

# 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”)

v2.0 [2020/03/25]

---





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 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  $\mathrm{p}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}\text{-ng}$  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}}
```

`\jsSetQHLlength` `\jsSetQHLlength\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\jsSetQHLlength#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

```

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

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

---

BXJS クラスでは、nojslogo を既定とする。

---

```
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{pdflatex}{%
381   \def\bxjs@engine@opt{pdflatex}%
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}

```

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

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

```

425 \DeclareOption{dviPDFmx-if-dvi}{%
426   \setkeys{bxjs}{dvi=dviPDFmx}}

```

---

## ■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。

```
427 \newif\ifbxjs@bigcode \bxjs@bigcodetrue
```

`nobigcode` / `bigcode` オプションの定義。

```
428 \DeclareOption{nobigcode}{%
429   \bxjs@bigcodefalse}
430 \DeclareOption{bigcode}{%
431   \bxjs@bigcodetrue}
```

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

```
432 \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。

```
433 \DeclareOption{nooldfontcommands}{%
434   \bxjs@oldfontcommandsfalse}
435 \DeclareOption{oldfontcommands}{%
436   \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。

```
437 \DeclareOption*{%
438   \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
439   \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。

```
440 \def\bxjs@safe@setkeys#1#2{%
441   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
442   \setkeys{#1}{#2}%
443   \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。

```
444 \@onlypreamble\bxjs@declare@enum@option
445 \def\bxjs@declare@enum@option#1#2{%
446   \define@key{bxjs}{#1}{%
447     \expandafter\ifx\csname bxjs@#2@@##1\endcsname\relax
448     \bxjs@error@keyval{#1}{##1}%
449   \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@@##1}%
450   \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。

```
451 \@onlypreamble\bxjs@declare@bool@option
452 \def\bxjs@declare@bool@option#1#2{%
453   \define@key{bxjs}{#1}[true]{%
454     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
455     \bxjs@error@keyval{#1}{##1}%
456   \else \@nameuse{bxjs@#2##1}%
457   \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。

```
458 \def\bxjs@set@keyval#1#2#3{%
459   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
460   \ifx\bxjs@next\relax
461     \bxjs@error@keyval{#1}{#2}%
462     #3%
463   \else \bxjs@next
464   \fi}
465 \@onlypreamble\bxjs@error@keyval
466 \def\bxjs@error@keyval#1#2{%
467   \ClassError\bxjs@clsname
468   {Invalid value '#2' for option #1}\@ehc}
```

`\jsScale` 〔実数値マクロ〕 和文スケール値。

```
469 \def\jsScale{0.924715}
```

`\bxjs@base@opt` 明示された base オプションの値。

```
470 %\let\bxjs@base@opt\@undefined
```

base オプションの処理。

```
471 \define@key{bxjs}{base}{%
472   \edef\bxjs@base@opt{#1}%
473   \bxjs@setbasefontsize{#1}}
474 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}
```

`\bxjs@jbase@opt` 明示された jbase オプションの値。

```
475 %\let\bxjs@jbase@opt\@undefined
```

jbase オプションの処理。

```

476 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
477 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}

```

\bxjs@scale@opt 明示された scale オプションの値。

```

478 %\let\bxjs@scale@opt\@undefined

```

scale オプションの処理。

```

479 \define@key{bxjs}{scale}{%
480   \edef\bxjs@scale@opt{#1}%
481   \let\jsScale\bxjs@scale@opt}
482 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

```

noscale オプションの処理。

```

483 \DeclareOption{noscale}{\setkeys{bxjs}{scale=1}}

```

\bxjs@param@mag mag オプションの値。

```

484 \let\bxjs@param@mag\relax

```

mag オプションの処理。

```

485 \define@key{bxjs}{mag}{\edef\bxjs@param@mag{#1}}

```

paper オプションの処理。

```

486 \define@key{bxjs}{paper}{\edef\bxjs@param@paper{#1}}

```

\bxjs@jadriver 和文ドライバの名前。

```

487 \let\bxjs@jadriver\relax
488 %\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）。

```

489 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver@opt{#1}}
490 \define@key{bxjs}{ja}[\relax]{%
491   \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}

```

\jsJaFont 和文フォント設定の名前。

```

492 \let\jsJaFont\@empty

```

jafont オプションの処理。

```

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

```

\jsJaParam 和文ドライバパラメタの文字列。

```

494 \let\jsJaParam\@empty

```

japaram オプションの処理。

```

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

```

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

```
496 \let\bxjs@magstyle@mag=m
497 \let\bxjs@magstyle@real=r
498 \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。

```
499 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
500 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
501 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

`\bxjs@magstyle@default` は既定の値を表す。

```
502 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
503 \ifx \l\jsEngine \ifnum\luatexversion>86
504   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
505 \fi\fi
506 \ifjsWithpTeXng
507   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
508 \fi
509 \let\bxjs@magstyle\bxjs@magstyle@default
```

magstyle オプションの処理。

```
510 \define@key{bxjs}{magstyle}{%
511   \expandafter\let\expandafter\bxjs@magstyle\csname
512     bxjs@magstyle@#1\endcsname
513   \ifx\bxjs@magstyle\relax
514     \ClassError\bxjs{clsname
515       {Invalid value '#1' for option magstyle}}\@ehc
516     \let\bxjs@magstyle\bxjs@magstyle@default
517   \fi}
```

`\bxjs@geometry` geometry オプションの値。

```
518 \let\bxjs@geometry@class=c
519 \let\bxjs@geometry@user=u
520 \let\bxjs@geometry\bxjs@geometry@class
```

geometry オプションの処理。

```
521 \define@key{bxjs}{geometry}{%
522   \expandafter\let\expandafter\bxjs@geometry\csname
523     bxjs@geometry@#1\endcsname
524   \ifx\bxjs@geometry\relax
525     \ClassError\bxjs{clsname
526       {Invalid value '#1' for option geometry}}\@ehc
527     \let\bxjs@geometry\bxjs@geometry@class
528   \fi}
```

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

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



fancyhdr オプションの処理。

```
530 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
531 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
532 \define@key{bxjs}{fancyhdr}[true]{%
533   \bxjs@set@keyval{fancyhdr}{#1}{}}
```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```
534 \newif\ifbxjs@dvi@opt
```

DVI モードのドライバとドライバ種別との対応。

```
535 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
536 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
537 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
538 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
539 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none
```

dvi オプションの処理。

```
540 \define@key{bxjs}{dvi}{%
541   \expandafter\let\expandafter\bxjs@tmpa\csname
542     bxjs@dvidriver@@#1\endcsname
543   \ifx\bxjs@tmpa\relax
544     \ClassError\bxjs{clsname
545       {Invalid value '#1' for option dvi}}\@ehc
546   \else
```

\bxjs@driver@given を未定義にしていることに注意。

```
547   \def\bxjs@driver@opt{#1}%
548   \let\bxjs@driver@given\undefined
549   \bxjs@dvi@opttrue
550 \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。

```
551 \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。

```
552 \newif\ifbxjs@force@chapterabstract
553 %<book>\bxjs@force@chapterabstracttrue
```

layout オプションの処理。

```
554 \@namedef{bxjs@kv@layout@v1}{%
555 %<book>\bxjs@layout@buggyhmargintrue
556 %<report>\bxjs@force@chapterabstracttrue
557 }
558 \@namedef{bxjs@kv@layout@v2}{%
559 %<book>\bxjs@layout@buggyhmarginfalse
560 %<report>\bxjs@force@chapterabstractfalse
561 }
```

```

562 \define@key{bxjs}{layout}{%
563   \bxjs@set@keyval{layout}{#1}{}}

```

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

```

564 %\let\bxjs@textwidth@limit@opt\@undefined
565 \define@key{bxjs}{textwidth-limit}{%
566   \edef\bxjs@textwidth@limit@opt{#1}}

```

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

```

567 %\let\bxjs@textwidth@opt\@undefined
568 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
569 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

```

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

```

570 %\let\bxjs@number@of@lines@opt\@undefined
571 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
572 \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。

```

573 %\let\bxjs@paragraph@mark\@undefined
574 \define@key{bxjs}{paragraph-mark}{%
575   \edef\bxjs@paragraph@mark{#1}}

```

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

```

576 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
577 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
578 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
579 \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。

```

580 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
581 \let\bxjs@kv@jaspacecmd@true\bxjs@jaspace@cmdtrue
582 \let\bxjs@kv@jaspacecmd@false\bxjs@jaspace@cmdfalse
583 \define@key{bxjs}{jaspace-cmd}[true]{\bxjs@set@keyval{jaspacecmd}{#1}{}}
584 \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。

```

585 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
586 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
587 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
588 \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。

```

589 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
590 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
591 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
592 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}

```

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

```

593 \chardef\bxjs@everyparhook@none=0
594 \chardef\bxjs@everyparhook@compat=1
595 \chardef\bxjs@everyparhook@modern=2
596 \if j\jsEngine
597   \let\bxjs@everyparhook\bxjs@everyparhook@compat
598 \else
599   \let\bxjs@everyparhook\bxjs@everyparhook@modern
600 \fi
601 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
602 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
603 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
604 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}

```

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

```

605 \chardef\bxjs@label@section@none=0
606 \chardef\bxjs@label@section@compat=1
607 \chardef\bxjs@label@section@modern=2
608 \let\bxjs@label@section\bxjs@label@section@compat
609 \def\bxjs@kv@labelsection@none{\let\bxjs@label@section\bxjs@label@section@none}
610 \def\bxjs@kv@labelsection@compat{\let\bxjs@label@section\bxjs@label@section@compat}
611 \def\bxjs@kv@labelsection@modern{\let\bxjs@label@section\bxjs@label@section@modern}
612 \define@key{bxjs}{label-section}{\bxjs@set@keyval{labelsection}{#1}{}}

```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```

613 \newif\ifbxjs@usezw \bxjs@usezwtrue
614 \bxjs@declare@bool@option{use-zw}{usezw}
615 \DeclareOption{noz}{\setkeys{bxjs}{use-zw=false}}
616 \DeclareOption{zw}{\setkeys{bxjs}{use-zw=true}}

```

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

```

617 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
618 \bxjs@declare@bool@option{disguise-js}{disguise@js}
619 \DeclareOption{nojs}{\setkeys{bxjs}{disguise-js=false}}
620 \DeclareOption{js}{\setkeys{bxjs}{disguise-js=true}}

```

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

```

621 \newif\ifbxjs@precisetext
622 \bxjs@declare@bool@option{precise-text}{precisetext}
623 \DeclareOption{noprecisetext}{\setkeys{bxjs}{precise-text=false}}
624 \DeclareOption{precisetext}{\setkeys{bxjs}{precise-text=true}}

```

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

```

625 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
626 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}
627 \DeclareOption{nosimplejasetup}{\setkeys{bxjs}{simple-ja-setup=false}}
628 \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。

```
629 \let\bxjs@org@removeelement\@removeelement
630 \def\@removeelement#1#2#3{%
631   \def\reserved@a{#2}%
632   \ifx\reserved@a\@empty \let#3\@empty
633   \else \bxjs@org@removeelement{#1}{#2}{#3}%
634   \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。

```
635 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
636 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
637 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
638 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
639 \ProcessOptions\relax
640 \bxjs@post@option@hook
```

後処理

```
641 \if@slide
642   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
643 \fi
644 \if@landscape
645   \setlength\@tempdima {\paperheight}
646   \setlength\paperheight{\paperwidth}
647   \setlength\paperwidth {\@tempdima}
648 \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。

```
649 \def\bxjs@tmpdo{%
650   \def\bxjs@tmpa{\gobble}%
651   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
652   \let\@classoptionslist\bxjs@tmpa}
653 \def\bxjs@tmpdo@a#1,{%
654   \ifx\@nil#1\relax\else
655     \bxjs@tmpdo@b#1{\@nil
```

```

656 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
657 \expandafter\bxjs@tmpdo@a
658 \fi}
659 \def\bxjs@tmpdo@b#1#{\bxjs@tmpdo@c}
660 \def\bxjs@tmpdo@c#1\@nil{%
661 \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
662 \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。

```

663 \@expandtwoargs\@removeelement
664 {papersize}\@classoptionslist\@classoptionslist
665 \@expandtwoargs\@removeelement
666 {10pt}\@classoptionslist\@classoptionslist
667 \@expandtwoargs\@removeelement
668 {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。

```

669 \ifx\bxjs@jadriver@opt\@undefined\else
670 \let\bxjs@jadriver\bxjs@jadriver@opt
671 \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。

```

672 \let\bxjs@tmpb\jsEngine
673 \ifx j\bxjs@tmpb\ifjsWithpTeXng
674 \let\bxjs@tmpb=g
675 \fi\fi
676 \ifx j\bxjs@tmpb\ifjsWithupTeX
677 \let\bxjs@tmpb=u
678 \fi\fi
679 \ifx p\bxjs@tmpb\ifjsInPdfMode\else

```

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

```
681 \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。)

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

```
683 \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。

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

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

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

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

```
688 \fi\fi
```

```
689 \fi
```

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

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

```
692 \ClassError\bxjs@clsname
```

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

```
694 \fi
```


```
695 \fi
```

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

```
696 \ifjsWithpTeXng
```

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

```
698 \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。

```
699 \@tempswatrue
```

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

```
701 \ifjsInPdfMode
```

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

```
703 \@tempswafalse
```

```
704 \fi
```

```
705 \else\ifx x\jsEngine
```

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

```
707 \@tempswafalse
```

```
708 \fi
```

```
709 \else
```

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

```
711 \@tempswafalse
```

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

```
713 \@tempswafalse
```

```
714 \fi\fi
```

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

```
716 \@tempswafalse
```

```
717 \fi\fi
```

```

718 \fi\fi
719 \fi
720 \if@tempswa\else
721 \ClassError\bxjs@clsname
722 {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
723 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

724 \ifjsInPdfMode \@tempswafalse
725 \else\ifx x\jsEngine \@tempswafalse
726 \else\ifjsWithpTeXng \@tempswafalse
727 \else \@tempwatrue
728 \fi\fi\fi
729 \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)。

```

730 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@opt\@undefined\else
731 \ClassWarningNoLine\bxjs@clsname
732 {No driver option is given}
733 \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。)

```

734 \ifbxjs@dvi@opt
735 \edef\bxjs@next{%
736 \let\noexpand\bxjs@driver@given
737 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
738 \noexpand\g@addto@macro\noexpand\@classoptionslist
739 {,\bxjs@driver@opt}%
740 }\bxjs@next
741 \fi
742 \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。

```

743 \ifjsWithpTeXng
744 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
745 \let\bxjs@platexng@nodrv\@undefined
746 \else\ifx t\bxjs@platexng@nodrv\else
747 \g@addto@macro\@classoptionslist{,dvipdfmx}
748 \fi\fi
749 \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。

```

750 \ifx\bxjs@driver@given\bxjs@driver@@none
751 \bxjs@papersizefalse
752 \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。

```

753 \def\bxjs@@minimal{minimal}
754 \ifx\bxjs@jadriver\relax
755   \ifx j\jsEngine
756     \def\bxjs@jadriver{standard}
757   \else
758     \let\bxjs@jadriver\bxjs@@minimal
759   \fi
760 \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。

```

761 \ifx\bxjs@jadriver@opt\@undefined\else
762   \ifx\bxjs@engine@given\@undefined
763     \ClassError\bxjs@clsname
764     {An engine option must be explicitly given}%
765     {When you use a Japanese-driver you must specify a correct\MessageBreak
766       engine option.\MessageBreak\@ehc}
767 \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。)

```

768 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
769   \ifx\bxjs@magstyle\bxjs@magstyle@mag
770     \let\bxjs@magstyle\bxjs@magstyle@default
771     \ClassError\bxjs@clsname
772     {The engine does not support 'magstyle=usemag'}%
773     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
774       The default value 'nomag*' is used instead.\MessageBreak \@ehc}
775   \fi
776 \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。

```

777 \ifx\bxjs@jbase@opt\@undefined\else
778   \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。

```

779     \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
780     \bxjs@invscale\@tempdima\jsScale
781     \bxjs@setbasefontsize{\@tempdima}%
782   \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。



```

783 \ifx\bxjs@scale@opt\@undefined\else
784 \ClassWarningNoLine\bxjs@clsname
785 {Redundant 'scale' option is ignored}%
786 \fi
787 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
788 \@tempdimb=\bxjs@param@basefontsize\relax
789 \edef\jsScale{\strip@pt\@tempdimb}%
790 \bxjs@invscale\@tempdima\jsScale
791 \edef\jsScale{\strip@pt\@tempdima}%
792 \fi
793 \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。

```

794 \let\Cjascale\jsScale

```

---



---

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

```

795 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
796 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
797 \catcode\@tempcnta\active
798 \advance\@tempcnta\@ne
799 \repeat
800 \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。

```

801 \ifbxjs@disguise@js
802 %<book|report>\def\bxjs@js@clsname{jsbook}
803 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
804 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
805 \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。

```

806 \g@addto@macro\@classoptionslist{,nosetpagesize}

```

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

```

807 \ifbxjs@oldfontcommands
808 \AtEndOfClass{\allowoldfontcommands}
809 \fi

```

---



---

■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 p<sub>La</sub>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。

---

```
810 \newdimen\stockwidth \newdimen\stockheight
811 \begingroup\expandafter\expandafter\expandafter\endgroup
812 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
813 % \newdimen\stockwidth \newdimen\stockheight
814 \setlength{\stockwidth}{\paperwidth}
815 \setlength{\stockheight}{\paperheight}
816 \advance \stockwidth 2in
817 \advance \stockheight 2in
818 \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。

```
819 %<slide>\def\n@baseline{13}%
820 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
821 %<!slide>\else \def\n@baseline{16}\fi
```

#### ■拡大率の設定

---

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

```
822 \ifx\bxjs@magstyle\bxjs@magstyle@mag
823 \jsc@magtrue
824 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
825 \jsc@mag@xrealtrue
826 \fi\fi
```

---

サイズの変更は T<sub>E</sub>X のプリミティブ `\mag` を使って行います。9 ポイントについては行送りも若干縮めました。サイズについては全面的に見直しました。

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

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

---

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

```
827 \ifx\bxjs@param@mag\relax
828   \@tempdima=\bxjs@param@basefontsize
829   \advance\@tempdima.001pt \multiply\@tempdima25
830   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
831   \edef\bxjs@param@mag{\the\@tempcnta}
832 \else
833 % mag 値が直接指定された場合
834 \bxjs@gset@tempcnta{\bxjs@param@mag}
835 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
836 % 有効な mag 値の範囲は 1--32768
837 \edef\bxjs@param@mag{\the\@tempcnta}
838 \advance\@tempcnta100000
839 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
840 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
841 \edef\bxjs@param@basefontsize{\the\@tempdima}
842 \fi
843 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
844 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
845 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
846 \edef\jsc@magscale{\strip@pt\@tempdima}
847 \let\jsBaseFontSize\bxjs@param@basefontsize
```

---

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

---

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

```
848 \newdimen\jsc@mpt
849 \newdimen\jsc@mmm
850 \ifjsc@mag
851   \jsc@mpt=1\p@
```

```

852 \jsc@mmm=1mm
853 \catcode\?=9 % \p@? read as \p@
854 \else
855 \jsc@mpt=\jsc@magscale\p@
856 \jsc@mmm=\jsc@magscale mm
857 \catcode\?=11 \let\p@?\jsc@mpt
858 \fi
859 \chardef\bxjs@qmc=\catcode\?\relax
860 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode\?=12\relax}

```

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

noz<sub>w</sub> 非指定時は \zw を \jsZw と同義にする。

```

861 \newdimen\jsZw
862 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
863 \ifbxjs@usezw
864 \providecommand*\zw{\jsZw}
865 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

867 \ifjsc@mag@xreal
868 \RequirePackage{type1cm}
869 \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

---

```

870 \ifbxjs@TUenc
871 \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
872 \else
873 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
874 \fi
875 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
876 \let\jsc@get@external@font\get@external@font
877 \def\get@external@font{%
878 \jsc@preadjust@extract@font
879 \jsc@get@external@font}
880 \def\jsc@fstrunc#1{%
881 \edef\jsc@tmpa{\strip@pt#1}%
882 \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
883 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
884 \if#5*\else
885 \edef\jsc@tmpa{#1%
886 \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
887 \fi}
888 \def\jsc@preadjust@extract@font{%
889 \let\jsc@req@size\f@size

```

```

890 \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
891 \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
892 \let\jsc@ref@size\jsc@tmpa
893 \let\f@size\jsc@ref@size}
894 \def\execute@size@function#1{%
895 \let\jsc@cref@size\f@size
896 \let\f@size\jsc@req@size
897 \csname s@fct@#1\endcsname}
898 \let\jsc@DeclareErrorFont\DeclareErrorFont
899 \def\DeclareErrorFont#1#2#3#4#5{%
900 \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
901 \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
902 \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
903 \def\gen@sfcnt{%
904 \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
905 \empty@sfcnt}
906 \def\genb@sfcnt{%
907 \edef\mandatory@arg{%
908 \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@@}%
909 \empty@sfcnt}
910 \ifbxjs@TUenc\else
911 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
912 \fi
913 \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 914 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 915 %\def\jsc@medskip{\vspace\jsc@medskipamount}
916 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 917 \newskip\jsc@smallskipamount
918 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
\jsc@bigskipamount 919 %\newskip\jsc@medskipamount
920 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
921 %\newskip\jsc@bigskipamount
922 %\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] トンボオブションが指定されているとき「だけ」`\stockwidth`, `\stockheight` が定義されています。

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

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

```
923 % \ifpapersize
924 %   \setlength{\@tempdima}{\paperwidth}
925 %   \setlength{\@tempdimb}{\paperheight}
926 %   \iftombow
927 %     \advance \@tempdima 2truein
928 %     \advance \@tempdimb 2truein
929 %   \fi
930 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
931 % \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` とピリオドのペアカーニングが効かなくなります。そこで、次のような定義に直し、`I watch TV.\@` と書くことにします。

[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。

---

```
932 \chardef\bxjs@periodchar=`\ .
933 \bxjs@protected\def\bxjs@SE{%
934   \ifnum\spacefactor<\@m \spacefactor\@m
935   \else \spacefactor\sfcode\bxjs@periodchar
936   \fi}
937 \ifbxjs@fix@at@cmd
938   \def\@{\bxjs@SE{}}
939 \fi
```

## 4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

---

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

```
940 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
941 \expandafter\bxjs@tmpa\expandafter{%
942   \set@fontsize{#1}{#2}{#3}%
943 % 末尾にコードを追加
944   \expandafter\def\expandafter\size@update\expandafter{%
945     \size@update
946     \jsFontSizeChanged}%
947 }
```

`\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。

```
948 \newcommand*\jsFontSizeChanged{%
949   \jsZw=\f@size\p@
950   \jsZw=\jsScale \jsZw
951   \ifdim\parindent>\z@
952     \if@english \parindent=1em
953     \else       \parindent=1\jsZw
954   \fi
955   \fi\relax
956   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
957 \newcommand*\jsResetDimen{}
```

---

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

```
958 \ifjsc@mag
959   \let\jsc@setfontsize\@setfontsize
960 \else
961   \def\jsc@setfontsize#1#2#3{%
962     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
963 % microtype 対策
964 \ifjsWithTeX\if j\jsEngine\else
965   \def\jsc@setfontsize#1#2#3{%
966     \edef\bxjs@sfs@next{%
967       \unexpanded{\@setfontsize#1}%
968       {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
969     }\bxjs@sfs@next}
970 \fi\fi
971 \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。

---

```
972 \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

`\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、しっぱ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。

---

```
973 \newif\ifnarrowbaselines
974 \if@english
975   \narrowbaselinestrue
976 \fi
977 \def\narrowbaselines{%
978   \narrowbaselinestrue
979   \skip0=\abovedisplayskip
980   \skip2=\abovedisplayshortskip
981   \skip4=\belowdisplayskip
982   \skip6=\belowdisplayshortskip
983 % 一時的に警告を無効化する
984   \let\bxjs@save@nomath\@nomath
985   \let\@nomath\@gobble
986   \@currsize\selectfont
987   \let\@nomath\bxjs@save@nomath
988   \abovedisplayskip=\skip0
989   \abovedisplayshortskip=\skip2
990   \belowdisplayskip=\skip4
991   \belowdisplayshortskip=\skip6\relax}
992 \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。

```
993 \def\bxjs@if@narrowbaselines{%
994   \ifnarrowbaselines\expandafter\@firstoftwo
995   \else \expandafter\@secondoftwo
996   \fi
997 }
```

---

`\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。

---

```

998 \renewcommand{\normalsize}{%
999   \bxjs@if@narrowbaselines{%
1000     \jsc@setfontsize\normalsize\@xpt\@xipt
1001   }{%else
1002     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1003   }%

```

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] `TEX 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。

```

1004 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1005 \abovedisplayshortskip \z@ \@plus3\p@?
1006 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1007 \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。

```

1008 \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。

```

1009 \normalsize

```

`\Cht` 基準となる長さの設定をします。p $\text{\LaTeX}$  2 <sub>$\epsilon$</sub>  カーネル (`plfonts.dtx`) で宣言されているパラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (1zw) です。

`\Cwd` [2017-08-31] 基準とする文字を「全角空白」(EUC コード 0xA1A1) から「漢」(JIS コード 0x3441) へ変更しました。

---

`\Chs`

`\Cwd` 等の変数は p $\text{\TeX}$  系以外では未定義なのでここで定義する。

```

1010 \ifx\Cht\@undefined \newdimen\Cht \fi
1011 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1012 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1013 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1014 \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。

---

```

1015 \setlength\Cht{0.88\jsZw}
1016 \setlength\Cdp{0.12\jsZw}

```

---

```

1017 \setlength\Cwd{1\jsZw}
1018 \setlength\Cvs{\baselineskip}
1019 \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。

---

```

1020 \newcommand{\small}{%
1021   \bxjs@if@narrowbaselines{%
1022     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1023     %<kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1024   }{%else
1025     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1026     %<kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1027   }%
1028   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1029   \abovedisplayshortskip \z@ \@plus3\p@?
1030   \belowdisplayskip \abovedisplayskip
1031   \belowdisplayshortskip \belowdisplayskip
1032   \def\@listi{\leftmargin\leftmargini
1033             \topsep \z@
1034             \parsep \z@
1035             \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。

```

1036 \newcommand{\footnotesize}{%
1037   \bxjs@if@narrowbaselines{%
1038     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1039     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1040   }{%else
1041     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1042     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1043   }%
1044   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1045   \abovedisplayshortskip \z@ \@plus2\p@?
1046   \belowdisplayskip \abovedisplayskip
1047   \belowdisplayshortskip \belowdisplayskip
1048   \def\@listi{\leftmargin\leftmargini
1049             \topsep \z@
1050             \parsep \z@
1051             \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

```

\tiny
\large
\Large
\LARGE
\huge

```

ます。特に注意すべきは `\large` で、これは二段組のときに節見出しのフォントとして使い、行送りを `\normalsize` と同じにすることによって、節見出しが複数行にわたっても段間で行が揃うようにします。

[2004-11-03] `\HUGE` を追加。

```

1052 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1053 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
1054 \if@twocolumn
1055 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
1056 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1057 \else
1058 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
1059 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1060 \fi
1061 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1062 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1063 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1064 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1065 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1066 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

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

```

1067 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}

```

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

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

```

1068 % \newcommand{\headfont}{\bfseries}
1069 \newcommand{\headfont}{\sffamily}
1070 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

## 5 レイアウト

### ■二段組

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

```

1071 %<kiyou>\setlength\columnsep{2\Cwd}
1072 %<kiyou>\setlength\columnsep{28truebp}
1073 \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ます。元は 0pt ですが 1pt に変更しました。normal... の付いた方は保存用です。

```

\lineskiplimit 1074 \setlength\lineskip{1\jsc@empt}
1075 \setlength\normallineskip{1\jsc@empt}
\normallineskiplimit 1076 \setlength\lineskiplimit{1\jsc@empt}
1077 \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、たとえば `\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。

```
1078 \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。

```

1079 \setlength\parskip{\z@}
1080 \if@slide
1081   \setlength\parindent{0\p@}
1082 \else
1083   \setlength\parindent{1\Cwd}
1084 \fi

```

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

```

\n@highpenalty 1085 \@lowpenalty 51
1086 \@medpenalty 151
1087 \@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。

```
1088 % \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。

```
1089 % \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。

```
1090 \onlypreamble\bxjs@bd@pre@geometry@hook
1091 \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。

```
1092 \ifjsc@mag
1093 \mag=\bxjs@param@mag
1094 \fi
1095 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLLength` のための和文単位の定義。

```
1096 \def\bxjs@unit@trueQ{0.25trueemm}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1097 \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。

```
1098 \def\bxjs@tmpdo{\futurelet\bxjs@tmpa\bxjs@tmpdo@a}
1099 \def\bxjs@tmpdo@a{%
1100   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@tmpdo@b
1101   \else \expandafter\bxjs@tmpdo@c \fi}
1102 \def\bxjs@tmpdo@b#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1103 \def\bxjs@tmpdo@c#1\@nil{\bxjs@tmpdo@d#1,,\@nil}
1104 \def\bxjs@tmpdo@d#1,#2,#3\@nil{%
1105   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1106 \expandafter\bxjs@tmpdo\bxjs@param@paper\@empty\@empty\@nil
```

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

```
1107 \edef\bxjs@layout@paper{%
1108   \ifjsc@mag truedimen,\fi
1109   \if@landscape landscape,\fi
1110   \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。

```
1111 %<*article|report>
1112 \def\bxjs@layout@base{%
1113   headheight=\topskip,footskip=0.03367\paperheight,%
1114   headsep=\footskip-\topskip,includeheadfoot,%
1115 }
1116 \edef\bxjs@layout{\bxjs@layout@base
1117   hscale=0.76,hmarginratio=1:1,%
1118   vscale=0.83,vmarginratio=1:1,%
1119 }
1120 %</article|report>
```

```

1121 %<*book>
1122 \def\bxjs@layout@base{%
1123   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1124 }
1125 \ifbxjs@layout@buggyhmargin    %---
1126 % アレ
1127 \edef\bxjs@layout{\bxjs@layout@base
1128   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1129   vscale=0.83,vmarginratio=1:1,%
1130 }
1131 \else    %---
1132 % 非アレ
1133 \edef\bxjs@layout{\bxjs@layout@base
1134   hmargin=18\jsc@mmm,%
1135   vscale=0.83,vmarginratio=1:1,%
1136 }
1137 \fi    %---
1138 %</book>
1139 %<*slide>
1140 \def\bxjs@layout@base{%
1141   noheadfoot,%
1142 }
1143 \edef\bxjs@layout{\bxjs@layout@base
1144   hscale=0.9,hmarginratio=1:1,%
1145   vscale=0.944,vmarginratio=1:1,%
1146 }
1147 %</slide>

```

textwidth オプションの設定を反映する。

```

1148 %<!*book>
1149 \ifx\bxjs@textwidth@opt\undefined\else
1150   \jsSetQHLlength\@tempdima{\bxjs@textwidth@opt}
1151   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1152 \fi
1153 %</!book>
1154 \ifx\bxjs@number@of@lines@opt\undefined\else
1155   \bxjs@gsset@tempcnta{\bxjs@number@of@lines@opt}
1156   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1157 \fi

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1158 \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。

```

1159 %<*book>
1160 \newcommand\jsTextWidthLimit{40}
1161 \@tempdima=\jsTextWidthLimit\Cwd

```

```

1162 \ifx\bxjs@textwidth@limit@opt\undefined\else
1163   \bxjs@egset@tempcnta{\bxjs@textwidth@limit@opt}
1164   \@tempdima=\@tempcnta\Cwd
1165 \fi
1166 \ifx\bxjs@textwidth@opt\undefined\else
1167   \jsSetQHLlength\@tempdima{\bxjs@textwidth@opt}
1168 \fi
1169 \edef\bxjs@textwidth@limit{\the\@tempdima}
1170 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1171   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1172   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1173 \fi
1174 %</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。

```

1175 \def\bxjs@preproc@layout{%
1176   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mpt}

```

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

```

1177 \def\bxjs@postproc@layout{%
    geometry のドライバを再設定する。
1178   \ifx\bxjs@geometry@driver\relax\else
1179     \let\Gm@driver\bxjs@geometry@driver
1180   \fi
    \ht\strutbox の値を元に戻す。
1181   \ht\strutbox=\bxjs@save@ht@strutbox\relax
    \textwidth の値を補正する。
1182   \ifbxjs@whole@zw@lines
1183     \@tempdimb=\textwidth
1184     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1185     \advance\textwidth.005pt\relax
1186     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1187     \advance\@tempdimb-\textwidth
1188     \advance\oddsidemargin 0.5\@tempdimb
1189     \advance\evensidemargin 0.5\@tempdimb
1190   \fi
1191   \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。

```

1192 %<*book>
1193   \@tempdima=\bxjs@textwidth@limit\relax
1194   \ifbxjs@whole@zw@lines

```



```

1195 \advance\@tempdima.005pt\relax
1196 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1197 \fi
1198 \ifdim\textwidth>\@tempdima
1199 \textwidth=\@tempdima
1200 \addtolength\evensidemargin{\fullwidth-\textwidth}
1201 \fi
1202 %</book>

\textheight 関連の調整。
1203 \@tempdimb=\textheight
1204 \advance\textheight-\topskip
1205 \advance\textheight.005pt\relax
1206 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1207 \advance\textheight\topskip
1208 \advance\@tempdimb-\textheight
1209 \advance\topmargin0.5\@tempdimb

\headheight 関連の調整。
1210 \@tempdima=\topskip
1211 \advance\headheight\@tempdima
1212 \advance\topmargin-\@tempdima

marginpar 関連の調整。
1213 \setlength\marginparsep{\columnsep}
1214 \setlength\marginparpush{\baselineskip}
1215 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1216 -\textwidth-10\jsc@mmm-\marginparsep}
1217 \ifbxjs@whole@zw@lines
1218 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1219 \fi

連動する変数。
1220 \maxdepth=.5\topskip
1221 \stockwidth=\paperwidth
1222 \stockheight=\paperheight
1223 }

```

`\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。

```

1224 \edef\jsGeometryOptions{%
1225 \bxjs@layout@paper,\bxjs@layout}

```

---

## ■ geometry パッケージを読み込む

---

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

```

1226 \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。

```

1227 \ifbxjs@papersize
1228   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1229     \PassOptionsToPackage{dvipdfm}{geometry}
1230   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1231     \PassOptionsToPackage{dvipdfm}{geometry}
1232   \fi\fi
1233   \let\bxPapersizeSpecialDone=t
1234 \else
1235   \PassOptionsToPackage{driver=none}{geometry}
1236 \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。

```

1237 \AtBeginDocument{\bxjs@bd@pre@geometry@hook}
1238 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1239 \bxjs@preproc@layout
1240 \edef\bxjs@next{%
1241   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1242 }\bxjs@next
1243 \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。

```

1244 \let\bxjs@geometry@driver\Gm@driver
1245 \bxjs@postproc@layout

```

`geometry` のドライバ自動判別に対する前処理。

```

1246 \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。

```

1247   \@ifpackagelater{geometry}{2010/02/12}{\}%else
1248   \PackageError\bxjs@clsname
1249     {Your 'geometry' package is too old (< v5.0)}%
1250     {\@ehc}%
1251   \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。

```

1252   \ifjsWithpTeXng
1253     \ifx\Gm@driver\@empty
1254       \def\Gm@driver{pdftex}%
1255     \fi
1256   \fi}

```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```

1257 \def\setpagelayout{%
1258   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1259     \@ifstar{\bxjs@setpagelayout@a@ne}{\bxjs@setpagelayout@a@z@}}
1260 \def\bxjs@setpagelayout@a#1#2{%
1261   \ifcase#1% modify
1262     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1263   \or% reset(*)
1264     \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1265   \or% semireset(+)
1266     \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1267   \fi
1268   \bxjs@preproc@layout
1269   \edef\bxjs@next{%
1270     \noexpand\geometry{\bxjs@next}%
1271   }\bxjs@next
1272   \bxjs@postproc@layout}

```

---

## ■ geometry パッケージを読み込まない

---

geometry=user の場合の処理。

```

1273 \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。

```

1274 \g@addto@macro\bxjs@begin@document@hook{%
1275   \ifdim\textwidth=.5\maxdimen
1276     \ClassError\bxjs@clsname
1277       {Page layout is not properly set}%
1278     {\@ehd}%
1279   \fi}
1280 \def\jsUseMinimalPageLayout{%
1281   \setlength{\textwidth}{6.5in}%
1282   \setlength{\textheight}{8in}}

```

\setpagelayout はとりあえず無効にしておく。

```

1283 \let\bxjs@geometry@driver\relax
1284 \def\setpagelayout{%
1285   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1286     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1287 \def\bxjs@pagelayout@a#1{%
1288   \ClassError\bxjs@clsname
1289   {Command '\string\setpagelayout' is not supported,\MessageBreak
1290     because 'geometry' value is not 'class'}\@eha}
1291 %
1292 \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。

---

```
1293 %<*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。

```
1294 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1295 \if@slide
1296   \setlength\headheight{0\jsc@empt}
1297 \else
1298   \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1299 \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。

```
1300 %<*article|kiyou>
1301 \if@slide
1302   \setlength\footskip{0pt}
1303 \else
1304   \setlength\footskip{0.03367\paperheight}
1305   \ifdim\footskip<\baselineskip
1306     \setlength\footskip{\baselineskip}
1307   \fi
1308 \fi
1309 %</article|kiyou>
```

```

1310 %<jspf>\setlength\footskip{9\jsc@mmm}
1311 %<*book>
1312 \if@report
1313   \setlength\footskip{0.03367\paperheight}
1314   \ifdim\footskip<\baselineskip
1315     \setlength\footskip{\baselineskip}
1316   \fi
1317 \else
1318   \setlength\footskip{0pt}
1319 \fi
1320 %</book>
1321 %<*report>
1322 \setlength\footskip{0.03367\paperheight}
1323 \ifdim\footskip<\baselineskip
1324   \setlength\footskip{\baselineskip}
1325 \fi
1326 %</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)。

```

1327 %<*article>
1328 \if@slide
1329   \setlength\headsep{0\jsc@empt}
1330   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1331   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1332 \else
1333   \setlength\headsep{\footskip}
1334   \addtolength\headsep{-\topskip}
1335 \fi
1336 %</article>
1337 %<*book>
1338 \if@report
1339   \setlength\headsep{\footskip}
1340   \addtolength\headsep{-\topskip}
1341 \else
1342   \setlength\headsep{6\jsc@mmm}
1343   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1344   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1345 \fi
1346 %</book>
1347 %<*report>
1348 \setlength\headsep{\footskip}
1349 \addtolength\headsep{-\topskip}
1350 %</report>
1351 %<*jspf>

```

```

1352 \setlength\headsep{9\jsc@mmm}
1353 \addtolength\headsep{-\topskip}
1354 %</jspf>
1355 %<*kiyou>
1356 \setlength\headheight{0\jsc@empt}
1357 \setlength\headsep{0\jsc@empt}
1358 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1359 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1360 %</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。

```

1361 \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。

```

1362 \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。

```

1363 %<*article>
1364 \if@slide
1365   \setlength\fullwidth{0.9\paperwidth}
1366 \else
1367   \setlength\fullwidth{0.76\paperwidth}
1368 \fi
1369 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1370 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1371 \setlength\textwidth{\fullwidth}
1372 %</article>
1373 %<*book>
1374 \if@report
1375   \setlength\fullwidth{0.76\paperwidth}
1376 \else
1377   \setlength\fullwidth{\paperwidth}

```

```

1378 \addtolength\fullwidth{-36\jsc@mmm}
1379 \fi
1380 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1381 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1382 \setlength\textwidth{\fullwidth}
1383 \if@report \else
1384 \if@twocolumn \else
1385 \ifdim \fullwidth>40zw
1386 \setlength\textwidth{40zw}
1387 \fi
1388 \fi
1389 \fi
1390 %</book>
1391 %<*report>
1392 \setlength\fullwidth{0.76\paperwidth}
1393 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1394 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1395 \setlength\textwidth{\fullwidth}
1396 %</report>
1397 %<*jspf>
1398 \setlength\fullwidth{50zw}
1399 \addtolength\fullwidth{8\jsc@mmm}
1400 \setlength\textwidth{\fullwidth}
1401 %</jspf>
1402 %<*kiyou>
1403 \setlength\fullwidth{48zw}
1404 \addtolength\fullwidth{\columnsep}
1405 \setlength\textwidth{\fullwidth}
1406 %</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)。

```

1407 %<*article|book|report>
1408 \if@slide

```

```

1409 \setlength{\textheight}{0.95\paperheight}
1410 \else
1411 \setlength{\textheight}{0.83\paperheight}
1412 \fi
1413 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1414 \addtolength{\textheight}{-\headsep}
1415 \addtolength{\textheight}{-\footskip}
1416 \addtolength{\textheight}{-\topskip}
1417 \divide\textheight\baselineskip
1418 \multiply\textheight\baselineskip
1419 %</article|book|report>
1420 %<jspf>\setlength{\textheight}{51\baselineskip}
1421 %<kiyou>\setlength{\textheight}{47\baselineskip}
1422 \addtolength{\textheight}{\topskip}
1423 \addtolength{\textheight}{0.1\jsc@mpt}
1424 %<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。

```

1425 \def\flushbottom{%
1426     \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1427     \let\@texttop\relax}

```

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

```

1428 \setlength\marginparsep{\columnsep}
1429 \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。

```

1430 \setlength{\oddsidemargin}{\paperwidth}
1431 \addtolength{\oddsidemargin}{-\fullwidth}
1432 \setlength{\oddsidemargin}{.5\oddsidemargin}
1433 \iftombow
1434     \addtolength{\oddsidemargin}{-1in}
1435 \else
1436     \addtolength{\oddsidemargin}{-\inv@mag in}
1437 \fi
1438 \setlength{\evensidemargin}{\oddsidemargin}

```



```

1439 \if@mparswitch
1440   \addtolength{\evensidemargin}{\fullwidth}
1441   \addtolength{\evensidemargin}{-\textwidth}
1442 \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。

```

1443 \setlength\marginparwidth{\paperwidth}
1444 \addtolength\marginparwidth{-\oddsidemargin}
1445 \addtolength\marginparwidth{-\inv@mag in}
1446 \addtolength\marginparwidth{-\textwidth}
1447 \addtolength\marginparwidth{-10\jsc@mmm}
1448 \addtolength\marginparwidth{-\marginparsep}
1449 \@tempdima=1zw
1450 \divide\marginparwidth\@tempdima
1451 \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)。

```

1452 \setlength\topmargin{\paperheight}
1453 \addtolength\topmargin{-\textheight}
1454 \if@slide
1455   \addtolength\topmargin{-\headheight}
1456 \else
1457   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1458 \fi
1459 \addtolength\topmargin{-\headsep}
1460 \addtolength\topmargin{-\footskip}
1461 \setlength\topmargin{0.5\topmargin}
1462 %<kiyou>\setlength\topmargin{81truebp}
1463 \iftombow
1464   \addtolength\topmargin{-1in}
1465 \else
1466   \addtolength\topmargin{-\inv@mag in}
1467 \fi
1468 %</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。

---

```
1469 \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。

```
1470 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

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

```
1471 \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。

```
1472 \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。

```
1473 \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。

```
1474 \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。

```
1475 \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。

```
1476 \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。

```
1477 \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。

```
1478 \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。

```
1479 \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。

```
1480 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1481 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1482 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1483 \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 1484 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1485 \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 1486 \setlength\@fptop{0\p@? \@plus 1fil}
```

```
1487 \setlength\@fpsep{8\p@? \@plus 2fil}
```

```
1488 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
\@dblfpsep 1489 \setlength\@dblfpsep{0\p@? \@plus 1fil}
```

```
\@dblfpbot 1490 \setlength\@dblfpsep{8\p@? \@plus 2fil}
```

```
1491 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

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

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

`\pltx@cleartooddpage`

`\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。

```

1492 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1493 % \ifodd\c@page
1494 % \iftdir
1495 % \hbox{}\thispagestyle{empty}\newpage
1496 % \if@twocolumn\hbox{}\newpage\fi
1497 % \fi
1498 % \else
1499 % \ifydir
1500 % \hbox{}\thispagestyle{empty}\newpage
1501 % \if@twocolumn\hbox{}\newpage\fi
1502 % \fi
1503 % \fi\fi}
1504 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1505 % \ifodd\c@page
1506 % \ifydir
1507 % \hbox{}\thispagestyle{empty}\newpage
1508 % \if@twocolumn\hbox{}\newpage\fi
1509 % \fi
1510 % \else
1511 % \iftdir
1512 % \hbox{}\thispagestyle{empty}\newpage
1513 % \if@twocolumn\hbox{}\newpage\fi
1514 % \fi
1515 % \fi\fi}
1516 %\def\pltx@cleartooddpage{\clearpage\if@twoside
1517 % \ifodd\c@page\else
1518 % \hbox{}\thispagestyle{empty}\newpage
1519 % \if@twocolumn\hbox{}\newpage\fi
1520 % \fi\fi}
1521 %\def\pltx@cleartoevenpage{\clearpage\if@twoside
1522 % \ifodd\c@page
1523 % \hbox{}\thispagestyle{empty}\newpage
1524 % \if@twocolumn\hbox{}\newpage\fi
1525 % \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。

```

1526 \let\pltx@cleartorightpage\pltx@cleartooddpage
1527 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

---

`\cleardoublepage` [2017-02-24] コミュニティ版 p<sup>A</sup>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。

```

1528 %<*book|report>
1529 \if@openleft
1530 \let\cleardoublepage\pltx@cleartoleftpage
1531 \else\if@openright
1532 \let\cleardoublepage\pltx@cleartorightpage

```

```

1533 \fi\fi
1534 %</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` で変更する場合がこれにあたります。しかし、同一ページに複数の `\markboth` があると、おかしい結果になることがあります。

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

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

```

1535 % \def\ps@empty{%
1536 %   \let\mkboth\gobbletwo
1537 %   \let\oddhead\empty
1538 %   \let\oddfoot\empty
1539 %   \let\evenhead\empty
1540 %   \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。

```

1541 \def\ps@plainfoot{%
1542   \let\mkboth\gobbletwo

```

```

1543 \let\@oddhead\@empty
1544 \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1545 \let\@evenhead\@empty
1546 \let\@evenfoot\@oddfoot}
1547 \def\ps@plainhead{%
1548 \let\@mkboth\@gobbletwo
1549 \let\@oddfoot\@empty
1550 \let\@evenfoot\@empty
1551 \def\@evenhead{%
1552 \if@mparswitch \hss \fi
1553 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1554 \if@mparswitch\else \hss \fi}%
1555 \def\@oddhead{%
1556 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1557 %<book>\let\ps@plain\ps@plainhead
1558 %<!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。

```

1559 %<*article|slide>
1560 \if@twoside
1561 \def\ps@headings{%
1562 \let\@oddfoot\@empty
1563 \let\@evenfoot\@empty
1564 \def\@evenhead{\if@mparswitch \hss \fi
1565 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1566 \if@mparswitch\else \hss \fi}%
1567 \def\@oddhead{%
1568 \underline{%
1569 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1570 \let\@mkboth\markboth
1571 \def\sectionmark##1{\markboth{%
1572 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1573 ##1}{}}%
1574 \def\subsectionmark##1{\markright{%
1575 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1576 ##1}}%
1577 }
1578 \else % if not twoside
1579 \def\ps@headings{%
1580 \let\@oddfoot\@empty
1581 \def\@oddhead{%
1582 \underline{%
1583 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1584 \let\@mkboth\markboth
1585 \def\sectionmark##1{\markright{%
1586 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi

```

```

1587         ##1}}
1588 \fi
1589 %</article|slide>

```

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

---

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

---

```

1590 %<*book|report>
1591 \def\bxjs@maybe@autoxspacing{%
1592   \ifx\autoxspacing\undefined\else \autoxspacing \fi}
1593 \newif\if@omit@number
1594 \def\ps@headings{%
1595   \let\@oddfoot\@empty
1596   \let\@evenfoot\@empty
1597   \def\@evenhead{%
1598     \if@mparswitch \hss \fi
1599     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1600       \textbf{\thepage}\hfil\leftmark}}}%
1601     \if@mparswitch\else \hss \fi}%
1602   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1603     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1604   \let\@mkboth\markboth
1605   \def\chaptermark##1{\markboth{%
1606     \ifnum \c@secnumdepth >\m@ne
1607       \if@mainmatter
1608         \if@omit@number\else
1609           \@chapapp\thechapter\@chappos\hskip1\jsZw
1610         \fi
1611       \fi
1612       \fi
1613     ##1}{}}%
1614   \def\sectionmark##1{\markright{%
1615     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1616     ##1}}}%
1617 %</book|report>

```

最後は学会誌の場合です。

```

1618 %<*jspf>
1619 \def\ps@headings{%
1620   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1621   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1622   \def\@oddhead{\normalfont\hfil \@title \hfil}
1623   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1624 %</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。

```
1625 \def\ps@myheadings{%
1626   \let\@oddfoot\@empty\let\@evenfoot\@empty
1627   \def\@evenhead{%
1628     \if@mparswitch \hss \fi%
1629     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1630     \if@mparswitch\else \hss \fi}%
1631   \def\@oddhead{%
1632     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1633   \let\@mkboth\@gobbletwo
1634 %<book|report> \let\chaptermark\@gobble
1635   \let\sectionmark\@gobble
1636 %<!book&!report> \let\subsectionmark\@gobble
1637 }
```

## 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 ます。
\date 1638 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1639 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1640 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1641 % \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。

```
1642 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1643 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1644 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1645 \AtBeginDocument{\bxjs@decl@subtitle}
1646 \def\bxjs@decl@subtitle{%
1647   \global\let\bxjs@decl@subtitle\relax
1648   \ifx\subtitle\@undefined
1649     \global\let\subtitle\jsSubtitle
1650   \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。

```
1651 \def\bxjs@annihilate@subtitle{%
1652   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1653   \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 1654 %<*jspf>
1655 \newcommand*\etitle[1]{\gdef\@etitle{#1}}
\keywords 1656 \newcommand*\eauthor[1]{\gdef\@eauthor{#1}}
1657 \newcommand*\keywords[1]{\gdef\@keywords{#1}}
1658 \newcommand*\email[1]{\gdef\authors@mail{#1}}
1659 \newcommand*\AuthorsEmail[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1660 %</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。

```
1661 \def\plainifnotempty{%
1662   \ifx \@oddhead \@empty
1663     \ifx \@oddfoot \@empty
1664       \else
1665         \thispagestyle{plainfoot}%
1666       \fi
1667     \else
1668       \thispagestyle{plainhead}%
1669     \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。

```
1670 %<*article|book|report|slide>
1671 \if@titlepage
1672   \newcommand*\maketitle{%
1673     \begin{titlepage}%
1674       \let\footnotesize\small
1675       \let\footnoterule\relax
1676       \let\footnote\thanks
1677       \null\vfil
1678       \if@slide
1679         {\footnotesize \@date}%
1680       \begin{center}
```

```

1681         \mbox{} \[1\jsZw]
1682     \large
1683     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1684     \jsc@smallskip
1685     \@title
1686     \ifx\bxjs@subtitle\@undefined\else
1687         \par\vskip\z@
1688         {\small \bxjs@subtitle\par}
1689     \fi
1690     \jsc@smallskip
1691     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1692     \vfill
1693     {\small \@author}%
1694 \end{center}
1695 \else
1696 \vskip 60\p@?
1697 \begin{center}%
1698 {\LARGE \@title \par}%
1699 \ifx\bxjs@subtitle\@undefined\else
1700     \vskip5\p@?
1701     {\normalsize \bxjs@subtitle\par}
1702 \fi
1703 \vskip 3em%
1704 {\large
1705     \lineskip .75em
1706     \begin{tabular}[t]{c}%
1707         \@author
1708     \end{tabular}\par}%
1709 \vskip 1.5em
1710 {\large \@date \par}%
1711 \end{center}%
1712 \fi
1713 \par
1714 \@thanks\vfil\null
1715 \end{titlepage}%
1716 \setcounter{footnote}{0}%
1717 \global\let\thanks\relax
1718 \global\let\maketitle\relax
1719 \global\let\@thanks\@empty
1720 \global\let\@author\@empty
1721 \global\let\@date\@empty
1722 \global\let\@title\@empty
1723 \global\let\title\relax
1724 \global\let\author\relax
1725 \global\let\date\relax
1726 \global\let\and\relax
1727 \bxjs@annihilate@subtitle
1728 }%
1729 \else

```

```

1730 \newcommand{\maketitle}{\par
1731   \begin{group}
1732     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1733     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1734     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1735       \parindent 1\jsZw\noindent
1736       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1737     \if@twocolumn
1738       \ifnum \col@number=\@ne
1739         \@maketitle
1740       \else
1741         \twocolumn[\@maketitle]%
1742       \fi
1743     \else
1744       \newpage
1745       \global\@topnum\z@ % Prevents figures from going at top of page.
1746       \@maketitle
1747     \fi
1748     \plainifnotempty
1749     \@thanks
1750   \end{group}
1751   \setcounter{footnote}{0}%
1752   \global\let\thanks\relax
1753   \global\let\maketitle\relax
1754   \global\let\@thanks\@empty
1755   \global\let\@author\@empty
1756   \global\let\@date\@empty
1757   \global\let\@title\@empty
1758   \global\let\title\relax
1759   \global\let\author\relax
1760   \global\let\date\relax
1761   \global\let\and\relax
1762   \bxjs@annihilate@subtitle
1763 }

```

`\@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。

```

1764 \def\@maketitle{%
1765   \newpage\null
1766   \vskip 2em
1767   \begin{center}%
1768     \let\footnote\thanks
1769     {\LARGE \@title \par}%
1770     \ifx\bxjs@subtitle\@undefined\else
1771       \vskip3\p@?
1772       {\normalsize \bxjs@subtitle\par}
1773     \fi
1774     \vskip 1.5em
1775     {\large
1776       \lineskip .5em

```

```

1777         \begin{tabular}[t]{c}%
1778         \@author
1779         \end{tabular}\par}%
1780         \vskip 1em
1781         {\large \@date}%
1782     \end{center}%
1783     \par\vskip 1.5em
1784 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1785 }
1786 \fi
1787 %</article|book|report|slide>
1788 %<*jspf>
1789 \newcommand{\maketitle}{\par
1790     \begin{group}
1791         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1792         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1793         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1794             \parindent 1\jsZw\noindent
1795             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1796         \twocolumn[\@maketitle]%
1797         \plainifnotempty
1798         \@thanks
1799     \end{group}
1800     \setcounter{footnote}{0}%
1801     \global\let\thanks\relax
1802     \global\let\maketitle\relax
1803     \global\let\@thanks\@empty
1804     \global\let\@author\@empty
1805     \global\let\@date\@empty
1806 % \global\let\@title\@empty % \@title は柱に使う
1807     \global\let\title\relax
1808     \global\let\author\relax
1809     \global\let\date\relax
1810     \global\let\and\relax
1811     \ifx\authors@mail\@undefined\else{%
1812         \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1813         \footnotetext[0]{\itshape\authors@mail}%
1814     }\fi
1815     \global\let\authors@mail\@undefined}
1816 \def\@maketitle{%
1817     \newpage\null
1818     \vskip 6em % used to be 2em
1819     \begin{center}
1820         \let\footnote\thanks
1821         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1822         \lineskip .5em
1823         \ifx\@author\@undefined\else
1824             \vskip 1em
1825             \begin{tabular}[t]{c}%

```

```

1826      \@author
1827      \end{tabular}\par
1828      \fi
1829      \ifx\@etitle\@undefined\else
1830          \vskip 1em
1831          {\large \@etitle \par}%
1832      \fi
1833      \ifx\@eauthor\@undefined\else
1834          \vskip 1em
1835          \begin{tabular}[t]{c}%
1836              \@eauthor
1837          \end{tabular}\par
1838      \fi
1839      \vskip 1em
1840      \@date
1841      \end{center}
1842      \vskip 1.5em
1843      \centerline{\box\@abstractbox}
1844      \ifx\@keywords\@undefined\else
1845          \vskip 1.5em
1846          \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1847      \fi
1848      \vskip 1.5em}
1849 %</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。

```

1850 \def\bxjs@label@sect#1{%
1851     \expandafter\ifx\csname label#1\endcsname\relax
1852         \csname the#1\endcsname
1853     \else \csname label#1\endcsname
1854     \fi}
1855 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1856 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1857 \def\@secapp{\presectionname}
1858 \def\@secpos{\postsectionname}
1859 \fi

```

`\labelsection` 節番号の出力書式。

```

1860 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1861 \def\labelsection{\@secapp\thesection\@secpos}
1862 \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。

```

1863 \def\@startsection#1#2#3#4#5#6{%
1864   \if@noskipsec \leavevmode \fi
1865   \par
1866 % 見出し上の空きを \@tempskipa にセットする
1867   \@tempskipa #4\relax
1868 % \@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
1869   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1870 % 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
1871   \ifdim \@tempskipa <\z@
1872     \@tempskipa -\@tempskipa \@afterindentfalse
1873   \fi
1874   \if@nobreak
1875 %   \everypar{\everyparhook}% これは間違い
1876     \everypar{}%
1877   \else

```

```

1878 \addpenalty\@secpenalty
1879 % 次の行は削除
1880 % \addvspace\@tempskipa
1881 % 次の \noindent まで追加
1882 \ifdim \@tempskipa >\z@
1883 \if@slide\else
1884 \null
1885 \vspace*{-\baselineskip}%
1886 \fi
1887 \vskip\@tempskipa
1888 \fi
1889 \fi
1890 \noindent
1891 % 追加終わり
1892 \@ifstar
1893 {\@ssect{#3}{#4}{#5}{#6}}%
1894 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

\@sect と \@xsect は、前のアキがちょうどゼロの場合にもうまくいくように、多少変えてあります。 \everyparhook も挿入しています。

---

\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。

```

1895 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1896 \let\bxjs@if@ceph\@firstofone
1897 \else \let\bxjs@if@ceph\@gobble
1898 \fi

```

---

```

1899 \def\@sect#1#2#3#4#5#6[#7]#8{%
1900 \ifnum #2>\c@secnumdepth
1901 \let\@svsec\@empty
1902 \else
1903 \refstepcounter{#1}%
1904 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1905 \fi
1906 % 見出し後の空きを \@tempskipa にセット
1907 \@tempskipa #5\relax
1908 % 条件判断の順序を入れ替えました
1909 \ifdim \@tempskipa<\z@
1910 \def\@svsechd{%
1911 #6{\hskip #3\relax
1912 \@svsec #8}%
1913 \csname #1mark\endcsname{#7}}%
1914 \addcontentsline{toc}{#1}{%
1915 \ifnum #2>\c@secnumdepth \else
1916 \protect\numberline{\bxjs@label@sect{#1}}}%
1917 \fi

```

```

1918      #7}}% 目次にフルネームを載せるなら #8
1919 \else
1920   \begingroup
1921     \interlinepenalty \@M % 下から移動
1922     #6{%
1923       \@hangfrom{\hskip #3\relax\@svsec}%
1924 %     \interlinepenalty \@M % 上に移動
1925     #8\@@par}%
1926   \endgroup
1927   \csname #1mark\endcsname{#7}%
1928   \addcontentsline{toc}{#1}{%
1929     \ifnum #2>\c@secnumdepth \else
1930       \protect\numberline{\bxjs@label@sect{#1}}%
1931     \fi
1932     #7}}% 目次にフルネームを載せるならここは #8
1933 \fi
1934 \@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。

```

1935 \def\@xsect#1{%
1936 % 見出しの後ろの空きを \@tempskipa にセット
1937   \@tempskipa #1\relax
1938 % 条件判断の順序を変えました
1939   \ifdim \@tempskipa<\z@
1940     \@nbreakfalse
1941     \global\@noskipsecttrue
1942     \everypar{%
1943       \if@noskipsec
1944         \global\@noskipsecfalse
1945         {\setbox\z@\lastbox}%
1946         \clubpenalty \@M
1947         \begingroup \@svsechd \endgroup
1948         \unskip
1949         \@tempskipa #1\relax
1950         \hskip -\@tempskipa
1951       \else
1952         \clubpenalty \@clubpenalty
1953         \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1954         \fi\bxjs@if@ceph\everyparhook}%
1955   \else
1956     \par \nbreak
1957     \vskip \@tempskipa
1958     \@afterheading
1959   \fi
1960   \if@slide

```



```

1961      {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1962      \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1963      \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1964      \fi
1965      \par % 2000-12-18
1966      \ignorespaces}
1967 \def\@ssect#1#2#3#4#5{%
1968   \@tempskipa #3\relax
1969   \ifdim \@tempskipa<\z@
1970     \def\@svsechd{#4{\hskip #1\relax #5}}%
1971   \else
1972     \begingroup
1973       #4{%
1974         \@hangfrom{\hskip #1}%
1975         \interlinepenalty \@M #5\@@par}%
1976     \endgroup
1977   \fi
1978   \@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  
`\sectionmark` 定義済みです。

```

\subsectionmark 1979 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1980 % \newcommand*\sectionmark[1]{}
\subsubsectionmark 1981 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1982 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1983 % \newcommand*\paragraphmark[1]{}
1984 % \newcommand*\subparagraphmark[1]{}

```

#### ■カウンタの定義

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

```

1985 %<!book&!report>\setcounter{secnumdepth}{3}
1986 %<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 1987 \newcounter{part}
\c@subsubsection 1988 %<book|report>\newcounter{chapter}
1989 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1990 %<!book&!report>\newcounter{section}
\c@subparagraph 1991 \newcounter{subsection}[section]
1992 \newcounter{subsubsection}[subsection]
1993 \newcounter{paragraph}[subsubsection]
1994 \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`

`\thesubparagraph`

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。

```
1995 \renewcommand{\thepart}{\@Roman\c@part}
1996 %<*!book&!report>
1997 \ifnum\bxjs@label@section=\bxjs@label@section@compat
1998 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1999 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2000 \else
2001 \renewcommand{\thesection}{\@arabic\c@section}
2002 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2003 \fi
2004 %</!book&!report>
2005 %<*book|report>
2006 \renewcommand{\thechapter}{\@arabic\c@chapter}
2007 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2008 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2009 %</book|report>
2010 \renewcommand{\thesubsubsection}{\%
2011 \thesubsection.\@arabic\c@subsubsection}
2012 \renewcommand{\theparagraph}{\%
2013 \thesubsubsection.\@arabic\c@paragraph}
2014 \renewcommand{\thesubparagraph}{\%
2015 \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。

```
2016 %<book|report>\newcommand{\@chapapp}{\prechaptername}
```

```
2017 %<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)。

```
2018 %<*book|report>
2019 \newcommand\frontmatter{%
2020   \pltx@cleartooddpage
2021   \@mainmatterfalse
2022   \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。

```
2023 \newcommand\mainmatter{%
2024   \pltx@cleartooddpage
2025   \@mainmattertrue
2026   \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。

```
2027 \newcommand\backmatter{%
2028   \if@openleft
2029     \cleardoublepage
2030   \else\if@openright
2031     \cleardoublepage
2032   \else
2033     \clearpage
2034   \fi\fi
2035   \@mainmatterfalse}
2036 %</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。

```
2037 %<*&!book&!report>
2038 \newcommand\part{%
2039   \if@noskipsec \leavevmode \fi
2040   \par
2041   \addvspace{4ex}%
2042   \if@english \@afterindentfalse \else \@afterindenttrue \fi
```

```

2043 \secdef\@part\@spart}
2044 %</*!book&!report>

book および report クラスの場合は、少し複雑です。

2045 %<*book|report>
2046 \newcommand\part{%
2047 \if@openleft
2048 \cleardoublepage
2049 \else\if@openright
2050 \cleardoublepage
2051 \else
2052 \clearpage
2053 \fi\fi
2054 \thispagestyle{empty}% 欧文用標準スタイルでは plain
2055 \if@twocolumn
2056 \onecolumn
2057 \@restonecoltrue
2058 \else
2059 \@restonecolfalse
2060 \fi
2061 \null\vfil
2062 \secdef\@part\@spart}
2063 %</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。

```

2064 %</*!book&!report>
2065 \def\@part[#1]#2{%
2066 \ifnum \c@secnumdepth >\m@ne
2067 \refstepcounter{part}%
2068 \addcontentsline{toc}{part}{%
2069 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2070 \else
2071 \addcontentsline{toc}{part}{#1}%
2072 \fi
2073 \markboth{}{}%
2074 {\parindent\z@
2075 \raggedright
2076 \interlinepenalty \@M
2077 \normalfont
2078 \ifnum \c@secnumdepth >\m@ne
2079 \Large\headfont\prepartname\thepart\postpartname
2080 \par\nobreak
2081 \fi
2082 \huge \headfont #2%
2083 \markboth{}{}\par}%
2084 \nobreak
2085 \vskip 3ex

```

```

2086 \afterheading}
2087 %<*/!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。

```

2088 %<*book|report>
2089 \def\@part[#1]#2{%
2090   \ifnum \c@secnumdepth >-2\relax
2091     \refstepcounter{part}%
2092     \addcontentsline{toc}{part}{%
2093       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2094   \else
2095     \addcontentsline{toc}{part}{#1}%
2096   \fi
2097   \markboth{}{}%
2098   {\centering
2099     \interlinepenalty \@M
2100     \normalfont
2101     \ifnum \c@secnumdepth >-2\relax
2102       \huge\headfont \prepartname\thepart\postpartname
2103       \par\vskip20\p@?
2104     \fi
2105     \Huge \headfont #2\par}%
2106   \@endpart}
2107 %</book|report>

```

\@spart 番号を付けない部です。

```

2108 %<*!book&!report>
2109 \def\@spart#1{%
2110   \parindent \z@ \raggedright
2111   \interlinepenalty \@M
2112   \normalfont
2113   \huge \headfont #1\par}%
2114   \nobreak
2115   \vskip 3ex
2116   \@afterheading}
2117 %<*/!book&!report>
2118 %<*book|report>
2119 \def\@spart#1{%
2120   \centering
2121   \interlinepenalty \@M
2122   \normalfont
2123   \Huge \headfont #1\par}%
2124   \@endpart}
2125 %</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。

```
2126 %<*book|report>
2127 \def\@endpart{\vfil\newpage
2128   \if@twoside
2129     \if@openleft %% added (2017/02/24)
2130       \null\thispagestyle{empty}\newpage
2131     \else\if@openright %% added (2016/12/13)
2132       \null\thispagestyle{empty}\newpage
2133     \fi\fi %% added (2016/12/13, 2017/02/24)
2134   \fi
2135   \if@restonecol
2136     \twocolumn
2137   \fi}
2138 %</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。

```
2139 %<*book|report>
2140 \newcommand{\chapter}{%
2141   \if@openleft\cleardoublepage\else
2142   \if@openright\cleardoublepage\else\clearpage\fi\fi
2143   \plainifnotempty % 元: \thispagestyle{plain}
2144   \global\@topnum\z@
2145   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2146   \secdef
2147     {\@omit@numberfalse\@chapter}%
2148     {\@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。

```
2149 \def\@chapter[#1]#2{%
2150   \ifnum \c@secnumdepth >\m@ne
2151     \if@mainmatter
2152       \refstepcounter{chapter}%
2153       \typeout{\@chapapp\thechapter\@chappos}%
2154       \addcontentsline{toc}{chapter}%
2155         {\protect\numberline
2156 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2157         {\@chapapp\thechapter\@chappos}%
2158         #1}%
2159     \else\addcontentsline{toc}{chapter}{#1}\fi
2160   \else
2161     \addcontentsline{toc}{chapter}{#1}%
2162   \fi
2163   \chaptermark{#1}%
2164   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}}
```

```

2165 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}}%
2166 \if@twocolumn
2167 \topnewpage[\@makechapterhead{#2}]%
2168 \else
2169 \makechapterhead{#2}%
2170 \afterheading
2171 \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。

```

2172 \def\@makechapterhead#1{%
2173 \vspace*{2\Cvs}% 欧文は 50pt
2174 {\parindent \z@ \raggedright \normalfont
2175 \ifnum \c@secnumdepth >\m@ne
2176 \if@mainmatter
2177 \huge\headfont \@chapapp\thechapter\@chappos
2178 \par\nobreak
2179 \vskip \Cvs % 欧文は 20pt
2180 \fi
2181 \fi
2182 \interlinepenalty\@M
2183 \Huge \headfont #1\par\nobreak
2184 \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。

```

2185 \def\@schapter#1{%
2186 \chaptermark{#1}%
2187 \if@twocolumn
2188 \topnewpage[\@makeschapterhead{#1}]%
2189 \else
2190 \makeschapterhead{#1}\@afterheading
2191 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2192 \def\@makeschapterhead#1{%
2193 \vspace*{2\Cvs}% 欧文は 50pt
2194 {\parindent \z@ \raggedright
2195 \normalfont
2196 \interlinepenalty\@M
2197 \Huge \headfont #1\par\nobreak
2198 \vskip 3\Cvs}} % 欧文は 40pt
2199 %</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。

```

2200 \if@twocolumn

```

```

2201 \newcommand{\section}{%
2202 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2203 \startsection{section}{1}{\z@}%
2204 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2205 %<kiyou> {\Cvs}{0.5\Cvs}%
2206 % {\normalfont\large\headfont\@secapp}%
2207 {\normalfont\large\headfont\raggedright}}
2208 \else
2209 \newcommand{\section}{%
2210 \if@slide\clearpage\fi
2211 \startsection{section}{1}{\z@}%
2212 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2213 {.5\Cvs \@plus.3\Cdp}% 後アキ
2214 % {\normalfont\Large\headfont\@secapp}%
2215 {\normalfont\Large\headfont\raggedright}}
2216 \fi

```

\subsection 同上です。

```

2217 \if@twocolumn
2218 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2219 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2220 {\normalfont\normalsize\headfont}}
2221 \else
2222 \newcommand{\subsection}{\startsection{subsection}{2}{\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}}
2226 \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)。

```

2227 \if@twocolumn
2228 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2229 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2230 {\normalfont\normalsize\headfont}}
2231 \else
2232 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2233 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2234 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2235 {\normalfont\normalsize\headfont}}
2236 \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。

---

```

2237 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2238 \let\bxjs@org@paragraph@mark\jsParagraphMark
2239 \ifx\bxjs@paragraph@mark\@empty
2240   \let\jsParagraphMark\@empty
2241 \else\ifx\bxjs@paragraph@mark\@undefined\else
2242   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2243 \fi\fi
2244 \let\jsJaChar\@empty
2245 \if@twocolumn
2246   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2247     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2248 %<jspf>    {\normalfont\normalsize\headfont}}
2249 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2250 \else
2251   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2252     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2253     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2254 %<jspf>    {\normalfont\normalsize\headfont}}
2255 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2256 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2257 \if@twocolumn
2258   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2259     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2260     {\normalfont\normalsize\headfont}}
2261 \else
2262   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2263     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2264     {\normalfont\normalsize\headfont}}
2265 \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。

```
2266 \if@slide
2267   \setlength\leftmargini{1\jsZw}
2268 \else
2269   \if@twocolumn
2270     \setlength\leftmargini{2\jsZw}
2271   \else
2272     \setlength\leftmargini{3\jsZw}
2273   \fi
2274 \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 2275 \if@slide
\leftmarginv 2276   \setlength\leftmarginii {1\jsZw}
2277   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2278   \setlength\leftmarginiv {1\jsZw}
2279   \setlength\leftmarginv {1\jsZw}
2280   \setlength\leftmarginvi {1\jsZw}
2281 \else
2282   \setlength\leftmarginii {2\jsZw}
2283   \setlength\leftmarginiii{2\jsZw}
2284   \setlength\leftmarginiv {2\jsZw}
2285   \setlength\leftmarginv {1\jsZw}
2286   \setlength\leftmarginvi {1\jsZw}
2287 \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。

```
2288 \setlength \labelsep {0.5\jsZw} % .5em
2289 \setlength \labelwidth{\leftmargini}
2290 \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。

```
2291 \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 2292 \@beginparpenalty -\@lowpenalty
\@itempenalty 2293 \@endparpenalty -\@lowpenalty
2294 \@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。

```
2295 \def\@listi{\leftmargin\leftmarginI
2296   \parsep \z@
2297   \topsep 0.5\baselineskip
2298   \itemsep \z@ \relax}
2299 \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）。

```
2300 \@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 2301 \def\@listii{\leftmargin\leftmarginii
\@listiv 2302   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2303   \topsep \z@
\@listv 2304   \parsep \z@
\@listvi 2305   \itemsep\parsep}
2306 \def\@listiii{\leftmargin\leftmarginiii
2307   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2308   \topsep \z@
2309   \parsep \z@
2310   \itemsep\parsep}
2311 \def\@listiv {\leftmargin\leftmarginiv
2312   \labelwidth\leftmarginiv
2313   \advance\labelwidth-\labelsep}
2314 \def\@listv {\leftmargin\leftmarginv
2315   \labelwidth\leftmarginv
2316   \advance\labelwidth-\labelsep}
2317 \def\@listvi {\leftmargin\leftmarginvi
2318   \labelwidth\leftmarginvi
2319   \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。

```
2320 \renewcommand{\theenumi}{\@arabic\c@enumi}
2321 \renewcommand{\theenumii}{\@alph\c@enumii}
2322 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2323 \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。

---

```

2324 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2325 \newcommand*\jsInJaParen}[1]{%
2326   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2327 \newcommand{\labelenumi}{\theenumi.}
2328 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2329 \newcommand{\labelenumiii}{\theenumiii.}
2330 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2331 \renewcommand{\p@enumii}{\theenumi}
2332 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2333 \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 2334 \newcommand\labelitemi{\textbullet}
\labelitemiii 2335 \newcommand\labelitemii{\normalfont\bfseries \textendash}
2336 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2337 \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。

```

2338 \newenvironment{description}{%
2339   \list{}{%
2340     \labelwidth=\leftmargin
2341     \labelsep=1\jsZw
2342     \advance \labelwidth by -\labelsep
2343     \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。

```

2344 \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。

```
2345 %<*book|report>
2346 \newenvironment{chapterabstract}{%
2347   \begin{list}{}{%
2348     \listparindent=1\jsZw
2349     \itemindent=\listparindent
2350     \rightmargin=0pt
2351     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2352 %</book|report>
```

“普通の” **abstract** 環境の定義。

```
2353 %<*article|report|slide>
2354 \newbox\@abstractbox
2355 \if@titlepage
2356   \newenvironment{abstract}{%
2357     \titlepage
2358     \null\vfil
2359     \@beginparpenalty\@lowpenalty
2360     \begin{center}%
2361       \headfont \abstractname
2362       \@endparpenalty\@M
2363     \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。

```
2364   \par}%
2365   {\par\vfil\@null\endtitlepage}
2366 \else
2367   \newenvironment{abstract}{%
2368     \if@twocolumn
2369       \ifx\maketitle\relax
2370       \section*{\abstractname}%
```

```

2371 \else
2372 \global\setbox\@abstractbox\hbox\bgroup
2373 \begin{minipage}[b]{\textwidth}
2374 \small\parindent1\jsZw
2375 \begin{center}%
2376 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2377 \end{center}%
2378 \list{}{%
2379 \listparindent\parindent
2380 \itemindent \listparindent
2381 \rightmargin \leftmargin}%
2382 \item\relax
2383 \fi
2384 \else
2385 \small
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}{\if@twocolumn
2395 \ifx\maketitle\relax
2396 \else
2397 \endlist\end{minipage}\egroup
2398 \fi
2399 \else
2400 \endlist
2401 \fi}
2402 \fi
2403 %</article|report|slide>
2404 %<*jspf>
2405 \newbox\@abstractbox
2406 \newenvironment{abstract}{%
2407 \global\setbox\@abstractbox\hbox\bgroup
2408 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2409 \small
2410 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2411 {\end{minipage}\egroup}
2412 %</jspf>

```

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

```

2413 %<*book|report>
2414 \ifbxjs@force@chapterabstract
2415 \let\abstract\chapterabstract
2416 \let\endabstract\endchapterabstract

```

```

2417 \fi
2418 %</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。

```

2419 %<*jspf>
2420 %\newbox\@keywordsbox
2421 %\newenvironment{keywords}{%
2422 % \global\setbox\@keywordsbox\hbox\bgroup
2423 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2424 % \small\parindent0\jsZw}%
2425 % {\end{minipage}\egroup}
2426 %</jspf>

```

## ■verse 環境

**verse** 詩のための `verse` 環境です。

```

2427 \newenvironment{verse}{%
2428 \let \=\@centercr
2429 \list{}{%
2430 \itemsep \z@
2431 \itemindent -2\jsZw % 元: -1.5em
2432 \listparindent\itemindent
2433 \rightmargin \z@
2434 \advance\leftmargin 2\jsZw}% 元: 1.5em
2435 \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。

```

2436 \newenvironment{quotation}{%
2437 \list{}{%
2438 \listparindent\parindent
2439 \itemindent\listparindent
2440 \rightmargin \z@}%
2441 \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。

```

2442 \newenvironment{quote}%
2443 {\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。

```
2444 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2445   \item[\hskip \labelsep{\headfont #1\ #2}]}
2446 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2447   \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 p<sub>La</sub>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 p<sub>La</sub>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。

```
2448 \newenvironment{titlepage}{%
2449 %<book> \pltx@cleartooddpage %% 2017-02-24
2450   \if@twocolumn
2451     \@restonecoltrue\onecolumn
2452   \else
2453     \@restonecolfalse\newpage
2454   \fi
2455   \thispagestyle{empty}%
2456   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2457   24
2458 }%
2459 {\if@restonecol\twocolumn \else \newpage \fi
2460   \if@twoside\else
2461     \setcounter{page}\@ne
2462   \fi}
```

## ■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2462 %<*&!book&!report>
2463 \newcommand{\appendix}{\par
2464   \setcounter{section}{0}%
2465   \setcounter{subsection}{0}%
2466   \ifnum\bxjs@label@section=\bxjs@label@section@compat
```



```

2467 \gdef\presectionname{\appendixname}%
2468 \gdef\postsectionname{}%
2469 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2470 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2471 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2472 \else
2473 \gdef\@secapp{\appendixname}%
2474 \gdef\@secpos{}%
2475 \gdef\thesection{\@Alph\c@section}%
2476 \fi}
2477 %</!book&!report>
2478 %<*book|report>
2479 \newcommand{\appendix}{\par
2480 \setcounter{chapter}{0}%
2481 \setcounter{section}{0}%
2482 \gdef\@chapapp{\appendixname}%
2483 \gdef\@chappos{}%
2484 \gdef\thechapter{\@Alph\c@chapter}}
2485 %</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。

```
2486 \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。

```
2487 \setlength\tabcolsep{6\p@?}
```

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

```
2488 \setlength\arrayrulewidth{.4\p@}
```

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

```
2489 \setlength\doublerulesep{2\p@}
```

### ■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2490 \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。

```
2491 \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。

```
2492 \setlength\fboxsep{3\p@?}
2493 \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。

```
2494 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2495 %<*book|report>
2496 \@addtoreset{equation}{chapter}
2497 \renewcommand\theequation
2498 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2499 %</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。

```
2500 % \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。

```
2501 % \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。

```
2502 % \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。

```
2503 %<!*book&!report>
2504 \newcounter{figure}
2505 \renewcommand \thefigure {\@arabic\c@figure}
2506 %</!*book&!report>
2507 %<*book|report>
2508 \newcounter{figure}[chapter]
2509 \renewcommand \thefigure
2510     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2511 %</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 2512 \def\fps@figure{tbp}
2513 \def\ftype@figure{1}
\fnun@figure 2514 \def\ext@figure{lof}
2515 \def\fnun@figure{\figurename\nobreak\thefigure}
```

`figure` \* 形式は段抜きのフロートです。

```
figure* 2516 \newenvironment{figure}%
2517     {\@float{figure}}%
2518     {\end@float}
2519 \newenvironment{figure*}%
2520     {\@dblfloat{figure}}%
2521     {\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。

```
2522 %<!*book&!report>
2523 \newcounter{table}
2524 \renewcommand\thetable{\@arabic\c@table}
2525 %</!*book&!report>
2526 %<*book|report>
2527 \newcounter{table}[chapter]
2528 \renewcommand \thetable
2529     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2530 %</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 2531 \def\fps@table{tbp}
2532 \def\ftype@table{2}
\fnun@table 2533 \def\ext@table{lot}
2534 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` \* は段抜きのフロートです。

`table*`

```

2535 \newenvironment{table}%
2536         {\@float{table}}%
2537         {\end@float}
2538 \newenvironment{table*}%
2539         {\@dblfloat{table}}%
2540         {\end@dblfloat}

```

## 8.6 キャプション

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

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

```

2541 \newlength\abovecaptionskip
2542 \newlength\belowcaptionskip
2543 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2544 \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。

```

2545 %<!*jspf>
2546 % \long\def\@makecaption#1#2{{\small
2547 %   \advance\leftskip10\jsc@mmm
2548 %   \advance\rightskip10\jsc@mmm
2549 %   \vskip\abovecaptionskip
2550 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2551 %   \ifdim \wd\@tempboxa >\hsize
2552 %     #1\hskip1\jsZw\relax #2\par
2553 %   \else
2554 %     \global \minipagefalse
2555 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2556 %   \fi
2557 %   \vskip\belowcaptionskip}}
2558 \long\def\@makecaption#1#2{{\small
2559   \advance\leftskip .0628\linewidth
2560   \advance\rightskip .0628\linewidth
2561   \vskip\abovecaptionskip
2562   \sbox\@tempboxa{#1\zwspace#2}%
2563   \ifdim \wd\@tempboxa <\hsize \centering \fi
2564   #1\zwspace#2\par
2565   \vskip\belowcaptionskip}}
2566 %</!*jspf>
2567 %<!*jspf>
2568 \long\def\@makecaption#1#2{%

```

```

2569 \vskip\abovecaptionskip
2570 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2571 \ifdim \wd\@tempboxa >\hsize
2572     {\small\sffamily
2573         \list{#1}{%
2574             \renewcommand{\makelabel}[1]{##1\hfil}
2575             \itemsep \z@
2576             \itemindent \z@
2577             \labelsep \z@
2578             \labelwidth 11\jsc@mmm
2579             \listparindent\z@
2580             \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2581 \else
2582     \global \@minipagefalse
2583     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2584 \fi
2585 \vskip\belowcaptionskip}
2586 %</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。


---


2587 \newif\if@jsc@warnoldfontcmd
2588 \@jsc@warnoldfontcmdtrue
2589 \newif\if@jsc@warnoldfontcmdexception
2590 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand
2591 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2592     \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2593     \DeclareOldFontCommand{#1}{%
2594         \bxjs@oldfontcmd{#1}#2%
2595     }{%

```

```

2596 \bxjs@oldfontcmd{#1}#3%
2597 }%
2598 }
2599 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2600 \ClassInfo\bxjs@clsname
2601 {Old font command '\string#1' is used!!\MessageBreak
2602 The first occurrence is}%
2603 }

```

---

`\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。

```

2604 \newcommand*{\allowoldfontcommands}{%
2605 \jsc@warnoldfontcmdexceptiontrue}
2606 \newcommand*{\disallowoldfontcommands}{%
2607 \jsc@warnoldfontcmdexceptionfalse}

2608 \let\bxjs@oldfontcmd@list\@empty
2609 \def\bxjs@oldfontcmd#1{%
2610 \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2611 \def\bxjs@oldfontcmd@a#1#2{%
2612 \if@jsc@warnoldfontcmdexception\else
2613 \global\@jsc@warnoldfontcmdfalse
2614 \ifx#1\relax
2615 \global\let#1=t%
2616 \jsc@warnoldfontcmd{#2}%
2617 \fi
2618 \fi}
2619 \def\bxjs@warnoldfontcmd@final{%
2620 % \par
2621 \global\let\bxjs@warnoldfontcmd@final\@empty
2622 \let\@tempa\@empty
2623 \def\do##1{%
2624 \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2625 \edef\@tempa{\@tempa \space\string##1}\fi}
2626 \bxjs@oldfontcmd@list
2627 \ifx\@tempa\@empty\else
2628 \ClassWarningNoLine\bxjs@clsname
2629 {Some old font commands were used in text:\MessageBreak
2630 \space\@tempa\MessageBreak
2631 You should note, that since 1994 LaTeX2e provides a\MessageBreak
2632 new font selection scheme called NFSS2 with several\MessageBreak
2633 new, combinable font commands. The
2634 class provides\MessageBreak
2635 the old font commands only for compatibility}
2636 \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。

```
2637 \def\bxjs@warnoldfontcmd@kick@final{%
2638   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2639 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}
```

---

`\mc` フォントファミリーを変更します。

```
\gt 2640 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2641 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2642 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2643 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2644 \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。

```
2645 \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。

```
2646 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2647 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2648 \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 2649 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2650 \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。

```
2651 \newcommand\@pnumwidth{1.55em}
2652 \newcommand\@tocrmarg{2.55em}
2653 \newcommand\@dotsep{4.5}
2654 %<!book&!report>\setcounter{tocdepth}{2}
2655 %<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)

```
2656 \newdimen\jsc@tocl@width
2657 \newcommand{\tableofcontents}{%
2658 %<*book|report>
2659 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2660 \settowidth\@tempdima{\headfont\appendixname}%
2661 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2662 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2663 \if@twocolumn
2664 \@restonecoltrue\onecolumn
2665 \else
2666 \@restonecolfalse
2667 \fi
2668 \chapter*{\contentsname}%
2669 \@mkboth{\contentsname}{%}
```



```

2670 %</book|report>
2671 %<*&book&!report>
2672 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2673 \settowidth\@tempdima{\headfont\appendixname}%
2674 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2675 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2676 \section*{\contentsname}%
2677 \@mkboth{\contentsname}{\contentsname}%
2678 %</!book&!report>
2679 \@starttoc{toc}%
2680 %<book|report> \if@restonecol\twocolumn\fi
2681 }

```

\l@part 部の目次です。

```

2682 \newcommand*{\l@part}[2]{%
2683 \ifnum \c@tocdepth >-2\relax
2684 %<!book&!report> \addpenalty\@secpenalty
2685 %<book|report> \addpenalty{-\@highpenalty}%
2686 \addvspace{2.25em \@plus\p@}%
2687 \begingroup
2688 \parindent \z@
2689 % \@pnumwidth should be \@tocrmarg
2690 % \rightskip \@pnumwidth
2691 \rightskip \@tocrmarg
2692 \parfillskip -\rightskip
2693 {\leavevmode
2694 \large \headfont
2695 \setlength\@lnumwidth{4\jsZw}%
2696 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2697 \nobreak
2698 %<book|report> \global\@nobreaktrue
2699 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2700 \endgroup
2701 \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)

```

2702 %<*&book|report>
2703 \newcommand*{\l@chapter}[2]{%
2704 \ifnum \c@tocdepth >\m@ne
2705 \addpenalty{-\@highpenalty}%
2706 \addvspace{1.0em \@plus\p@}%
2707 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2708 \begingroup
2709 \parindent\z@
2710 % \rightskip\@pnumwidth
2711 \rightskip\@tocrmarg
2712 \parfillskip-\rightskip

```

```

2713 \leavevmode\headfont
2714 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2715 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2716 \advance\leftskip\@lnumwidth \hskip-\leftskip
2717 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2718 \penalty\@highpenalty
2719 \endgroup
2720 \fi}
2721 %</book|report>

```

\l@section 節の目次です。

```

2722 %<*&!book&!report>
2723 \newcommand*{\l@section}[2]{%
2724 \ifnum \c@tocdepth >\z@
2725 \addpenalty{\@secpenalty}%
2726 \addvspace{1.0em \@plus\p@}%
2727 \begingroup
2728 \parindent\z@
2729 % \rightskip\@pnumwidth
2730 \rightskip\@tocrmarg
2731 \parfillskip-\rightskip
2732 \leavevmode\headfont
2733 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2734 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2735 \advance\leftskip\@lnumwidth \hskip-\leftskip
2736 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2737 \endgroup
2738 \fi}
2739 %</!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。

```

2740 %<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 2741 %<*&!book&!report>
2742 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2743 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2744 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2745 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2746 %
2747 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2748 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2749 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2750 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2751 %

```

```

2752 \newcommand*{\l@section}{%
2753     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2754     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2755 \newcommand*{\l@subsubsection}{%
2756     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2757     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2758 \newcommand*{\l@paragraph}{%
2759     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2760     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2761 \newcommand*{\l@subparagraph}{%
2762     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2763     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2764 %</!book&!report>
2765 %<*book|report>
2766 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2767 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2768 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2769 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2770 \newcommand*{\l@section}{%
2771     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2772     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2773 \newcommand*{\l@subsubsection}{%
2774     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2775     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2776 \newcommand*{\l@subsubsection}{%
2777     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2778     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2779 \newcommand*{\l@paragraph}{%
2780     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2781     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2782 \newcommand*{\l@subparagraph}{%
2783     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2784     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2785 %</book|report>

```

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

```

2786 \newdimen\@lnumwidth
2787 \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。

```
2788 \def\jsTocLine{\leaders\hbox{%
2789   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2790 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2791   \vskip \z@ \@plus.2\p@?
2792   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2793     \parindent #2\relax\@afterindenttrue
2794     \interlinepenalty\@M
2795     \leavevmode
2796     \@lnumwidth #3\relax
2797     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2798     {#4}\nobreak
2799     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2800       \hfil\normalfont \normalcolor #5}\par}\fi}
```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```
2801 \newcommand{\listoffigures}{%
2802   %<*book|report>
2803   \if@twocolumn\@restonecoltrue\onecolumn
2804   \else\@restonecolfalse\fi
2805   \chapter*{\listfigurename}%
2806   \@mkboth{\listfigurename}{}%
2807   %</book|report>
2808   %<!*book&!report>
2809   \section*{\listfigurename}%
2810   \@mkboth{\listfigurename}{\listfigurename}%
2811   %</!*book&!report>
2812   \@starttoc{lof}%
2813   %<book|report> \if@restonecol\twocolumn\fi
2814 }
```

`\l@figure` 図目次の項目を出力します。

```
2815 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```
2816 \newcommand{\listoftables}{%
2817   %<*book|report>
2818   \if@twocolumn\@restonecoltrue\onecolumn
2819   \else\@restonecolfalse\fi
2820   \chapter*{\listtablename}%
2821   \@mkboth{\listtablename}{}%
2822   %</book|report>
2823   %<!*book&!report>
2824   \section*{\listtablename}%
2825   \@mkboth{\listtablename}{\listtablename}%
```

```

2826 %</!book&!report>
2827 \starttoc{lot}%
2828 %<book|report> \if@restonecol\twocolumn\fi
2829 }

```

`\l@table` 表目次は図目次と同じです。

```

2830 \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。

```

2831 \newdimen\bibindent
2832 \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。

```

2833 \newenvironment{thebibliography}[1]{%
2834   \@jsc@warnoldfontcmdexceptiontrue
2835   \global\let\presectionname\relax
2836   \global\let\postsectionname\relax
2837   %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2838   %<*kiyou>
2839   \vspace{1.5\baselineskip}
2840   \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2841   \vspace{0.5\baselineskip}
2842   %</kiyou>
2843   %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2844   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2845   \list{\@biblabel{\@arabic{c@enumiv}}}%
2846     {\settowidth\labelwidth{\@biblabel{#1}}%
2847     \leftmargin\labelwidth
2848     \advance\leftmargin\labelsep
2849     \@openbib@code
2850     \usecounter{enumiv}%
2851     \let\p@enumiv\@empty
2852     \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2853   %<kiyou> \small
2854   \sloppy
2855   \clubpenalty4000
2856   \@clubpenalty\clubpenalty
2857   \widowpenalty4000%
2858   \sfcode`\.\@m}
2859   {\def\@noitemerr
2860     {\@latex@warning{Empty `thebibliography' environment}}}%
2861   \endlist}

```

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

```
2862 \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。

```
2863 \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。

```
2864 % \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。

```
2865 % \def\@citex[#1]#2{\leavevmode
2866 %   \let\@citea\empty
2867 %   \@cite{\@for\@citeb:=#2\do
2868 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2869 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2870 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2871 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2872 %       \G@refundefinedtrue
2873 %       \@latex@warning
2874 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2875 %       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2876 % \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。

```
2877 % \DeclareRobustCommand\cite{\unskip
2878 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2879 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempwa
2880 %   , \jsInhibitGlue\ #2\fi}}}$}
```

## 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)。

```
2881 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2882   \if@twocolumn
2883     \onecolumn\@restonecolfalse
2884   \else
2885     \clearpage\@restonecoltrue
2886   \fi
```

```

2887 \columnseprule.4pt \columnsep 2\jsZw
2888 \ifx\multicols\@undefined
2889 %<book|report> \twocolumn[\@makeschapterhead{\indexname}%
2890 %<book|report> \addcontentsline{toc}{chapter}{\indexname}}%
2891 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2892 %<!book&!report> \twocolumn[\section*{\indexname}]%
2893 \else
2894 \ifdim\textwidth<\fullwidth
2895 \setlength{\evensidemargin}{\oddsidemargin}
2896 \setlength{\textwidth}{\fullwidth}
2897 \setlength{\linewidth}{\fullwidth}
2898 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}%
2899 %<book|report> \addcontentsline{toc}{chapter}{\indexname}}%
2900 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2901 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]%
2902 \else
2903 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}%
2904 %<book|report> \addcontentsline{toc}{chapter}{\indexname}}%
2905 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2906 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]%
2907 \fi
2908 \fi
2909 %<book|report> \mkboth{\indexname}{}%
2910 %<!book&!report> \mkboth{\indexname}{\indexname}%
2911 \plainifnotempty % \thispagestyle{plain}
2912 \parindent\z@
2913 \parskip\z@ \@plus .3\p@?\relax
2914 \let\item\@idxitem
2915 \raggedright
2916 \footnotesize\narrowbaselines
2917 }{
2918 \ifx\multicols\@undefined
2919 \if@restonecol\onecolumn\fi
2920 \else
2921 \end{multicols}
2922 \fi
2923 \clearpage
2924 }

```

`\@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 2925 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2926 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2927 \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。

```

2928 \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。

```
2929 \newcommand\seename{\if@english see\else →\fi}  
2930 \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、`\inhibitglue` を入れることにします。p<sub>La</sub>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。

---

```
2931 \ifx\pltx@foot@penalty\@undefined  
2932 \let\footnotes@ve=\footnote  
2933 \def\footnote{\jsInhibitGlue\footnotes@ve}  
2934 \let\footnotemarks@ve=\footnotemark  
2935 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}  
2936 \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 p<sub>T</sub>E<sub>X</sub> 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 p<sub>La</sub>T<sub>E</sub>X の変更に追随しました (Thanks: 角藤さん)。p<sub>La</sub>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。

---

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

---

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

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

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

```
2937 \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。

```
2938 % \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。

```
2939 \renewcommand{\footnoterule}{%
2940   \kern-2.6\p@? \kern-.4\p@
2941   \hrule width .4\columnwidth
2942   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2943 %<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。

```
2944 \long\def\@footnotetext{%
2945   \insert\footins\bgroup
2946     \normalfont\footnotesize
2947     \interlinepenalty\interfootnotelinepenalty
2948     \splittopskip\footnotesep
2949     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2950     \hsize\columnwidth \@parboxrestore
2951     \protected@edef\@currentlabel{%
2952       \csname p@footnote\endcsname\@thefnmark
2953     }%
2954     \color@begingroup
2955     \makefnintext{%
2956       \rule\z@\footnotesep\ignorespaces}%
2957     \futurelet\jsc@next\jsc@fo@t}
2958 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@fo@t
2959               \else \let\jsc@next\jsc@fo@t\fi \jsc@next}
2960 \def\jsc@fo@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
2961 \def\jsc@fo@t#1{#1\jsc@@foot}
2962 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
2963   \ifx\pltx@foot@penalty\@undefined\else
2964     \ifhmode\null\fi
2965     \ifnum\pltx@foot@penalty=\z@\else
2966       \penalty\pltx@foot@penalty
2967       \pltx@foot@penalty\z@
2968     \fi
2969   \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。

```
2970 \newcommand\@makefntext[1]{%
2971   \advance\leftskip 3\jsZw
2972   \parindent 1\jsZw
2973   \noindent
2974   \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。

```
2975 % \def\@xfootnotenext[#1]{%
2976 %   \begingroup
2977 %     \ifnum#1>\z@
2978 %       \csname c@\@mpfn\endcsname #1\relax
2979 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2980 %     \else
2981 %       \unrestored@protected@xdef\@thefnmark{}%
2982 %     \fi
2983 %   \endgroup
2984 %   \@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。

```
2985 \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。

```
2986 \def\everyparhook{\jsInhibitGlueAtParTop}
2987 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2988 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
2989 \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。

---

```
2990 \ifx j\jsEngine
2991 \def\@inhibitglue{%
2992   \futurelet\@let@token\@@inhibitglue}
2993 \begingroup
2994 \let\GDEF=\gdef
2995 \let\CATCODE=\catcode
2996 \let\ENDGROUP=\endgroup
2997 \CATCODE`k=12
2998 \CATCODE`a=12
2999 \CATCODE`n=12
3000 \CATCODE`j=12
3001 \CATCODE`i=12
3002 \CATCODE`c=12
3003 \CATCODE`h=12
3004 \CATCODE`r=12
3005 \CATCODE`t=12
3006 \CATCODE`e=12
3007 \GDEF\KANJI@CHARACTER{kanji character }
3008 \ENDGROUP
3009 \def\@@inhibitglue{%
3010   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3011   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
3012   \def\jsc@ig@temp{#1}%
3013   \ifx\jsc@ig@temp\@empty
3014     \ifnum\the\inhibitxspcode`#2=2\relax
3015       \inhibitglue
3016     \fi
3017   \fi}
3018 \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。

---

```
3019 \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。

```
3020 \def\@doendpe{%
3021   \@endpetrue
3022   \def\par{%
3023     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3024   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] `minipage` 環境にも対策します。

```
3025 \def\@setminipage{%
3026   \@minipagetrue
3027   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3028 }
```

`\item` 命令の直後です。

```
3029 \def\@item[#1]{%
3030   \if@noparitem
3031     \@donoparitem
3032   \else
3033     \if@inlabel
3034       \indent \par
3035     \fi
3036     \ifhmode
3037       \unskip\unskip \par
3038     \fi
3039     \if@newlist
3040       \if@nobreak
3041         \@nbitem
3042       \else
3043         \addpenalty\@beginparpenalty
3044         \addvspace\@topsep
3045         \addvspace{-\parskip}%
3046       \fi
3047     \else
3048       \addpenalty\@itempenalty
3049       \addvspace\itemsep
3050     \fi
3051     \global\@inlabeltrue
3052   \fi
3053   \everypar{%
3054     \@minipagefalse
3055     \global\@newlistfalse
3056     \if@inlabel
3057       \global\@inlabelfalse
3058     {\setbox\z@\lastbox
```

```

3059      \ifvoid\z@
3060      \kern-\itemindent
3061      \fi}%
3062      \box\@labels
3063      \penalty\z@
3064      \fi
3065      \if@nobreak
3066      \@nobreakfalse
3067      \clubpenalty \@M
3068      \else
3069      \clubpenalty \@clubpenalty
3070      \everypar{\everyparhook}%
3071      \fi
3072      \everyparhook}%
3073      \if@noitemarg
3074      \@noitemargfalse
3075      \if@nmbrlist
3076      \refstepcounter\@listctr
3077      \fi
3078      \fi
3079      \sbox\@tempboxa{\makelabel{#1}}}%
3080      \global\setbox\@labels\hbox{%
3081      \unhbox\@labels
3082      \hskip \itemindent
3083      \hskip -\labelwidth
3084      \hskip -\labelsep
3085      \ifdim \wd\@tempboxa >\labelwidth
3086      \box\@tempboxa
3087      \else
3088      \hbox to\labelwidth {\unhbox\@tempboxa}%
3089      \fi
3090      \hskip \labelsep}%
3091      \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。

```

3092 \def\@afterheading{%
3093   \@nobreaktrue
3094   \everypar{%
3095     \if@nobreak
3096     \@nobreakfalse
3097     \clubpenalty \@M
3098     \if@afterindent \else
3099     {\setbox\z@\lastbox}%
3100     \fi
3101     \else
3102     \clubpenalty \@clubpenalty
3103     \everypar{\everyparhook}%
3104     \fi\everyparhook}}

```

---

「\everyparhook 用のパッチ処理」はここまで。

---

3105 \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。

---

```
3106 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3107 \AtEndOfPackage{%
3108 \def\@gnewline #1{%
3109   \ifvmode
3110     \@nolnerr
3111   \else
3112     \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
3113     \jsInhibitGlue \ignorespaces
3114   \fi}
3115 }
3116 \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。

---

```
3117 \if@jslogo
3118   \IfFileExists{jslogo.sty}{%
3119     \RequirePackage{jslogo}%
3120   }{%
3121     \ClassWarningNoLine\bxjs@clsname
3122       {The package 'jslogo' is not installed.\MessageBreak
3123       It is included in the recent release of\MessageBreak
```

```

3124         the 'jsclasses' bundle}
3125     }
3126 \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。

```

3127 %\let\ltx@ifnextchar\@ifnextchar
3128 %\def\ProvidesFile#1{%
3129 %   \begingroup
3130 %       \catcode`\ 10 %
3131 %       \ifnum \endlinechar<256 %
3132 %           \ifnum \endlinechar>\m@ne
3133 %               \catcode\endlinechar 10 %
3134 %       \fi
3135 %       \fi
3136 %       \@makeother\/%
3137 %       \@makeother\&%
3138 %       \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 3139 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3140 \newcommand{\postpartname}{\if@english\else 部\fi}
3141 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3142 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3143 \newcommand{\presectionname}{}% 第
\postsectionname 3144 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3145 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
3146 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
\listtablename 3147 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname
\indexname

```

```

3148 \newcommand{\refname}{\if@english References\else 参考文献\fi}
3149 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3150 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3151 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3152 %<jspf>\newcommand{\figurename}{Fig.~}
3153 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3154 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3155 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3156 \newcommand{\appendixname}{\if@english \else 付録\fi}
3157 %<!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。

---

```

3158 \@onlypreamble\bxjs@decl@Seireki@cmds
3159 \@tempswafalse
3160 \if p\jsEngine \@tempswatruetrue \fi
3161 \if n\jsEngine \@tempswatruetrue \fi
3162 \bxjs@cond\if@tempswa\fi{%
3163 % 欧文 8bitTeX の場合
3164 \newif\ifjsSeireki \jsSeirekitruetrue
3165 \def\bxjs@decl@Seireki@cmds{%
3166   \def\西暦{\jsSeirekitruetrue}%
3167   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3168 \def\Seireki{\jsSeirekitruetrue}
3169 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3170 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3171 \def\bxjs@iai{\noexpand~}
3172 }{%
3173 \newif\if西暦 \西暦 true
3174 \def\bxjs@decl@Seireki@cmds{%
3175   \def\西暦{\西暦 true}%
3176   \def\和暦{\西暦 false\bxjs@wareki@used}}
3177 \def\Seireki{\西暦 true}
3178 \def\Wareki{\西暦 false\bxjs@wareki@used}
3179 \def\bxjs@if@use@seireki{\bxjs@cond\if西暦\fi}
3180 \let\bxjs@iai\@empty
3181 }

```



```

3182 \bxjs@decl@Seireki@cmds
3183 \let\bxjs@unxp@\firstofone \let\bxjs@onxp@\firstofone
3184 \bxjs@test@engine\unexpanded{%
3185   \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。

```

3186 \newif\ifbxjs@bxwareki@avail
3187 \IfFileExists{bxwareki.sty}{%
3188   \RequirePackage{bxwareki}[]%
3189   \bxjs@bxwareki@availtrue}{%

```

\bxjs@wareki@used 和暦が非対応の場合に警告を出す。

```

3190 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\@empty
3191 \else
3192   \bxjs@robust@def\bxjs@wareki@used{%
3193     \global\let\bxjs@wareki@used\@empty
3194     \ClassWarning\bxjs@clsname
3195       {Wareki mode is not supported, since\MessageBreak
3196        'bxwareki' is unavailable, found}}
3197   \g@addto@macro\bxjs@begin@document@hook{%
3198     \let\bxjs@wareki@used\@empty}
3199 \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。

```

3200 \ifbxjs@bxwareki@avail
3201   \let\jayear\warekiyear
3202   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3203     \newcount\heisei \heisei=\value{warekiyear}
3204   \fi

```

bxwareki が使えない場合は西暦表示にする。

```

3205 \else
3206   \edef\jayear{\the\year \bxjs@iai}
3207 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3208 \let\bxjs@next\relax
3209 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3210   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3211 \fi\fi
3212 \edef\bxjs@today{%
3213   \if@english
3214     \ifcase\month\or
3215       January\or February\or March\or April\or May\or June\or
3216       July\or August\or September\or October\or November\or December\fi
3217     \space\number\day, \number\year
3218   \else

```

```

3219 \ifx\bxjs@next\relax \expandafter\@firstoftwo
3220 \else \noexpand\bxjs@if@use@seireki
3221 \fi {%
3222 \number\year\bxjs@iai\bxjs@unxp{年}%
3223 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3224 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3225 }{\bxjs@next}%
3226 \fi}
3227 \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。

```

3228 \g@addto@macro\bxjs@begin@document@hook{%
3229 \ifx\bbl@jpn@maybekansuji\@undefined\else
3230 \bxjs@decl@Seireki@cmds
3231 \g@addto@macro\datejapanese{%
3232 \let\today\bxjs@today}%
3233 \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)

```

3234 \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。

```

3235 %<slide>\pagestyle{empty}%
3236 %<article|report>\pagestyle{plain}%
3237 %<book>\pagestyle{headings}%
3238 \pagenumbering{arabic}
3239 \if@twocolumn
3240 \twocolumn
3241 \sloppy
3242 \flushbottom
3243 \else
3244 \onecolumn
3245 \raggedbottom
3246 \fi
3247 %<slide>
3248 \renewcommand\familydefault{\sfdefault}
3249 \raggedright
3250 %</slide>

```

■BXJS 独自の追加処理 

---

フックを実行する。

3251 \bxjs@pre@jadriver@hook

和文ドライバのファイルを読み込む。

3252 \input{bxjsja-\bxjs@jadriver.def}

おしまい。

3253 %</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。

```
3254 %<*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>TeX</sub>-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>TeX</sub> 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

```
3255 %<*minimal>
```

```
3256 %% 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.
```

```
3257 \def\DeclareJaTextFontCommand#1#2{%
```

```
3258   \DeclareRobustCommand#1[1]{%
```

```
3259     \relax
```

```
3260     \ifmmode \expandafter\nfss@text \fi
```

```
3261     {#2##1}}%
```

```
3262 }
```

```
\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。
```

```
3263 \def\DeclareJaMathFontCommand#1#2{%
```

```
3264   \DeclareRobustCommand#1[1]{%
```

```
3265     \relax
```

```
3266     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3267     \nfss@text{\fontfamily\familydefault
```

```
3268       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3269       #2##1}}%
```

```

3270 }%
3271 }

```

`\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。

```

3272 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3273 \@onlypreamble\bxjs@if@sf@default
3274 \def\bxjs@if@sf@default#1{%
3275   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3276   \g@addto@macro\bxjs@begin@document@hook{%
3277     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3278 }

```

`\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。

```

3279 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3280 \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。

```

3281 \def\jsLetHeadChar#1#2{%
3282   \begingroup
3283     \escapechar=`\ %
3284     \let\bxjs@tmpa={% brace-match-hack
3285       \bxjs@let@hchar@exp#2}%
3286   \endgroup
3287   \let#1\bxjs@g@tmpa}
3288 \def\bxjs@let@hchar@exp{%
3289   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3290 \def\bxjs@let@hchar@exp@a{%
3291   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3292   \bxjs@let@hchar@out\let\relax
3293   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3294   \bxjs@let@hchar@out\let\space%
3295   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3296   \bxjs@let@hchar@out\let\@backslashchar
3297   }{\bxjs@let@hchar@exp@b}}}%
3298 \def\bxjs@let@hchar@exp@b#1{%
3299   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3300 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3301   %\message{<#1#2>}%
3302   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3303 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3304 \bxjs@let@hchar@out\let\relax
3305 }{%else
3306 \expandafter\bxjs@let@hchar@exp
3307 }%
3308 }{%else
3309 \bxjs@let@hchar@chr#1%
3310 }}
3311 \def\bxjs@let@hchar@chr#1{%
3312 \bxjs@let@hchar@out\def{{#1}}}%
3313 \def\bxjs@let@hchar@out#1#2{%
3314 \global#1\bxjs@g@tmpa#2\relax
3315 \toks@{\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3316 \chardef\bxjs@let@hchar@csta=128
3317 \chardef\bxjs@let@hchar@cstb=192
3318 \chardef\bxjs@let@hchar@cstc=224
3319 \chardef\bxjs@let@hchar@cstd=240
3320 \chardef\bxjs@let@hchar@cste=248
3321 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3322 \def\bxjs@let@hchar@chr@ue#1{%
3323 \@tempcnta=`#1\relax
3324 %\message{\the\@tempcnta}%
3325 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3326 \bxjs@let@hchar@chr@ue@a#1%
3327 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3328 \bxjs@let@hchar@out\let\relax
3329 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3330 \bxjs@let@hchar@chr@ue@b
3331 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3332 \bxjs@let@hchar@chr@ue@c
3333 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3334 \bxjs@let@hchar@chr@ue@d
3335 }{%else
3336 \bxjs@let@hchar@out\let\relax
3337 }}}}
3338 \def\bxjs@let@hchar@chr@ue@a#1{%
3339 \bxjs@let@hchar@out\def{{#1}}}%
3340 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3341 \bxjs@let@hchar@out\def{{#1#2}}}%
3342 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3343 \bxjs@let@hchar@out\def{{#1#2#3}}}%
3344 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3345 \bxjs@let@hchar@out\def{{#1#2#3#4}}}%

```

## B.2 (u)pTeX 用の設定

```
3346 \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。

```
3347 \def\bxjs@let@hchar@chr@pp#1#2{%
3348   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3349 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3350   %\message{(\meaning#3:\meaning#4)}%
3351   \bxjs@cond\if#1k\fi{%
3352     \bxjs@let@hchar@out\def{#4}}%
3353   }{%else
3354     \bxjs@let@hchar@chr@ue#3#4%
3355   }}
3356 \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。

```
3357 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3358 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3359 \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。

```
3360 \@onlypreamble\bxjs@declarefontshape
3361 \ifjsWithupTeX
3362 \def\bxjs@declarefontshape{%
3363   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3364   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3365   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3366   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3367 }
3368 \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。

```
3369 \else
3370 \def\bxjs@declarefontshape{%
3371   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3372   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3373   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```



```

3374 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3375 }
3376 \def\bxjs@sizereference{jis}
3377 \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。

```

3378 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3379   \def\bxjs@tmpb{#5}}
3380 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3381 \else
3382 \expandafter\expandafter\expandafter\bxjs@next
3383 \expandafter\string\the\jfont\relax
3384 \fi
3385 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3386                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3387   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3388    \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。

```

3389 \begingroup
3390 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3391 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3392 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3393 % 幅が丁度 10pt なら補正は不要
3394 \ifdim\wd\z@=10pt
3395   \global\let\bxjs@scale\jsScale
3396 \else
3397 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3398   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3399   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3400   \bxjs@invscale\@tempdima\bxjs@tmpa
3401   \xdef\bxjs@scale{\strip@pt\@tempdima}
3402 \fi
3403 \endgroup
3404 %\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。

```

3405 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3406 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3407 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3408 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3409 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3410 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3411 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3412 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3413 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3414 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3415 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3416 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3417 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3418 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3419 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3420 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{-}
3421 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3422 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
3423 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
3424 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3425 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3426 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3427 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3428 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3429 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3430 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3431 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3432 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3433 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3434 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3435 \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。

```

3436 \ifx\@rmfamilyhook\@undefined % old
3437 \DeclareRobustCommand\rmfamily
3438 {\not@math@alphabet\rmfamily\mathrm
3439 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3440 \DeclareRobustCommand\sffamily
3441 {\not@math@alphabet\sffamily\mathsf
3442 \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3443 \DeclareRobustCommand\ttfamily
3444 {\not@math@alphabet\ttfamily\mathtt
3445 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3446 \g@addto@macro\bxjs@begin@document@hook{%
3447 \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3448 % my definitions above should have been overwritten, recover it!
3449 % \selectfont is executed twice but I don't care about speed...
3450 \expandafter\g@addto@macro\csname rmfamily \endcsname
3451 {\kanjifamily\mcdefault\selectfont}%
3452 \expandafter\g@addto@macro\csname sffamily \endcsname
3453 {\kanjifamily\gtdefault\selectfont}%

```

```

3454 \expandafter\g@addto@macro\csname ttfamily \endcsname
3455 {\kanjifamily\gtdefault\selectfont}%
3456 \fi}
3457 \else % 2020-02-02
3458 \g@addto@macro\rmfamilyhook
3459 {\prepare@family@series@update@kanji{mc}\mcdefault}
3460 \g@addto@macro\sffamilyhook
3461 {\prepare@family@series@update@kanji{gt}\gtdefault}
3462 \g@addto@macro\ttfamilyhook
3463 {\prepare@family@series@update@kanji{gt}\gtdefault}
3464 \fi
3465 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3466 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3467 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3468 \fi
3469 \bxjs@if@sf@default{%
3470 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3471 \selectfont

```

`\bxjs@parse@qh` の処理は pTeX 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）。

```

3472 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3473 \let\bxjs@parse@qh@a\undefined
3474 \let\bxjs@parse@qh@b\undefined

```

#### ■パラメタの設定

```

3475 \prebreakpenalty\jis"2147=10000
3476 \postbreakpenalty\jis"2148=10000
3477 \prebreakpenalty\jis"2149=10000
3478 \inhibitxspcode`!=1
3479 \inhibitxspcode`〒=2
3480 \xspcode`+=3
3481 \xspcode`%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3482 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3483 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3484 \let\jsInhibitGlueAtParTop\@inhibitglue

```

`\jsResetDimen` は空のままでよい。

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

```

3485 \begingroup
3486 \catcode`\!=0

```

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

```
3487 \gdef\bxjs@ptex@dir{%
3488   !iftdir t%
3489   !else!ifydir y%
3490   !else ?%
3491   !fi!fi}
```

新版の p<sub>T</sub>E<sub>X</sub> で脚注番号の周囲の空きが過大になる現象への対処。

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

```
3492 % 古い \@makefnmark の定義
3493 \long\def\bxjs@tmpa{\hbox{%
3494   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3495   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3496 \ifx\@makefnmark\bxjs@tmpa
3497 \long\gdef\@makefnmark{%
3498   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3499   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3500 \fi
3501 \endgroup
```

### B.3 pdf<sub>T</sub>E<sub>X</sub> 用の処理

```
3502 \else\ifx p\jsEngine
3503 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3504 \@onlypreamble\bxjs@cjk@loaded
3505 \def\bxjs@cjk@loaded{%
3506   \def\@footnotemark{%
3507     \leavevmode
3508     \ifhmode
3509       \edef\@x@sf{\the\spacefactor}%
3510       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3511         \unkern\unkern
3512         \ifdim\lastskip>\z@ \unskip \fi
3513       \fi\fi
3514       \nobreak
3515     \fi
3516     \@makefnmark
3517     \ifhmode \spacefactor\@x@sf \fi
3518     \relax}%
3519 \let\bxjs@cjk@loaded\relax
3520 }
3521 \g@addto@macro\bxjs@begin@document@hook{%
3522   \@ifpackageloaded{CJK}{%
3523     \bxjs@cjk@loaded
3524   }{}%
3525 }
```

## B.4 X<sub>Y</sub>TeX 用の処理

3526 \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。

```
3527 \def\bxjs@let@hchar@chr#1{%
3528   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3529   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3530     \bxjs@let@hchar@chr@xe
3531   }\bxjs@let@hchar@out\def{{#1}}}}
3532 \def\bxjs@let@hchar@chr@xe#1{%
3533   \lccode`0=`#1\relax
3534   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3535 \@onlypreamble\bxjs@do@precisetext
3536 \ifx\XeTeXgenerateactualtext\@undefined\else
3537   \def\bxjs@do@precisetext{%
3538     \XeTeXgenerateactualtext=\@ne}
3539 \fi
```

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

```
3540 \@onlypreamble\bxjs@do@simplejasetup
3541 \def\bxjs@do@simplejasetup{%
3542   \ifnum\XeTeXinterchartokenstate>\z@
3543   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3544     \jsSimpleJaSetup
3545     \ClassInfo\bxjs@clsname
3546     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3547   \fi\fi}
```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```
3548 \newcommand*{\jsSimpleJaSetup}{%
3549   \XeTeXlinebreaklocale "ja"\relax
3550   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3551   \XeTeXlinebreakpenalty=0\relax}
```

## B.5 後処理（エンジン共通）

3552 \fi\fi\fi

simplejasetup オプションの処理。

```
3553 \ifx\bxjs@do@simplejasetup\@undefined\else
3554   \g@addto@macro\bxjs@begin@document@hook{%
3555     \ifbxjs@simplejasetup
3556       \bxjs@do@simplejasetup
3557     \fi}
3558 \fi
```

precisetext オプションの処理。

```

3559 \ifbxjs@precisetext
3560   \ifx\bxjs@do@precisetext\@undefined
3561     \ClassWarning\bxjs@clsname
3562       {The current engine does not support the\MessageBreak
3563         'precise-text' option\@gobble}
3564   \else
3565     \bxjs@do@precisetext
3566   \fi
3567 \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。

```

3568 \g@addto@macro\bxjs@begin@document@hook{%
3569   \ifx\jsInhibitGlueAtParTop\@empty
3570     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3571     \ifx\everyparhook\bxjs@tmpa
3572       \let\everyparhook\@empty
3573     \fi
3574   \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。

```

3575 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3576   \let\bxjs@everypar\everypar
3577   \newtoks\everypar
3578   \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。

```

3579   \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3580 \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。

```

3581 \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。

```

3582 \@onlypreamble\bxjs@adjust@fancyhdr
3583 \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。

```

3584 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3585 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3586 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3587 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3588 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3589 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3590 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3591 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3592 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3593 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3594 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3595 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3596 \def\bxjs@tmpa{\rm\thepage\strut}%
3597 \def\bxjs@tmpb{\thepage\strut}%
3598 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3599 \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。

```

3600 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3601   \setlength{\@tempdima}{\fullwidth-\textwidth}%
3602   \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3603   }\bxjs@tmpa
3604 \fi\fi
3605 \PackageInfo\bxjs@clsname
3606 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3607 \def\bxjs@pagestyle@hook{%
3608   \@ifpackageloaded{fancyhdr}{%
3609     \bxjs@adjust@fancyhdr
3610     \global\let\bxjs@adjust@fancyhdr\relax
3611   }{}}

```

\pagestyle にフックを入れ込む。

```

3612 \let\bxjs@org@pagestyle\pagestyle
3613 \def\pagestyle{%
3614   \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。

```

3615 \g@addto@macro\bxjs@begin@document@hook{%
3616   \bxjs@pagestyle@hook
3617   \global\let\bxjs@pagestyle@hook\relax}

```

## ■和文空白命令

```

3618 \ifbxjs@jaspace@cmd

```

`\jaenspace` 半角幅の水平空き。

```
3619 \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。

```
3620 \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。

```
3621 \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3622 \DeclareRobustCommand*{\jaspace}[1]{%
3623   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3624     \ClassError\bxjs@clsname
3625     {Unknown jaspaces: #1}{\@eha}%
3626   \else
3627     \csname bxjs@jaspace@@#1\endcsname
3628   \fi}
3629 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3630 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3631 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3632 \fi
```

終わり。

```
3633 \fi
```

以上で終わり。

```
3634 %</minimal>
```

## 付録 C 和文ドライバ：standard

標準のドライバ。

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

■和文フォント指定の扱い `standard` 和文ドライバでは `\jsJaFont` の値を和文フォントの“プリセット”の指定として用いる。プリセットの値は、`TeX Live` の `kanji-config-`



updmap コマンドで使う“ファミリ”と同じにすることを想定する。特別な値として、auto は kanji-config-updmap で現在指定されているファミリを表す。

## C.1 共通処理 (1)

まず minimal ドライバを読み込む。

```
3635 %<*standard>
3636 %% このファイルは日本語文字を含みます
3637 \input{bxjsja-minimal.def}
```

simplejasetup は standard では無効になる。

```
3638 \bxjs@simplejasetupfalse
```

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

\ifbxjs@jp@jismmiv 2004JIS 字形を優先させるか。

```
3639 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。

3640 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3641 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3642 \define@key{bxjsStd}{jis2004}[true]{%
3643   \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。

```
3644 \newif\ifbxjs@jp@units

units オプションの処理。

3645 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3646 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3647 \define@key{bxjsStd}{units}[true]{%
3648   \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。

```
3649 \let\bxjs@jp@font\@empty

font オプションの処理。

3650 \define@key{bxjsStd}{font}{%
3651   \edef\bxjs@jp@font{#1}}

実際の japaram の値を適用する。

3652 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3653 \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。

```

3654 \ifbxjs@jp@jismmiv
3655   \g@addto@macro\@classoptionslist{,jis2004}
3656 % \@ifpackagewith 判定への対策
3657   \PassOptionsToPackage{jis2004}{otf}
3658 \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。

```

3659 \if j\jsEngine
3660   \bxjs@jp@unitsfalse
3661 \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。

```

3662 \ifbxjs@jp@units
3663   \IfFileExists{bxcalc.sty}{%
3664     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3665     \ifx\usepTeXunits\undefined
3666       \PackageWarningNoLine\bxjs@clsname
3667         {Cannot support pTeX units (zw etc.), since\MessageBreak
3668           the package 'bxcalc' is too old}%
3669       \bxjs@jp@unitsfalse
3670     \else \usepTeXunits
3671     \fi
3672   }{%else
3673     \PackageWarningNoLine\bxjs@clsname
3674       {Cannot support pTeX units (zw etc.), since\MessageBreak
3675         the package 'bxcalc' is unavailable}%
3676     \bxjs@jp@unitsfalse
3677   }
3678 \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。

```

3679 \ifbxjs@jp@units
3680 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3681 \let\bxjs@parse@qh@a\undefined
3682 \let\bxjs@parse@qh@b\undefined
3683 \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。

```

3684 \ifbxjs@jp@units
3685   \def\bxjs@let@lenexpr#1#2{%
3686     \edef#1{#2}%
3687     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3688 \else
3689   \def\bxjs@let@lenexpr{\edef}
3690 \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。

```
3691 \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。

```
\jH3692 \@tempdima=0.25mm
3693 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3694 \let\jH\jQ
```

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

```
\trueH3695 \ifjsc@mag
3696 \@tempdimb=\jsBaseFontSize\relax
3697 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3698 \@tempdima=2.5mm
3699 \bxjs@invscale\@tempdima\bxjs@tmpa
3700 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3701 \@tempdima=10pt
3702 \bxjs@invscale\@tempdima\bxjs@tmpa
3703 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3704 \else \let\trueQ\jQ \let\bxjs@truept\p@
3705 \fi
3706 \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。

```
3707 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3708 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3709 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3710 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3711 \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。

```
3712 \def\jafontsize#1#2{%
3713 \begingroup
3714 \bxjs@jafontsize@a{#1}%
3715 \@tempdimb\jsInverseScale\@tempdima
3716 \bxjs@jafontsize@a{#2}%
3717 \xdef\bxjs@g@tmpa{%
3718 \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3719 \endgroup\bxjs@g@tmpa}
3720 \def\bxjs@jafontsize@a#1{%
```

```

3721 \bxjs@parse@qh{#1}%
3722 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3723 \@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。

```
3724 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```

3725 \newcommand*\setkanjiskip[1]{%
3726 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3727 \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3728 \newcommand*\getkanjiskip{%
3729 \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。

```
3730 \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 3731 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3732 \bxjs@kanjiskip@enabledtrue
3733 \bxjs@reset@kanjiskip}
3734 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3735 \bxjs@kanjiskip@enabledfalse
3736 \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。

```

3737 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3738 \ifbxjs@kanjiskip@enabled
3739 \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3740 \else \@tempskipa\z@
3741 \fi
3742 \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 3743 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3744 \newcommand*\setxkanjiskip[1]{%
```

```

3745 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3746 \bxjs@reset@xkanjiskip}

```

```
\bxjs@enable@xkanjiskip 3747 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3748 \bxjs@xkanjiskip}
```

```
3749 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3750 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3751 \bxjs@xkanjiskip@enabledtrue
```

```
3752 \bxjs@reset@xkanjiskip}
```

```

3753 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3754   \bxjs@xkanjiskip@enabledfalse
3755   \bxjs@reset@xkanjiskip}
3756 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3757   \ifbxjs@xkanjiskip@enabled
3758     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3759   \else \@tempskipa\z@
3760   \fi
3761   \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。

```

3762 \g@addto@macro\jsResetDimen{%
3763   \bxjs@reset@kanjiskip
3764   \bxjs@reset@xkanjiskip}
3765 \let\bxjs@apply@kanjiskip\relax
3766 \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。

```

3767 \@onlypreamble\bxjs@adjust@jafont
3768 \def\bxjs@adjust@jafont#1{%
3769   \ifx\jsJaFont\bxjs@@auto
3770     \bxjs@get@kanjiEmbed
3771     \ifx\bxjs@kanjiEmbed\relax
3772       \let\bxjs@tmpa\@empty
3773     \else
3774       \let\bxjs@tmpa\bxjs@kanjiEmbed
3775     \fi
3776   \else
3777     \let\bxjs@tmpa\jsJaFont
3778   \fi
3779   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3780     \ClassWarningNoLine\bxjs@clsname
3781     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3782      not available on the current situation}%
3783     \let\bxjs@tmpa\@empty
3784   \fi\fi
3785 }
3786 \def\bxjs@@auto{auto}
3787 \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。

```

3788 \let\bxjs@kanjiEmbed\relax

```

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

```
3789 \@onlypreamble\bxjs@get@kanjiEmbed
3790 \def\bxjs@get@kanjiEmbed{%
3791   \begingroup\setbox\z@=\hbox{%
3792     \global\let\bxjs@g@tmpa\relax
3793     \endlinechar\m@ne
3794     \let\do\@makeother\dospecials
3795     \catcode32=10 \catcode12=10 %form-feed
3796     \let\bxjs@tmpa\@empty
3797     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3798     \ifeof\@inputcheck\else
3799       \read\@inputcheck to\bxjs@tmpa
3800       \closein\@inputcheck
3801     \fi
3802     \ifx\bxjs@tmpa\@empty\else
3803       \openin\@inputcheck="\bxjs@tmpa"\relax
3804       \@tempswatrue
3805       \loop\if@tempswa
3806         \read\@inputcheck to\bxjs@tmpa
3807         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3808         \ifx\bxjs@tmpb\relax\else
3809           \global\let\bxjs@g@tmpa\bxjs@tmpb
3810           \@tempswafalse
3811         \fi
3812         \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3813         \ifx\bxjs@tmpb\relax\else
3814           \global\let\bxjs@g@tmpa\bxjs@tmpb
3815           \@tempswafalse
3816         \fi
3817         \ifeof\@inputcheck \@tempswafalse \fi
3818       \repeat
3819     \fi
3820   }\endgroup
3821   \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3822 }
3823 \@onlypreamble\bxjs@get@ke@a
3824 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3825   \ifx$#1$\def\bxjs@tmpb{#2}%
3826   \else \let\bxjs@tmpb\relax
3827   \fi}
3828 \@onlypreamble\bxjs@get@ke@b
3829 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3830   \ifx$#1$\def\bxjs@tmpb{#2}%
3831   \else \let\bxjs@tmpb\relax
3832   \fi}
```

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

```
3833 \newcommand*\jachar[1]{%
3834   \begingroup
```

\jsLetHeadChar で先頭の“文字”を拾ってそれを \bxjs@jachar に渡す。

```
3835 \jsLetHeadChar\bxjs@tmpa{#1}%
3836 \ifx\bxjs@tmpa\relax
3837 \ClassWarningNoLine\bxjs@clsname
3838 {Illegal argument given to \string\jachar}%
3839 \else
3840 \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3841 \fi
3842 \endgroup}
```

\jsJaChar を \jachar と等価にする。

```
3843 \let\jsJaChar\jachar
```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```
3844 \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。

```
3845 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3846 \@onlypreamble\bxjs@fix@hyperref@unicode
3847 \def\bxjs@fix@hyperref@unicode#1{%
3848 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3849 \@namedef{KV@Hyp@bxjs/hook}##1{%
3850 \KV@Hyp@unicode{##1}%
3851 \def\KV@Hyp@unicode####1{%
3852 \expandafter\ifx\csname if##1\expandafter\endcsname
3853 \csname if####1\endcsname\else
3854 \ClassWarningNoLine\bxjs@clsname
3855 {Blcoked hyperref option 'unicode=####1'}%
3856 \fi
3857 }%
3858 }%
3859 }
```

\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。

```
3860 \@onlypreamble\jsCheckHyperrefUnicode
3861 \let\jsCheckHyperrefUnicode\@empty
3862 \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。

```
3863 \@onlypreamble\bxjs@check@hyperref@unicode
3864 \def\bxjs@check@hyperref@unicode#1{%
3865 \g@addto@macro\jsCheckHyperrefUnicode{%
3866 \@tempwafalse
3867 \begingroup
3868 \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3869 \aftergroup\@tempwatruue \fi
```

```

3870 \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3871 \csname if#1\endcsname
3872 \aftergroup\@tempwattrue \fi
3873 \endgroup
3874 \if@tempswa\else
3875 \ClassError\bxjs@clsname
3876 {The value of hyperref 'unicode' key is not suitable\MessageBreak
3877 for the present engine (must be #1)}%
3878 {\@ehc}%
3879 \fi}}

```

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

```

3880 \@onlypreamble\bxjs@urgent@special
3881 \def\bxjs@urgent@special#1{%
3882 \AtBeginDvi{\special{#1}}%
3883 \g@addto@macro\bxjs@begin@document@hook{%
3884 \@ifpackageloaded{atbegshi}{%
3885 \begingroup
3886 \toks\z@{\special{#1}}%
3887 \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3888 \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3889 \endgroup
3890 }{}%
3891 }%
3892 }

```

`\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。

```

3893 \@onlypreamble\bxjs@resolve@jafont@paren
3894 \def\bxjs@resolve@jafont@paren#1{%
3895 \def\bxjs@tmpb{\let#1}%
3896 \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3897 \@onlypreamble\bxjs@resolve@jafont@paren@a
3898 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3899 \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3900 \else
3901 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3902 \bxjs@tmpb\bxjs@tmpa
3903 \fi}

```

## C.2 pTeX 用設定

```
3904 \if j\jsEngine
```

### ■ 共通命令の実装

```

3905 \def\bxjs@apply@kanjiskip{%
3906 \kanjiskip\@tempskipa}
3907 \def\bxjs@apply@xkanjiskip{%
3908 \xkanjiskip\@tempskipa}

```



`\jaJaChar` のサブマクロ。

```
3909 \def\bxjs@jachar#1{%
3910   \bxjs@jachar@a#1...\@nil}
3911 \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。

```
3912   \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。

```
3913   \else\ifx.#3%
3914     \@tempcnta`#1 \multiply\@tempcnta64
3915     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3916     \bxjs@jachar@b
3917   \else\ifx.#4%
3918     \@tempcnta`#1 \multiply\@tempcnta64
3919     \advance\@tempcnta`#2 \multiply\@tempcnta64
3920     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3921     \bxjs@jachar@b
3922   \else
3923     \@tempcnta`#1 \multiply\@tempcnta64
3924     \advance\@tempcnta`#2 \multiply\@tempcnta64
3925     \advance\@tempcnta`#3 \multiply\@tempcnta64
3926     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3927     \bxjs@jachar@b
3928   \fi\fi\fi}
```

符号値が `\@tempcnta` の和文文字を出力する処理。

```
3929 \ifjsWithupTeX
3930   \def\bxjs@jachar@b{\kchar\@tempcnta}
3931 \else
3932   \def\bxjs@jachar@b{%
3933     \ifx\bxUInt\@undefined\else
3934       \bxUInt{\@tempcnta}%
3935     \fi}
3936 \fi
```

和欧文間空白の命令 `\jathinspace` の実装。

```
3937 \ifbxjs@jaspace@cmd
3938   \def\jathinspace{\hskip\xkanjiskip}
3939 \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。

```
3940 \ifbxjs@jp@jismmiv
3941   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3942   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3943 \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。

```

3944 \let\bxjs@jafont@paren\@firstofone
3945 \let\bxjs@tmpa\jsJaFont
3946 \ifx\bxjs@tmpa\bxjs@@auto
3947   \let\bxjs@tmpa\@empty
3948 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3949   \def\bxjs@tmpa{noembed}
3950 \fi\fi
3951 \bxjs@resolve@jafont@paren\bxjs@tmpa
3952 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3953 \ifx\bxjs@tmpa\@empty\else
3954   \edef\bxjs@next{%
3955     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3956   }\bxjs@next
3957 \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……。)

```

3958 \begingroup
3959   \global\let\bxjs@g@tmpa\relax
3960   \catcode`\|=0 \catcode`\|=12
3961   |def|bxjs@tmpdo#1|@nil{%
3962     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
3963   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
3964     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
3965   |catcode`\|=0 \catcode`\|=12
3966   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
3967     \ifx$#2$\else
3968       \xdef\bxjs@g@tmpa{%
3969         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3970     \fi}
3971 \@firstofone{%
3972   \catcode10=12 \endlinechar\m@ne
3973   \let\do\@makeother \dospecials \catcode32=10
3974   \openin\@inputcheck=otf.sty\relax
3975   \@tempswatrue
3976   \loop\if@tempswa
3977     \ifeof\@inputcheck \@tempswafalse \fi
3978     \if@tempswa
3979       \read\@inputcheck to\bxjs@next

```

```

3980 \expandafter\bxjs@tmpdo\bxjs@next\@nil
3981 \fi
3982 \repeat
3983 \closein\@inputcheck
3984 \endgroup}
3985 \bxjs@g@tmpa

```

■hyperref 対策 unicode にはいけない。

```

3986 \ifbxjs@hyperref@enc
3987 \bxjs@check@hyperref@unicode{false}
3988 \fi

```

tounicode special 命令を出力する。

```

3989 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3990 \else\ifjsWithpTeXng T\else F\fi\fi T%
3991 \ifnum\jis"2121="A1A1 %euc
3992 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3993 \else\ifnum\jis"2121="8140 %sjis
3994 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3995 \else\ifnum\jis"2121="3000 %uptex
3996 \ifbxjs@bigcode
3997 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3998 \PassOptionsToPackage{bigcode}{pxjahyper}
3999 \else
4000 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4001 \fi
4002 \fi\fi\fi
4003 \let\bxToUnicodeSpecialDone=t
4004 \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。

```

4005 \ifx f\bxjs@enablejfam\else
4006 \@enablejfamtrue
4007 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

4008 \if@enablejfam
4009 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4010 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4011 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4012 \jfam\symmincho
4013 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4014 \g@addto@macro\bxjs@begin@document@hook{%
4015 \ifx\reDeclareMathAlphabet\undefined\else
4016 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4017 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4018 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4019 \fi}

```

4020 \fi

### C.3 pdf<sub>TEX</sub> 用設定 : CJK + bxcjkatype

4021 \else\if p\jsEngine

■bxcjkatype パッケージの読み込 \jsJaFont が指定されている場合は、その値を bxcjkatype のオプション（プリセット指定）に渡す。（auto ならば \bxjs@get@kanjiEmbed を実行する。）スケール値（\jsScale）の反映は bxcjkatype の側で行われる。

※Pandoc モードでは autotilde を指定しない。

```
4022 \bxjs@adjust@jafont{f}
4023 \let\bxjs@jafont@paren\@firstofone
4024 \bxjs@resolve@jafont@paren\bxjs@tmpa
4025 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4026 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4027 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4028   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4029 \fi
4030 \edef\bxjs@next{%
4031   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4032 }\bxjs@next
4033 \bxjs@cjk@loaded
```

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

※取りあえず固定はしない。

```
4034 \ifbxjs@hyperref@enc
4035   \PassOptionsToPackage{unicode}{hyperref}
4036 \fi
```

\hypersetup 命令で（CJK\* 環境に入れなくても）日本語文字を含む文書情報を設定できるようにするための細工。

※bxcjkatype を whole 付きで使っていることが前提。

※パッケージオプションでの指定に対応するのは、「アクティブな高位バイトトークンがその場で展開されてしまう」ため困難である。

```
4037 \ifx\bxchkatypeHyperrefPatchDone\@undefined
4038 \begingroup
4039   \CJK@input{UTF8.bdg}
4040 \endgroup
4041 \g@addto@macro\pdfstringdefPreHook{%
4042   \@nameuse{CJK@UTF8Binding}%
4043 }
4044 \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。

```
4045 \ifx\bxchkatypeHyperrefPatchDone\@undefined
4046 \g@addto@macro\pdfstringdefPreHook{%
```

```

4047 \ifx~\bxjs@@CJKtilde
4048 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4049 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4050 \let~\@empty
4051 \fi
4052 }
4053 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4054 \def\bxjs@@tildecmd{~}
4055 \def\bxjs@LetUnexpandableSpace#1{%
4056 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4057 \bxjs@org@LetUnexpandableSpace#1%
4058 \fi}
4059 \fi

```

### ■ 共通命令の実装

```

4060 \newskip\jsKanjiSkip
4061 \newskip\jsXKanjiSkip
4062 \ifx\CJKecglue\@undefined
4063 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4064 \fi
4065 \let\autospacing\bxjs@enable@kanjiskip
4066 \let\noautospacing\bxjs@disable@kanjiskip
4067 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4068 \def\bxjs@apply@kanjiskip{%
4069 \jsKanjiSkip\@tempskipa
4070 \let\CJKglue\bxjs@CJKglue}
4071 \let\autoxspacing\bxjs@enable@xkanjiskip
4072 \let\noautoxspacing\bxjs@disable@xkanjiskip
4073 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4074 \def\bxjs@apply@xkanjiskip{%
4075 \jsXKanjiSkip\@tempskipa
4076 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4077 \def\bxjs@jachar#1{%
4078 \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4079 \ifbxjs@jaspace@cmd
4080 \protected\def\jathinspace{\CJKecglue}
4081 \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。

```

4082 \ifx t\bxjs@enablejfam
4083 \ClassWarningNoLine\bxjs@clsname
4084 {You cannot use 'enablejfam=true', since the\MessageBreak
4085 CJK package does not support Japanese math}
4086 \fi

```

## C.4 X<sub>Y</sub>TeX 用設定 : xeCJK + zxjatype

```
4087 \else\if x\jsEngine
```

■zxjatype パッケージの読込 スケール値 (\jsScale) の反映は zxjatype の側で行われる。

```
4088 \RequirePackage{zxjatype}
4089 \PassOptionsToPackage{no-math}{fontspec}%!
4090 \PassOptionsToPackage{xetex}{graphicx}%!
4091 \PassOptionsToPackage{xetex}{graphics}%!
4092 \ifx\zxJaFamilyName\@undefined
4093   \ClassError{bxjs}{clsname
4094     {xeCJK or zxjatype is too old}}\@ehc
4095 \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。

```
4096 \bxjs@adjust@jafont{f}
4097 \let\bxjs@jafont@paren\@gobble
4098 \bxjs@resolve@jafont@paren\bxjs@tmpa
4099 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4100 \ifx\bxjs@tmpa\@empty
4101   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4102   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4103 \else
4104   \edef\bxjs@next{%
4105     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4106   }\bxjs@next
4107 \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。

```
4108 \ifnum\stricmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4109   \ifbxjs@hyperref@enc
4110     \PassOptionsToPackage{unicode}{hyperref}
```

```
4111 \fi
4112 \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。

```
4113 \AtEndOfPackage{%
4114 \def\@gnewline #1{%
4115 \ifvmode \@nolnerr
4116 \else
4117 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4118 \nobreak \hskip-1sp\hskip1sp\relax
4119 \ignorespaces
4120 \fi}
4121 }
```

#### ■共通命令の実装

```
4122 \newskip\jsKanjiSkip
4123 \newskip\jsXKanjiSkip
4124 \ifx\CJKecglue\@undefined
4125 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4126 \fi
4127 \let\autospacing\bxjs@enable@kanjiskip
4128 \let\noautospacing\bxjs@disable@kanjiskip
4129 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4130 \def\bxjs@apply@kanjiskip{%
4131 \jsKanjiSkip\@tempskipa
4132 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4133 \let\autoxspacing\bxjs@enable@xkanjiskip
4134 \let\noautoxspacing\bxjs@disable@xkanjiskip
4135 \protected\def\bxjs@CJKeclue{\hskip\jsXKanjiSkip}
4136 \def\bxjs@apply@xkanjiskip{%
4137 \jsXKanjiSkip\@tempskipa
4138 \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。

```
4139 \ifx\mcfamily\@undefined
4140 \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4141 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4142 \fi
```

\jachar のサブマクロの実装。

```
4143 \def\bxjs@jachar#1{%
4144 \xeCJKDeclareCharClass{CJK}{`#1}\relax
```

```
4145 #1}
```

`\jathinspace` の実装。

```
4146 \ifbxjs@jaspace@cmd
```

```
4147 \protected\def\jathinspace{\CJKecglue}
```

```
4148 \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。

```
4149 \ifx t\bxjs@enablejfam
```

```
4150 \@enablejfamtrue
```

```
4151 \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。

```
4152 \if@enablejfam
```

```
4153 \xeCJKsetup{CJKmath=true}
```

```
4154 \fi
```

## C.5 Lua<sub>T</sub><sub>E</sub>X 用設定 : Lua<sub>T</sub><sub>E</sub>X-ja

```
4155 \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。

```
4156 \let\zw\@undefined
```

```
4157 \RequirePackage{luatexja}
```

```
4158 \edef\bxjs@next{%
```

```
4159 \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
```

```
4160 }\bxjs@next
```

フォント代替の明示的定義。

```
4161 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{}
```

```
4162 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{}
```

```
4163 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{}
```

```
4164 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{}
```

```
4165 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{}
```

```
4166 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{}
```

```
4167 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{}
```

```
4168 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{}
```

```
4169 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{}
```

```
4170 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{}
```

```
4171 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{}
```



```

4172 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4173 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4174 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4175 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4176 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4177 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4178 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4179 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4180 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4181 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4182 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4183 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4184 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4185 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4186 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4187 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4188 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4189 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4190 \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。

```

4191 \bxjs@adjust@jafont{t}
4192 \ifx\bxjs@tmpa\bxjs@@noEmbed
4193   \def\bxjs@tmpa{noembed}
4194 \fi
4195 \let\bxjs@jafont@paren\@gobble
4196 \bxjs@resolve@jafont@paren\bxjs@tmpa
4197 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4198 \ifx\bxjs@tmpa\@empty
4199   \defaultjfontfeatures{ Kerning=Off }
4200   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4201   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4202 \else
4203   \edef\bxjs@next{%
4204     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4205   }\bxjs@next
4206 \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。

```

4207 \DeclareRobustCommand\rmfamily
4208   {\not@math@alphabet\rmfamily\mathrm
4209     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}

```

```

4210 \DeclareRobustCommand\sffamily
4211   {\not@math@alphabet\sffamily\mathsf
4212     \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
4213 \DeclareRobustCommand\ttfamily
4214   {\not@math@alphabet\ttfamily\mathtt
4215     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4216 \g@addto@macro\bxjs@begin@document@hook{%
4217   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4218   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4219   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4220 \bxjs@if@sfddefault{%
4221   \renewcommand\kanjifamilydefault{\gtdefault}}

```

#### ■和文パラメタの設定

```

4222 % 次の 3 つは既定値の通り
4223 \ltjsetparameter{prebreakpenalty={`',10000}}
4224 %\ltjsetparameter{postbreakpenalty={`",10000}}
4225 %\ltjsetparameter{prebreakpenalty={`",10000}}
4226 \ltjsetparameter{jaxspmode={`!,1}}
4227 \ltjsetparameter{jaxspmode={`〒,2}}
4228 \ltjsetparameter{alxspmode={`+,3}}
4229 \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。

```

4230 \ifx\ltjfakeparbegin\@undefined
4231   \protected\def\ltjfakeparbegin{%
4232     \ifhmode
4233       \relax\directlua{%
4234         luatexja.jfmglue.create_beginpar_node()}}
4235   \fi}
4236 \fi

```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```

4237 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4238 \begingroup
4239   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4240   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4241   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4242     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4243   \directlua{
4244     local function patchcmd(cs, code, from, to)
4245       tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4246         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."")
4247     end

```

```

4248   patchcmd(\@xsect, [[\meaning\xsect]],
4249     \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4250   patchcmd(\@item, [[\meaning@item]],
4251     \bxjs@tmpa, \@ltjfakeparbegin)}
4252 \endgroup
4253 \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。

```

4254 \ifbxjs@hyperref@enc
4255   \PassOptionsToPackage{unicode}{hyperref}
4256   \bxjs@check@hyperref@unicode{true}
4257 \fi

```

■共通命令の実装

```

4258 \protected\def\autospacing{%
4259   \ltjsetparameter{autospacing=true}}
4260 \protected\def\noautospacing{%
4261   \ltjsetparameter{autospacing=false}}
4262 \protected\def\autoxspacing{%
4263   \ltjsetparameter{autoxspacing=true}}
4264 \protected\def\noautoxspacing{%
4265   \ltjsetparameter{autoxspacing=false}}
4266 \def\bxjs@apply@kanjiskip{%
4267   \ltjsetparameter{kanjiskip={\@tempskipa}}}
4268 \def\bxjs@apply@xkanjiskip{%
4269   \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4270 \def\bxjs@jachar#1{%
4271   \ltjjachar`#1\relax}

```

\jathinspace の実装。

```

4272 \ifbxjs@jaspace@cmd
4273   \protected\def\jathinspace{%
4274     \hskip\ltjgetparameter{xkanjiskip}\relax}
4275 \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。

```

4276 \ifx f\bxjs@enablejfam
4277   \ClassWarningNoLine\bxjs@clsname
4278     {You cannot use 'enablejfam=false', since the\MessageBreak
4279     LuaTeX-ja always provides Japanese math families}
4280 \fi

```

## C.6 共通処理 (2)

```
4281 \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 4282 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4283 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4284 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4285 \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。

```
4286 \ifx\mathmc\undefined
4287 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4288 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4289 \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)。

```
4290 \ifbxjs@jaspace@cmd
4291 \bxjs@protected\def\bxjs@choice@jathinspace{%
4292 \relax\ifmmode \mskip\medmuskip
4293 \else \jathinspace\ignorespaces
4294 \fi}
4295 \jsAtEndOfClass{%
4296 \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4297 \else \def\>{\protect\bxjs@choice@jathinspace}%
4298 \fi}
4299 \fi
```

### ■和文・和欧文間空白の初期値

```
4300 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4301 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4302 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4303 \fi
```

以上で終わり。

```
4304 %</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。

```
4305 %<*modern>
4306 \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。

```
4307 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4308 \def\encodingdefault{T1}%
4309 \input{t1enc.def}%
4310 \fontencoding\encodingdefault\selectfont
4311 \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。

```
4312 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4313 \renewcommand{\rmdefault}{lmr}
4314 \renewcommand{\sfdefault}{lms}
4315 \renewcommand{\ttdefault}{lmtt}
4316 \fi
```

大型演算子用の数式フォントの設定。

※`amsmaths` パッケージと同等にする。

```
4317 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4318   <-7.5>cmex7<7.5-8.5>cmex8%
4319   <8.5-9.5>cmex9<9.5->cmex10}{}%
4320 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
    amsmath 読込時に上書きされるのを防ぐ。
4321 \def\cmex@opt{10}
```

## D.2 fixltx2e 読込

※`fixltx2e` 廃止前の L<sup>A</sup>T<sub>E</sub>X カーネルの場合。

```
4322 \ifx\@IncludeInRelease\@undefined
4323 \RequirePackage{fixltx2e}
4324 \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。

```
4325 \RequirePackage{bxjsjckcat}
```

## D.4 完了

おしまい。

```
4326 %</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。

```
4327 %<*pandoc>
```

```
4328 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4329 \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。)

```
4330 \@onlypreamble\bxjs@endpreamble@hook
```

```
4331 \let\bxjs@endpreamble@hook\empty
```

```
4332 \ifjsWithTeX
```

```
4333 \RequirePackage{etoolbox}[2010/08/21]% v2.0
```

```
4334 \AtEndPreamble{\bxjs@endpreamble@hook}
```

```
4335 \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込が `\@filewithoptions` で指示されて、しかもそのファイルが読込済である場合に、オプション重複検査をスキップして、代わりに `〈定義本体〉` のコードを実行する。このコード中で `#1` は渡されたオプション列のテキストに置換される。

```
4336 \@onlypreamble\bxjs@set@dupload@proc
```

```
4337 \def\bxjs@set@dupload@proc#1{%
```

```
4338 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
4339 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
4340 \def\bxjs@set@dupload@proc@a#1{%
```

```
4341 \@onlypreamble#1\def#1##1}
```

```
4342 \def\bxjs@unset@dupload@proc#1{%
```

```
4343 \expandafter\let\csname bxjs@dlp/#1\endcsname\@undefined}
```

`\@if@ptions` `\@if@ptions` の再定義。

```
4344 \@onlypreamble\bxjs@org@if@ptions
```

```
4345 \let\bxjs@org@if@ptions\@if@ptions
```

```
4346 \@onlypreamble\bxjs@org@reset@ptions
```

```
4347 \let\bxjs@org@reset@ptions\relax
```

```
4348 \def\@if@ptions#1#2#3{%
```

```
4349 \let\bxjs@next\@secondoftwo
```

```
4350 \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
```

```

4351 \ifx\bxjs@tmpa\bxjs@tmpb
4352 \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4353 \let\bxjs@next\@firstoftwo \fi
4354 \fi
4355 \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4356 \g@addto@macro\bxjs@begin@document@hook{%
4357 \let\@if@options\bxjs@org@if@options}
4358 \@onlypreamble\bxjs@do@dupload@proc
4359 \def\bxjs@do@dupload@proc#1#2#3{%
4360 \ifx\bxjs@org@reset@options\relax
4361 \let\bxjs@org@reset@options\@reset@options
4362 \fi
4363 \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4364 \def\@reset@options{%
4365 \let\@reset@options\bxjs@org@reset@options
4366 \@reset@options
4367 \bxjs@next{#3}}%
4368 \@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。

```
4369 \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。

```
4370 \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。

```
4371 \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。

```

4372 \pandocSkipLoadPackage{polyglossia}
4373 \bxjs@set@dupload@proc{polyglossia.sty}{%
4374 \bxjs@unset@dupload@proc{polyglossia.sty}%
4375 \ClassWarning\bxjs@clsname
4376 {Package polyglossia is required}%
4377 \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。

```
4378 \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。

```
4379 \ifblank{##2}{%
4380   \ClassWarning\bxjs@clsname
4381     {Main language is invalid, thus the fallback\MessageBreak
4382       definitions will be employed}%
4383   \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。

```
4384 }{%else
4385   \ClassWarning\bxjs@clsname
4386     {Main language is valid (##2),\MessageBreak
4387       thus polyglossia will be loaded}%
4388   \csundef{ver@polyglossia.sty}%
4389   \edef\bxjs@next{%
4390     \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4391   }\bxjs@next
4392   \setmainlanguage[##1]{##2}%
4393 }
```

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。

```
4394 \g@addto@macro\bxjs@endpreamble@hook{%
4395   \ifx\bxjs@polyglossia@options\relax
4396     \csundef{ver@polyglossia.sty}%
4397   \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。

```
4398 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4399 \def\bxjs@pandoc@polyglossia@ja{%
4400   \renewcommand*\setmainlanguage[2][]{}%
4401   \newcommand*\setotherlanguage[2][]{%
4402     \ifblank{##2}{\else
4403       \cslet{##2}\@empty \cslet{end##2}\@empty
4404       \cslet{text##2}\@firstofone}}%
4405   \newcommand*\setotherlanguages[2][]{%
4406     \@for\bxjs@tmpa:={##2}\do{%
4407       \setotherlangauge{\bxjs@tmpa}}}%
4408 }
```

`Polyglossia` の読込済マークは外れるようにしておく。

```
4408 \let\bxjs@polyglossia@options\relax}%
4409 \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。



```
4410 \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。

```
4411 \pandocSkipLoadPackage{babel}  
4412 \bxjs@set@dupload@proc{babel.sty}{%  
4413   \bxjs@unset@dupload@proc{babel.sty}%  
4414   \ClassWarning\bxjs@clsname  
4415   {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。

```
4416   \@tempswafalse \let\bxjs@babel@options\@empty  
4417   \def\bxjs@tmpb{main=}%  
4418   \@for\bxjs@tmpa:=#1\do{%  
4419     \ifx\bxjs@tmpa\bxjs@tmpb \@tempwattrue  
4420     \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%  
4421     \fi}%  
4422   \if@tempswa  
4423     \ClassWarning\bxjs@clsname  
4424     {Main language is invalid, so fallback\MessageBreak  
4425     definitions will be employed}%  
4426     \bxjs@pandoc@polyglossia@ja
```

ない場合は、本来の babel の処理を実行する。

```
4427   \else  
4428     \ClassWarning\bxjs@clsname  
4429     {Main language is valid,\MessageBreak  
4430     thus babel will be loaded}%  
4431     \expandafter\let\csname ver@babel.sty\endcsname\@undefined  
4432     \RequirePackage[#1]{babel}[]%  
4433   \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。

```
4434 \g@addto@macro\bxjs@endpreamble@hook{%  
4435   \ifx\bxjs@babel@options\relax  
4436     \expandafter\let\csname ver@babel.sty\endcsname\@undefined  
4437   \fi}
```

\bxjs@pandoc@babel@ja Pandoc 側で lang=ja が指定されていた場合の処理。

```
4438 \@onlypreamble\bxjs@pandoc@babel@ja  
4439 \def\bxjs@pandoc@babel@ja{%  
4440   \expandafter\let\csname ver@babel.sty\endcsname\@undefined  
4441   \edef\bxjs@next{%  
4442     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%  
4443   }\bxjs@next  
4444   \if j\jsEngine  
4445     \RequirePackage[main=japanese]{pxbabel}[]%  
4446   \else  
4447     \RequirePackage{bxorigcapt}[]%
```

```
4448 \fi}
```

lang 対策はこれで終わり。

```
4449 \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。

```
4450 \bxjs@set@dupload@proc{geometry.sty}{%
4451 \setpagelayout*{#1}}
```

### E.4 CJKmainfont 変数

LuaTeX (+ LuaTeX-jan) の場合に CJKmainfont 変数が指定された場合は `\setmainfont` の指定にまわす。

```
4452 \if 1\jsEngine
4453 \pandocSkipLoadPackage{xeCJK}
4454 \providecommand*{\setCJKmainfont}{\setmainfont}
4455 \fi
```

ムニャムニャムニャ。

```
4456 \if x\jsEngine
4457 \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4458 ,space}
4459 \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。

```
4460 \g@addto@macro\bxjs@begin@document@hook{%
4461 \@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。

```
4462 \ifx\oldparagraph\undefined\else
4463 \@tempswattrue
4464 \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4465 \ifnum\c@secnumdepth>3
4466     \@tempwattrue
4467 \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。

```
4468 \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4469     \let\jsParagraphMark\@empty
4470 \fi\fi}
```

## E.6 全角空白文字

```
4471 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4472 \catcode"3000=\active
4473 \begingroup \catcode`\!=7
4474 \protected\gdef!!!!3000{\zwspace}
4475 \endgroup
4476 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4477 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4478 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4479 \fi\fi
```

■hyperref 対策 hyperref の unicode オプションの固定を行う。

```
4480 \if j\jsEngine
4481 \bxjs@fix@hyperref@unicode{false}
4482 \else
4483 \bxjs@fix@hyperref@unicode{true}
4484 \fi
```

## E.7 完了

おしまい。

```
4485 %</pandoc>
```

和文ドライバ実装はここまで。

```
4486 %</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。

```
4487 %<*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

```
4488 %<*compat>
4489 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4490 \let\bxac@engine=n
4491 \def\bxac@do#1#2{%
4492   \edef\bxac@tmpa{\string#1}%
4493   \edef\bxac@tmpb{\meaning#1}%
4494   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
4495 \bxac@do\XeTeXversion{\let\bxac@engine=x}
4496 \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。

```
4497 \ifx\jsAtEndOfClass\@undefined
4498   \let\bxac@delayed@if@bxjs\@firstofone
4499 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
4500 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4501 \newif\ifbxac@in@old@behavior
4502 \let\ImposeOldLuaTeXBehavior\relax
4503 \let\RevokeOldLuaTeXBehavior\relax
```

### G.2 XeTeX 部分

```
4504 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。

4505 \@onlypreamble\bxac@adjust@charclass
4506 \bxac@delayed@if@bxjs{%
4507   \@ifpackageloaded{xeCJK}{-}{%else
4508     \ifx\xe@alloc@intercharclass\@undefined\else
4509       \ifnum\xe@alloc@intercharclass=z@
4510         \PackageInfo\bxac@pkgname
4511           {Setting up interchar class for CJK...\@gobble}%
4512         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4513           \xe@alloc@intercharclass=3
4514         }{%else
4515           \PackageWarning\bxac@pkgname
4516             {Cannot find file 'load-unicode-xetex-classes.tex'%
```

```

4517         \@gobble}%
4518     }%
4519     \fi\fi
4520     \ifnum\XeTeXcharclass"3041=\z@
4521         \PackageInfo\bxac@pkgname
4522         {Adjusting interchar class for CJK...\@gobble}%
4523         \@for\bxac@x={%
4524             3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4525             3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4526             30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4527             31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4528             31FF%
4529         }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4530     \fi
4531 }%
4532 }

```

以上。

```
4533 \fi
```

### G.3 LuaTeX 部分

```

4534 \ifx l\bxac@engine
    ムニヤムニヤ。
4535 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4536     \chardef\pdftexversion=200
4537     \def\pdfTeXrevision{0}
4538     \let\pdfTeXbanner\luatexbanner
4539 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior 4540 \begingroup\expandafter\expandafter\expandafter\endgroup
4541 \expandafter\ifx\cscname outputmode\endcsname\relax\else
4542 \def\bxac@ob@list{%
4543     \do{\let}\pdfoutput{\outputmode}%
4544     \do{\let}\pdfpagewidth{\pagewidth}%
4545     \do{\let}\pdfpageheight{\pageheight}%
4546     \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
4547     \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
4548 \def\bxac@ob@do#1#2{\begingroup
4549     \expandafter\bxac@ob@do@a\cscname bxac@\string#2\endcsname{#1}#2}
4550 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4551     \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4552     \else \let#3#1\relax \let#1\@undefined
4553     \fi}
4554 \protected\def\ImposeOldLuaTeXBehavior{%
4555     \unless\ifbxac@in@old@behavior
4556         \bxac@in@old@behaviortrue
4557         \let\do\bxac@ob@do \bxac@ob@list
4558     \fi}

```

```

4559 \protected\def\RevokeOldLuaTeXBehavior{%
4560   \ifbxac@in@old@behavior
4561     \bxac@in@old@behaviorfalse
4562     \let\do\bxac@ob@do \bxac@ob@list
4563   \fi}
4564 \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。

```

4565 \ifnum\luatexversion>64 \directlua{
4566   local function range(cs, ce, cc, ff)
4567     if ff or not tex.getcatcode(cs) == cc then
4568       local setcc = tex.setcatcode
4569       for c = cs, ce do setcc(c, cc) end
4570     end
4571   end
4572   range(0x3400, 0x4DB5, 11, false)
4573   range(0x4DB5, 0x4DBF, 11, true)
4574   range(0x4E00, 0x9FCC, 11, false)
4575   range(0x9FCD, 0x9FFF, 11, true)
4576   range(0xAC00, 0xD7A3, 11, false)
4577   range(0x20000, 0x2A6D6, 11, false)
4578   range(0x2A6D7, 0x2A6FF, 11, true)
4579   range(0x2A700, 0x2B734, 11, false)
4580   range(0x2B735, 0x2B73F, 11, true)
4581   range(0x2B740, 0x2B81D, 11, false)
4582   range(0x2B81E, 0x2B81F, 11, true)
4583   range(0x2B820, 0x2CEA1, 11, false)
4584   range(0x2CEA2, 0x2FFFD, 11, true)
4585 } \fi

```

以上。

```

4586 \fi

```

## G.4 完了

おしまい。

```

4587 %</compat>

```

## 付録 H 補助パッケージ : bxjscjkcat

modern ドライバ用の和文カテゴリを適用する。

### H.1 準備

```

4588 %<*cjkcat>
4589 \def\bxjx@pkgname{bxjscjkcat}
4590 \newcount\bxjx@canta

```

```

4591 \@onlypreamble\bxjx@tmpdo
4592 \@onlypreamble\bxjx@tmpdo@a
4593 \@onlypreamble\bxjx@tmpdo@b

```

`\bxjx@engine` エンジンの種別。

```

4594 \let\bxjx@engine=n
4595 \def\bxjx@tmpdo#1#2{%
4596   \edef\bxjx@tmpa{\string#1}%
4597   \edef\bxjx@tmpb{\meaning#1}%
4598   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4599 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4600 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4601 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4602 \bxjx@tmpdo\pdftexversion{\let\bxjx@engine=p}
4603 \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。

```

4604 \def\bxjx@tmpdo#1#2{%
4605   \if#1\bxjx@engine
4606     \@ifpackageloaded{#2}{\}%else
4607     \PackageError\bxjx@pkgname
4608       {Package '#2' must be loaded}%
4609       {Package loading is aborted.\MessageBreak\@ehc}%
4610     \endinput}
4611   \fi}
4612 \bxjx@tmpdo{p}{bxcjkatype}
4613 \bxjx@tmpdo{x}{xeCJK}
4614 \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。

```

4615 \ifx\TextOrMath\@undefined
4616   \RequirePackage{fixltx2e}
4617 \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。

```

4618 \if u\bxjx@engine
4619 \@for\bxjx@tmpa:={%
4620 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4621 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4622 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4623 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4624 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4625 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%

```

```

4626 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4627 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4628 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4629 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4630 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4631 FE70,FF00,%
4632 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4633 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4634 10480,104B0,10500,10530,10600,10800,10840,10860,%
4635 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4636 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4637 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4638 11200,11280,112B0,11300,11400,11480,11580,11600,%
4639 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4640 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4641 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4642 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4643 1EE00,1F000,1F030,1FOA0,1F300,1F600,1F650,1F680,%
4644 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4645 00C0%
4646 }\do{%
4647 \@tempcnta="\bxjx@tmpa\relax
4648 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4649 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4650 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4651 \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。

```

4652 \@onlypreamble\bxjx@grkcyr@list
4653 \def\bxjx@grkcyr@list{%
4654 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4655 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4656 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4657 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4658 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4659 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA

```



4660	$\backslash\mathrm{do}\{0397\}\{\mathrm{LGR}\}\{\mathrm{textEta}\}\{\mathrm{H}\}\%$	% GR. C. L. ETA
4661	$\backslash\mathrm{do}\{0398\}\{\mathrm{LGR}\}\{\mathrm{textTheta}\}\{\mathrm{Theta}\}\%$	% GR. C. L. THETA
4662	$\backslash\mathrm{do}\{0399\}\{\mathrm{LGR}\}\{\mathrm{textIota}\}\{\mathrm{I}\}\%$	% GR. C. L. IOTA
4663	$\backslash\mathrm{do}\{039\mathrm{A}\}\{\mathrm{LGR}\}\{\mathrm{textKappa}\}\{\mathrm{K}\}\%$	% GR. C. L. KAPPA
4664	$\backslash\mathrm{do}\{039\mathrm{B}\}\{\mathrm{LGR}\}\{\mathrm{textLambda}\}\{\mathrm{Lambda}\}\%$	% GR. C. L. LAMDA
4665	$\backslash\mathrm{do}\{039\mathrm{C}\}\{\mathrm{LGR}\}\{\mathrm{textMu}\}\{\mathrm{M}\}\%$	% GR. C. L. MU
4666	$\backslash\mathrm{do}\{039\mathrm{D}\}\{\mathrm{LGR}\}\{\mathrm{textNu}\}\{\mathrm{N}\}\%$	% GR. C. L. NU
4667	$\backslash\mathrm{do}\{039\mathrm{E}\}\{\mathrm{LGR}\}\{\mathrm{textXi}\}\{\mathrm{Xi}\}\%$	% GR. C. L. XI
4668	$\backslash\mathrm{do}\{039\mathrm{F}\}\{\mathrm{LGR}\}\{\mathrm{textOmicron}\}\{\mathrm{O}\}\%$	% GR. C. L. OMICRON
4669	$\backslash\mathrm{do}\{03\mathrm{A}0\}\{\mathrm{LGR}\}\{\mathrm{textPi}\}\{\mathrm{Pi}\}\%$	% GR. C. L. PI
4670	$\backslash\mathrm{do}\{03\mathrm{A}1\}\{\mathrm{LGR}\}\{\mathrm{textRho}\}\{\mathrm{P}\}\%$	% GR. C. L. RHO
4671	$\backslash\mathrm{do}\{03\mathrm{A}3\}\{\mathrm{LGR}\}\{\mathrm{textSigma}\}\{\mathrm{Sigma}\}\%$	% GR. C. L. SIGMA
4672	$\backslash\mathrm{do}\{03\mathrm{A}4\}\{\mathrm{LGR}\}\{\mathrm{textTau}\}\{\mathrm{T}\}\%$	% GR. C. L. TAU
4673	$\backslash\mathrm{do}\{03\mathrm{A}5\}\{\mathrm{LGR}\}\{\mathrm{textUpsilon}\}\{\mathrm{Upsilon}\}\%$	% GR. C. L. UPSILON
4674	$\backslash\mathrm{do}\{03\mathrm{A}6\}\{\mathrm{LGR}\}\{\mathrm{textPhi}\}\{\mathrm{Phi}\}\%$	% GR. C. L. PHI
4675	$\backslash\mathrm{do}\{03\mathrm{A}7\}\{\mathrm{LGR}\}\{\mathrm{textChi}\}\{\mathrm{X}\}\%$	% GR. C. L. CHI
4676	$\backslash\mathrm{do}\{03\mathrm{A}8\}\{\mathrm{LGR}\}\{\mathrm{textPsi}\}\{\mathrm{Psi}\}\%$	% GR. C. L. PSI
4677	$\backslash\mathrm{do}\{03\mathrm{A}9\}\{\mathrm{LGR}\}\{\mathrm{textOmega}\}\{\mathrm{Omega}\}\%$	% GR. C. L. OMEGA
4678	$\backslash\mathrm{do}\{03\mathrm{B}1\}\{\mathrm{LGR}\}\{\mathrm{textalpha}\}\{\mathrm{alpha}\}\%$	% GR. S. L. ALPHA
4679	$\backslash\mathrm{do}\{03\mathrm{B}2\}\{\mathrm{LGR}\}\{\mathrm{textbeta}\}\{\mathrm{beta}\}\%$	% GR. S. L. BETA
4680	$\backslash\mathrm{do}\{03\mathrm{B}3\}\{\mathrm{LGR}\}\{\mathrm{textgamma}\}\{\mathrm{gamma}\}\%$	% GR. S. L. GAMMA
4681	$\backslash\mathrm{do}\{03\mathrm{B}4\}\{\mathrm{LGR}\}\{\mathrm{textdelta}\}\{\mathrm{delta}\}\%$	% GR. S. L. DELTA
4682	$\backslash\mathrm{do}\{03\mathrm{B}5\}\{\mathrm{LGR}\}\{\mathrm{textepsilon}\}\{\mathrm{epsilon}\}\%$	% GR. S. L. EPSILON
4683	$\backslash\mathrm{do}\{03\mathrm{B}6\}\{\mathrm{LGR}\}\{\mathrm{textzeta}\}\{\mathrm{zeta}\}\%$	% GR. S. L. ZETA
4684	$\backslash\mathrm{do}\{03\mathrm{B}7\}\{\mathrm{LGR}\}\{\mathrm{texteta}\}\{\mathrm{eta}\}\%$	% GR. S. L. ETA
4685	$\backslash\mathrm{do}\{03\mathrm{B}8\}\{\mathrm{LGR}\}\{\mathrm{texttheta}\}\{\mathrm{theta}\}\%$	% GR. S. L. THETA
4686	$\backslash\mathrm{do}\{03\mathrm{B}9\}\{\mathrm{LGR}\}\{\mathrm{textiota}\}\{\mathrm{iota}\}\%$	% GR. S. L. IOTA
4687	$\backslash\mathrm{do}\{03\mathrm{B}\mathrm{A}\}\{\mathrm{LGR}\}\{\mathrm{textkappa}\}\{\mathrm{kappa}\}\%$	% GR. S. L. KAPPA
4688	$\backslash\mathrm{do}\{03\mathrm{B}\mathrm{B}\}\{\mathrm{LGR}\}\{\mathrm{textlambda}\}\{\mathrm{lambda}\}\%$	% GR. S. L. LAMDA
4689	$\backslash\mathrm{do}\{03\mathrm{B}\mathrm{C}\}\{\mathrm{LGR}\}\{\mathrm{textmu}\}\{\mathrm{mu}\}\%$	% GR. S. L. MU
4690	$\backslash\mathrm{do}\{03\mathrm{B}\mathrm{D}\}\{\mathrm{LGR}\}\{\mathrm{textnu}\}\{\mathrm{nu}\}\%$	% GR. S. L. NU
4691	$\backslash\mathrm{do}\{03\mathrm{B}\mathrm{E}\}\{\mathrm{LGR}\}\{\mathrm{textxi}\}\{\mathrm{xi}\}\%$	% GR. S. L. XI
4692	$\backslash\mathrm{do}\{03\mathrm{B}\mathrm{F}\}\{\mathrm{LGR}\}\{\mathrm{textomicron}\}\{\mathrm{o}\}\%$	% GR. S. L. OMICRON
4693	$\backslash\mathrm{do}\{03\mathrm{C}0\}\{\mathrm{LGR}\}\{\mathrm{textpi}\}\{\mathrm{pi}\}\%$	% GR. S. L. PI
4694	$\backslash\mathrm{do}\{03\mathrm{C}1\}\{\mathrm{LGR}\}\{\mathrm{textrho}\}\{\mathrm{rho}\}\%$	% GR. S. L. RHO
4695	$\backslash\mathrm{do}\{03\mathrm{C}2\}\{\mathrm{LGR}\}\{\mathrm{textvarsigma}\}\{\mathrm{varsigma}\}\%$	% GR. S. L. FINAL SIGMA
4696	$\backslash\mathrm{do}\{03\mathrm{C}3\}\{\mathrm{LGR}\}\{\mathrm{textsigma}\}\{\mathrm{sigma}\}\%$	% GR. S. L. SIGMA
4697	$\backslash\mathrm{do}\{03\mathrm{C}4\}\{\mathrm{LGR}\}\{\mathrm{texttau}\}\{\mathrm{tau}\}\%$	% GR. S. L. TAU
4698	$\backslash\mathrm{do}\{03\mathrm{C}5\}\{\mathrm{LGR}\}\{\mathrm{textupsilon}\}\{\mathrm{upsilon}\}\%$	% GR. S. L. UPSILON
4699	$\backslash\mathrm{do}\{03\mathrm{C}6\}\{\mathrm{LGR}\}\{\mathrm{textphi}\}\{\mathrm{phi}\}\%$	% GR. S. L. PHI
4700	$\backslash\mathrm{do}\{03\mathrm{C}7\}\{\mathrm{LGR}\}\{\mathrm{textchi}\}\{\mathrm{chi}\}\%$	% GR. S. L. CHI
4701	$\backslash\mathrm{do}\{03\mathrm{C}8\}\{\mathrm{LGR}\}\{\mathrm{textpsi}\}\{\mathrm{psi}\}\%$	% GR. S. L. PSI
4702	$\backslash\mathrm{do}\{03\mathrm{C}9\}\{\mathrm{LGR}\}\{\mathrm{textomega}\}\{\mathrm{omega}\}\%$	% GR. S. L. OMEGA
4703	$\backslash\mathrm{do}\{0401\}\{\mathrm{T}2\mathrm{A}\}\{\mathrm{CYR}0\}\{\}\%$	% CY. C. L. IO
4704	$\backslash\mathrm{do}\{0410\}\{\mathrm{T}2\mathrm{A}\}\{\mathrm{CYR}\mathrm{A}\}\{\}\%$	% CY. C. L. A
4705	$\backslash\mathrm{do}\{0411\}\{\mathrm{T}2\mathrm{A}\}\{\mathrm{CYR}\mathrm{B}\}\{\}\%$	% CY. C. L. BE
4706	$\backslash\mathrm{do}\{0412\}\{\mathrm{T}2\mathrm{A}\}\{\mathrm{CYR}\mathrm{V}\}\{\}\%$	% CY. C. L. VE
4707	$\backslash\mathrm{do}\{0413\}\{\mathrm{T}2\mathrm{A}\}\{\mathrm{CYR}\mathrm{G}\}\{\}\%$	% CY. C. L. GHE
4708	$\backslash\mathrm{do}\{0414\}\{\mathrm{T}2\mathrm{A}\}\{\mathrm{CYR}\mathrm{D}\}\{\}\%$	% CY. C. L. DE

4709 \do{0415}{T2A}{\CYRE}{}%	% CY. C. L. IE
4710 \do{0416}{T2A}{\CYRZH}{}%	% CY. C. L. ZHE
4711 \do{0417}{T2A}{\CYRZ}{}%	% CY. C. L. ZE
4712 \do{0418}{T2A}{\CYRI}{}%	% CY. C. L. I
4713 \do{0419}{T2A}{\CYRISHRT}{}%	% CY. C. L. SHORT I
4714 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
4715 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
4716 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
4717 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
4718 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
4719 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
4720 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
4721 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
4722 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
4723 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
4724 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
4725 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
4726 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
4727 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
4728 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
4729 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
4730 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
4731 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
4732 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
4733 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
4734 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
4735 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
4736 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
4737 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
4738 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
4739 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
4740 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
4741 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
4742 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
4743 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
4744 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
4745 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
4746 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
4747 \do{043B}{T2A}{\cyrl}{}%	% CY. S. L. EL
4748 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
4749 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
4750 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
4751 \do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
4752 \do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
4753 \do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES
4754 \do{0442}{T2A}{\cyrt}{}%	% CY. S. L. TE
4755 \do{0443}{T2A}{\cyru}{}%	% CY. S. L. U
4756 \do{0444}{T2A}{\cyrf}{}%	% CY. S. L. EF
4757 \do{0445}{T2A}{\cyrh}{}%	% CY. S. L. HA

```

4758 \do{0446}{T2A}{\cyrcl}{}%           % CY. S. L. TSE
4759 \do{0447}{T2A}{\cyrch}{}%           % CY. S. L. CHE
4760 \do{0448}{T2A}{\cyrsh}{}%           % CY. S. L. SHA
4761 \do{0449}{T2A}{\cyrshch}{}%         % CY. S. L. SHCHA
4762 \do{044A}{T2A}{\cyrhrdsn}{}%        % CY. S. L. HARD SIGN
4763 \do{044B}{T2A}{\cyrery}{}%         % CY. S. L. YERU
4764 \do{044C}{T2A}{\cyrstsn}{}%        % CY. S. L. SOFT SIGN
4765 \do{044D}{T2A}{\cyrerev}{}%        % CY. S. L. E
4766 \do{044E}{T2A}{\cyrlyu}{}%         % CY. S. L. YU
4767 \do{044F}{T2A}{\cyrly}{}%          % CY. S. L. YA
4768 \do{0451}{T2A}{\cyrlyo}{}%         % CY. S. L. IO
4769 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4770 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4771 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4772 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
4773 \do{00B4}{TS1}{\textasciicircum}{}% % ACUTE ACCENT
4774 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4775 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4776 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
4777 }

```

`\mathdegree` 面倒なので補っておく。

```

4778 \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。

```

4779 \newif\ifbxjx@gcc@CJK

```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```

4780 \newcommand*\greekasCJK{%
4781   \bxjx@gcc@CJKtrue}

```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```

4782 \newcommand*\nogreekasCJK{%
4783   \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。

```

4784 \def\bxjx@tmpdo#1\relax{%
4785   \def\bxjx@fake@grk##1##2{%
4786     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4787   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4788     \ifx\@##1\%
4789       \bxjx@canta##4\divide\bxjx@canta\@cclvi

```

```

4790 \multiply\bxjx@cmta\@cclvi \advance\bxjx@cmta`##3\relax
4791 \mathchar\bxjx@cmta
4792 \else ##3\fi}
4793 }\expandafter\bxjx@tmpdo\string\mathchar\relax

```

### ■pdf $\TeX$ ・up $\TeX$ の場合

```

4794 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0

```

- $\backslash\text{bxjx@KC}/\langle\text{符号値}\rangle$  : 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  $\text{L}\text{A}\text{T}\text{E}\text{X}$  においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```

4795 \@ifpackageloaded{inputenc}{\fi}{%else
4796 \RequirePackage[utf8]{inputenc}}
4797 \def\bxjx@tmpa{utf8}
4798 \ifx\bxjx@tmpa\inputencdoingname
4799 \PackageWarningNoLine\bxjx@pkgname
4800 {Input encoding changed to utf8}%
4801 \inputencoding{utf8}%
4802 \fi

```

up $\TeX$  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。

```

4803 \if u\bxjx@engine
4804 \kcatcode"0370=15
4805 \kcatcode"0400=15
4806 \kcatcode"0500=15
4807 \fi

```

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

```

4808 \def\bxjx@tmpdo#1{%
4809 \@tempcnta="#1\relax
4810 \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4811 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

引数 =  $\backslash\text{bxjx@KC}/\langle\text{符号値}\rangle\{\langle\text{符号値}\rangle\}\{\langle\text{fontenc}\rangle\}\{\langle\text{LICR}\rangle\}\{\langle\text{数式 LICR}\rangle\}$

“数式中の動作”を決定する。 $\langle\text{数式 LICR}\rangle$  が空（数式非対応）なら警告を出す。

```

4812 \ifx\#5\%
4813 \def\bxjx@tmpa{\@inmathwarn#4}%

```

$\langle\text{数式 LICR}\rangle$  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。

```

4814 \else\ifcat A\noexpand#5%
4815 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4816 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%

```

それ以外は  $\langle\text{数式 LICR}\rangle$  をそのまま実行する。

```

4817 \else \def\bxjx@tmpa{#5}%
4818 \fi\fi
4819 \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4820 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

4821 \if u\bxjx@engine
4822 \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。

```

4823 \kchardef#1=\@tempcnta
4824 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4825 \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。

```

4826 \else\if p\bxjx@engine
4827 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4828 \mathchardef#1=\@tempcnta
4829 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4830 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4831 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

4832 \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制したもの。

```

4833 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4834 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4835 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4836 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4837 \count@="#1\relax
4838 \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4839 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4840 \else
4841 \wlog{\space\space skipped defining Unicode char U+#1}%
4842 \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。

```

4843 \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。

```
4844 \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。

```
4845 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
```

```
4846 \else \UseTextSymbol{#2}{#3}%
```

```
4847 \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。

```
4848 \begingroup
```

```
4849 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
```

```
4850 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
```

```
4851 \noexpand\bxjx@swap@DUC@cmd
```

```
4852 \the\toks@
```

```
4853 \noexpand\bxjx@swap@DUC@cmd}}}
```

```
4854 \endgroup\next
```

```
4855 \def\bxjx@swap@DUC@cmd{%
```

```
4856 \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```
4857 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
```

```
4858 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
```

```
4859 \let\bxjx@tmpa\relax}
```

以上。

## ■X<sub>Y</sub>TeX・LuaTeX の場合

```
4860 \else\ifnum0\if x\bxjx@engine1\fi\if 1\bxjx@engine1\fi>0
```

各文字について、数式中の動作を定義する。

```
4861 \def\bxjx@tmpdo#1{%
```

```
4862 \bxjx@cmta="#1\relax
```

```
4863 \begingroup
```

```
4864 \lccode`~=\bxjx@cmta
```

```
4865 \lowercase{\endgroup
```

```
4866 \bxjx@tmpdo@a{~}}{#1}}
```

```
4867 \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。

```
4868 \ifx\\#5\\\let\bxjx@tmpa\relax
```

```
4869 \else\ifcat A\noexpand#5%
```

```
4870 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
```

```
4871 {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
```

```
4872 \else \def\bxjx@tmpa{#5}%
```

```
4873 \fi\fi
```

```
4874 \ifx\bxjx@tmpa\relax\else
```

```
4875 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
```

```
4876 \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。

```
4877 \mathchardef\bxjx@tmpa="119
4878 \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。

```
4879 \if 1\bxjx@engine
4880   \protected\def\greekasCJK{%
4881     \bxjx@gcc@cjctrue
4882     \ltjsetparameter{jacharrange={+2, +8}}
4883   \protected\def\nogreekasCJK{%
4884     \bxjx@gcc@cjcfalse
4885     \ltjsetparameter{jacharrange={-2, -8}}
4886 \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。

```
4887 \if x\bxjx@engine
4888   \let\bxjx@gcc@cjkl@list\@empty
4889   \def\do#1#2#3#4{%
4890     \edef\bxjx@gcc@cjkl@list{\bxjx@gcc@cjkl@list
4891       \noexpand\XeTeXcharclass"#1\bxjx@canta}}
4892   \bxjx@grkcyr@list
4893   \protected\def\greekasCJK{%
4894     \bxjx@gcc@cjctrue
4895     \bxjx@canta=\@ne \bxjx@gcc@cjkl@list}
4896   \protected\def\nogreekasCJK{%
4897     \bxjx@gcc@cjcfalse
4898     \bxjx@canta=\z@ \bxjx@gcc@cjkl@list}
4899 \fi
```

以上。

```
4900 \fi\fi
```

## H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
4901 \nogreekasCJK
```

## H.5 完了

おしまい。

```
4902 %</cjkat>
```

## 付録 I 補助パッケージ : bxjspancod

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

```
4903 %<*ancpandoc>
4904 %% このファイルは日本語文字を含みます.
4905 \def\bxjsp@pkgname{bxjscjkat}

\bxjsp@engine エンジンの種別。

4906 \let\bxjsp@engine=n
4907 \@onlypreamble\bxjsp@do
4908 \def\bxjsp@do#1#2{%
4909   \edef\bxjsp@tmpa{\string#1}%
4910   \edef\bxjsp@tmpb{\meaning#1}%
4911   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
4912 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
4913 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
4914 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
4915 \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。

```
4916 \newif\ifbxjsp@english
4917 \DeclareOption{english}{\bxjsp@englishtrue}

オプション定義はおしまい。

4918 \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。

```
4919 \@onlypreamble\pandocSkipLoadFile
4920 \newcommand*\pandocSkipLoadFile[1]{%
```



```

4921 \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4922 \def\bxjsp@skip@load@file@a#1#2{%
4923   \ifx#1\relax
4924     \def#1{2001/01/01}%
4925     \PackageInfo\bxjsp@pkgname
4926       {File '#2' marked as loaded\@gobble}%
4927   \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。

```

4928 \@onlypreamble\pandocSkipLoadPackage
4929 \newcommand*\pandocSkipLoadPackage[1]{%
4930   \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  $\text{\LaTeX}$  ではこれで警告が出る。これを抑止する。

$\text{\LaTeX}$  カーネルが新しい場合は `fixltx2e` を読込済にする。

```

4931 \ifx\@IncludeInRelease\@undefined\else
4932   \pandocSkipLoadPackage{fixltx2e}
4933 \fi

```

## 1.5 cmap パッケージ

エンジンが (u)p $\text{\LaTeX}$  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。）

```

4934 \if j\bxjsp@engine
4935   \pandocSkipLoadPackage{cmap}
4936 \fi

```

## 1.6 microtype パッケージ

警告が多すぎなので消す。

```

4937 \if j\bxjsp@engine \else
4938   \PassOptionsToPackage{verbose=silent}{microtype}
4939 \fi

```

エンジンが (u)p $\text{\LaTeX}$  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。

```

4940 \if j\bxjsp@engine
4941   \pandocSkipLoadPackage{microtype}
4942   \newcommand*\UseMicrotypeSet[2][]{\fi}

```

4943 \fi

## 1.7 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。

$\cdots \rightarrow \text{\ldots}$  ‘ $\rightarrow$ ’ ‘ $\rightarrow$ ’ “ $\rightarrow$ ” “ $\rightarrow$ ”

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、「 $\cdots$ 」の置換を抑止する機能はないようである。そこで、「 $\text{\ldots}$ 」を『 $\cdots$ 』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  $\cdots$  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  $\cdots$  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  $\cdots$  に戻す動作に変更した。

```
4944 \DeclareRobustCommand{\pandocLdots}{%
4945   \relax\ifmmode \expandafter\bxjsp@org@ldots
4946   \else \expandafter\bxjsp@ja@ellipsis
4947   \fi}
4948 \def\bxjsp@ja@ellipsis{…}
4949 \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。

```
4950 \AtBeginDocument{%
4951   \let\bxjsp@org@ldots\ldots
4952   \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。

```
4953 \chardef\bxjsp@cc@other=12
4954 \@onlypreamble\bxjsp@makeother@range
4955 \def\bxjsp@makeother@range#1#2{%
4956   \@tempcnta"#1\relax \@tempcntb"#2\relax
4957   \loop\ifnum\@tempcnta<\@tempcntb
4958     \catcode\@tempcnta\bxjsp@cc@other
4959     \advance\@tempcnta\@ne
4960   \repeat}
4961 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
```

```

4962 \catcode"1F23B=\bxjsp@cc@other
4963 \bxjsp@makeother@range{9FCD}{A000}
4964 \bxjsp@makeother@range{1B002}{1B170}
4965 \bxjsp@makeother@range{2B820}{2EBF0}
4966 \fi

```

## 1.8 PandoLa モジュール

インストール済であれば読み込む。

```

4967 \IfFileExists{bxpandola.sty}{%
4968 \RequirePackage{bxpandola}\relax
4969 \PackageInfo{bxjsp@pkgname
4970 {PandoLa module is loaded\@gobble}
4971 }{}

```

## 1.9 完了

おしまい。

```

4972 %</ancpandoc>

```

補助パッケージ実装はここまで。

```

4973 %</anc>

```