```

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

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

```

1374 %<*article>
1375 \if@slide
1376   \setlength\fullwidth{0.9\paperwidth}
1377 \else
1378   \setlength\fullwidth{0.76\paperwidth}
1379 \fi
1380 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1381 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1382 \setlength\textwidth{\fullwidth}
1383 %</article>
1384 %<*book>
1385 \if@report

```

```

1386 \setlength\fullwidth{0.76\paperwidth}
1387 \else
1388 \setlength\fullwidth{\paperwidth}
1389 \addtolength\fullwidth{-36\jsc@mmm}
1390 \fi
1391 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1392 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1393 \setlength\textwidth{\fullwidth}
1394 \if@report \else
1395 \if@twocolumn \else
1396 \ifdim \fullwidth>40zw
1397 \setlength\textwidth{40zw}
1398 \fi
1399 \fi
1400 \fi
1401 %</book>
1402 %<*report>
1403 \setlength\fullwidth{0.76\paperwidth}
1404 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1405 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1406 \setlength\textwidth{\fullwidth}
1407 %</report>
1408 %<*jspf>
1409 \setlength\fullwidth{50zw}
1410 \addtolength\fullwidth{8\jsc@mmm}
1411 \setlength\textwidth{\fullwidth}
1412 %</jspf>
1413 %<*kiyou>
1414 \setlength\fullwidth{48zw}
1415 \addtolength\fullwidth{\columnsep}
1416 \setlength\textwidth{\fullwidth}
1417 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

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

[2016-10-08] `article` の `slide` のときに `\headheight` はゼロなので、さらに修正しました

(2016-08-17 での修正漏れ)。

```
1418 %<*article|book|report>
1419 \if@slide
1420   \setlength{\textheight}{0.95\paperheight}
1421 \else
1422   \setlength{\textheight}{0.83\paperheight}
1423 \fi
1424 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1425 \addtolength{\textheight}{-\headsep}
1426 \addtolength{\textheight}{-\footskip}
1427 \addtolength{\textheight}{-\topskip}
1428 \divide\textheight\baselineskip
1429 \multiply\textheight\baselineskip
1430 %</article|book|report>
1431 %<jspf>\setlength{\textheight}{51\baselineskip}
1432 %<kiyou>\setlength{\textheight}{47\baselineskip}
1433 \addtolength{\textheight}{\topskip}
1434 \addtolength{\textheight}{0.1\jsc@mpt}
1435 %<jspf>\setlength{\mathindent}{10\jsc@mmm}
```

`\flushbottom` [2016-07-18] `\textheight` に念のため 0.1 ポイント余裕を持たせているのと同様に,  
`\flushbottom` にも余裕を持たせます。元の L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> での完全な `\flushbottom` の定  
義は

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

ですが、次のようにします。

```
1436 \def\flushbottom{%
1437   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1438   \let\@texttop\relax}
```

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

```
1439 \setlength\marginparsep{\columnsep}
1440 \setlength\marginparpush{\baselineskip}
```

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

```
1441 \setlength{\oddsidemargin}{\paperwidth}
1442 \addtolength{\oddsidemargin}{-\fullwidth}
1443 \setlength{\oddsidemargin}{.5\oddsidemargin}
1444 \iftombow
1445   \addtolength{\oddsidemargin}{-1in}
1446 \else
```



```

1447 \addtolength{\oddsidemargin}{-\inv@mag in}
1448 \fi
1449 \setlength{\evensidemargin}{\oddsidemargin}
1450 \if@mparswitch
1451 \addtolength{\evensidemargin}{\fullwidth}
1452 \addtolength{\evensidemargin}{-\textwidth}
1453 \fi

```

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

```

1454 \setlength\marginparwidth{\paperwidth}
1455 \addtolength\marginparwidth{-\oddsidemargin}
1456 \addtolength\marginparwidth{-\inv@mag in}
1457 \addtolength\marginparwidth{-\textwidth}
1458 \addtolength\marginparwidth{-10\jsc@mmm}
1459 \addtolength\marginparwidth{-\marginparsep}
1460 \@tempdima=1zw
1461 \divide\marginparwidth\@tempdima
1462 \multiply\marginparwidth\@tempdima

```

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

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

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

```

1463 \setlength\topmargin{\paperheight}
1464 \addtolength\topmargin{-\textheight}
1465 \if@slide
1466 \addtolength\topmargin{-\headheight}
1467 \else
1468 \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1469 \fi
1470 \addtolength\topmargin{-\headsep}
1471 \addtolength\topmargin{-\footskip}
1472 \setlength\topmargin{0.5\topmargin}
1473 %<kiyou>\setlength\topmargin{81truebp}
1474 \iftombow
1475 \addtolength\topmargin{-1in}
1476 \else
1477 \addtolength\topmargin{-\inv@mag in}
1478 \fi
1479 %</jsclasses>

```

■脚注

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

---

ここは元々は

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

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

---

```
1480 \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。

```
1481 \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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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。

```
1482 \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。

```
1483 \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。

```
1484 \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。

```
1485 \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。

```
1486 \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。

```
1487 \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。

```
1488 \renewcommand{\floatpagefraction}{.8}
```

`\cdbltopnumber` 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。

```
1489 \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。

```
1490 \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。

```
1491 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1492 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1493 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1494 \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 1495 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1496 \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 1497 \setlength\@fptop{0\p@? \@plus 1fil}
1498 \setlength\@fpsep{8\p@? \@plus 2fil}
1499 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpptop` 段抜きフロートについての値です。

```
\@dblfpsep 1500 \setlength\@dblfpptop{0\p@? \@plus 1fil}
\@dblfpbot 1501 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1502 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

## 6 改ページ（日本語 TeX 開発コミュニティ版のみ）

`\pltx@cleartorightpage` [2017-02-24] コミュニティ版 pLaTeX の標準クラス 2017/02/15 に合わせて、同じ命令を追

`\pltx@cleartoleftpage` 加しました。

`\pltx@cleartooddpage`

`\pltx@cleartoevenpage`

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

となっています。

```
1503 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1504 % \ifodd\c@page
1505 % \iftdir
1506 % \hbox{}\thispagestyle{empty}\newpage
1507 % \if@twocolumn\hbox{}\newpage\fi
1508 % \fi
1509 % \else
1510 % \ifydir
1511 % \hbox{}\thispagestyle{empty}\newpage
1512 % \if@twocolumn\hbox{}\newpage\fi
1513 % \fi
1514 % \fi\fi}
1515 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1516 % \ifodd\c@page
1517 % \ifydir
1518 % \hbox{}\thispagestyle{empty}\newpage
1519 % \if@twocolumn\hbox{}\newpage\fi
1520 % \fi
1521 % \else
1522 % \iftdir
1523 % \hbox{}\thispagestyle{empty}\newpage
1524 % \if@twocolumn\hbox{}\newpage\fi
1525 % \fi
1526 % \fi\fi}
1527 \def\pltx@cleartooddpage{\clearpage\if@twoside
1528 \ifodd\c@page\else
1529 \hbox{}\thispagestyle{empty}\newpage
1530 \if@twocolumn\hbox{}\newpage\fi
1531 \fi\fi}
1532 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1533 \ifodd\c@page
1534 \hbox{}\thispagestyle{empty}\newpage
1535 \if@twocolumn\hbox{}\newpage\fi
1536 \fi\fi}
```

---

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

```
1537 \let\pltx@cleartorightpage\pltx@cleartooddpage
1538 \let\pltx@cleartoleftpage\pltx@cleartoevenpage
```

---

`\cleardoublepage` [2017-02-24] コミュニティ版 p<sub>A</sub>T<sub>E</sub>X の標準クラス 2017/02/15 に合わせて、`report` と `book` クラスの場合に `\cleardoublepage` を再定義します。

```
1539 %<*book|report>
1540 \if@openleft
1541 \let\cleardoublepage\pltx@cleartoleftpage
1542 \else\if@openright
```

```

1543 \let\cleardoublepage\pltx@cleartorightpage
1544 \fi\fi
1545 %</book|report>

```

## 7 ページスタイル

ページスタイルとして、 $\text{\LaTeX 2}_\epsilon$  (欧文版) の標準クラスでは `empty`, `plain`, `headings`, `myheadings` があります。このうち `empty`, `plain` スタイルは  $\text{\LaTeX 2}_\epsilon$  本体で定義されています。

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

`\@evenhead` `\@oddhead`, `\@oddfoot`, `\@evenhead`, `\@evenfoot` は偶数・奇数ページの柱 (ヘッダ, フッタ) を出力する命令です。これらは `\fullwidth` 幅の `\hbox` の中で呼び出されます。  
`\@evenfoot` `\ps@...` の中で定義しておきます。  
`\@oddfoot` 柱の内容は, `\chapter` が呼び出す `\chaptermark{何々}`, `\section` が呼び出す `\sectionmark{何々}` で設定します。柱を扱う命令には次のものがあります。

```

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

```

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

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

`\ps@empty` `empty` ページスタイルの定義です。 $\text{\LaTeX}$  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。

```

1546 % \def\ps@empty{%
1547 %   \let\mkboth\gobbletwo
1548 %   \let\oddhead\empty
1549 %   \let\oddfoot\empty
1550 %   \let\evenhead\empty
1551 %   \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。

```

1552 \def\ps@plainfoot{%

```

```

1553 \let\@mkboth\@gobbletwo
1554 \let\@oddhead\@empty
1555 \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1556 \let\@evenhead\@empty
1557 \let\@evenfoot\@oddfoot}
1558 \def\ps@plainhead{%
1559 \let\@mkboth\@gobbletwo
1560 \let\@oddfoot\@empty
1561 \let\@evenfoot\@empty
1562 \def\@evenhead{%
1563 \if@mparswitch \hss \fi
1564 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1565 \if@mparswitch\else \hss \fi}%
1566 \def\@oddhead{%
1567 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1568 %<book>\let\ps@plain\ps@plainhead
1569 %<!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。

```

1570 %<*article|slide>
1571 \if@twoside
1572 \def\ps@headings{%
1573 \let\@oddfoot\@empty
1574 \let\@evenfoot\@empty
1575 \def\@evenhead{\if@mparswitch \hss \fi
1576 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1577 \if@mparswitch\else \hss \fi}%
1578 \def\@oddhead{%
1579 \underline{%
1580 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1581 \let\@mkboth\markboth
1582 \def\sectionmark##1{\markboth{%
1583 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1584 ##1}{}}%
1585 \def\subsectionmark##1{\markright{%
1586 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1587 ##1}}%
1588 }
1589 \else % if not twoside
1590 \def\ps@headings{%
1591 \let\@oddfoot\@empty
1592 \def\@oddhead{%
1593 \underline{%
1594 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1595 \let\@mkboth\markboth
1596 \def\sectionmark##1{\markright{%

```

```

1597         \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1598         ##1}}}}
1599 \fi
1600 %</article|slide>

```

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

---

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

---

```

1601 %<*book|report>
1602 \def\bxjs@maybe@autoxspacing{%
1603   \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1604 \newif\if@omit@number
1605 \def\ps@headings{%
1606   \let\@oddfoot\@empty
1607   \let\@evenfoot\@empty
1608   \def\@evenhead{%
1609     \if@mparswitch \hss \fi
1610     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1611       \textbf{\thepage}\hfil\leftmark}}}%
1612     \if@mparswitch\else \hss \fi}%
1613   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1614     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1615   \let\@mkboth\markboth
1616   \def\chaptermark##1{\markboth{%
1617     \ifnum \c@secnumdepth >\m@ne
1618       \if@mainmatter
1619         \if@omit@number\else
1620           \@chapapp\thechapter\@chappos\hskip1\jsZw
1621         \fi
1622       \fi
1623     \fi
1624     ##1-}}}%
1625   \def\sectionmark##1{\markright{%
1626     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1627     ##1}}}%
1628 %</book|report>

```

最後は学会誌の場合です。

```

1629 %<*jspf>
1630 \def\ps@headings{%
1631   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1632   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1633   \def\@oddhead{\normalfont\hfil \@title \hfil}
1634   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1635 %</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。

```
1636 \def\ps@myheadings{%
1637   \let\@oddfoot\@empty\let\@evenfoot\@empty
1638   \def\@evenhead{%
1639     \if@mparswitch \hss \fi%
1640     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1641     \if@mparswitch\else \hss \fi}%
1642   \def\@oddhead{%
1643     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1644   \let\mkboth\@gobbletwo
1645   %<book|report> \let\chaptermark\@gobble
1646   \let\sectionmark\@gobble
1647   %<!book&!report> \let\subsectionmark\@gobble
1648 }
```

## 8 文書のマークアップ

### 8.1 表題

`\title` これらは L<sup>A</sup>T<sub>E</sub>X 本体で次のように定義されています。ここではコメントアウトした形で示します。

```
\author
\date 1649 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1650 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1651 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1652 % \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。

```
1653 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1654 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。
1655 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1656 \AtBeginDocument{\bxjs@decl@subtitle}
1657 \def\bxjs@decl@subtitle{%
1658   \global\let\bxjs@decl@subtitle\relax
1659   \ifx\subtitle\@undefined
1660     \global\let\subtitle\jsSubtitle
1661   \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。

```
1662 \def\bxjs@annihilate@subtitle{%
1663   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1664   \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。

```
\eauthor 1665 %<*jspf>
1666 \newcommand*\etitle[1]{\gdef\etitle{#1}}
\keywords 1667 \newcommand*\eauthor[1]{\gdef\eauthor{#1}}
1668 \newcommand*\keywords[1]{\gdef\keywords{#1}}
1669 \newcommand*\email[1]{\gdef\authors@mail{#1}}
1670 \newcommand*\AuthorsEmail[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1671 %</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。

```
1672 \def\plainifnotempty{%
1673   \ifx \@oddhead \@empty
1674     \ifx \@oddfoot \@empty
1675       \else
1676         \thispagestyle{plainfoot}%
1677       \fi
1678     \else
1679       \thispagestyle{plainhead}%
1680     \fi}
```

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

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

```
1681 %<*article|book|report|slide>
1682 \if@titlepage
1683   \newcommand*\maketitle{%
1684     \begin{titlepage}%
1685       \let\footnotesize\small
1686       \let\footnoterule\relax
1687       \let\footnote\thanks
1688       \null\vfil
1689       \if@slide
1690         {\footnotesize \@date}%
```

```

1691 \begin{center}
1692 \mbox{} \\\[1\jsZw]
1693 \large
1694 {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1695 \jsc@smallskip
1696 \@title
1697 \ifx\bxjs@subtitle\@undefined\else
1698 \par\vskip\z@
1699 {\small \bxjs@subtitle\par}
1700 \fi
1701 \jsc@smallskip
1702 {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1703 \vfill
1704 {\small \@author}%
1705 \end{center}
1706 \else
1707 \vskip 60\p@?
1708 \begin{center}%
1709 {\LARGE \@title \par}%
1710 \ifx\bxjs@subtitle\@undefined\else
1711 \vskip5\p@?
1712 {\normalsize \bxjs@subtitle\par}
1713 \fi
1714 \vskip 3em%
1715 {\large
1716 \lineskip .75em
1717 \begin{tabular}[t]{c}%
1718 \@author
1719 \end{tabular}\par}%
1720 \vskip 1.5em
1721 {\large \@date \par}%
1722 \end{center}%
1723 \fi
1724 \par
1725 \@thanks\vfil\null
1726 \end{titlepage}%
1727 \setcounter{footnote}{0}%
1728 \global\let\thanks\relax
1729 \global\let\maketitle\relax
1730 \global\let\@thanks\@empty
1731 \global\let\@author\@empty
1732 \global\let\@date\@empty
1733 \global\let\@title\@empty
1734 \global\let\title\relax
1735 \global\let\author\relax
1736 \global\let\date\relax
1737 \global\let\and\relax
1738 \bxjs@annihilate@subtitle
1739 }%

```

```

1740 \else
1741   \newcommand{\maketitle}{\par
1742     \begin{group}
1743       \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1744       \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1745       \long\def\@makefnmark##1{\advance\leftskip 3\jsZw
1746         \parindent 1\jsZw\noindent
1747         \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1748       \if@twocolumn
1749         \ifnum \col@number=\@one
1750           \@maketitle
1751         \else
1752           \twocolumn[\@maketitle]%
1753         \fi
1754       \else
1755         \newpage
1756         \global\@topnum\z@ % Prevents figures from going at top of page.
1757         \@maketitle
1758       \fi
1759       \plainifnotempty
1760       \@thanks
1761     \end{group}
1762     \setcounter{footnote}{0}%
1763     \global\let\thanks\relax
1764     \global\let\maketitle\relax
1765     \global\let\@thanks\@empty
1766     \global\let\@author\@empty
1767     \global\let\@date\@empty
1768     \global\let\@title\@empty
1769     \global\let\title\relax
1770     \global\let\author\relax
1771     \global\let\date\relax
1772     \global\let\and\relax
1773     \bxjs@annihilate@subtitle
1774   }

```

`\@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。

```

1775 \def\@maketitle{%
1776   \newpage\null
1777   \vskip 2em
1778   \begin{center}%
1779     \let\footnote\thanks
1780     {\LARGE \@title \par}%
1781     \ifx\bxjs@subtitle\@undefined\else
1782       \vskip3\p@?
1783       {\normalsize \bxjs@subtitle\par}
1784     \fi
1785     \vskip 1.5em
1786     {\large

```

```

1787         \lineskip .5em
1788         \begin{tabular}[t]{c}%
1789             \@author
1790         \end{tabular}\par}%
1791     \vskip 1em
1792     {\large \@date}%
1793 \end{center}%
1794 \par\vskip 1.5em
1795 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1796 }
1797 \fi
1798 %</article|book|report|slide>
1799 %<*jspf>
1800 \newcommand\maketitle{\par
1801     \begingroup
1802         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1803         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1804         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1805             \parindent 1\jsZw\noindent
1806             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1807         \twocolumn[\@maketitle]%
1808         \plainifnotempty
1809         \@thanks
1810     \endgroup
1811     \setcounter{footnote}{0}%
1812     \global\let\thanks\relax
1813     \global\let\maketitle\relax
1814     \global\let\@thanks\@empty
1815     \global\let\@author\@empty
1816     \global\let\@date\@empty
1817 % \global\let\@title\@empty % \@title は柱に使う
1818     \global\let\title\relax
1819     \global\let\author\relax
1820     \global\let\date\relax
1821     \global\let\and\relax
1822     \ifx\authors@mail\@undefined\else{%
1823         \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1824         \footnotetext[0]{\itshape\authors@mail}%
1825     }\fi
1826     \global\let\authors@mail\@undefined}
1827 \def\@maketitle{%
1828     \newpage\null
1829     \vskip 6em % used to be 2em
1830     \begin{center}
1831         \let\footnote\thanks
1832         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1833         \lineskip .5em
1834         \ifx\@author\@undefined\else
1835             \vskip 1em

```

```

1836     \begin{tabular}[t]{c}%
1837         \@author
1838     \end{tabular}\par
1839 \fi
1840 \ifx\@etitle\@undefined\else
1841     \vskip 1em
1842     {\large \@etitle \par}%
1843 \fi
1844 \ifx\@eauthor\@undefined\else
1845     \vskip 1em
1846     \begin{tabular}[t]{c}%
1847         \@eauthor
1848     \end{tabular}\par
1849 \fi
1850 \vskip 1em
1851 \@date
1852 \end{center}
1853 \vskip 1.5em
1854 \centerline{\box\@abstractbox}
1855 \ifx\@keywords\@undefined\else
1856     \vskip 1.5em
1857     \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1858 \fi
1859 \vskip 1.5em}
1860 %</jspf>

```

## 8.2 章・節

---

ムニャムニャ……。

`\bxjs@label@sect` 節付 #1 の番号を出力する。節付 XXX に対して、`\labelXXX` が定義済ならそれが出力書式を表す。未定義ならばカウンタの出力書式 `\theXXX` が使われる。

```

1861 \def\bxjs@label@sect#1{%
1862     \expandafter\ifx\csname label#1\endcsname\relax
1863         \csname the#1\endcsname
1864     \else \csname label#1\endcsname
1865     \fi}
1866 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1867 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1868 \def\@secapp{\presectionname}
1869 \def\@secpos{\postsectionname}
1870 \fi

```

`\labelsection` 節番号の出力書式。

```
1871 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1872 \def\labelsection{\@secapp\thesection\@secpos}
1873 \fi
```

---

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

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

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```
1874 \def\@startsection#1#2#3#4#5#6{%
1875   \if@noskipsec \leavevmode \fi
1876   \par
1877 % 見出し上の空きを \@tempskipa にセットする
1878   \@tempskipa #4\relax
1879 % \@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
1880   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1881 % 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
1882   \ifdim \@tempskipa <\z@
1883     \@tempskipa -\@tempskipa \@afterindentfalse
1884   \fi
1885   \if@nobreak
1886 %   \everypar{\everyparhook}% これは間違い
1887   \everypar{}\%
```

```

1888 \else
1889 \addpenalty\@secpenalty
1890 % 次の行は削除
1891 % \addvspace\@tempskipa
1892 % 次の \noindent まで追加
1893 \ifdim \@tempskipa >\z@
1894 \if@slide\else
1895 \null
1896 \vspace*{-\baselineskip}%
1897 \fi
1898 \vskip\@tempskipa
1899 \fi
1900 \fi
1901 \noindent
1902 % 追加終わり
1903 \@ifstar
1904 {\@ssect{#3}{#4}{#5}{#6}}%
1905 {\@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いくように、多少変えてあります。 \everyparhook も挿入しています。

---

\everyparhook の挿入は everyparhook=compat の時のみ行う。

\bxjs@if@ceph everyparhook=compat である場合にのみ直後のトークンを実行する。

```

1906 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1907 \let\bxjs@if@ceph\@firstofone
1908 \else \let\bxjs@if@ceph\@gobble
1909 \fi

```

---

```

1910 \def\@sect#1#2#3#4#5#6[#7]#8{%
1911 \ifnum #2>\c@secnumdepth
1912 \let\@svsec\@empty
1913 \else
1914 \refstepcounter{#1}%
1915 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1916 \fi
1917 % 見出し後の空きを \@tempskipa にセット
1918 \@tempskipa #5\relax
1919 % 条件判断の順序を入れ替えました
1920 \ifdim \@tempskipa<\z@
1921 \def\@svsechd{%
1922 #6{\hskip #3\relax
1923 \@svsec #8}%
1924 \csname #1mark\endcsname{#7}}%
1925 \addcontentsline{toc}{#1}{%
1926 \ifnum #2>\c@secnumdepth \else
1927 \protect\numberline{\bxjs@label@sect{#1}}%

```

```

1928     \fi
1929     #7}}% 目次にフルネームを載せるなら #8
1930 \else
1931     \begingroup
1932     \interlinepenalty \@M % 下から移動
1933     #6{%
1934         \@hangfrom{\hskip #3\relax\@svsec}%
1935 %     \interlinepenalty \@M % 上に移動
1936     #8\@@par}%
1937 \endgroup
1938 \csname #1mark\endcsname{#7}%
1939 \addcontentsline{toc}{#1}{%
1940     \ifnum #2>\c@secnumdepth \else
1941         \protect\numberline{\bxjs@label@sect{#1}}%
1942     \fi
1943     #7}}% 目次にフルネームを載せるならここは #8
1944 \fi
1945 \@xsect{#5}}

```

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

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

```

1946 \def\@xsect#1{%
1947 % 見出しの後ろの空きを \@tempskipa にセット
1948     \@tempskipa #1\relax
1949 % 条件判断の順序を変えました
1950     \ifdim \@tempskipa<\z@
1951         \@nobreakfalse
1952         \global\@noskipsectrue
1953         \everypar{%
1954             \if@noskipsec
1955                 \global\@noskipsecfalse
1956                 {\setbox\z@\lastbox}%
1957                 \clubpenalty\@M
1958                 \begingroup \@svsechd \endgroup
1959                 \unskip
1960                 \@tempskipa #1\relax
1961                 \hskip -\@tempskipa
1962             \else
1963                 \clubpenalty \@clubpenalty
1964                 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1965                 \fi\bxjs@if@ceph\everyparhook}%
1966     \else
1967         \par \nobreak
1968         \vskip \@tempskipa
1969         \@afterheading
1970     \fi

```



```

1971 \if@slide
1972 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1973 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1974 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1975 \fi
1976 \par % 2000-12-18
1977 \ignorespaces}
1978 \def\@ssect#1#2#3#4#5{%
1979 \@tempskipa #3\relax
1980 \ifdim \@tempskipa<\z@
1981 \def\@svsechd{#4{\hskip #1\relax #5}}%
1982 \else
1983 \begingroup
1984 #4{%
1985 \@hangfrom{\hskip #1}%
1986 \interlinepenalty \@M #5\@@par}%
1987 \endgroup
1988 \fi
1989 \@xsect{#3}}

```

### ■柱関係の命令

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

```

\subsectionmark 1990 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1991 % \newcommand*{\sectionmark}[1]{}
1992 % \newcommand*{\subsectionmark}[1]{}
\paragraphmark 1993 % \newcommand*{\subsubsectionmark}[1]{}
\subparagraphmark 1994 % \newcommand*{\paragraphmark}[1]{}
1995 % \newcommand*{\subparagraphmark}[1]{}

```

### ■カウンタの定義

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

```

1996 %<!book&!report>\setcounter{secnumdepth}{3}
1997 %<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 1998 \newcounter{part}
1999 %<book|report>\newcounter{chapter}
\c@subsubsection 2000 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2001 %<!book&!report>\newcounter{section}
\c@subparagraph 2002 \newcounter{subsection}[section]
2003 \newcounter{subsubsection}[subsection]
2004 \newcounter{paragraph}[subsubsection]
2005 \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`

`\thesection`

`\thesubsection`

`\thesubsubsection`

`\theparagraph`

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>\arabic{COUNTER}</code>	1, 2, 3, ...
<code>\roman{COUNTER}</code>	i, ii, iii, ...
<code>\Roman{COUNTER}</code>	I, II, III, ...
<code>\alph{COUNTER}</code>	a, b, c, ...
<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。

```
2006 \renewcommand{\thepart}{\@Roman\c@part}
2007 %<*!book&!report>
2008 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2009 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2010 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2011 \else
2012 \renewcommand{\thesection}{\@arabic\c@section}
2013 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2014 \fi
2015 %</!book&!report>
2016 %<*book|report>
2017 \renewcommand{\thechapter}{\@arabic\c@chapter}
2018 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2019 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2020 %</book|report>
2021 \renewcommand{\thesubsubsection}{\%
2022   \thesubsection.\@arabic\c@subsubsection}
2023 \renewcommand{\theparagraph}{\%
2024   \thesubsubsection.\@arabic\c@paragraph}
2025 \renewcommand{\thesubparagraph}{\%
2026   \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。

```
2027 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2028 %<book|report>\newcommand{\@chappos}{\postchaptername}
```

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

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

[2017-03-05] `\frontmatter` と `\mainmatter` の2つの命令は, 改丁または改ページした後で `\pagenumbering{...}` でノンブルを1にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが, これではノンブルをリセットする際に偶奇逆転が起こる場合がありました。 `openany` かどうかにかかわらず奇数ページまで繰るように

修正することで、問題を解消しました。実は、 $\text{\LaTeX}$  の標準クラスでは 1998 年に修正されていた問題です (コミュニティ版  $\text{p}\text{\LaTeX}$  の標準クラス 2017/03/05 も参照)。

```
2029 %<*book|report>
2030 \newcommand\frontmatter{%
2031   \pltx@cleartooddpage
2032   \@mainmatterfalse
2033   \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。

```
2034 \newcommand\mainmatter{%
2035   \pltx@cleartooddpage
2036   \@mainmattertrue
2037   \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。

```
2038 \newcommand\backmatter{%
2039   \if@openleft
2040     \cleardoublepage
2041   \else\if@openright
2042     \cleardoublepage
2043   \else
2044     \clearpage
2045   \fi\fi
2046   \@mainmatterfalse}
2047 %</book|report>
```

## ■部

`\part` 新しい部を始めます。

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

```
\secdef{星なし}{星あり}
```

星なし \* のない形の定義です。

星あり \* のある形の定義です。

`\secdef` は次のようにして使います。

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

まず `book` と `report` のクラス以外です。

```
2048 %<*&!book&!report>
2049 \newcommand\part{%
2050   \if@noskipsec \leavevmode \fi
2051   \par
2052   \addvspace{4ex}%
2053   \if@english \@afterindentfalse \else \@afterindenttrue \fi
```

```

2054 \secdef\@part\@spart}
2055 %</*!book&!report>

book および report クラスの場合は、少し複雑です。

2056 %<*book|report>
2057 \newcommand\part{%
2058 \if@openleft
2059 \cleardoublepage
2060 \else\if@openright
2061 \cleardoublepage
2062 \else
2063 \clearpage
2064 \fi\fi
2065 \thispagestyle{empty}% 欧文用標準スタイルでは plain
2066 \if@twocolumn
2067 \onecolumn
2068 \@restonecoltrue
2069 \else
2070 \@restonecolfalse
2071 \fi
2072 \null\vfil
2073 \secdef\@part\@spart}
2074 %</book|report>

```

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

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

```

2075 %</*!book&!report>
2076 \def\@part[#1]#2{%
2077 \ifnum \c@secnumdepth >\m@ne
2078 \refstepcounter{part}%
2079 \addcontentsline{toc}{part}{%
2080 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2081 \else
2082 \addcontentsline{toc}{part}{#1}%
2083 \fi
2084 \markboth{}{}%
2085 {\parindent\z@
2086 \raggedright
2087 \interlinepenalty \@M
2088 \normalfont
2089 \ifnum \c@secnumdepth >\m@ne
2090 \Large\headfont\prepartname\thepart\postpartname
2091 \par\nobreak
2092 \fi
2093 \huge \headfont #2%
2094 \markboth{}{}\par}%
2095 \nobreak
2096 \vskip 3ex

```

```

2097 \afterheading}
2098 %</!book&!report>

```

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

```

2099 %<*book|report>
2100 \def\@part[#1]#2{%
2101   \ifnum \c@secnumdepth >-2\relax
2102     \refstepcounter{part}%
2103     \addcontentsline{toc}{part}{%
2104       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2105   \else
2106     \addcontentsline{toc}{part}{#1}%
2107   \fi
2108   \markboth{}{}%
2109   {\centering
2110     \interlinepenalty \@M
2111     \normalfont
2112     \ifnum \c@secnumdepth >-2\relax
2113       \huge\headfont \prepartname\thepart\postpartname
2114       \par\vskip20\p@?
2115     \fi
2116     \Huge \headfont #2\par}%
2117   \@endpart}
2118 %</book|report>

```

\@spart 番号を付けない部です。

```

2119 %<*!book&!report>
2120 \def\@spart#1{%
2121   \parindent \z@ \raggedright
2122   \interlinepenalty \@M
2123   \normalfont
2124   \huge \headfont #1\par}%
2125   \nobreak
2126   \vskip 3ex
2127   \afterheading}
2128 %</!book&!report>
2129 %<*book|report>
2130 \def\@spart#1{%
2131   \centering
2132   \interlinepenalty \@M
2133   \normalfont
2134   \Huge \headfont #1\par}%
2135   \@endpart}
2136 %</book|report>

```

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

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

います。

```
2137 %<*book|report>
2138 \def\@endpart{\vfil\newpage
2139   \if@twoside
2140     \if@openleft %% added (2017/02/24)
2141       \null\thispagestyle{empty}\newpage
2142     \else\if@openright %% added (2016/12/13)
2143       \null\thispagestyle{empty}\newpage
2144     \fi\fi %% added (2016/12/13, 2017/02/24)
2145   \fi
2146   \if@restonecol
2147     \twocolumn
2148   \fi}
2149 %</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。

```
2150 %<*book|report>
2151 \newcommand{\chapter}{%
2152   \if@openleft\cleardoublepage\else
2153   \if@openright\cleardoublepage\else\clearpage\fi\fi
2154   \plainifnotempty % 元: \thispagestyle{plain}
2155   \global\@topnum\z@
2156   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2157   \secdef
2158     {\@omit@numberfalse\@chapter}%
2159     {\@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。

```
2160 \def\@chapter[#1]#2{%
2161   \ifnum \c@secnumdepth >\m@ne
2162     \if@mainmatter
2163       \refstepcounter{chapter}%
2164       \typeout{\@chapapp\thechapter\@chappos}%
2165       \addcontentsline{toc}{chapter}%
2166         {\protect\numberline
2167 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2168         {\@chapapp\thechapter\@chappos}%
2169         #1}%
2170     \else\addcontentsline{toc}{chapter}{#1}\fi
2171   \else
2172     \addcontentsline{toc}{chapter}{#1}%
2173   \fi
2174   \chaptermark{#1}%
2175   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}}
```

```

2176 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}}%
2177 \if@twocolumn
2178   \topnewpage[\@makechapterhead{#2}]%
2179 \else
2180   \@makechapterhead{#2}%
2181   \@afterheading
2182 \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。

```

2183 \def\@makechapterhead#1{%
2184   \vspace*{2\Cvs}% 欧文は 50pt
2185   {\parindent \z@ \raggedright \normalfont
2186     \ifnum \c@secnumdepth >\m@ne
2187       \if@mainmatter
2188         \huge\headfont \@chapapp\thechapter\@chappos
2189         \par\nobreak
2190         \vskip \Cvs % 欧文は 20pt
2191       \fi
2192     \fi
2193     \interlinepenalty\@M
2194     \Huge \headfont #1\par\nobreak
2195     \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。

```

2196 \def\@schapter#1{%
2197   \chaptermark{#1}%
2198   \if@twocolumn
2199     \topnewpage[\@makeschapterhead{#1}]%
2200   \else
2201     \@makeschapterhead{#1}\@afterheading
2202   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2203 \def\@makeschapterhead#1{%
2204   \vspace*{2\Cvs}% 欧文は 50pt
2205   {\parindent \z@ \raggedright
2206     \normalfont
2207     \interlinepenalty\@M
2208     \Huge \headfont #1\par\nobreak
2209     \vskip 3\Cvs}} % 欧文は 40pt
2210 %</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。

```

2211 \if@twocolumn

```

```

2212 \newcommand{\section}{%
2213 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2214 \startsection{section}{1}{\z@}%
2215 %<kiyou> {0.6\Cvs}{0.4\Cvs}%
2216 %<kiyou> {\Cvs}{0.5\Cvs}%
2217 % {\normalfont\large\headfont\@secapp}%
2218 {\normalfont\large\headfont\raggedright}}
2219 \else
2220 \newcommand{\section}{%
2221 \if@slide\clearpage\fi
2222 \startsection{section}{1}{\z@}%
2223 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2224 {.5\Cvs \@plus.3\Cdp}% 後アキ
2225 % {\normalfont\Large\headfont\@secapp}%
2226 {\normalfont\Large\headfont\raggedright}}
2227 \fi

```

\subsection 同上です。

```

2228 \if@twocolumn
2229 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2230 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2231 {\normalfont\normalsize\headfont}}
2232 \else
2233 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2234 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2235 {.5\Cvs \@plus.3\Cdp}% 後アキ
2236 {\normalfont\large\headfont}}
2237 \fi

```

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

```

2238 \if@twocolumn
2239 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2240 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2241 {\normalfont\normalsize\headfont}}
2242 \else
2243 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2244 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2245 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2246 {\normalfont\normalsize\headfont}}
2247 \fi

```

\paragraph 見出しの後ろで改行されません。

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

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



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

---

※ BXJS クラスでは、1.1 版 [2016-02-14] から `\jsParagraphMark` をサポートしている。  
 段落のマーク (■) が必ず和文フォントで出力されるようにする。

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

---

```

2248 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2249 \let\bxjs@org@paragraph@mark\jsParagraphMark
2250 \ifx\bxjs@paragraph@mark\empty
2251   \let\jsParagraphMark\empty
2252 \else\ifx\bxjs@paragraph@mark\undefined\else
2253   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2254 \fi\fi
2255 \let\jsJaChar\empty
2256 \if@twocolumn
2257   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2258     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2259 %<jspf>    {\normalfont\normalsize\headfont}}
2260 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2261 \else
2262   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2263     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2264     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2265 %<jspf>    {\normalfont\normalsize\headfont}}
2266 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2267 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2268 \if@twocolumn
2269   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2270     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2271     {\normalfont\normalsize\headfont}}
2272 \else
2273   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2274     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2275     {\normalfont\normalsize\headfont}}
2276 \fi

```

### 8.3 リスト環境

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

```
2277 \if@slide
2278   \setlength\leftmargini{1\jsZw}
2279 \else
2280   \if@twocolumn
2281     \setlength\leftmargini{2\jsZw}
2282   \else
2283     \setlength\leftmargini{3\jsZw}
2284   \fi
2285 \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 2286 \if@slide
\leftmarginv 2287   \setlength\leftmarginii {1\jsZw}
2288   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2289   \setlength\leftmarginiv {1\jsZw}
2290   \setlength\leftmarginv {1\jsZw}
2291   \setlength\leftmarginvi {1\jsZw}
2292 \else
2293   \setlength\leftmarginii {2\jsZw}
2294   \setlength\leftmarginiii{2\jsZw}
2295   \setlength\leftmarginiv {2\jsZw}
2296   \setlength\leftmarginv {1\jsZw}
2297   \setlength\leftmarginvi {1\jsZw}
2298 \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。

```
2299 \setlength \labelsep {0.5\jsZw} % .5em
2300 \setlength \labelwidth{\leftmargini}
2301 \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。

```
2302 \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 2303 \@beginparpenalty -\@lowpenalty
\@itempenalty 2304 \@endparpenalty -\@lowpenalty
2305 \@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  ${}^{+0.2}_{-0.1}$  `\baselineskip` を思い切って外しました。

```
2306 \def\@listI{\leftmargin\leftmarginI
2307   \parsep \z@
2308   \topsep 0.5\baselineskip
2309   \itemsep \z@ \relax}
2310 \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）。

```
2311 \@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 2312 \def\@listii{\leftmargin\leftmarginii
\@listiv 2313   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2314   \topsep \z@
\@listv 2315   \parsep \z@
\@listvi 2316   \itemsep\parsep}
2317 \def\@listiii{\leftmargin\leftmarginiii
2318   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2319   \topsep \z@
2320   \parsep \z@
2321   \itemsep\parsep}
2322 \def\@listiv {\leftmargin\leftmarginiv
2323   \labelwidth\leftmarginiv
2324   \advance\labelwidth-\labelsep}
2325 \def\@listv {\leftmargin\leftmarginv
2326   \labelwidth\leftmarginv
2327   \advance\labelwidth-\labelsep}
2328 \def\@listvi {\leftmargin\leftmarginvi
2329   \labelwidth\leftmarginvi
2330   \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。

```
2331 \renewcommand{\theenumi}{\@arabic\c@enumi}
2332 \renewcommand{\theenumii}{\@alph\c@enumii}
2333 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2334 \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。

---

```

2335 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2336 \newcommand*\jsInJaParen}[1]{%
2337   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2338 \newcommand{\labelenumi}{\theenumi.}
2339 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2340 \newcommand{\labelenumiii}{\theenumiii.}
2341 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2342 \renewcommand{\p@enumii}{\theenumi}
2343 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2344 \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 2345 \newcommand\labelitemi{\textbullet}
2346 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 2347 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2348 \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。

```

2349 \newenvironment{description}{%
2350   \list{}{%
2351     \labelwidth=\leftmargin
2352     \labelsep=1\jsZw
2353     \advance \labelwidth by -\labelsep
2354     \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。

```

2355 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}

```

## ■概要

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

JSPF スタイルでは実際の出力は `\maketitle` で行われます。

---

**bxjsreport** クラスの **abstract** 環境は：

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

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

```
2356 %<*book|report>
2357 \newenvironment{chapterabstract}{%
2358   \begin{list}{}{%
2359     \listparindent=1\jsZw
2360     \itemindent=\listparindent
2361     \rightmargin=0pt
2362     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2363 %</book|report>
```

“普通の” **abstract** 環境の定義。

```
2364 %<*article|report|slide>
2365 \newbox\@abstractbox
2366 \if@titlepage
2367   \newenvironment{abstract}{%
2368     \titlepage
2369     \null\vfil
2370     \@beginparpenalty\@lowpenalty
2371     \begin{center}%
2372       \headfont \abstractname
2373       \@endparpenalty\@M
2374     \end{center}%
```

BXJS クラスでは、概要の最初の段落に段落下げが入るようにする。

```
2375   \par}%
2376   {\par\vfil\null\endtitlepage}
2377 \else
2378   \newenvironment{abstract}{%
2379     \if@twocolumn
2380       \ifx\maketitle\relax
2381         \section*{\abstractname}%
```

```

2382 \else
2383 \global\setbox\@abstractbox\hbox\bgroup
2384 \begin{minipage}[b]{\textwidth}
2385 \small\parindent1\jsZw
2386 \begin{center}%
2387 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2388 \end{center}%
2389 \list{}{%
2390 \listparindent\parindent
2391 \itemindent \listparindent
2392 \rightmargin \leftmargin}%
2393 \item\relax
2394 \fi
2395 \else
2396 \small
2397 \begin{center}%
2398 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2399 \end{center}%
2400 \list{}{%
2401 \listparindent\parindent
2402 \itemindent \listparindent
2403 \rightmargin \leftmargin}%
2404 \item\relax
2405 \fi}{\if@twocolumn
2406 \ifx\maketitle\relax
2407 \else
2408 \endlist\end{minipage}\egroup
2409 \fi
2410 \else
2411 \endlist
2412 \fi}
2413 \fi
2414 %</article|report|slide>
2415 %<*jspf>
2416 \newbox\@abstractbox
2417 \newenvironment{abstract}{%
2418 \global\setbox\@abstractbox\hbox\bgroup
2419 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2420 \small
2421 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2422 {\end{minipage}\egroup}
2423 %</jspf>

```

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

```

2424 %<*book|report>
2425 \ifbxjs@force@chapterabstract
2426 \let\abstract\chapterabstract
2427 \let\endabstract\endchapterabstract

```

```

2428 \fi
2429 %</book|report>

```

---

## ■キーワード

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

```

2430 %<*jspf>
2431 %\newbox\@keywordsbox
2432 %\newenvironment{keywords}{%
2433 % \global\setbox\@keywordsbox\hbox\bgroup
2434 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2435 % \small\parindent0\jsZw}%
2436 % {\end{minipage}\egroup}
2437 %</jspf>

```

## ■verse 環境

**verse** 詩のための `verse` 環境です。

```

2438 \newenvironment{verse}{%
2439 \let \=\@centercr
2440 \list{}{%
2441 \itemsep \z@
2442 \itemindent -2\jsZw % 元: -1.5em
2443 \listparindent\itemindent
2444 \rightmargin \z@
2445 \advance\leftmargin 2\jsZw}% 元: 1.5em
2446 \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。

```

2447 \newenvironment{quotation}{%
2448 \list{}{%
2449 \listparindent\parindent
2450 \itemindent\listparindent
2451 \rightmargin \z@}%
2452 \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。

```

2453 \newenvironment{quote}%
2454 {\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。

```
2455 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2456   \item[\hskip \labelsep{\headfont #1\ #2}]}
2457 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2458   \item[\hskip \labelsep{\headfont #1\ #2 (#3)}]}
```

`titlepage` タイトルを独立のページに出力するのに使われます。

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

```
2459 \newenvironment{titlepage}{%
2460 %<book> \pltx@cleartooddpage %% 2017-02-24
2461   \if@twocolumn
2462     \@restonecoltrue\onecolumn
2463   \else
2464     \@restonecolfalse\newpage
2465   \fi
2466   \thispagestyle{empty}%
2467   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2468   24
2469 }%
2470 {\if@restonecol\twocolumn \else \newpage \fi
2471   \if@twoside\else
2472     \setcounter{page}\@ne
2473   \fi}
```

## ■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2473 %<*&!book&!report>
2474 \newcommand{\appendix}{\par
2475   \setcounter{section}{0}%
2476   \setcounter{subsection}{0}%
2477   \ifnum\bxjs@label@section=\bxjs@label@section@compat
```



```

2478 \gdef\presectionname{\appendixname}%
2479 \gdef\postsectionname{}%
2480 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2481 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2482 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2483 \else
2484 \gdef\@secapp{\appendixname}%
2485 \gdef\@secpos{}%
2486 \gdef\thesection{\@Alph\c@section}%
2487 \fi}
2488 %</!book&!report>
2489 %<*book|report>
2490 \newcommand{\appendix}{\par
2491 \setcounter{chapter}{0}%
2492 \setcounter{section}{0}%
2493 \gdef\@chapapp{\appendixname}%
2494 \gdef\@chappos{}%
2495 \gdef\thechapter{\@Alph\c@chapter}}
2496 %</book|report>

```

## 8.4 パラメータの設定

### ■array と tabular 環境

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

```
2497 \setlength\arraycolsep{5\p@?}
```

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

```
2498 \setlength\tabcolsep{6\p@?}
```

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

```
2499 \setlength\arrayrulewidth{.4\p@}
```

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

```
2500 \setlength\doublerulesep{2\p@}
```

### ■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2501 \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。

```
2502 \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。

```
2503 \setlength\fboxsep{3\p@?}
2504 \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。

```
2505 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2506 %<*book|report>
2507 \@addtoreset{equation}{chapter}
2508 \renewcommand\theequation
2509 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2510 %</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。

```
2511 % \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。

```
2512 % \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。

```
2513 % \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip\@italiccorr )}}
```

## 8.5 フロート

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

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

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

`\ext@TYPE` フロートの目次を出力するファイルの拡張子です。

`\fnum@TYPE` キャプション用の番号を生成するマクロです。

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

## ■figure 環境

`\c@figure` 図番号のカウンタです。

`\thefigure` 図番号を出力するコマンドです。

```
2514 %<!*book&!report>
2515 \newcounter{figure}
2516 \renewcommand \thefigure {\@arabic\c@figure}
2517 %</!*book&!report>
2518 %<*book|report>
2519 \newcounter{figure}[chapter]
2520 \renewcommand \thefigure
2521     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2522 %</book|report>
```

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

```
\ext@figure 2523 \def\fps@figure{tbp}
2524 \def\ftype@figure{1}
\fnun@figure 2525 \def\ext@figure{lof}
2526 \def\fnun@figure{\figurename\nobreak\thefigure}
```

`figure` \* 形式は段抜きのフロートです。

```
figure* 2527 \newenvironment{figure}%
2528     {\@float{figure}}%
2529     {\end@float}
2530 \newenvironment{figure*}%
2531     {\@dblfloat{figure}}%
2532     {\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。

```
2533 %<!*book&!report>
2534 \newcounter{table}
2535 \renewcommand\thetable{\@arabic\c@table}
2536 %</!*book&!report>
2537 %<*book|report>
2538 \newcounter{table}[chapter]
2539 \renewcommand \thetable
2540     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2541 %</book|report>
```

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

```
\ext@table 2542 \def\fps@table{tbp}
2543 \def\ftype@table{2}
\fnun@table 2544 \def\ext@table{lot}
2545 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` \* は段抜きのフロートです。

`table*`

```

2546 \newenvironment{table}%
2547         {\@float{table}}%
2548         {\end@float}
2549 \newenvironment{table*}%
2550         {\@dblfloat{table}}%
2551         {\end@dblfloat}

```

## 8.6 キャプション

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

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

```

2552 \newlength\abovecaptionskip
2553 \newlength\belowcaptionskip
2554 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2555 \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。

```

2556 %<!*jspf>
2557 % \long\def\@makecaption#1#2{{\small
2558 %   \advance\leftskip10\jsc@mmm
2559 %   \advance\rightskip10\jsc@mmm
2560 %   \vskip\abovecaptionskip
2561 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2562 %   \ifdim \wd\@tempboxa >\hsize
2563 %     #1\hskip1\jsZw\relax #2\par
2564 %   \else
2565 %     \global \@minipagefalse
2566 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2567 %   \fi
2568 %   \vskip\belowcaptionskip}}
2569 \long\def\@makecaption#1#2{{\small
2570   \advance\leftskip .0628\linewidth
2571   \advance\rightskip .0628\linewidth
2572   \vskip\abovecaptionskip
2573   \sbox\@tempboxa{#1\zwspace#2}%
2574   \ifdim \wd\@tempboxa <\hsize \centering \fi
2575   #1\zwspace#2\par
2576   \vskip\belowcaptionskip}}
2577 %</!*jspf>
2578 %<!*jspf>
2579 \long\def\@makecaption#1#2{%

```

```

2580 \vskip\abovecaptionskip
2581 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2582 \ifdim \wd\@tempboxa >\hsize
2583   {\small\sffamily
2584     \list{#1}{%
2585       \renewcommand{\makelabel}[1]{##1\hfil}
2586       \itemsep \z@
2587       \itemindent \z@
2588       \labelsep \z@
2589       \labelwidth 11\jsc@mmm
2590       \listparindent\z@
2591       \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2592 \else
2593   \global \@minipagefalse
2594   \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2595 \fi
2596 \vskip\belowcaptionskip}
2597 %</jspf>

```

## 9 フォントコマンド

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

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

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

```

\if@jsc@warnoldfontcmd


---


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


---


2598 \newif\if@jsc@warnoldfontcmd
2599 \@jsc@warnoldfontcmdtrue
2600 \newif\if@jsc@warnoldfontcmdexception
2601 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand
2602 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2603   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2604   \DeclareOldFontCommand{#1}{%
2605     \bxjs@oldfontcmd{#1}#2%
2606   }{%

```

```

2607 \bxjs@oldfontcmd{#1}#3%
2608 }%
2609 }
2610 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2611 \ClassInfo\bxjs@clsname
2612 {Old font command '\string#1' is used!!\MessageBreak
2613 The first occurrence is}%
2614 }

```

---

`\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。

```

2615 \newcommand*{\allowoldfontcommands}{%
2616 \@jsc@warnoldfontcmdexceptiontrue}
2617 \newcommand*{\disallowoldfontcommands}{%
2618 \@jsc@warnoldfontcmdexceptionfalse}

2619 \let\bxjs@oldfontcmd@list\@empty
2620 \def\bxjs@oldfontcmd#1{%
2621 \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2622 \def\bxjs@oldfontcmd@a#1#2{%
2623 \if@jsc@warnoldfontcmdexception\else
2624 \global\@jsc@warnoldfontcmdfalse
2625 \ifx#1\relax
2626 \global\let#1=t%
2627 \jsc@warnoldfontcmd{#2}%
2628 \fi
2629 \fi}
2630 \def\bxjs@warnoldfontcmd@final{%
2631 % \par
2632 \global\let\bxjs@warnoldfontcmd@final\@empty
2633 \let\@tempa\@empty
2634 \def\do##1{%
2635 \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2636 \edef\@tempa{\@tempa \space\string##1}\fi}
2637 \bxjs@oldfontcmd@list
2638 \ifx\@tempa\@empty\else
2639 \ClassWarningNoLine\bxjs@clsname
2640 {Some old font commands were used in text:\MessageBreak
2641 \space\@tempa\MessageBreak
2642 You should note, that since 1994 LaTeX2e provides a\MessageBreak
2643 new font selection scheme called NFSS2 with several\MessageBreak
2644 new, combinable font commands. The
2645 class provides\MessageBreak
2646 the old font commands only for compatibility}
2647 \fi}

```

単純に `\AtEndDocument` のフックの中で `\bxjs@warnoldfontcmd@final` を実行した場

合、最終ページのヘッダ・フッタの中にある二文字フォント命令はそれより後に実行されるため捕捉できない。これに対処するため、`\end{document}` 中に実行される `\clearpage` の処理の直後に `\bxjs....final` が呼ばれるようにする。

```
2648 \def\bxjs@warnoldfontcmd@kick@final{%
2649   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2650 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}
```

---

`\mc` フォントファミリーを変更します。

```
\gt 2651 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2652 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2653 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2654 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2655 \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。

```
2656 \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プスは数式中では何もしま  
`\sl` せん（警告メッセージを出力します）。通常のアップライト体に戻すコマンドは `\upshape`  
`\sc` です。

```
2657 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2658 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2659 \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 2660 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2661 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

## 10 相互参照

### 10.1 目次の類

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

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

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

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

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

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

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

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

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

という書式です。

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

**インデント** 左側の字下げ量です。

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

`\@pnumwidth` ページ番号の入る箱の幅です。

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

`\@dotsep` 点の間隔です (単位 mu)。

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

```
2662 \newcommand\@pnumwidth{1.55em}
2663 \newcommand\@tocrmarg{2.55em}
2664 \newcommand\@dotsep{4.5}
2665 %<!book&!report>\setcounter{tocdepth}{2}
2666 %<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)

```
2667 \newdimen\jsc@tocl@width
2668 \newcommand{\tableofcontents}{%
2669 %<*book|report>
2670 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2671 \settowidth\@tempdima{\headfont\appendixname}%
2672 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2673 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2674 \if@twocolumn
2675 \@restonecoltrue\onecolumn
2676 \else
2677 \@restonecolfalse
2678 \fi
2679 \chapter*{\contentsname}%
2680 \@mkboth{\contentsname}{%}
```



```

2681 %</book|report>
2682 %<!*book&!report>
2683 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2684 \settowidth\@tempdima{\headfont\appendixname}%
2685 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2686 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2687 \section*{\contentsname}%
2688 \@mkboth{\contentsname}{\contentsname}%
2689 %</!book&!report>
2690 \@starttoc{toc}%
2691 %<book|report> \if@restonecol\twocolumn\fi
2692 }

```

\l@part 部の目次です。

```

2693 \newcommand*{\l@part}[2]{%
2694 \ifnum \c@tocdepth >-2\relax
2695 %<!book&!report> \addpenalty\@secpenalty
2696 %<book|report> \addpenalty{-\@highpenalty}%
2697 \addvspace{2.25em \@plus\p@?}%
2698 \begingroup
2699 \parindent \z@
2700 % \@pnumwidth should be \@tocrmarg
2701 % \rightskip \@pnumwidth
2702 \rightskip \@tocrmarg
2703 \parfillskip -\rightskip
2704 {\leavevmode
2705 \large \headfont
2706 \setlength\@lnumwidth{4\jsZw}%
2707 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2708 \nobreak
2709 %<book|report> \global\@nobreaktrue
2710 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2711 \endgroup
2712 \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)

```

2713 %<*book|report>
2714 \newcommand*{\l@chapter}[2]{%
2715 \ifnum \c@tocdepth >\m@ne
2716 \addpenalty{-\@highpenalty}%
2717 \addvspace{1.0em \@plus\p@?}
2718 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2719 \begingroup
2720 \parindent\z@
2721 % \rightskip\@pnumwidth
2722 \rightskip\@tocrmarg
2723 \parfillskip-\rightskip

```

```

2724 \leavevmode\headfont
2725 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2726 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2727 \advance\leftskip\@lnumwidth \hskip-\leftskip
2728 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2729 \penalty\@highpenalty
2730 \endgroup
2731 \fi}
2732 %</book|report>

```

\l@section 節の目次です。

```

2733 %<*!book&!report>
2734 \newcommand*{\l@section}[2]{%
2735 \ifnum \c@tocdepth >\z@
2736 \addpenalty{\@secpenalty}%
2737 \addvspace{1.0em \@plus\p@}%
2738 \begingroup
2739 \parindent\z@
2740 % \rightskip\@pnumwidth
2741 \rightskip\@tocrmarg
2742 \parfillskip-\rightskip
2743 \leavevmode\headfont
2744 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2745 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2746 \advance\leftskip\@lnumwidth \hskip-\leftskip
2747 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2748 \endgroup
2749 \fi}
2750 %</!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。

```

2751 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

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

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

```

\l@subparagraph 2752 %<*!book&!report>
2753 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2754 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2755 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2756 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2757 %
2758 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2759 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2760 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2761 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2762 %

```

```

2763 \newcommand*{\l@section}{%
2764     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2765     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2766 \newcommand*{\l@subsubsection}{%
2767     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2768     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2769 \newcommand*{\l@paragraph}{%
2770     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2771     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2772 \newcommand*{\l@subparagraph}{%
2773     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2774     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2775 %</!book&!report>
2776 %<*book|report>
2777 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2778 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2779 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2780 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2781 \newcommand*{\l@section}{%
2782     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2783     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2784 \newcommand*{\l@subsection}{%
2785     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2786     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2787 \newcommand*{\l@subsubsection}{%
2788     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2789     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2790 \newcommand*{\l@paragraph}{%
2791     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2792     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2793 \newcommand*{\l@subparagraph}{%
2794     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2795     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2796 %</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。

```

2797 \newdimen\@lnumwidth
2798 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

`\@dottedtocline` L<sup>A</sup>T<sub>E</sub>X 本体 (ltsect.dtx 参照) での定義と同じですが、`\@tempdima` を `\@lnumwidth` に `\jsTocLine` 変えています。

[2018-06-23] デフォルトでは . . . . . のようにベースラインになります。

これを変更可能にするため、`\jsTocLine` というマクロに切り出しました。例えば、仮想ボディの中央 . . . . . に変更したい場合は

```
\renewcommand{\jsTocLine}{\leaders \hbox {\hss · \hss}\hfill}
```

とします。

```
2799 \def\jsTocLine{\leaders\hbox{%
2800   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2801 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2802   \vskip \z@ \@plus.2\p@?
2803   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2804     \parindent #2\relax\@afterindenttrue
2805     \interlinepenalty\@M
2806     \leavevmode
2807     \@lnumwidth #3\relax
2808     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2809     {#4}\nobreak
2810     \jsTocLine \nobreak\hbext@\@pnumwidth{%
2811       \hfil\normalfont \normalcolor #5}\par}\fi}
```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```
2812 \newcommand{\listoffigures}{%
2813   %<*book|report>
2814   \if@twocolumn\@restonecoltrue\onecolumn
2815   \else\@restonecolfalse\fi
2816   \chapter*{\listfigurename}%
2817   \@mkboth{\listfigurename}{}%
2818   %</book|report>
2819   %<!*book&!report>
2820   \section*{\listfigurename}%
2821   \@mkboth{\listfigurename}{\listfigurename}%
2822   %</!*book&!report>
2823   \@starttoc{lof}%
2824   %<book|report> \if@restonecol\twocolumn\fi
2825 }
```

`\l@figure` 図目次の項目を出力します。

```
2826 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```
2827 \newcommand{\listoftables}{%
2828   %<*book|report>
2829   \if@twocolumn\@restonecoltrue\onecolumn
2830   \else\@restonecolfalse\fi
2831   \chapter*{\listtablename}%
2832   \@mkboth{\listtablename}{}%
2833   %</book|report>
2834   %<!*book&!report>
2835   \section*{\listtablename}%
2836   \@mkboth{\listtablename}{\listtablename}%

```

```

2837 %</!book&!report>
2838 \starttoc{lot}%
2839 %<book|report> \if@restonecol\twocolumn\fi
2840 }

```

`\l@table` 表目次は図目次と同じです。

```

2841 \let\l@table\l@figure

```

## 10.2 参考文献

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

```

2842 \newdimen\bibindent
2843 \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。

```

2844 \newenvironment{thebibliography}[1]{%
2845   \@jsc@warnoldfontcmdexceptiontrue
2846   \global\let\presectionname\relax
2847   \global\let\postsectionname\relax
2848   %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2849   %<*kiyou>
2850   \vspace{1.5\baselineskip}
2851   \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2852   \vspace{0.5\baselineskip}
2853   %</kiyou>
2854   %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2855   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2856   \list{\@biblabel{\@arabic{c@enumiv}}}%
2857     {\settowidth\labelwidth{\@biblabel{#1}}%
2858     \leftmargin\labelwidth
2859     \advance\leftmargin\labelsep
2860     \@openbib@code
2861     \usecounter{enumiv}%
2862     \let\p@enumiv\@empty
2863     \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2864   %<kiyou> \small
2865   \sloppy
2866   \clubpenalty4000
2867   \@clubpenalty\clubpenalty
2868   \widowpenalty4000%
2869   \sfcode`\.\@m}
2870 {\def\@noitemerr
2871   {\@latex@warning{Empty `thebibliography' environment}}}%
2872 \endlist}

```

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

```
2873 \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。

```
2874 \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。

```
2875 % \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、必要

`\@citex` に応じて生かしてください。かつこの前後に入るグルーを `\jsInhibitGlue` で取っていますので、オリジナル同様、Knuth-`\cite{knu}` のように半角空白で囲んでください。

```
2876 % \def\@citex[#1]#2{\leavevmode
2877 %   \let\@citea\empty
2878 %   \@cite{\@for\@citeb:=#2\do
2879 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2880 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2881 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2882 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2883 %       \G@refundefinedtrue
2884 %       \@latex@warning
2885 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2886 %       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2887 % \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。

```
2888 % \DeclareRobustCommand\cite{\unskip
2889 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2890 %   \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2891 %     , \jsInhibitGlue\ #2\fi}}}$}
```

### 10.3 索引

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

```
2892 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2893   \if@twocolumn
2894     \onecolumn\@restonecolfalse
2895   \else
2896     \clearpage\@restonecoltrue
2897   \fi
```

```

2898 \columnseprule.4pt \columnsep 2\jsZw
2899 \ifx\multicols\@undefined
2900 %<book|report> \twocolumn[\@makeschapterhead{\indexname}%
2901 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2902 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2903 %<!book&!report> \twocolumn[\section*{\indexname}]]%
2904 \else
2905 \ifdim\textwidth<\fullwidth
2906 \setlength{\evensidemargin}{\oddsidemargin}
2907 \setlength{\textwidth}{\fullwidth}
2908 \setlength{\linewidth}{\fullwidth}
2909 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}%
2910 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2911 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2912 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]]%
2913 \else
2914 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}%
2915 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2916 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2917 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]]%
2918 \fi
2919 \fi
2920 %<book|report> \mkboth{\indexname}{}%
2921 %<!book&!report> \mkboth{\indexname}{\indexname}%
2922 \plainifnotempty % \thispagestyle{plain}
2923 \parindent\z@
2924 \parskip\z@ \@plus .3\p@?\relax
2925 \let\item\@idxitem
2926 \raggedright
2927 \footnotesize\narrowbaselines
2928 }{
2929 \ifx\multicols\@undefined
2930 \if@restonecol\onecolumn\fi
2931 \else
2932 \end{multicols}
2933 \fi
2934 \clearpage
2935 }

```

\@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 2936 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2937 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2938 \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。

```
2939 \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。

```
2940 \newcommand\seename{\if@english see\else →\fi}  
2941 \newcommand\alsoname{\if@english see also\else →\fi}
```

## 10.4 脚注

`\footnote` 和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため、`\footnotemark` `\inhibitglue` を入れることにします。pL<sup>A</sup>T<sub>E</sub>X の日付が 2016/09/03 より新しい場合は、このパッチが不要なのであてません。

---

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

---

```
2942 \ifx\pltx@foot@penalty\@undefined  
2943 \let\footnotes@ve=\footnote  
2944 \def\footnote{\jsInhibitGlue\footnotes@ve}  
2945 \let\footnotemarks@ve=\footnotemark  
2946 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}  
2947 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

---

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

---

`\thefootnote` 脚注番号に \* 印が付くようにしました。ただし、番号がゼロのときは \* 印も脚注番号も付きません。

[2003-08-15] `\textasteriskcentered` ではフォントによって下がりすぎるので変更しました。

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

```
2948 \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。

```
2949 % \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。

```
2950 \renewcommand{\footnoterule}{%
2951   \kern-2.6\p@? \kern-.4\p@
2952   \hrule width .4\columnwidth
2953   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2954 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

[2018-03-11] `\next` などいくつかの内部命令を `\jsc@...` 付きのユニークな名前にしました。

```
2955 \long\def\@footnotetext{%
2956   \insert\footins\bgroup
2957     \normalfont\footnotesize
2958     \interlinepenalty\interfootnotelinepenalty
2959     \splittopskip\footnotesep
2960     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2961     \hsize\columnwidth \@parboxrestore
2962     \protected@edef\@currentlabel{%
2963       \csname p@footnote\endcsname\@thefnmark
2964     }%
2965     \color@begingroup
2966     \makefnintext{%
2967       \rule\z@\footnotesep\ignorespaces}%
2968     \futurelet\jsc@next\jsc@fo@t}
2969 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@fo@t
2970               \else \let\jsc@next\jsc@fo@t\fi \jsc@next}
2971 \def\jsc@fo@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
2972 \def\jsc@fo@t#1{#1\jsc@@foot}
2973 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
2974   \ifx\pltx@foot@penalty\@undefined\else
2975     \ifhmode\null\fi
2976     \ifnum\pltx@foot@penalty=\z@\else
2977       \penalty\pltx@foot@penalty
2978       \pltx@foot@penalty\z@
2979     \fi
2980   \fi}
```

`\@makefntext` 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。

```
2981 \newcommand\@makefntext[1]{%
2982   \advance\leftskip 3\jsZw
2983   \parindent 1\jsZw
2984   \noindent
2985   \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。

```
2986 % \def\@xfootnotenext[#1]{%
2987 %   \begingroup
2988 %     \ifnum#1>\z@
2989 %       \csname c@\@mpfn\endcsname #1\relax
2990 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2991 %     \else
2992 %       \unrestored@protected@xdef\@thefnmark{}%
2993 %     \fi
2994 %   \endgroup
2995 %   \@footnotetext}
```

---

ここまでのコードは JS クラスを踏襲する。

---

## 11 段落の頭へのグルー挿入禁止

段落頭のかぎ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。

```
2996 \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。

```
2997 \def\everyparhook{\jsInhibitGlueAtParTop}
2998 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2999 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3000 \fi
```

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

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

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

---

次の `\@inhibitglue` は JS クラスでの `\jsInhibitGlueAtParTop` の実装である。エンジンが (u)platex の場合はこれを採用する。

---

```
3001 \ifx j\jsEngine
3002 \def\@inhibitglue{%
3003   \futurelet\@let@token\@@inhibitglue}
3004 \begingroup
3005 \let\GDEF=\gdef
3006 \let\CATCODE=\catcode
3007 \let\ENDGROUP=\endgroup
3008 \CATCODE`k=12
3009 \CATCODE`a=12
3010 \CATCODE`n=12
3011 \CATCODE`j=12
3012 \CATCODE`i=12
3013 \CATCODE`c=12
3014 \CATCODE`h=12
3015 \CATCODE`r=12
3016 \CATCODE`t=12
3017 \CATCODE`e=12
3018 \GDEF\KANJI@CHARACTER{kanji character }
3019 \ENDGROUP
3020 \def\@@inhibitglue{%
3021   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3022   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
3023   \def\jsc@ig@temp{#1}%
3024   \ifx\jsc@ig@temp\@empty
3025     \ifnum\the\inhibitxspcode`#2=2\relax
3026       \inhibitglue
3027     \fi
3028   \fi}
3029 \fi
```

---

ここからしばらく「(本物の) `\everypar` に追加した `\everyparhook` を保持する」ためのパッチ処理が続く。これは、`everyparhook=compat` の場合にのみ実行する。

---

```
3030 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
```

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

まず、環境の直後の段落です。

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

```
3031 \def\@doendpe{%
3032   \@endpetrue
3033   \def\par{%
3034     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3035   \everypar{{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] `minipage` 環境にも対策します。

```
3036 \def\@setminipage{%
3037   \@minipagetrue
3038   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3039 }
```

`\item` 命令の直後です。

```
3040 \def\@item[#1]{%
3041   \if@noparitem
3042     \@donoparitem
3043   \else
3044     \if@inlabel
3045       \indent \par
3046     \fi
3047     \ifhmode
3048       \unskip\unskip \par
3049     \fi
3050     \if@newlist
3051       \if@nobreak
3052         \@nbitem
3053       \else
3054         \addpenalty\@beginparpenalty
3055         \addvspace\@topsep
3056         \addvspace{-\parskip}%
3057       \fi
3058     \else
3059       \addpenalty\@itempenalty
3060       \addvspace\itemsep
3061     \fi
3062     \global\@inlabeltrue
3063   \fi
3064   \everypar{%
3065     \@minipagefalse
3066     \global\@newlistfalse
3067     \if@inlabel
3068       \global\@inlabelfalse
3069     {\setbox\z@\lastbox
```

```

3070      \ifvoid\z@
3071      \kern-\itemindent
3072      \fi}%
3073      \box\@labels
3074      \penalty\z@
3075      \fi
3076      \if@nobreak
3077      \@nobreakfalse
3078      \clubpenalty \@M
3079      \else
3080      \clubpenalty \@clubpenalty
3081      \everypar{\everyparhook}%
3082      \fi
3083      \everyparhook}%
3084      \if@noitemarg
3085      \@noitemargfalse
3086      \if@nmbrlist
3087      \refstepcounter\@listctr
3088      \fi
3089      \fi
3090      \sbox\@tempboxa{\makelabel{#1}}}%
3091      \global\setbox\@labels\hbox{%
3092      \unhbox\@labels
3093      \hskip \itemindent
3094      \hskip -\labelwidth
3095      \hskip -\labelsep
3096      \ifdim \wd\@tempboxa >\labelwidth
3097      \box\@tempboxa
3098      \else
3099      \hbox to\labelwidth {\unhbox\@tempboxa}%
3100      \fi
3101      \hskip \labelsep}%
3102      \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。

```

3103 \def\@afterheading{%
3104   \@nobreaktrue
3105   \everypar{%
3106     \if@nobreak
3107     \@nobreakfalse
3108     \clubpenalty \@M
3109     \if@afterindent \else
3110     {\setbox\z@\lastbox}%
3111     \fi
3112     \else
3113     \clubpenalty \@clubpenalty
3114     \everypar{\everyparhook}%
3115     \fi\everyparhook}}

```

---

「\everyparhook 用のパッチ処理」はここまで。

---

3116 \fi

\@gnewline についてはちょっと複雑な心境です。もともとの p $\text{\LaTeX}$  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。

---

※`luatexja` を読みこんだ場合に `lltjcore.sty` によって上書きされるのを防ぐため遅延させる。

---

```
3117 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3118 \AtEndOfPackage{%
3119 \def\@gnewline #1{%
3120   \ifvmode
3121     \@nolnerr
3122   \else
3123     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3124     \jsInhibitGlue \ignorespaces
3125   \fi}
3126 }
3127 \fi
```

## 12 いろいろなロゴ

$\text{\LaTeX}$  関連のロゴを作り直します。

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

---

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

※\小、\上小 の制御綴は定義しない。

---

```
3128 \if@jslogo
3129   \IfFileExists{jslogo.sty}{%
3130     \RequirePackage{jslogo}%
3131   }{%
3132     \ClassWarningNoLine\bxjs@clsname
3133       {The package 'jslogo' is not installed.\MessageBreak
3134       It is included in the recent release of\MessageBreak
```

```

3135         the 'jsclasses' bundle}
3136     }
3137 \fi

```

## 13 amsmath との衝突の回避

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

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

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

```

3138 %\let\ltx@ifnextchar\@ifnextchar
3139 %\def\ProvidesFile#1{%
3140 %   \begingroup
3141 %     \catcode\ 10 %
3142 %     \ifnum \endlinechar<256 %
3143 %       \ifnum \endlinechar>\m@ne
3144 %         \catcode\endlinechar 10 %
3145 %       \fi
3146 %     \fi
3147 %     \@makeother\/%
3148 %     \@makeother\&%
3149 %     \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 3150 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3151 \newcommand{\postpartname}{\if@english\else 部\fi}
3152 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3153 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3154 \newcommand{\presectionname}{}% 第
\postsectionname 3155 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3156 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3157 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3158 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname
\indexname

```

```

3159 \newcommand{\refname}{\if@english References\else 参考文献\fi}
3160 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3161 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 3162 %<!\jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3163 %<\jspf>\newcommand{\figurename}{Fig.~}
3164 %<!\jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3165 %<\jspf>\newcommand{\tablename}{Table~}

\appendixname

\abstractname 3166 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3167 \newcommand{\appendixname}{\if@english \else 付録\fi}
3168 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

---

環境変数 `SOURCE_DATE_EPOCH` / `FORCE_SOURCE_DATE` が設定されている場合は“今日”が過去・未来の日付になる可能性がある。BXJS クラスでは、和暦の扱いは `bxwareki` パッケージに任せる。

※ 2.0 版より、完全に `bxwareki` に任せる。

---

```

3169 \@onlypreamble\bxjs@decl@Seireki@cmds
3170 \@tempswafalse
3171 \if p\jsEngine \@tempswatruetrue \fi
3172 \if n\jsEngine \@tempswatruetrue \fi
3173 \bxjs@cond\if@tempswa\fi{%
3174 % 欧文 8bitTeX の場合
3175 \newif\ifjsSeireki \jsSeirekitruetrue
3176 \def\bxjs@decl@Seireki@cmds{%
3177   \def\西暦{\jsSeirekitruetrue}%
3178   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3179 \def\Seireki{\jsSeirekitruetrue}
3180 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3181 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3182 \def\bxjs@iai{\noexpand~}
3183 }{%
3184 \newif\if西暦 \西暦 true
3185 \def\bxjs@decl@Seireki@cmds{%
3186   \def\西暦{\西暦 true}%
3187   \def\和暦{\西暦 false\bxjs@wareki@used}}
3188 \def\Seireki{\西暦 true}
3189 \def\Wareki{\西暦 false\bxjs@wareki@used}
3190 \def\bxjs@if@use@seireki{\bxjs@cond\if西暦\fi}
3191 \let\bxjs@iai\@empty
3192 }

```



```

3193 \bxjs@decl@Seireki@cmds
3194 \let\bxjs@unxp@\firstofone \let\bxjs@onxp@\firstofone
3195 \bxjs@test@engine\unexpanded{%
3196   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}

```

\ifbxjs@bxwareki@avail bxwareki パッケージが利用できるか。

```

3197 \newif\ifbxjs@bxwareki@avail
3198 \IfFileExists{bxwareki.sty}{%
3199   \RequirePackage{bxwareki}[]%
3200   \bxjs@bxwareki@availtrue}{%

```

\bxjs@wareki@used 和暦が非対応の場合に警告を出す。

```

3201 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\@empty
3202 \else
3203   \bxjs@robust@def\bxjs@wareki@used{%
3204     \global\let\bxjs@wareki@used\@empty
3205     \ClassWarning\bxjs@clsname
3206       {Wareki mode is not supported, since\MessageBreak
3207         'bxwareki' is unavailable, found}}
3208   \g@addto@macro\bxjs@begin@document@hook{%
3209     \let\bxjs@wareki@used\@empty}
3210 \fi

```

\jayear 和暦における年の表記（元号 + 年数）。

\heisei 年数を表す整数レジスタで、元号が「平成」である場合にのみ定義される。

```

3211 \ifbxjs@bxwareki@avail
3212   \let\jayear\warekiyear
3213   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3214     \newcount\heisei \heisei=\value{warekiyear}
3215   \fi

```

bxwareki が使えない場合は西暦表示にする。

```

3216 \else
3217   \edef\jayear{\the\year \bxjs@iai}
3218 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3219 \let\bxjs@next\relax
3220 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3221   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3222 \fi\fi
3223 \edef\bxjs@today{%
3224   \if@english
3225     \ifcase\month\or
3226       January\or February\or March\or April\or May\or June\or
3227       July\or August\or September\or October\or November\or December\fi
3228     \space\number\day, \number\year
3229   \else

```

```

3230 \ifx\bxjs@next\relax \expandafter\@firstoftwo
3231 \else \noexpand\bxjs@if@use@seireki
3232 \fi {%
3233 \number\year\bxjs@iai\bxjs@unxp{年}%
3234 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3235 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3236 }{\bxjs@next}%
3237 \fi}
3238 \let\today\bxjs@today

```

---

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

```

3239 \g@addto@macro\bxjs@begin@document@hook{%
3240 \ifx\bbl@jpn@maybekansuji\@undefined\else
3241 \bxjs@decl@Seireki@cmds
3242 \g@addto@macro\datejapanese{%
3243 \let\today\bxjs@today}%
3244 \fi}

```

---

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

```

3245 \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。

```

3246 %<slide>\pagestyle{empty}%
3247 %<article|report>\pagestyle{plain}%
3248 %<book>\pagestyle{headings}%
3249 \pagenumbering{arabic}
3250 \if@twocolumn
3251 \twocolumn
3252 \sloppy
3253 \flushbottom
3254 \else
3255 \onecolumn
3256 \raggedbottom
3257 \fi
3258 %<*slide>
3259 \renewcommand\familydefault{\sfdefault}
3260 \raggedright
3261 %</slide>

```

■BXJS 独自の追加処理 

---

フックを実行する。

3262 \bxjs@pre@jadriver@hook

和文ドライバのファイルを読み込む。

3263 \input{bxjsja-\bxjs@jadriver.def}

おしまい。

3264 %</cls>

---

以上です。

## 付録 A 和文ドライバの仕様

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

- `\jsDocClass` [文字トークンの `let`] 文書クラスの種類を示し、次のいずれかと一致する (`\if` で判定可能)。
  - `\jsArticle` `bxjsarticle` クラス
  - `\jsBook` `bxjsbook` クラス
  - `\jsReport` `bxjsreport` クラス
  - `\jsSlide` `bxjsslide` クラス
- `\jsEngine` [文字トークンの `let`] 使用されているエンジンの種別。 (`\if` で判定可能)。
  - `p` `pdfTeX` (DVI モードも含む)
  - `l` `LuaTeX` (＃)
  - `x` `XYTeX`
  - `j` `pTeX` または `upTeX`
  - `n` 以上の何れでもない
- `\ifjsWithupTeX` [スイッチ] 使用されているエンジンが `upTeX` であるか。
- `\ifjsWitheTeX` [スイッチ] 使用されているエンジンが  $\epsilon$ -`TeX` 拡張であるか。
- `\ifjsInPdfMode` [スイッチ] 使用されているエンジンが (`pdfTeX`・`LuaTeX` の) PDF モードであるか。
- `\jsUnusualPtSize` [整数定数を表す文字列のマクロ] 基底フォントサイズが 10pt、11pt、12pt のいずれでもない場合の `\@ptsize` の値。 (`\@ptsize` 自体があまり有用でないと思われる。)
- `\jsScale` [実数を表す文字列のマクロ] 和文フォントサイズの要求サイズに対するスケール。クラスオプション `scale` で指定される。(既定値は 0.924715。)
- `\jsJaFont` [マクロ] 和文フォント設定を表す文字列。クラスオプション `jafont` で指定された値。
- `\jsJaParam` [マクロ] 和文モジュールに渡すパラメタを表す文字列。この値が何を表すかは決まっておらず、各々の和文モジュールが独自に解釈する。クラスオプション `japaram` で指定された値。
- `\jsInhibitGlue` [マクロ] `\inhibitglue` という命令が定義されていればそれを実行し、そうでなければ何もしない。JS クラスで `\inhibitglue` を用いている箇所は全て `\jsInhibitGlue` に置き換えられている。従って、`\inhibitglue` は未定義でも動作するが、その実装がある場合は BXJS クラスはそれを活用する。
- `\jsInhibitGlueAtParTop` [マクロ] ★ 段落先頭におけるカギ括弧の位置調整を行うマクロ。全ての段落先頭で呼び出される。
- `\jsZw` [内部寸法値] 「現在の全角幅」を表す変数。JS クラスで `zw` 単位で設定されている長さパラメタはこの変数を単位として設定されている。この変数の値は実際に

用いられる「和文フォント」のメトリックに基づくのではなく、機械的に `\jsScale` × (フォントサイズ) であると定められている (フォントサイズ変更の度に再設定される)。従って、「和文コンポーネント」はこの設定と辻褄が合うように和文フォントサイズを調整する必要がある。ほとんどの場合、和文フォントを NFSS で規定する際に `\jsScale` の値をスケール値として与えれば上手くいく。

- `\jsFontSizeChanged` [マクロ] フォントサイズが変更された時に必ず呼び出されるマクロ。
- `\jsResetDimen` [マクロ] ★ 上記 `\jsFontSizeChanged` の中で呼び出される、ユーザ (和文モジュール) 用のフック。フォントサイズに依存するパラメタをここで設定することができる。既定の定義は空。

以下で標準で用意されている和文ドライバの実装を示す。

```
3265 %<*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 Lua<sub>T</sub><sub>E</sub>X-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)p<sub>T</sub><sub>E</sub>X 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

```
3266 %<*minimal>
```

```
3267 %% 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.
```

```
3268 \def\DeclareJaTextFontCommand#1#2{%
```

```
3269   \DeclareRobustCommand#1[1]{%
```

```
3270     \relax
```

```
3271     \ifmmode \expandafter\nfss@text \fi
```

```
3272     {#2##1}}%
```

```
3273 }
```

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

```
3274 \def\DeclareJaMathFontCommand#1#2{%
```

```
3275   \DeclareRobustCommand#1[1]{%
```

```
3276     \relax
```

```
3277     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3278     \nfss@text{\fontfamily\familydefault
```

```
3279       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3280     #2##1}}%
```

```

3281 }%
3282 }

```

`\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。

```

3283 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3284 \@onlypreamble\bxjs@if@sf@default
3285 \def\bxjs@if@sf@default#1{%
3286   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3287   \g@addto@macro\bxjs@begin@document@hook{%
3288     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3289 }

```

`\jsInverseScale` `\jsScale` の逆数。

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

```

3290 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3291 \edef\jsInverseScale{\strip@pt\@tempdima}

```

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

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

※文字トークンは “`\the-文字列`” のカテゴリコードをもつ。

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

```

3292 \def\jsLetHeadChar#1#2{%
3293   \begingroup
3294     \escapechar=`\ %
3295     \let\bxjs@tmpa={% brace-match-hack
3296       \bxjs@let@hchar@exp#2}%
3297   \endgroup
3298   \let#1\bxjs@g@tmpa}
3299 \def\bxjs@let@hchar@exp{%
3300   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3301 \def\bxjs@let@hchar@exp@a{%
3302   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
3303     \bxjs@let@hchar@out\let\relax
3304   }\bxjs@cond@ifcat\noexpand\@let@token\sptoken\fi{% 空白
3305     \bxjs@let@hchar@out\let\space%
3306   }\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
3307     \bxjs@let@hchar@out\let\@backslashchar
3308   }\bxjs@let@hchar@exp@b}}}%
3309 \def\bxjs@let@hchar@exp@b#1{%
3310   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3311 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3312   %\message{<#1#2>}%
3313   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴

```

```

3314 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3315 \bxjs@let@hchar@out\let\relax
3316 }{%else
3317 \expandafter\bxjs@let@hchar@exp
3318 }%
3319 }{%else
3320 \bxjs@let@hchar@chr#1%
3321 }}
3322 \def\bxjs@let@hchar@chr#1{%
3323 \bxjs@let@hchar@out\def{{#1}}}}
3324 \def\bxjs@let@hchar@out#1#2{%
3325 \global#1\bxjs@g@tmpa#2\relax
3326 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3327 \chardef\bxjs@let@hchar@csta=128
3328 \chardef\bxjs@let@hchar@cstb=192
3329 \chardef\bxjs@let@hchar@cstc=224
3330 \chardef\bxjs@let@hchar@cstd=240
3331 \chardef\bxjs@let@hchar@cste=248
3332 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3333 \def\bxjs@let@hchar@chr@ue#1{%
3334 \@tempcnta=`#1\relax
3335 %\message{\the\@tempcnta}%
3336 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3337 \bxjs@let@hchar@chr@ue@a#1%
3338 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3339 \bxjs@let@hchar@out\let\relax
3340 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3341 \bxjs@let@hchar@chr@ue@b
3342 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3343 \bxjs@let@hchar@chr@ue@c
3344 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3345 \bxjs@let@hchar@chr@ue@d
3346 }{%else
3347 \bxjs@let@hchar@out\let\relax
3348 }}}}}
3349 \def\bxjs@let@hchar@chr@ue@a#1{%
3350 \bxjs@let@hchar@out\def{{#1}}}}
3351 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3352 \bxjs@let@hchar@out\def{{#1#2}}}}
3353 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3354 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3355 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3356 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

## B.2 (u)pTeX 用の設定

```
3357 \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。

```
3358 \def\bxjs@let@hchar@chr@pp#1#2{%
3359   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3360 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3361   %\message{(\meaning#3:\meaning#4)}%
3362   \bxjs@cond\if#1k\fi{%
3363     \bxjs@let@hchar@out\def{#{#4}}%
3364   }{%else
3365     \bxjs@let@hchar@chr@ue#3#4%
3366   }%
3367 \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。

```
3368 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3369 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3370 \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。

```
3371 \@onlypreamble\bxjs@declarefontshape
3372 \ifjsWithupTeX
3373 \def\bxjs@declarefontshape{%
3374   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3375   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3376   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3377   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3378 }
3379 \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。

```
3380 \else
3381 \def\bxjs@declarefontshape{%
3382   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3383   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3384   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```



```

3385 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3386 }
3387 \def\bxjs@sizereference{jis}
3388 \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。

```

3389 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3390   \def\bxjs@tmpb{#5}}
3391 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3392 \else
3393 \expandafter\expandafter\expandafter\bxjs@next
3394 \expandafter\string\the\jfont\relax
3395 \fi
3396 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3397                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3398   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3399    \expandafter\let\csname\bxjs@tmpa/\bxjs@tmpb\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。

```

3400 \begingroup
3401 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3402 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3403 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3404 % 幅が丁度 10pt なら補正は不要
3405 \ifdim\wd\z@=10pt
3406   \global\let\bxjs@scale\jsScale
3407 \else
3408 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3409   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3410   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3411   \bxjs@invscale\@tempdima\bxjs@tmpa
3412   \xdef\bxjs@scale{\strip@pt\@tempdima}
3413 \fi
3414 \endgroup
3415 %\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。

```

3416 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3417 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3418 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3419 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3420 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3421 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3422 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3423 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3424 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3425 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3426 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3427 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3428 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3429 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3430 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3431 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{-}
3432 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3433 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
3434 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
3435 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3436 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3437 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3438 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3439 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3440 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3441 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3442 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3443 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3444 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3445 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3446 \DeclareFontShape{\jsc@JTn}{gt}{b}{sl}{<->ssub*gt/bx/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。

※ 2020-02-02 の NFSS の改修に対する jsclasses の対策を取り入れた。

```

3447 \ifx\@rmfamilyhook\@undefined % old
3448 \DeclareRobustCommand\rmfamily
3449 {\not@math@alphabet\rmfamily\mathrm
3450 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3451 \DeclareRobustCommand\sffamily
3452 {\not@math@alphabet\sffamily\mathsf
3453 \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3454 \DeclareRobustCommand\ttfamily
3455 {\not@math@alphabet\ttfamily\mathtt
3456 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3457 \g@addto@macro\bxjs@begin@document@hook{%
3458 \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3459 % my definitions above should have been overwritten, recover it!
3460 % \selectfont is executed twice but I don't care about speed...
3461 \expandafter\g@addto@macro\csname rmfamily \endcsname
3462 {\kanjifamily\mcdefault\selectfont}%
3463 \expandafter\g@addto@macro\csname sffamily \endcsname
3464 {\kanjifamily\gtdefault\selectfont}%

```

```

3465 \expandafter\g@addto@macro\csname ttfamily \endcsname
3466 {\kanjifamily\gtdefault\selectfont}%
3467 \fi}
3468 \else % 2020-02-02
3469 \g@addto@macro\@rmfamilyhook
3470 {\prepare@family@series@update@kanji{mc}\mcdefault}
3471 \g@addto@macro\@sffamilyhook
3472 {\prepare@family@series@update@kanji{gt}\gtdefault}
3473 \g@addto@macro\@ttfamilyhook
3474 {\prepare@family@series@update@kanji{gt}\gtdefault}
3475 \fi
3476 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3477 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3478 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3479 \fi
3480 \bxjs@if@sf@default{%
3481 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3482 \selectfont

```

`\bxjs@parse@qh` の処理は p<sub>T</sub>E<sub>X</sub> 系では不要になるので無効化する（つまり `\jsSetQHLlength` は `\setlength` と等価になる）。

```

3483 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3484 \let\bxjs@parse@qh@a\@undefined
3485 \let\bxjs@parse@qh@b\@undefined

```

#### ■パラメタの設定

```

3486 \prebreakpenalty\jis"2147=10000
3487 \postbreakpenalty\jis"2148=10000
3488 \prebreakpenalty\jis"2149=10000
3489 \inhibitxspcode`!=1
3490 \inhibitxspcode`〒=2
3491 \xspcode`+=3
3492 \xspcode`\%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3493 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3494 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3495 \let\jsInhibitGlueAtParTop\@inhibitglue

```

`\jsResetDimen` は空のままでよい。

■組方向依存の処理 組方向判定の `if-トークン` (`\if?dir`) は 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。

```

3496 \begingroup
3497 \catcode`\!=0

```

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

```
3498 \gdef\bxjs@ptex@dir{%
3499   !iftdir t%
3500   !else!ifydir y%
3501   !else ?%
3502   !fi!fi}
```

新版の pT<sub>E</sub>X 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 pL<sub>A</sub>T<sub>E</sub>X 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。

```
3503 % 古い \@makefnmark の定義
3504 \long\def\bxjs@tmpa{\hbox{%
3505   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3506   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3507 \ifx\@makefnmark\bxjs@tmpa
3508 \long\gdef\@makefnmark{%
3509   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3510   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3511 \fi
3512 \endgroup
```

### B.3 pdfT<sub>E</sub>X 用の処理

```
3513 \else\ifx p\jsEngine
3514 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3515 \@onlypreamble\bxjs@cjk@loaded
3516 \def\bxjs@cjk@loaded{%
3517   \def\@footnotemark{%
3518     \leavevmode
3519     \ifhmode
3520       \edef\@x@sf{\the\spacefactor}%
3521       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3522         \unkern\unkern
3523         \ifdim\lastskip>\z@ \unskip \fi
3524       \fi\fi
3525       \nobreak
3526     \fi
3527     \@makefnmark
3528     \ifhmode \spacefactor\@x@sf \fi
3529     \relax}%
3530 \let\bxjs@cjk@loaded\relax
3531 }
3532 \g@addto@macro\bxjs@begin@document@hook{%
3533   \@ifpackageloaded{CJK}{%
3534     \bxjs@cjk@loaded
3535   }{}%
3536 }
```

## B.4 X<sub>Y</sub>TeX 用の処理

3537 \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。

```
3538 \def\bxjs@let@hchar@chr#1{%
3539   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3540   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3541     \bxjs@let@hchar@chr@xe
3542   }\bxjs@let@hchar@out\def{{#1}}}}
3543 \def\bxjs@let@hchar@chr@xe#1{%
3544   \lccode`0=`#1\relax
3545   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3546 \@onlypreamble\bxjs@do@precisetext
3547 \ifx\XeTeXgenerateactualtext\@undefined\else
3548   \def\bxjs@do@precisetext{%
3549     \XeTeXgenerateactualtext=\@ne}
3550 \fi
```

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

```
3551 \@onlypreamble\bxjs@do@simplejasetup
3552 \def\bxjs@do@simplejasetup{%
3553   \ifnum\XeTeXinterchartokenstate>\z@
3554   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3555     \jsSimpleJaSetup
3556     \ClassInfo\bxjs@clsname
3557     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3558   \fi\fi}
```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```
3559 \newcommand*{\jsSimpleJaSetup}{%
3560   \XeTeXlinebreaklocale "ja"\relax
3561   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3562   \XeTeXlinebreakpenalty=0\relax}
```

## B.5 後処理（エンジン共通）

3563 \fi\fi\fi

simplejasetup オプションの処理。

```
3564 \ifx\bxjs@do@simplejasetup\@undefined\else
3565   \g@addto@macro\bxjs@begin@document@hook{%
3566     \ifbxjs@simplejasetup
3567       \bxjs@do@simplejasetup
3568     \fi}
3569 \fi
```

precisetext オプションの処理。

```

3570 \ifbxjs@precisetext
3571   \ifx\bxjs@do@precisetext\@undefined
3572     \ClassWarning\bxjs@clsname
3573       {The current engine does not support the\MessageBreak
3574         'precise-text' option\@gobble}
3575   \else
3576     \bxjs@do@precisetext
3577   \fi
3578 \fi

```

■段落頭でのグルー挿入禁止 本体開始時において `\everyparhook` を検査して、“結局何もしない” ことになっている場合は、副作用を完全に無くするために `\everyparhook` を空にする。

```

3579 \g@addto@macro\bxjs@begin@document@hook{%
3580   \ifx\jsInhibitGlueAtParTop\@empty
3581     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3582     \ifx\everyparhook\bxjs@tmpa
3583       \let\everyparhook\@empty
3584     \fi
3585   \fi}

```

`\everyparhook=modern` の場合の、`\everyparhook` の有効化の実装。

※本体開始時ではなく最初から有効化していることに注意。

```

3586 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3587 \let\bxjs@everypar\everypar
3588 \newtoks\everypar
3589 \everypar\bxjs@everypar

```

そして本物の `\everypar` では、最後で常に `\everyparhook` が実行されるようにする。

```

3590 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3591 \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。

```

3592 \ifbxjs@fancyhdr

```

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

```

3593 \@onlypreamble\bxjs@adjust@fancyhdr
3594 \def\bxjs@adjust@fancyhdr{%

```

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

※和文なので \sl は無い方がよいはず。

```
3595 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3596 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3597 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3598 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3599 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3600 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3601 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3602 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3603 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3604 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3605 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3606 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3607 \def\bxjs@tmpa{\rm\thepage\strut}%
3608 \def\bxjs@tmpb{\thepage\strut}%
3609 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3610 \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。

```
3611 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3612 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3613 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3614 }\bxjs@tmpa
3615 \fi\fi
3616 \PackageInfo\bxjs@clsname
3617 {Patch to fancyhdr is applied\@gobble}}
```

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

```
3618 \def\bxjs@pagestyle@hook{%
3619 \ifpackageloaded{fancyhdr}{%
3620 \bxjs@adjust@fancyhdr
3621 \global\let\bxjs@adjust@fancyhdr\relax
3622 }{}}
```

\pagestyle にフックを入れ込む。

```
3623 \let\bxjs@org@pagestyle\pagestyle
3624 \def\pagestyle{%
3625 \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。

```
3626 \g@addto@macro\bxjs@begin@document@hook{%
3627 \bxjs@pagestyle@hook
3628 \global\let\bxjs@pagestyle@hook\relax}
```

## ■和文空白命令

```
3629 \ifbxjs@jaspace@cmd
```

`\jaenspace` 半角幅の水平空き。

```
3630 \def\jaenspace{\hskip.5\jsZw\relax}
```

`\jathinspace` 和欧文間空白を入れるユーザ命令。

※ `minimal` ではダミー定義。

```
3631 \def\jathinspace{\hskip\z@skip}
```

`\_` 全角空白文字 1 つからなる名前の制御綴。 `\zwspace` と等価になる。

```
3632 \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3633 \DeclareRobustCommand*{\jaspace}[1]{%
3634   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3635     \ClassError\bxjs@clsname
3636     {Unknown jaspaces: #1}{\@eha}%
3637   \else
3638     \csname bxjs@jaspace@@#1\endcsname
3639   \fi}
3640 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3641 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3642 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3643 \fi
```

終わり。

```
3644 \fi
```

以上で終わり。

```
3645 %</minimal>
```

## 付録 C 和文ドライバ : standard

標準のドライバ。

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

■和文フォント指定の扱い `standard` 和文ドライバでは `\jsJaFont` の値を和文フォントの“プリセット”の指定として用いる。プリセットの値は、`TeX Live` の `kanji-config-`



updmap コマンドで使う“ファミリー”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。

```
3646 %<*standard>
3647 %% このファイルは日本語文字を含みます
3648 \input{bxjsja-minimal.def}
```

simplejasetup は standard では無効になる。

```
3649 \bxjs@simplejasetupfalse
```

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

\ifbxjs@jp@jismmiv 2004JIS 字形を優先させるか。

```
3650 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。

3651 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3652 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3653 \define@key{bxjsStd}{jis2004}[true]{%
3654   \bxjs@set@keyval{jis2004}{#1}{}}
```

\ifbxjs@jp@units 和文用単位 (zw、zh、(true)Q、(true)H) を使えるようにするか。

```
3655 \newif\ifbxjs@jp@units

units オプションの処理。

3656 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3657 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3658 \define@key{bxjsStd}{units}[true]{%
3659   \bxjs@set@keyval{units}{#1}{}}
```

\bxjs@jp@font フォントパッケージの追加オプション。

```
3660 \let\bxjs@jp@font\@empty

font オプションの処理。

3661 \define@key{bxjsStd}{font}{%
3662   \edef\bxjs@jp@font{#1}}

実際の japaram の値を適用する。

3663 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3664 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

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

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

```

3665 \ifbxjs@jp@jismmiv
3666 \g@addto@macro\@classoptionslist{,jis2004}
3667 % \@ifpackagewith 判定への対策
3668 \PassOptionsToPackage{jis2004}{otf}
3669 \fi

```

■和文用単位をサポート エンジンが (u)pT<sub>E</sub>X の場合は `units` を無効にする。

```

3670 \if j\jsEngine
3671 \bxjs@jp@unitsfalse
3672 \fi

```

`units` パラメタが有効の場合は、`bxcalc` パッケージの `\usepTeXunits` 命令を実行して和文用単位を有効化する。

```

3673 \ifbxjs@jp@units
3674 \IfFileExists{bxcalc.sty}{%
3675 \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3676 \ifx\usepTeXunits\undefined
3677 \PackageWarningNoLine\bxjs@clsname
3678 {Cannot support pTeX units (zw etc.), since\MessageBreak
3679 the package 'bxcalc' is too old}%
3680 \bxjs@jp@unitsfalse
3681 \else \usepTeXunits
3682 \fi
3683 }{%else
3684 \PackageWarningNoLine\bxjs@clsname
3685 {Cannot support pTeX units (zw etc.), since\MessageBreak
3686 the package 'bxcalc' is unavailable}%
3687 \bxjs@jp@unitsfalse
3688 }
3689 \fi

```

`bxcalc` で和文用単位をサポートした場合は、`\bxjs@parse@qh` の処理は不要になるので無効化する。

```

3690 \ifbxjs@jp@units
3691 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3692 \let\bxjs@parse@qh@a\undefined
3693 \let\bxjs@parse@qh@b\undefined
3694 \fi

```

`\bxjs@let@lenexpr \bxjs@let@lenexpr\CS{(長さ式)}` : 長さ式に `bxcalc` の展開を適用した結果のトークン列を `\CS` に代入する。

```

3695 \ifbxjs@jp@units
3696 \def\bxjs@let@lenexpr#1#2{%
3697 \edef#1{#2}%
3698 \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3699 \else
3700 \def\bxjs@let@lenexpr{\edef}
3701 \fi

```

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

```
3702 \ifjsWithTeX
```

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

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

```
\jH 3703 \@tempdima=0.25mm
      3704 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
      3705 \let\jH\jQ
```

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

```
\trueH 3706 \ifjsc@mag
      3707 \@tempdimb=\jsBaseFontSize\relax
      3708 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
      3709 \@tempdima=2.5mm
      3710 \bxjs@invscale\@tempdima\bxjs@tmpa
      3711 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
      3712 \@tempdima=10pt
      3713 \bxjs@invscale\@tempdima\bxjs@tmpa
      3714 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
      3715 \else \let\trueQ\jQ \let\bxjs@truept\p@
      3716 \fi
      3717 \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。

```
3718 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3719 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3720 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3721 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3722 \fi
```

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

```
3723 \def\jafontsize#1#2{%
3724 \begingroup
3725 \bxjs@jafontsize@a{#1}%
3726 \@tempdimb\jsInverseScale\@tempdima
3727 \bxjs@jafontsize@a{#2}%
3728 \xdef\bxjs@g@tmpa{%
3729 \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3730 \endgroup\bxjs@g@tmpa}
3731 \def\bxjs@jafontsize@a#1{%
```

```

3732 \bxjs@parse@qh{#1}%
3733 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3734 \@defaultunits\@tempdima\bxjs@tmpb pt\relax\@nnil}

```

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

`\bxjs@kanjiskip` 和文間空白の量を表すテキスト。

```
3735 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```

3736 \newcommand*\setkanjiskip[1]{%
3737 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3738 \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3739 \newcommand*\getkanjiskip{%
3740 \bxjs@kanjiskip}

```

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

```
3741 \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。(p<sub>TEX</sub> 以外)

```

\bxjs@disable@kanjiskip 3742 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3743 \bxjs@kanjiskip@enabledtrue
3744 \bxjs@reset@kanjiskip}
3745 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3746 \bxjs@kanjiskip@enabledfalse
3747 \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。

```

3748 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3749 \ifbxjs@kanjiskip@enabled
3750 \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3751 \else \@tempskipa\z@
3752 \fi
3753 \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 3754 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3755 \newcommand*\setxkanjiskip[1]{%
```

```
3756 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3757 \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3758 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3759 \bxjs@xkanjiskip}
```

```
3760 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3761 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3762 \bxjs@xkanjiskip@enabledtrue
```

```
3763 \bxjs@reset@xkanjiskip}
```

```

3764 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3765   \bxjs@xkanjiskip@enabledfalse
3766   \bxjs@reset@xkanjiskip}
3767 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3768   \ifbxjs@xkanjiskip@enabled
3769     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3770   \else \@tempskipa\z@
3771   \fi
3772   \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。

```

3773 \g@addto@macro\jsResetDimen{%
3774   \bxjs@reset@kanjiskip
3775   \bxjs@reset@xkanjiskip}
3776 \let\bxjs@apply@kanjiskip\relax
3777 \let\bxjs@apply@xkanjiskip\relax

```

#### ■和文フォント指定の扱い

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

```

3778 \@onlypreamble\bxjs@adjust@jafont
3779 \def\bxjs@adjust@jafont#1{%
3780   \ifx\jsJaFont\bxjs@@auto
3781     \bxjs@get@kanjiEmbed
3782     \ifx\bxjs@kanjiEmbed\relax
3783       \let\bxjs@tmpa\@empty
3784     \else
3785       \let\bxjs@tmpa\bxjs@kanjiEmbed
3786     \fi
3787   \else
3788     \let\bxjs@tmpa\jsJaFont
3789   \fi
3790   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3791     \ClassWarningNoLine\bxjs@clsname
3792     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3793     not available on the current situation}%
3794     \let\bxjs@tmpa\@empty
3795   \fi\fi
3796 }
3797 \def\bxjs@@auto{auto}
3798 \def\bxjs@@noEmbed{noEmbed}

```

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

```

3799 \let\bxjs@kanjiEmbed\relax

```

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

```
3800 \@onlypreamble\bxjs@get@kanjiEmbed
3801 \def\bxjs@get@kanjiEmbed{%
3802   \begingroup\setbox\z@=\hbox{%
3803     \global\let\bxjs@g@tmpa\relax
3804     \endlinechar\m@ne
3805     \let\do\@makeother\dospecials
3806     \catcode32=10 \catcode12=10 %form-feed
3807     \let\bxjs@tmpa\@empty
3808     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3809     \ifeof\@inputcheck\else
3810       \read\@inputcheck to\bxjs@tmpa
3811       \closein\@inputcheck
3812     \fi
3813     \ifx\bxjs@tmpa\@empty\else
3814       \openin\@inputcheck="\bxjs@tmpa"\relax
3815       \@tempswatrue
3816       \loop\if@tempswa
3817         \read\@inputcheck to\bxjs@tmpa
3818         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3819         \ifx\bxjs@tmpb\relax\else
3820           \global\let\bxjs@g@tmpa\bxjs@tmpb
3821           \@tempswafalse
3822         \fi
3823         \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3824         \ifx\bxjs@tmpb\relax\else
3825           \global\let\bxjs@g@tmpa\bxjs@tmpb
3826           \@tempswafalse
3827         \fi
3828         \ifeof\@inputcheck \@tempswafalse \fi
3829       \repeat
3830     \fi
3831   }\endgroup
3832   \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3833 }
3834 \@onlypreamble\bxjs@get@ke@a
3835 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3836   \ifx$#1$\def\bxjs@tmpb{#2}%
3837   \else \let\bxjs@tmpb\relax
3838   \fi}
3839 \@onlypreamble\bxjs@get@ke@b
3840 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3841   \ifx$#1$\def\bxjs@tmpb{#2}%
3842   \else \let\bxjs@tmpb\relax
3843   \fi}
```

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

```
3844 \newcommand*\jachar[1]{%
3845   \begingroup
```

\jsLetHeadChar で先頭の“文字”を拾ってそれを \bxjs@jachar に渡す。

```
3846 \jsLetHeadChar\bxjs@tmpa{#1}%
3847 \ifx\bxjs@tmpa\relax
3848 \ClassWarningNoLine\bxjs@clsname
3849 {Illegal argument given to \string\jachar}%
3850 \else
3851 \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3852 \fi
3853 \endgroup}
```

\jsJaChar を \jachar と等価にする。

```
3854 \let\jsJaChar\jachar
```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```
3855 \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。

```
3856 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3857 \@onlypreamble\bxjs@fix@hyperref@unicode
3858 \def\bxjs@fix@hyperref@unicode#1{%
3859 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3860 \@namedef{KV@Hyp@bxjs/hook}##1{%
3861 \KV@Hyp@unicode{##1}%
3862 \def\KV@Hyp@unicode####1{%
3863 \expandafter\ifx\csname if##1\expandafter\endcsname
3864 \csname if####1\endcsname\else
3865 \ClassWarningNoLine\bxjs@clsname
3866 {Blcoked hyperref option 'unicode=####1'}%
3867 \fi
3868 }%
3869 }%
3870 }
```

\jsCheckHyperrefUnicode 「hyperref の unicode オプションの値を検証する」ための本体開始時のフック。

```
3871 \@onlypreamble\jsCheckHyperrefUnicode
3872 \let\jsCheckHyperrefUnicode\@empty
3873 \g@addto@macro\bxjs@begin@document@hook{\jsCheckHyperrefUnicode}
```

\bxjs@check@hyperref@unicode hyperref の unicode オプションの値を本体開始時に検証する。

```
3874 \@onlypreamble\bxjs@check@hyperref@unicode
3875 \def\bxjs@check@hyperref@unicode#1{%
3876 \g@addto@macro\jsCheckHyperrefUnicode{%
3877 \@tempwafalse
3878 \begingroup
3879 \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3880 \aftergroup\@tempwatruue \fi
```

```

3881 \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3882 \csname if#1\endcsname
3883 \aftergroup\@tempwattrue \fi
3884 \endgroup
3885 \if@tempswa\else
3886 \ClassError\bxjs@clsname
3887 {The value of hyperref 'unicode' key is not suitable\MessageBreak
3888 for the present engine (must be #1)}%
3889 {\@ehc}%
3890 \fi}}

```

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

```

3891 \@onlypreamble\bxjs@urgent@special
3892 \def\bxjs@urgent@special#1{%
3893 \AtBeginDvi{\special{#1}}%
3894 \g@addto@macro\bxjs@begin@document@hook{%
3895 \ifpackageloaded{atbegshi}{%
3896 \begingroup
3897 \toks\z@{\special{#1}}%
3898 \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3899 \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3900 \endgroup
3901 }{}%
3902 }%
3903 }

```

`\bxjs@resolve@jafont@paren` jafont パラメタ値内の ( ) を解決する。`\bxjs@resolve@jafont@paren\CS` で、`\CS` の内容中の ( ... ) を `\bxjs@jafont@paren{...}` に置き換える。

```

3904 \@onlypreamble\bxjs@resolve@jafont@paren
3905 \def\bxjs@resolve@jafont@paren#1{%
3906 \def\bxjs@tmpb{\let#1}%
3907 \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3908 \@onlypreamble\bxjs@resolve@jafont@paren@a
3909 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3910 \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3911 \else
3912 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3913 \bxjs@tmpb\bxjs@tmpa
3914 \fi}

```

## C.2 pTeX 用設定

```
3915 \if j\jsEngine
```

### ■ 共通命令の実装

```

3916 \def\bxjs@apply@kanjiskip{%
3917 \kanjiskip\@tempskipa}
3918 \def\bxjs@apply@xkanjiskip{%
3919 \xkanjiskip\@tempskipa}

```



\jaJaChar のサブマクロ。

```
3920 \def\bxjs@jachar#1{%
3921   \bxjs@jachar@a#1...\@nil}
3922 \def\bxjs@jachar@a#1#2#3#4#5\@nil{%
```

引数が単一トークンなら和文文字トークンが得られたと見なし、それをそのまま出力する。

```
3923   \ifx.#2#1%
```

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

```
3924   \else\ifx.#3%
3925     \@tempcnta`#1 \multiply\@tempcnta64
3926     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3927     \bxjs@jachar@b
3928   \else\ifx.#4%
3929     \@tempcnta`#1 \multiply\@tempcnta64
3930     \advance\@tempcnta`#2 \multiply\@tempcnta64
3931     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3932     \bxjs@jachar@b
3933   \else
3934     \@tempcnta`#1 \multiply\@tempcnta64
3935     \advance\@tempcnta`#2 \multiply\@tempcnta64
3936     \advance\@tempcnta`#3 \multiply\@tempcnta64
3937     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3938     \bxjs@jachar@b
3939   \fi\fi\fi}
```

符号値が \@tempcnta の和文文字を出力する処理。

```
3940 \ifjsWithupTeX
3941   \def\bxjs@jachar@b{\kchar\@tempcnta}
3942 \else
3943   \def\bxjs@jachar@b{%
3944     \ifx\bxUInt\@undefined\else
3945       \bxUInt{\@tempcnta}%
3946     \fi}
3947 \fi
```

和欧文間空白の命令 \jathinspace の実装。

```
3948 \ifbxjs@jaspace@cmd
3949   \def\jathinspace{\hskip\xkanjiskip}
3950 \fi
```

■jis2004 パラメタ pxchfon と pxbabel では 2004JIS を指定するオプションの名が prefer2004jis である。

```
3951 \ifbxjs@jp@jismmiv
3952   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3953   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3954 \fi
```

■和文フォント指定の扱い pTeX は既定で kanji-config-updmap の設定に従うため、`\jsJaFont` が `auto` の場合は何もする必要がない。無指定でも `auto` でもない場合は、`\jsJaFont` をオプションにして `pxchfon` パッケージを読み込む。ここで、和文ドライバパラメタ `font` が指定されている場合は、その値を `pxchfon` のオプションに追加する。

```

3955 \let\bxjs@jafont@paren\@firstofone
3956 \let\bxjs@tmpa\jsJaFont
3957 \ifx\bxjs@tmpa\bxjs@@auto
3958   \let\bxjs@tmpa\@empty
3959 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3960   \def\bxjs@tmpa{noembed}
3961 \fi\fi
3962 \bxjs@resolve@jafont@paren\bxjs@tmpa
3963 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3964 \ifx\bxjs@tmpa\@empty\else
3965   \edef\bxjs@next{%
3966     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3967   }\bxjs@next
3968 \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。

※`scale` 対応は 1.7b6 版 [2013/11/17] から。

※ `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……。)

```

3969 \begingroup
3970   \global\let\bxjs@g@tmpa\relax
3971   \catcode`\|=0 \catcode`\|=12
3972   |def|bxjs@tmpdo#1|@nil{%
3973     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
3974   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
3975     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
3976   |catcode`\|=0 \catcode`\|=12
3977   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
3978     \ifx$#2$\else
3979       \xdef\bxjs@g@tmpa{%
3980         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3981     \fi}
3982 \@firstofone{%
3983   \catcode10=12 \endlinechar\m@ne
3984   \let\do\@makeother \dospecials \catcode32=10
3985   \openin\@inputcheck=otf.sty\relax
3986   \@tempswatrue
3987   \loop\if@tempswa
3988     \ifeof\@inputcheck \@tempswafalse \fi
3989     \if@tempswa
3990       \read\@inputcheck to\bxjs@next

```

```

3991 \expandafter\bxjs@tmpdo\bxjs@next\@nil
3992 \fi
3993 \repeat
3994 \closein\@inputcheck
3995 \endgroup}
3996 \bxjs@g@tmpa

```

■hyperref 対策 unicode にはいけない。

```

3997 \ifbxjs@hyperref@enc
3998 \bxjs@check@hyperref@unicode{false}
3999 \fi

```

tounicode special 命令を出力する。

```

4000 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4001 \else\ifjsWithpTeXng T\else F\fi\fi T%
4002 \ifnum\jis"2121="A1A1 %euc
4003 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4004 \else\ifnum\jis"2121="8140 %sjis
4005 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4006 \else\ifnum\jis"2121="3000 %uptex
4007 \ifbxjs@bigcode
4008 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4009 \PassOptionsToPackage{bigcode}{pxjahyper}
4010 \else
4011 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4012 \fi
4013 \fi\fi\fi
4014 \let\bxToUnicodeSpecialDone=t
4015 \fi

```

■和文数式ファミリー 和文数式ファミリーは既定で有効とする。すなわち enablejfam=false 以外の場合は @enablejfam を真にする。

```

4016 \ifx f\bxjs@enablejfam\else
4017 \@enablejfamtrue
4018 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

4019 \if@enablejfam
4020 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4021 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4022 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4023 \jfam\symmincho
4024 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4025 \g@addto@macro\bxjs@begin@document@hook{%
4026 \ifx\reDeclareMathAlphabet\undefined\else
4027 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4028 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4029 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4030 \fi}

```

4031 \fi

### C.3 pdf<sub>TEX</sub> 用設定 : CJK + bxcjkjatype

4032 \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。

※Pandoc モードでは autotilde を指定しない。

```
4033 \bxjs@adjust@jafont{f}
4034 \let\bxjs@jafont@paren\@firstofone
4035 \bxjs@resolve@jafont@paren\bxjs@tmpa
4036 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4037 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4038 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4039   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4040 \fi
4041 \edef\bxjs@next{%
4042   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkjatype}[2013/10/15]% v0.2c
4043 }\bxjs@next
4044 \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。

```
4045 \ifbxjs@hyperref@enc
4046   \PassOptionsToPackage{unicode}{hyperref}
4047 \fi
```

\hypersetup 命令で（CJK\* 環境に入れなくても）日本語文字を含む文書情報を設定できるようにするための細工。

※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。

```
4048 \ifx\bxchkjatypeHyperrefPatchDone\@undefined
4049 \begingroup
4050   \CJK@input{UTF8.bdg}
4051 \endgroup
4052 \g@addto@macro\pdfstringdefPreHook{%
4053   \@nameuse{CJK@UTF8Binding}%
4054 }
4055 \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。

```
4056 \ifx\bxchkjatypeHyperrefPatchDone\@undefined
4057 \g@addto@macro\pdfstringdefPreHook{%
```

```

4058 \ifx~\bxjs@@CJKtilde
4059 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4060 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4061 \let~\@empty
4062 \fi
4063 }
4064 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4065 \def\bxjs@@tildecmd{~}
4066 \def\bxjs@LetUnexpandableSpace#1{%
4067 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4068 \bxjs@org@LetUnexpandableSpace#1%
4069 \fi}
4070 \fi

```

#### ■ 共通命令の実装

```

4071 \newskip\jsKanjiSkip
4072 \newskip\jsXKanjiSkip
4073 \ifx\CJKecglue\@undefined
4074 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4075 \fi
4076 \let\autospacing\bxjs@enable@kanjiskip
4077 \let\noautospacing\bxjs@disable@kanjiskip
4078 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4079 \def\bxjs@apply@kanjiskip{%
4080 \jsKanjiSkip\@tempskipa
4081 \let\CJKglue\bxjs@CJKglue}
4082 \let\autoxspacing\bxjs@enable@xkanjiskip
4083 \let\noautoxspacing\bxjs@disable@xkanjiskip
4084 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4085 \def\bxjs@apply@xkanjiskip{%
4086 \jsXKanjiSkip\@tempskipa
4087 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4088 \def\bxjs@jachar#1{%
4089 \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4090 \ifbxjs@jaspace@cmd
4091 \protected\def\jathinspace{\CJKecglue}
4092 \fi

```

■ 和文数式ファミリ CJK パッケージは（恐らく）数式文字として CJK 文字をサポートしていない。従って @enablejfam は常に偽になる。

```

4093 \ifx t\bxjs@enablejfam
4094 \ClassWarningNoLine\bxjs@clsname
4095 {You cannot use 'enablejfam=true', since the\MessageBreak
4096 CJK package does not support Japanese math}
4097 \fi

```

## C.4 X<sub>Y</sub>TeX 用設定 : xeCJK + zxjatype

```
4098 \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。

```
4099 \RequirePackage{zxjatype}
4100 \PassOptionsToPackage{no-math}{fontspec}%!
4101 \PassOptionsToPackage{xetex}{graphicx}%!
4102 \PassOptionsToPackage{xetex}{graphics}%!
4103 \ifx\zxJaFamilyName\@undefined
4104   \ClassError{bxjs}{clsname
4105     {xeCJK or zxjatype is too old}}\@ehc
4106 \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。非指定の場合は原ノ味フォントを使用する。

※ 2.0 版より既定を IPAex から原ノ味に変更。

```
4107 \bxjs@adjust@jafont{f}
4108 \let\bxjs@jafont@paren\@gobble
4109 \bxjs@resolve@jafont@paren\bxjs@tmpa
4110 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4111 \ifx\bxjs@tmpa\@empty
4112   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4113   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4114 \else
4115   \edef\bxjs@next{%
4116     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4117   }\bxjs@next
4118 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X<sub>Y</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>Y</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>Y</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。

```
4119 \ifnum\stricmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4120   \ifbxjs@hyperref@enc
4121     \PassOptionsToPackage{unicode}{hyperref}
```

```
4122 \fi
4123 \fi
```

■段落頭でのグルー挿入禁止 どうやら、zxjatype の \inhibitglue の実装が極めて杜撰なため、1.0 版での実装では全く期待通りの動作をしていないし、そもそも（少なくとも現状の）xeCJK では、段落頭での \inhibitglue は実行しないほうが JS クラスの出力に近いものが得られるらしい。

従って、\jsInhibitGlueAtParTop は結局何もしないことにする。

強制改行直後のグルー禁止処理、のような怪しげな何か。

```
4124 \AtEndOfPackage{%
4125 \def\@gnewline #1{%
4126 \ifvmode \@nolnerr
4127 \else
4128 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4129 \nobreak \hskip-1sp\hskip1sp\relax
4130 \ignorespaces
4131 \fi}
4132 }
```

#### ■共通命令の実装

```
4133 \newskip\jsKanjiSkip
4134 \newskip\jsXKanjiSkip
4135 \ifx\CJKecglue\@undefined
4136 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4137 \fi
4138 \let\autospacing\bxjs@enable@kanjiskip
4139 \let\noautospacing\bxjs@disable@kanjiskip
4140 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4141 \def\bxjs@apply@kanjiskip{%
4142 \jsKanjiSkip\@tempskipa
4143 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4144 \let\autoxspacing\bxjs@enable@xkanjiskip
4145 \let\noautoxspacing\bxjs@disable@xkanjiskip
4146 \protected\def\bxjs@CJKeclue{\hskip\jsXKanjiSkip}
4147 \def\bxjs@apply@xkanjiskip{%
4148 \jsXKanjiSkip\@tempskipa
4149 \xeCJKsetup{CJKeclue={\bxjs@CJKeclue}}}
```

\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。

```
4150 \ifx\mcfamily\@undefined
4151 \protected\def\mcfamily{\CJKfamily{\CJKrmddefault}}
4152 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4153 \fi
```

\jachar のサブマクロの実装。

```
4154 \def\bxjs@jachar#1{%
4155 \xeCJKDeclareCharClass{CJK}{`#1}\relax
```

```
4156 #1}
```

`\jathinspace` の実装。

```
4157 \ifbxjs@jaspace@cmd
```

```
4158 \protected\def\jathinspace{\CJKecglue}
```

```
4159 \fi
```

■和文数式ファミリー 和文数式ファミリーは既定で無効とする。すなわち `enablejfam=true` の場合にのみ `@enablejfam` を真にする。

```
4160 \ifx t\bxjs@enablejfam
```

```
4161 \@enablejfamtrue
```

```
4162 \fi
```

実際に和文用の数式ファミリーの設定を行う。

※ FIXME: 要検討。

```
4163 \if@enablejfam
```

```
4164 \xeCJKsetup{CJKmath=true}
```

```
4165 \fi
```

## C.5 Lua<sub>T</sub><sub>E</sub>X 用設定 : Lua<sub>T</sub><sub>E</sub>X-ja

```
4166 \else\if l\jsEngine
```

■Lua<sub>T</sub><sub>E</sub>X-ja パッケージの読込 `luatexja` とともに `luatexja-fontspec` パッケージを読み込む。

`luatexja` は自前の `\zw` (これは実際の現在和文フォントに基づく値を返す) を定義するので、`\zw` の定義を消しておく。なお、レイアウト定義の「全角幅」は「規定」に基づく `\jsZw` であることに注意が必要。

※ 1.0b 版から「graphics パッケージに `pdftex` オプションを渡す」処理を行っていたが、1.4 版で廃止された。

```
4167 \let\zw\@undefined
```

```
4168 \RequirePackage{luatexja}
```

```
4169 \edef\bxjs@next{%
```

```
4170 \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
```

```
4171 }\bxjs@next
```

フォント代替の明示的定義。

```
4172 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{}
```

```
4173 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{}
```

```
4174 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{}
```

```
4175 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{}
```

```
4176 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{}
```

```
4177 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{}
```

```
4178 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{}
```

```
4179 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{}
```

```
4180 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{}
```

```
4181 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{}
```

```
4182 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{}
```



```

4183 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4184 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4185 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4186 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4187 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4188 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4189 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4190 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4191 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4192 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4193 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4194 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4195 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4196 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4197 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4198 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4199 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4200 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4201 \DeclareFontShape{JT3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }

```

■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 luatexja-preset を読み込む。非指定の場合は原ノ味フォントを指定する (luatexja-preset は読み込まない)。

※ 2.0 版より既定を IPAex から原ノ味に変更。

```

4202 \bxjs@adjust@jafont{t}
4203 \ifx\bxjs@tmpa\bxjs@@noEmbed
4204   \def\bxjs@tmpa{noembed}
4205 \fi
4206 \let\bxjs@jafont@paren\@gobble
4207 \bxjs@resolve@jafont@paren\bxjs@tmpa
4208 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4209 \ifx\bxjs@tmpa\@empty
4210   \defaultjfontfeatures{ Kerning=Off }
4211   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4212   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4213 \else
4214   \edef\bxjs@next{%
4215     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4216   }\bxjs@next
4217 \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。

```

4218 \DeclareRobustCommand\rmfamily
4219   {\not@math@alphabet\rmfamily\mathrm
4220     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}

```

```

4221 \DeclareRobustCommand\sffamily
4222   {\not@math@alphabet\sffamily\mathsf
4223     \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
4224 \DeclareRobustCommand\ttfamily
4225   {\not@math@alphabet\ttfamily\mathtt
4226     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4227 \g@addto@macro\bxjs@begin@document@hook{%
4228   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4229   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4230   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4231 \bxjs@if@sf@default{%
4232   \renewcommand\kanjifamilydefault{\gtdefault}}

```

#### ■和文パラメタの設定

```

4233 % 次の 3 つは既定値の通り
4234 \ltjsetparameter{prebreakpenalty={`',10000}}
4235 %\ltjsetparameter{postbreakpenalty={`",10000}}
4236 %\ltjsetparameter{prebreakpenalty={`",10000}}
4237 \ltjsetparameter{jaxspmode={`!,1}}
4238 \ltjsetparameter{jaxspmode={`〒,2}}
4239 \ltjsetparameter{alxspmode={`+,3}}
4240 \ltjsetparameter{alxspmode={`%,3}}

```

■段落頭でのグルー挿入禁止 基本的に現状の `ltjs*` クラスの処理に合わせる。

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の `LuaTEX-ja` で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合にに備えて同等のものを用意する。

```

4241 \ifx\ltjfakeparbegin\@undefined
4242   \protected\def\ltjfakeparbegin{%
4243     \ifhmode
4244       \relax\directlua{%
4245         luatexja.jfmglue.create_beginpar_node()}}
4246   \fi}
4247 \fi

```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```

4248 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4249 \begingroup
4250   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4251   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4252   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4253     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4254   \directlua{
4255     local function patchcmd(cs, code, from, to)
4256       tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4257         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."")
4258     end

```

```

4259 patchcmd(\@xsect, [[\meaning\xsect]],
4260 \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4261 patchcmd(\@item, [[\meaning@item]],
4262 \bxjs@tmpa, \@ltjfakeparbegin)}
4263 \endgroup
4264 \fi

```

■hyperref 対策 unicode にするべき。

※ 1.6c 版より、固定ではなく既定設定+検証に切り替えた。

```

4265 \ifbxjs@hyperref@enc
4266 \PassOptionsToPackage{unicode}{hyperref}
4267 \bxjs@check@hyperref@unicode{true}
4268 \fi

```

■共通命令の実装

```

4269 \protected\def\autospacing{%
4270 \ltjsetparameter{autospacing=true}}
4271 \protected\def\noautospacing{%
4272 \ltjsetparameter{autospacing=false}}
4273 \protected\def\autoxspacing{%
4274 \ltjsetparameter{autoxspacing=true}}
4275 \protected\def\noautoxspacing{%
4276 \ltjsetparameter{autoxspacing=false}}
4277 \def\bxjs@apply@kanjiskip{%
4278 \ltjsetparameter{kanjiskip={\@tempskipa}}}
4279 \def\bxjs@apply@xkanjiskip{%
4280 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4281 \def\bxjs@jachar#1{%
4282 \ltjjachar`#1\relax}

```

\jathinspace の実装。

```

4283 \ifbxjs@jaspace@cmd
4284 \protected\def\jathinspace{%
4285 \hskip\ltjgetparameter{xkanjiskip}\relax}
4286 \fi

```

■和文数式ファミリ LuaTeX-ja では和文数式ファミリは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablejfam は常に真になる。

```

4287 \ifx f\bxjs@enablejfam
4288 \ClassWarningNoLine\bxjs@clsname
4289 {You cannot use 'enablejfam=false', since the\MessageBreak
4290 LuaTeX-ja always provides Japanese math families}
4291 \fi

```

## C.6 共通処理 (2)

```
4292 \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 4293 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4294 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4295 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4296 \fi
```

`\mathmc` この時点で未定義である場合に限り、`\DeclareJaMathFontCommand` を利用したフォール

`\mathgt` バックの定義を行う。

```
4297 \ifx\mathmc\undefined
4298 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4299 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4300 \fi
```

### ■和文空白命令

\> 非数式中では `\jathinspace` と等価になるように再定義する。

※数式中では従来通り (`\:` と等価)。

```
4301 \ifbxjs@jaspace@cmd
4302 \bxjs@protected\def\bxjs@choice@jathinspace{%
4303 \relax\ifmmode \mskip\medmuskip
4304 \else \jathinspace\ignorespaces
4305 \fi}
4306 \jsAtEndOfClass{%
4307 \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4308 \else \def\>\protect\bxjs@choice@jathinspace}%
4309 \fi}
4310 \fi
```

### ■和文・和欧文間空白の初期値

```
4311 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4312 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4313 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4314 \fi
```

以上で終わり。

```
4315 %</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。

```
4316 %<*modern>
4317 \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。

```
4318 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4319 \def\encodingdefault{T1}%
4320 \input{t1enc.def}%
4321 \fontencoding\encodingdefault\selectfont
4322 \fi
```

基本フォントを Latin Modern フォントファミリに変更する。

※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。

```
4323 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4324 \renewcommand{\rmdefault}{lmr}
4325 \renewcommand{\sfdefault}{lms}
4326 \renewcommand{\ttdefault}{lmtt}
4327 \fi
```

大型演算子用の数式フォントの設定。

※`amsfonts` パッケージと同等にする。

```
4328 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4329   <-7.5>cmex7<7.5-8.5>cmex8%
4330   <8.5-9.5>cmex9<9.5->cmex10}{}%
4331 \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。
4332 \def\cmex@opt{10}
```

## D.2 fixltx2e 読み込

※`fixltx2e` 廃止前の L<sup>A</sup>T<sub>E</sub>X カーネルの場合。

```
4333 \ifx\@IncludeInRelease\@undefined
4334 \RequirePackage{fixltx2e}
4335 \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。

```
4336 \RequirePackage{bxjsjckcat}
```

## D.4 完了

おしまい。

```
4337 %</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。

```
4338 %<*pandoc>
```

```
4339 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4340 \RequirePackage{bxjspandoc}
```

`\bxjs@endpreamble@hook` etoolbox の `\AtEndPreamble` で実行される BXJS クラス用のフック。

※  $\varepsilon$ -TeX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```
4341 \@onlypreamble\bxjs@endpreamble@hook
```

```
4342 \let\bxjs@endpreamble@hook\empty
```

```
4343 \ifjsWithTeX
```

```
4344 \RequirePackage{etoolbox}[2010/08/21]% v2.0
```

```
4345 \AtEndPreamble{\bxjs@endpreamble@hook}
```

```
4346 \fi
```

### 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み込みが `\@filewithoptions` で指示されて、しかもそのファイルが読み込み済である場合に、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。

```
4347 \@onlypreamble\bxjs@set@dupload@proc
```

```
4348 \def\bxjs@set@dupload@proc#1{%
```

```
4349 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
4350 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
4351 \def\bxjs@set@dupload@proc@a#1{%
```

```
4352 \@onlypreamble#1\def#1##1}
```

```
4353 \def\bxjs@unset@dupload@proc#1{%
```

```
4354 \expandafter\let\csname bxjs@dlp/#1\endcsname\@undefined}
```

`\@if@ptions` `\@if@ptions` の再定義。

```
4355 \@onlypreamble\bxjs@org@if@ptions
```

```
4356 \let\bxjs@org@if@ptions\@if@ptions
```

```
4357 \@onlypreamble\bxjs@org@reset@ptions
```

```
4358 \let\bxjs@org@reset@ptions\relax
```

```
4359 \def\@if@ptions#1#2#3{%
```

```
4360 \let\bxjs@next\@secondoftwo
```

```
4361 \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
```

```

4362 \ifx\bxjs@tmpa\bxjs@tmpb
4363 \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4364 \let\bxjs@next\@firstoftwo \fi
4365 \fi
4366 \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4367 \g@addto@macro\bxjs@begin@document@hook{%
4368 \let\@if@options\bxjs@org@if@options}
4369 \@onlypreamble\bxjs@do@dupload@proc
4370 \def\bxjs@do@dupload@proc#1#2#3{%
4371 \ifx\bxjs@org@reset@options\relax
4372 \let\bxjs@org@reset@options\@reset@ptions
4373 \fi
4374 \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4375 \def\@reset@ptions{%
4376 \let\@reset@ptions\bxjs@org@reset@ptions
4377 \@reset@ptions
4378 \bxjs@next{#3}}%
4379 \@firstoftwo}

```

## 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。これを防ぐための対策を行う。

`\bxjs@polyglossia@options` Polyglossia のオプション列のテキスト。“実際には読み込まれていない” 場合は `\relax` になる。

```
4380 \let\bxjs@polyglossia@options\relax
```

`\bxjs@babel@options` Babel のオプション列のテキスト。“実際には読み込まれていない” 場合は `\relax` になる。

```
4381 \let\bxjs@babel@options\relax
```

■Polyglossia について つまり  $\text{Xe}\text{L}\text{a}\text{T}\text{E}\text{X}$ ・ $\text{Lua}\text{L}\text{a}\text{T}\text{E}\text{X}$  の場合。

※この場合 `etoolbox` が使用可能になっている。

```
4382 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0
```

パッケージの読込を検知するため読込済のマークを付けて `dupload` の処理を仕込む。

```

4383 \pandocSkipLoadPackage{polyglossia}
4384 \bxjs@set@dupload@proc{polyglossia.sty}{%
4385 \bxjs@unset@dupload@proc{polyglossia.sty}%
4386 \ClassWarning\bxjs@clsname
4387 {Package polyglossia is required}%
4388 \def\bxjs@polyglossia@options{#1}%

```

`polyglossia` の読込が指示された場合、直後に `\setmainlanguage` が実行されることを想定して、フック用の `\setmainlanguage` を定義する。

※先に `\setmainlanguage` 以外が実行された場合はエラーになる。

```
4389 \newcommand*\setmainlanguage[2] []{%
```

もし、言語名が空の `\setmainlanguage{}` が実行された場合は、`lang=ja` が指定されたと見なす。

```
4390 \ifblank{##2}{%
4391   \ClassWarning\bxjs@clsname
4392     {Main language is invalid, thus the fallback\MessageBreak
4393       definitions will be employed}%
4394   \bxjs@pandoc@polyglossia@ja
```

言語名が非空ならば、本来の `polyglossia` の処理を実行する。

```
4395 }{%else
4396   \ClassWarning\bxjs@clsname
4397     {Main language is valid (##2),\MessageBreak
4398       thus polyglossia will be loaded}%
4399   \csundef{ver@polyglossia.sty}%
4400   \edef\bxjs@next{%
4401     \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4402   }\bxjs@next
4403   \setmainlanguage[##1]{##2}%
4404 }
```

プレアンブルで `polyglossia` の読込が指示されなかった場合、`Polyglossia` と連携するパッケージの誤動作を防ぐため、読込済マークを外す。

```
4405 \g@addto@macro\bxjs@endpreamble@hook{%
4406   \ifx\bxjs@polyglossia@options\relax
4407     \csundef{ver@polyglossia.sty}%
4408   \fi}
```

`\bxjs@pandoc@polyglossia@ja` Pandoc 側で `lang=ja` が指定されていた場合の処理。この場合は `Polyglossia` の処理を無効化するためにダミーの定義を行う。その時点でダミーの `\setotherlanguage(s)` を定義する。

```
4409 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4410 \def\bxjs@pandoc@polyglossia@ja{%
4411   \renewcommand*\setmainlanguage[2] []{%
4412     \newcommand*\setotherlanguage[2] []{%
4413       \ifblank{##2}{\else
4414         \cslet{##2}\@empty \cslet{end##2}\@empty
4415         \cslet{text##2}\@firstofone}}%
4416     \newcommand*\setotherlanguages[2] []{%
4417       \@for\bxjs@tmpa:={##2}\do{%
4418         \setotherlangauge{\bxjs@tmpa}}}%
4419   }
```

`Polyglossia` の読込済マークは外れるようにしておく。

```
4419 \let\bxjs@polyglossia@options\relax}%
4420 \fi
```

■Babel について `XYTEX` 以外の場合。

※ Pandoc 2.6 版において、`LuaTeX` で用いられる多言語パッケージが `Polyglossia` から `Babel` に変更された。



```
4421 \if x\jsEngine\else
```

パッケージの読込を検知するため読込済のマークを付けて dupload の処理を仕込む。

```
4422 \pandocSkipLoadPackage{babel}
4423 \bxjs@set@dupload@proc{babel.sty}{%
4424   \bxjs@unset@dupload@proc{babel.sty}%
4425   \ClassWarning\bxjs@clsname
4426   {Package babel is required}}%
```

パッケージオプションに言語名が空の main= があるかを調べる。ある場合は lang=ja 対策を実行する。

※\bxjs@babel@options には main= を除いたオプション列を格納する。

```
4427 \@tempswafalse \let\bxjs@babel@options\@empty
4428 \def\bxjs@tmpb{main=}%
4429 \@for\bxjs@tmpa:=#1\do{%
4430   \ifx\bxjs@tmpa\bxjs@tmpb \@tempswatrue
4431   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4432   \fi}%
4433 \if@tempswa
4434   \ClassWarning\bxjs@clsname
4435   {Main language is invalid, so fallback\MessageBreak
4436     definitions will be employed}%
4437   \bxjs@pandoc@polyglossia@ja
```

ない場合は、本来の babel の処理を実行する。

```
4438 \else
4439   \ClassWarning\bxjs@clsname
4440   {Main language is valid,\MessageBreak
4441     thus babel will be loaded}%
4442   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4443   \RequirePackage[#1]{babel}[]%
4444 \fi}
```

プレアンブルで babel の読込が指示されなかった場合、読込済マークを外す。

```
4445 \g@addto@macro\bxjs@endpreamble@hook{%
4446   \ifx\bxjs@babel@options\relax
4447   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4448   \fi}
```

\bxjs@pandoc@babel@ja Pandoc 側で lang=ja が指定されていた場合の処理。

```
4449 \@onlypreamble\bxjs@pandoc@babel@ja
4450 \def\bxjs@pandoc@babel@ja{%
4451   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4452   \edef\bxjs@next{%
4453     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4454   }\bxjs@next
4455   \if j\jsEngine
4456     \RequirePackage[main=japanese]{pxbabel}[]%
4457   \else
4458     \RequirePackage{bxorigcapt}[]%
```

```
4459 \fi}
```

lang 対策はこれで終わり。

```
4460 \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。

```
4461 \bxjs@set@dupload@proc{geometry.sty}{%
```

```
4462 \setpagelayout*{#1}}
```

### E.4 CJKmainfont 変数

LuaTeX (+ LuaTeX-ja) の場合に CJKmainfont 変数が指定された場合は `\setmainfont` の指定にまわす。

```
4463 \if 1\jsEngine
```

```
4464 \pandocSkipLoadPackage{xeCJK}
```

```
4465 \providecommand*{\setCJKmainfont}{\setmainfont}
```

```
4466 \fi
```

ムニャムニャムニャ。

```
4467 \if x\jsEngine
```

```
4468 \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
```

```
4469 ,space}
```

```
4470 \fi
```

### E.5 paragraph のマーク

BXJS クラスでは `\paragraph` の見出しの前に `\jsParagraphMark` で指定したマークが付加され、既定ではこれは“■”である。しかし、この規定は `\paragraph` が本来のレイアウトを保っている、すなわち「行内見出しである」「節番号が付かない」ことが前提になっていると考えられる。Pandoc はこの規定を変更することがある（特に既定で `\paragraph` を別行見出しに再定義する）ため、変更された場合は `\jsParagraphMark` の既定値を空にする。

Pandoc がプレアンブルで行う再定義の結果を調べるため、begin-document フックを利用する。

```
4471 \g@addto@macro\bxjs@begin@document@hook{%
```

```
4472 \@tempswafalse
```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```
4473 \ifx\oldparagraph\undefined\else
```

```
4474 \@tempswattrue
```

```
4475 \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4476 \ifnum\c@secnumdepth>3
4477 \@tempwattrue
4478 \fi
```

「マーク変更が必要」である場合、`\jsParagraphMark` が既定値のままであれば空に変更する。

```
4479 \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4480 \let\jsParagraphMark\@empty
4481 \fi\fi}
```

## E.6 全角空白文字

```
4482 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4483 \catcode"3000=\active
4484 \begingroup \catcode`\!=7
4485 \protected\gdef!!!!3000{\zwspace}
4486 \endgroup
4487 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4488 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4489 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4490 \fi\fi
```

■hyperref 対策 hyperref の unicode オプションの固定を行う。

```
4491 \if j\jsEngine
4492 \bxjs@fix@hyperref@unicode{false}
4493 \else
4494 \bxjs@fix@hyperref@unicode{true}
4495 \fi
```

## E.7 完了

おしまい。

```
4496 %</pandoc>
```

和文ドライバ実装はここまで。

```
4497 %</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。

- `bxjscompat` : ムニャムニャムニャ。
- `bxjscjkat` : modern ドライバ用の和文カテゴリを適用する。
- `bxjspandoc` : Pandoc 用のナニカ。

```
4498 %<*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

```
4499 %<*compat>
4500 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4501 \let\bxac@engine=n
4502 \def\bxac@do#1#2{%
4503   \edef\bxac@tmpa{\string#1}%
4504   \edef\bxac@tmpb{\meaning#1}%
4505   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
4506 \bxac@do\XeTeXversion{\let\bxac@engine=x}
4507 \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。

4508 \ifx\jsAtEndOfClass\@undefined
4509   \let\bxac@delayed@if@bxjs\@firstofone
4510 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
4511 \fi

\ImposeOldLuaTeXBehavior ムニャムニャ。

\RevokeOldLuaTeXBehavior 4512 \newif\ifbxac@in@old@behavior
4513 \let\ImposeOldLuaTeXBehavior\relax
4514 \let\RevokeOldLuaTeXBehavior\relax
```

### G.2 XeTeX 部分

```
4515 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。

4516 \@onlypreamble\bxac@adjust@charclass
4517 \bxac@delayed@if@bxjs{%
4518   \@ifpackageloaded{xeCJK}{-}{%else
4519     \ifx\xe@alloc@intercharclass\@undefined\else
4520       \ifnum\xe@alloc@intercharclass=z@
4521         \PackageInfo\bxac@pkgname
4522           {Setting up interchar class for CJK...\@gobble}%
4523         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4524           \xe@alloc@intercharclass=3
4525         }{%else
4526           \PackageWarning\bxac@pkgname
4527             {Cannot find file 'load-unicode-xetex-classes.tex'%
```

```

4528         \@gobble}%
4529     }%
4530 \fi\fi
4531 \ifnum\XeTeXcharclass"3041=\z@
4532     \PackageInfo\bxac@pkgname
4533     {Adjusting interchar class for CJK...\@gobble}%
4534     \@for\bxac@x={%
4535         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4536         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4537         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4538         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4539         31FF%
4540     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4541 \fi
4542 }%
4543 }

```

以上。

```
4544 \fi
```

### G.3 LuaTeX 部分

```

4545 \ifx l\bxac@engine
    ムニヤムニヤ。
4546 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4547     \chardef\pdftexversion=200
4548     \def\pdfTeXrevision{0}
4549     \let\pdfTeXbanner\luatexbanner
4550 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior
4551 \begingroup\expandafter\expandafter\expandafter\endgroup
4552 \expandafter\ifx\cscname outputmode\endcsname\relax\else
4553 \def\bxac@ob@list{%
4554     \do{\let}\pdfoutput{\outputmode}%
4555     \do{\let}\pdfpagewidth{\pagewidth}%
4556     \do{\let}\pdfpageheight{\pageheight}%
4557     \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
4558     \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
4559 \def\bxac@ob@do#1#2{\begingroup
4560     \expandafter\bxac@ob@do@a\cscname bxac@\string#2\endcsname{#1}#2}
4561 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4562     \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4563     \else \let#3#1\relax \let#1\@undefined
4564     \fi}
4565 \protected\def\ImposeOldLuaTeXBehavior{%
4566     \unless\ifbxac@in@old@behavior
4567         \bxac@in@old@behaviortrue
4568         \let\do\bxac@ob@do \bxac@ob@list
4569     \fi}

```

```

4570 \protected\def\RevokeOldLuaTeXBehavior{%
4571   \ifbxac@in@old@behavior
4572     \bxac@in@old@behaviorfalse
4573     \let\do\bxac@ob@do \bxac@ob@list
4574   \fi}
4575 \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。

```

4576 \ifnum\luatexversion>64 \directlua{
4577   local function range(cs, ce, cc, ff)
4578     if ff or not tex.getcatcode(cs) == cc then
4579       local setcc = tex.setcatcode
4580       for c = cs, ce do setcc(c, cc) end
4581     end
4582   end
4583   range(0x3400, 0x4DB5, 11, false)
4584   range(0x4DB5, 0x4DBF, 11, true)
4585   range(0x4E00, 0x9FCC, 11, false)
4586   range(0x9FCD, 0x9FFF, 11, true)
4587   range(0xAC00, 0xD7A3, 11, false)
4588   range(0x20000, 0x2A6D6, 11, false)
4589   range(0x2A6D7, 0x2A6FF, 11, true)
4590   range(0x2A700, 0x2B734, 11, false)
4591   range(0x2B735, 0x2B73F, 11, true)
4592   range(0x2B740, 0x2B81D, 11, false)
4593   range(0x2B81E, 0x2B81F, 11, true)
4594   range(0x2B820, 0x2CEA1, 11, false)
4595   range(0x2CEA2, 0x2FFFD, 11, true)
4596 } \fi

```

以上。

```

4597 \fi

```

## G.4 完了

おしまい。

```

4598 %</compat>

```

## 付録 H 補助パッケージ : bxjscjkat

modern ドライバ用の和文カテゴリを適用する。

### H.1 準備

```

4599 %<*cjkcat>
4600 \def\bxjx@pkgname{bxjscjkat}
4601 \newcount\bxjx@cna

```

```

4602 \@onlypreamble\bxjx@tmpdo
4603 \@onlypreamble\bxjx@tmpdo@a
4604 \@onlypreamble\bxjx@tmpdo@b

```

`\bxjx@engine` エンジンの種別。

```

4605 \let\bxjx@engine=n
4606 \def\bxjx@tmpdo#1#2{%
4607   \edef\bxjx@tmpa{\string#1}%
4608   \edef\bxjx@tmpb{\meaning#1}%
4609   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4610 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4611 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4612 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4613 \bxjx@tmpdo\pdftexversion{\let\bxjx@engine=p}
4614 \bxjx@tmpdo\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。

```

4615 \def\bxjx@tmpdo#1#2{%
4616   \if#1\bxjx@engine
4617     \@ifpackageloaded{#2}{\}%else
4618     \PackageError\bxjx@pkgname
4619       {Package '#2' must be loaded}%
4620       {Package loading is aborted.\MessageBreak\@ehc}%
4621     \endinput}
4622   \fi}
4623 \bxjx@tmpdo{p}{bxcjkatype}
4624 \bxjx@tmpdo{x}{xeCJK}
4625 \bxjx@tmpdo{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。

```

4626 \ifx\TextOrMath\@undefined
4627   \RequirePackage{fixltx2e}
4628 \fi

```

## H.2 和文カテゴリコードの設定

upL<sup>A</sup>T<sub>E</sub>X の場合、和文カテゴリコードの設定を LuaT<sub>E</sub>X-ja と（ほぼ）等価なものに変更  
する。

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

```

4629 \if u\bxjx@engine
4630 \@for\bxjx@tmpa:={%
4631 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4632 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4633 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4634 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4635 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4636 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%

```

```

4637 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4638 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4639 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4640 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4641 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4642 FE70,FF00,%
4643 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4644 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4645 10480,104B0,10500,10530,10600,10800,10840,10860,%
4646 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4647 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4648 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4649 11200,11280,112B0,11300,11400,11480,11580,11600,%
4650 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4651 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4652 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4653 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4654 1EE00,1F000,1F030,1FOA0,1F300,1F600,1F650,1F680,%
4655 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4656 00C0%
4657 }\do{%
4658 \@tempcnta="\bxjx@tmpa\relax
4659 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4660 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4661 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4662 \fi

```

### H.3 ギリシャ・キリル文字の扱い

「特定 CJK 曖昧文字」について、和文・欧文扱いを制御できるようにする。ここで「特定 CJK 曖昧文字」とは以下に該当する文字の集合を指す：

- Unicode と JIS X 0213 に共通して含まれるギリシャ文字・キリル文字。
- Latin-1 の上位部分と JIS X 0208 に共通して含まれる文字（LuaTeX-ja の定める“範囲 8”）。

`\bxjx@grkcyr@list` 「特定 CJK 曖昧文字」に関する情報をもつ `\do`-リスト。各項目の形式は以下の通り：

`\do{⟨Unicode 符号値⟩}{⟨対象 fontenc⟩}{⟨テキスト LICR⟩}{⟨数式 LICR⟩}`

※数式で使わない文字は `⟨数式 LICR⟩` を空にする。

```

4663 \@onlypreamble\bxjx@grkcyr@list
4664 \def\bxjx@grkcyr@list{%
4665 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4666 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4667 \do{0393}{LGR}{\textGamma}{\Gamma}%      % GR. C. L. GAMMA
4668 \do{0394}{LGR}{\textDelta}{\Delta}%      % GR. C. L. DELTA
4669 \do{0395}{LGR}{\textEpsilon}{E}%         % GR. C. L. EPSILON
4670 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA

```



4671	\do{0397}{LGR}{\textEta}{H}%	% GR. C. L. ETA
4672	\do{0398}{LGR}{\textTheta}{\Theta}%	% GR. C. L. THETA
4673	\do{0399}{LGR}{\textIota}{I}%	% GR. C. L. IOTA
4674	\do{039A}{LGR}{\textKappa}{K}%	% GR. C. L. KAPPA
4675	\do{039B}{LGR}{\textLambda}{\Lambda}%	% GR. C. L. LAMDA
4676	\do{039C}{LGR}{\textMu}{M}%	% GR. C. L. MU
4677	\do{039D}{LGR}{\textNu}{N}%	% GR. C. L. NU
4678	\do{039E}{LGR}{\textXi}{\Xi}%	% GR. C. L. XI
4679	\do{039F}{LGR}{\textOmicron}{O}%	% GR. C. L. OMICRON
4680	\do{03A0}{LGR}{\textPi}{\Pi}%	% GR. C. L. PI
4681	\do{03A1}{LGR}{\textRho}{P}%	% GR. C. L. RHO
4682	\do{03A3}{LGR}{\textSigma}{\Sigma}%	% GR. C. L. SIGMA
4683	\do{03A4}{LGR}{\textTau}{T}%	% GR. C. L. TAU
4684	\do{03A5}{LGR}{\textUpsilon}{\Upsilon}%	% GR. C. L. UPSILON
4685	\do{03A6}{LGR}{\textPhi}{\Phi}%	% GR. C. L. PHI
4686	\do{03A7}{LGR}{\textChi}{X}%	% GR. C. L. CHI
4687	\do{03A8}{LGR}{\textPsi}{\Psi}%	% GR. C. L. PSI
4688	\do{03A9}{LGR}{\textOmega}{\Omega}%	% GR. C. L. OMEGA
4689	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
4690	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
4691	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
4692	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4693	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4694	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4695	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4696	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4697	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4698	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4699	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4700	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4701	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4702	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4703	\do{03BF}{LGR}{\textomicron}{o}%	% GR. S. L. OMICRON
4704	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4705	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4706	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4707	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4708	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4709	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4710	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4711	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4712	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4713	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4714	\do{0401}{T2A}{\CYRYO}{}	% CY. C. L. IO
4715	\do{0410}{T2A}{\CYRA}{}	% CY. C. L. A
4716	\do{0411}{T2A}{\CYRB}{}	% CY. C. L. BE
4717	\do{0412}{T2A}{\CYRV}{}	% CY. C. L. VE
4718	\do{0413}{T2A}{\CYRG}{}	% CY. C. L. GHE
4719	\do{0414}{T2A}{\CYRD}{}	% CY. C. L. DE

4720 \do{0415}{T2A}{\CYRE}{}%	% CY. C. L. IE
4721 \do{0416}{T2A}{\CYRZH}{}%	% CY. C. L. ZHE
4722 \do{0417}{T2A}{\CYRZ}{}%	% CY. C. L. ZE
4723 \do{0418}{T2A}{\CYRI}{}%	% CY. C. L. I
4724 \do{0419}{T2A}{\CYRISHRT}{}%	% CY. C. L. SHORT I
4725 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
4726 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
4727 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
4728 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
4729 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
4730 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
4731 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
4732 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
4733 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
4734 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
4735 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
4736 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
4737 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
4738 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
4739 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
4740 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
4741 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
4742 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
4743 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
4744 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
4745 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
4746 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
4747 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
4748 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
4749 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
4750 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
4751 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
4752 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
4753 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
4754 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
4755 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
4756 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
4757 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
4758 \do{043B}{T2A}{\cyrl}{}%	% CY. S. L. EL
4759 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
4760 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
4761 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
4762 \do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
4763 \do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
4764 \do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES
4765 \do{0442}{T2A}{\cyrt}{}%	% CY. S. L. TE
4766 \do{0443}{T2A}{\cyru}{}%	% CY. S. L. U
4767 \do{0444}{T2A}{\cyrf}{}%	% CY. S. L. EF
4768 \do{0445}{T2A}{\cyrh}{}%	% CY. S. L. HA

```

4769 \do{0446}{T2A}{\cyrcl}{}%           % CY. S. L. TSE
4770 \do{0447}{T2A}{\cyrch}{}%           % CY. S. L. CHE
4771 \do{0448}{T2A}{\cyrsh}{}%           % CY. S. L. SHA
4772 \do{0449}{T2A}{\cyrshch}{}%         % CY. S. L. SHCHA
4773 \do{044A}{T2A}{\cyrhrdsn}{}%         % CY. S. L. HARD SIGN
4774 \do{044B}{T2A}{\cyrery}{}%          % CY. S. L. YERU
4775 \do{044C}{T2A}{\cyrstsn}{}%         % CY. S. L. SOFT SIGN
4776 \do{044D}{T2A}{\cyrerev}{}%         % CY. S. L. E
4777 \do{044E}{T2A}{\cyrlyu}{}%          % CY. S. L. YU
4778 \do{044F}{T2A}{\cyrly}{}%           % CY. S. L. YA
4779 \do{0451}{T2A}{\cyrlyo}{}%          % CY. S. L. IO
4780 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4781 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4782 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4783 \do{00B1}{TS1}{\textpm}{\pm}%        % PLUS-MINUS SIGN
4784 \do{00B4}{TS1}{\textasciicircum}{}%  % ACUTE ACCENT
4785 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4786 \do{00D7}{TS1}{\texttimes}{\times}%  % MULTIPLICATION SIGN
4787 \do{00F7}{TS1}{\textdiv}{\div}%      % DIVISION SIGN
4788 }

```

`\mathdegree` 面倒なので補っておく。

```

4789 \providecommand*\mathdegree{\circ}

```

`\ifbxjx@gcc@CJK` [スイッチ]「特定 CJK 曖昧文字」を和文扱いにするか。

```

4790 \newif\ifbxjx@gcc@CJK

```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```

4791 \newcommand*\greekasCJK{%
4792   \bxjx@gcc@CJKtrue}

```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```

4793 \newcommand*\nogreekasCJK{%
4794   \bxjx@gcc@CJKfalse}

```

`\bxjx@fake@grk` `\bxjx@fake@grk{〈出力文字〉}{〈基準文字〉}` : ラテン文字で代用される数式ギリシャ文字の出力を行う。〈基準文字〉(`\mathchardef`の制御綴)の数式クラスと数式ファミリを引き継いで、〈出力文字〉(ASCII 文字トークン)の文字コードの数式文字を出力する。例えば、`\Pi`の意味が `\mathchar"7005` である場合、`\bxjx@fake@grk{B}{\Pi}` は `\mathchar"7042` を実行する。

※フォントパッケージ使用時の再定義を考慮して、〈基準文字〉が `\mathchardef` であるかを検査し、そうでない場合はフォールバックとして単に〈出力文字〉を実行する。

```

4795 \def\bxjx@tmpdo#1\relax{%
4796   \def\bxjx@fake@grk##1##2{%
4797     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4798   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4799     \ifx\##1\%
4800       \bxjx@canta##4\divide\bxjx@canta\@cclvi

```

```

4801 \multiply\bxjx@cmta\@cclvi \advance\bxjx@cmta`##3\relax
4802 \mathchar\bxjx@cmta
4803 \else ##3\fi}
4804 }\expandafter\bxjx@tmpdo\string\mathchar\relax

```

### ■pdfTeX・upTeX の場合

```

4805 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0

```

- `\[bxjx@KC/<符号値>]` : その文字が「特定曖昧 CJK 文字」に該当する場合に定義済になる。

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。

※「既定 UTF-8 化」後の L<sup>A</sup>T<sub>E</sub>X においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```

4806 \@ifpackageloaded{inputenc}{\fi}{%else
4807 \RequirePackage[utf8]{inputenc}}
4808 \def\bxjx@tmpa{utf8}
4809 \ifx\bxjx@tmpa\inputencdoingname
4810 \PackageWarningNoLine\bxjx@pkgname
4811 {Input encoding changed to utf8}%
4812 \inputencoding{utf8}%
4813 \fi

```

upTeX の場合に、「特定曖昧 CJK 文字」を含むブロックの和文カテゴリコードを変更する。

```

4814 \if u\bxjx@engine
4815 \kcatcode"0370=15
4816 \kcatcode"0400=15
4817 \kcatcode"0500=15
4818 \fi

```

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

```

4819 \def\bxjx@tmpdo#1{%
4820 \@tempcnta="#1\relax
4821 \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4822 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

引数 = `\[bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式 LICR>}`

“数式中の動作”を決定する。`<数式 LICR>` が空（数式非対応）なら警告を出す。

```

4823 \ifx\#5\%
4824 \def\bxjx@tmpa{\@inmathwarn#4}%

```

`<数式 LICR>` が英字である場合は `\bxjx@fake@grk` で出力する。大文字なら `\Pi`、小文字なら `\pi` を基準文字にする。

```

4825 \else\ifcat A\noexpand#5%
4826 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4827 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%

```

それ以外は `<数式 LICR>` をそのまま実行する。

```

4828 \else \def\bxjx@tmpa{#5}%
4829 \fi\fi
4830 \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4831 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

4832 \if u\bxjx@engine
4833 \def\bxjx@tmpdo@b#1#2#3#4#5{%

```

引数 =  $\backslash\text{[bxjx@KC/}\langle\text{符号値}\rangle\text{]}\{\langle\text{符号値}\rangle\}\{\langle\text{fontenc}\rangle\}\{\langle\text{LICR}\rangle\}\{\langle\text{数式中の動作}\rangle\}$

当該の Unicode 文字の動作は「テキストでは  $\langle\text{LICR}\rangle$ 、数式では  $\langle\text{数式中の動作}\rangle$ 」となる。LICR は現在エンコーディングで有効な定義がある場合はそれが実行されるはずである。（つまり、現在が LGR である場合はギリシャ文字は常に欧文扱いになる。）それ以外の場合は LICR を  $\backslash\text{bxjx@ja@or@not}$  に帰着させる。この際に、和文用の定義として当該の `kchardef` を使用し、その制御綴として  $\backslash\text{[bxjx@KC/}\dots\text{]}$  を流用している。

```

4834 \kchardef#1=\@tempcnta
4835 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4836 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdfTeX の場合も処理はほとんど同じ。ただし、和文用の定義として  $\backslash\text{UTF}\{\langle\text{符号値}\rangle\}$  を使う（ $\backslash\text{UTF}$  は `bxckjatype` の命令）。 $\backslash\text{[bxjx@KC/}\dots\text{]}$  は使わないが定義済にする必要がある。

```

4837 \else\if p\bxjx@engine
4838 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4839 \mathchardef#1=\@tempcnta
4840 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4841 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4842 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

4843 \let\do\bxjx@tmpdo \bxjx@grkcyrc@list

```

$\backslash\text{bxjx@DeclareUnicodeCharacter}$   $\backslash\text{bxjx@DeclareUnicodeCharacter}$  を改変して、「特定 CJK 曖昧文字」の場合に再定義を抑止したもの。

```

4844 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4845 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4846 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4847 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4848 \count@="#1\relax
4849 \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4850 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4851 \else
4852 \wlog{ \space\space skipped defining Unicode char U+#1}%
4853 \fi}

```

$\backslash\text{bxjx@ja@or@not}$   $\backslash\text{bxjx@ja@or@not}\{\langle\text{和文用定義}\rangle\}\{\langle\text{対象 fontenc}\rangle\}\{\langle\text{LICR}\rangle\}$  :  $\backslash\text{[no]greekasCJK}$  の状態に応じて和文または欧文で文字を出力する。

```

4854 \def\bxjx@ja@or@not#1#2#3{%

```

$\backslash\text{greekasCJK}$  の場合は、無条件に  $\langle\text{和文用定義}\rangle$  を実行する。

```
4855 \ifbxjx@gcc@CJK #1%
```

\nogreekasCJK の場合は、対象のエンコーディングに変更して LICR 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）和文用定義を使う。

```
4856 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
```

```
4857 \else \UseTextSymbol{#2}{#3}%
```

```
4858 \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。

```
4859 \begingroup
```

```
4860 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
```

```
4861 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
```

```
4862 \noexpand\bxjx@swap@DUC@cmd
```

```
4863 \the\toks@
```

```
4864 \noexpand\bxjx@swap@DUC@cmd}}}
```

```
4865 \endgroup\next
```

```
4866 \def\bxjx@swap@DUC@cmd{%
```

```
4867 \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```
4868 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
```

```
4869 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
```

```
4870 \let\bxjx@tmpa\relax}
```

以上。

## ■X<sub>Y</sub>TeX・LuaTeX の場合

```
4871 \else\ifnum0\if x\bxjx@engine1\fi\if l\bxjx@engine1\fi>0
```

各文字について、数式中の動作を定義する。

```
4872 \def\bxjx@tmpdo#1{%
```

```
4873 \bxjx@cmta="#1\relax
```

```
4874 \begingroup
```

```
4875 \lccode`~=\bxjx@cmta
```

```
4876 \lowercase{\endgroup
```

```
4877 \bxjx@tmpdo@a{~}}{#1}}
```

```
4878 \def\bxjx@tmpdo@a#1#2#3#4#5{%
```

〈数式 LICR〉が空なら何もしない。空でない場合、upL<sup>A</sup>T<sub>E</sub>X の場合と同じ方法で“数式中の動作”を決定し、当該の文字を math active にしてその動作を設定する。

```
4879 \ifx\\#5\\\let\bxjx@tmpa\relax
```

```
4880 \else\ifcat A\noexpand#5%
```

```
4881 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
```

```
4882 {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
```

```
4883 \else \def\bxjx@tmpa{#5}%
```

```
4884 \fi\fi
```

```
4885 \ifx\bxjx@tmpa\relax\else
```

```
4886 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
```

```
4887 \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、以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```
4888 \mathchardef\bxjx@tmpa="119
4889 \ifx\bxjx@tmpa\pi \let\do\bxjx@tmpdo \bxjx@grkcyr@list \fi
```

次に、テキストにおいて「特定 CJK 曖昧文字」の扱いが `\[no]greekasCJK` で切り替わるようにする。

Lua<sub>T</sub><sub>E</sub>X の場合は、Lua<sub>T</sub><sub>E</sub>X-j<sub>a</sub> の `jacharrange` の設定を変更する。

※ “範囲 2” がギリシャ・キリル文字、“範囲 8” が Latin-1 の記号。

```
4890 \if 1\bxjx@engine
4891   \protected\def\greekasCJK{%
4892     \bxjx@gcc@cjctrue
4893     \ltjsetparameter{jacharrange={+2, +8}}
4894   \protected\def\nogreekasCJK{%
4895     \bxjx@gcc@cjcfalse
4896     \ltjsetparameter{jacharrange={-2, -8}}
4897 \fi
```

X<sub>Ǝ</sub><sub>T</sub><sub>E</sub>X の場合、`xeCJK` は X<sub>Ǝ</sub><sub>T</sub><sub>E</sub>X の文字クラス定義を参照しているので、対象文字の文字クラスを変更する。

```
4898 \if x\bxjx@engine
4899   \let\bxjx@gcc@cjkl@list\@empty
4900   \def\do#1#2#3#4{%
4901     \edef\bxjx@gcc@cjkl@list{\bxjx@gcc@cjkl@list
4902       \noexpand\XeTeXcharclass"#1\bxjx@canta}}
4903   \bxjx@grkcyr@list
4904   \protected\def\greekasCJK{%
4905     \bxjx@gcc@cjctrue
4906     \bxjx@canta=\@ne \bxjx@gcc@cjkl@list}
4907   \protected\def\nogreekasCJK{%
4908     \bxjx@gcc@cjcfalse
4909     \bxjx@canta=\z@ \bxjx@gcc@cjkl@list}
4910 \fi
```

以上。

```
4911 \fi\fi
```

## H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
4912 \nogreekasCJK
```

## H.5 完了

おしまい。

```
4913 %</cjkat>
```

## 付録 I 補助パッケージ : bxjspan doc

Pandoc の L<sup>A</sup>T<sub>E</sub>X 用標準テンプレートをより幸せに使うための設定。BXJS クラスの pandoc ドライバのコードの中の、“汎用的”に使える部分を切り出したもの。つまり現在の pandoc ドライバはこのパッケージを読みこむ。

※テンプレートの T<sub>E</sub>X コードより前に読み込む必要があるため、専ら文書クラス内での読込に限られる。

### I.1 準備

```
4914 %<*ancpandoc>
4915 %% このファイルは日本語文字を含みます.
4916 \def\bxjsp@pkgname{bxjscjkat}

\bxjsp@engine エンジンの種別。

4917 \let\bxjsp@engine=n
4918 \@onlypreamble\bxjsp@do
4919 \def\bxjsp@do#1#2{%
4920   \edef\bxjsp@tmpa{\string#1}%
4921   \edef\bxjsp@tmpb{\meaning#1}%
4922   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
4923 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
4924 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
4925 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
4926 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

### I.2 パッケージオプション

`english` オプションが指定されている場合、`\ldots` の調整を抑止する。

※つまり、「グローバルの `english` オプション」が指定されている場合も抑止の対象になる。BXJS クラスの英語モードを想定しているが、それ以外の場合でも、一般的な L<sup>A</sup>T<sub>E</sub>X の習慣として、グローバルの `english` は「その文書の基底言語が英語である」ことを示す。

```
4927 \newif\ifbxjsp@english
4928 \DeclareOption{english}{\bxjsp@englishtrue}

オプション定義はおしまい。

4929 \ProcessOptions*
```

### I.3 パッケージ読込の阻止

`\pandocSkipLoadFile` `\pandocSkipLoadFile{〈ファイル名〉}` : 特定のファイルを (`\@filewithoptions` の処理に関して) 読込済であるとマークする。

```
4930 \@onlypreamble\pandocSkipLoadFile
4931 \newcommand*\pandocSkipLoadFile[1]{%
```



```

4932 \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4933 \def\bxjsp@skip@load@file@a#1#2{%
4934 \ifx#1\relax
4935 \def#1{2001/01/01}%
4936 \PackageInfo\bxjsp@pkgname
4937 {File '#2' marked as loaded\@gobble}%
4938 \fi}

```

`\pandocSkipLoadPackage` `\pandocSkipLoadPackage{<パッケージ名>}` : `\pandocSkipLoadFile` の機能を用いてパッケージの読込を阻止する。

```

4939 \@onlypreamble\pandocSkipLoadPackage
4940 \newcommand*\pandocSkipLoadPackage[1]{%
4941 \pandocSkipLoadFile{#1.sty}}

```

## 1.4 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。

```

4942 \ifx\@IncludeInRelease\@undefined\else
4943 \pandocSkipLoadPackage{fixltx2e}
4944 \fi

```

## 1.5 cmap パッケージ

エンジンが (u)p`LATEX` 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。）

```

4945 \if j\bxjsp@engine
4946 \pandocSkipLoadPackage{cmap}
4947 \fi

```

## 1.6 microtype パッケージ

警告が多すぎなので消す。

```

4948 \if j\bxjsp@engine \else
4949 \PassOptionsToPackage{verbose=silent}{microtype}
4950 \fi

```

エンジンが (u)p`LATEX` のときに `microtype` パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は `standard` ドライバでこの処理を行っていたが、元来は `Pandoc` 用の処理なので、1.5 版で `pandoc` に移動。

```

4951 \if j\bxjsp@engine
4952 \pandocSkipLoadPackage{microtype}
4953 \newcommand*\UseMicrotypeSet[2][]{\fi}

```

## I.7 Unicode 文字変換対策

Pandoc で  $\text{\LaTeX}$  形式に書き出す場合は、元データ中の一部の Unicode 文字を「 $\text{\LaTeX}$  の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

…→ $\text{\ldots}$ { } ‘→` ’→’ “→` ”→’

日本語  $\text{\LaTeX}$  では「 $\text{\LaTeX}$  の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「 $\text{\ldots}$ 」を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の  $\text{\ldots}$  の実装。非数式である場合は代わりに … を実行する。

※以前は「Pandoc が必ず  $\text{\ldots}$  の形で書き出す」ことを利用して後続に { } があるかで「元が … であるか」を判断していた。ところが、Pandoc 2.7 版で { } を必ずしも付けなくなったため、1.9f 版で非数式の  $\text{\ldots}$  を全て … に戻す動作に変更した。

```
4955 \DeclareRobustCommand{\pandocLdots}{%
4956   \relax\ifmmode \expandafter\bxjsp@org@ldots
4957   \else \expandafter\bxjsp@ja@ellipsis
4958   \fi}
4959 \def\bxjsp@ja@ellipsis{…}
4960 \let\bxjsp@org@ldots\ldots
```

$\text{\ldots}$  の実装を置き換える。

※ Benglish オプション指定時は置き換えない。

```
4961 \AtBeginDocument{%
4962   \let\bxjsp@org@ldots\ldots
4963   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}
```

$\text{\ldots}$  の直後の文字が非英字の場合、Pandoc は「 $\text{\ldots}$ 。」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが  $\text{\XeTeX}$ ・ $\text{\LuaTeX}$  は英字と見なす（または将来その可能性がある）」文字で、特に日本語文書に現れるものについて、非英字扱いにしておく。

※ Pandoc は「Unicode 7.0 で GC が Letter」な文字を英字と判定している。

```
4964 \chardef\bxjsp@cc@other=12
4965 \@onlypreamble\bxjsp@makeother@range
4966 \def\bxjsp@makeother@range#1#2{%
4967   \@tempcnta"#1\relax \@tempcntb"#2\relax
4968   \loop\ifnum\@tempcnta<\@tempcntb
4969     \catcode\@tempcnta\bxjsp@cc@other
4970     \advance\@tempcnta\@ne
4971   \repeat}
4972 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
```

```

4973 \catcode"1F23B=\bxjsp@cc@other
4974 \bxjsp@makeother@range{9FCD}{A000}
4975 \bxjsp@makeother@range{1B002}{1B170}
4976 \bxjsp@makeother@range{2B820}{2EBF0}
4977 \fi

```

## I.8 PandoLa モジュール

インストール済であれば読み込む。

```

4978 \IfFileExists{bxpandola.sty}{%
4979 \RequirePackage{bxpandola}\relax
4980 \PackageInfo{bxjsp@pkgname
4981 {PandoLa module is loaded\@gobble}
4982 }{}

```

## I.9 完了

おしまい。

```

4983 %</ancpandoc>

```

補助パッケージ実装はここまで。

```

4984 %</anc>

```