

# 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”)

v2.5a [2021/05/18]






---





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

---

## 目次

1	はじめに	3
2	オプション	10
3	和文フォントの変更	40
4	フォントサイズ	40
5	レイアウト	46
5.1	ページレイアウト	47
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)	62
7	ページスタイル	63
8	文書のマークアップ	66
8.1	表題	66
8.2	章・節	72
8.3	リスト環境	84
8.4	パラメータの設定	91
8.5	フロート	93
8.6	キャプション	94
9	フォントコマンド	95

10	相互参照	98
10.1	目次の類 . . . . .	98
10.2	参考文献 . . . . .	103
10.3	索引 . . . . .	105
10.4	脚注 . . . . .	106
11	段落の頭へのグルー挿入禁止	109
12	いろいろなロゴ	113
13	amsmath との衝突の回避	113
14	初期設定	114
付録 A	和文ドライバの仕様 	118
付録 B	和文ドライバ : minimal 	119
B.1	補助マクロ . . . . .	119
B.2	(u)pTeX 用の設定 . . . . .	121
B.3	pdfTeX 用の処理 . . . . .	126
B.4	X <sub>Y</sub> TeX 用の処理 . . . . .	127
B.5	後処理 (エンジン共通) . . . . .	128
付録 C	和文ドライバ : standard 	131
C.1	準備 . . . . .	131
C.2	和文ドライバパラメタ . . . . .	131
C.3	共通処理 (1) . . . . .	132
C.4	pTeX 用設定 . . . . .	140
C.5	pdfTeX 用設定 : CJK + bxcjkatype . . . . .	143
C.6	X <sub>Y</sub> TeX 用設定 : xeCJK + zxjatype . . . . .	145
C.7	LuaTeX 用設定 : LuaTeX-ja . . . . .	148
C.8	共通処理 (2) . . . . .	151
付録 D	和文ドライバ : modern 	152
D.1	フォント設定 . . . . .	153
D.2	fixltx2e 読込 . . . . .	153
D.3	和文カテゴリコード . . . . .	153
D.4	完了 . . . . .	153
付録 E	和文ドライバ : pandoc 	154
E.1	準備 . . . . .	154
E.2	和文ドライバパラメタ . . . . .	154
E.3	dupload システム . . . . .	156

E.4	lang 変数 . . . . .	157
E.5	geometry 変数 . . . . .	160
E.6	CJKmainfont 変数 . . . . .	160
E.7	Option clash 対策 . . . . .	160
E.8	レイアウト上書き禁止 . . . . .	160
E.9	paragraph のマーク . . . . .	161
E.10	全角空白文字 . . . . .	162
E.11	hyperref 対策 . . . . .	162
E.12	Pandoc 要素に対する和文用の補正 . . . . .	162
E.13	ifPDFTeX スイッチ . . . . .	163
E.14	完了 . . . . .	164
付録 F	補助パッケージ一覧 	165
付録 G	補助パッケージ : bxjscompat 	165
G.1	準備 . . . . .	165
G.2	X <sub>Y</sub> TeX 部分 . . . . .	166
G.3	LuaTeX 部分 . . . . .	166
G.4	完了 . . . . .	168
付録 H	補助パッケージ : bxjscjkat 	168
H.1	準備 . . . . .	168
H.2	和文カテゴリコードの設定 . . . . .	169
H.3	ギリシャ・キリル文字の扱い . . . . .	169
H.4	初期設定 . . . . .	177
H.5	完了 . . . . .	177
付録 I	補助パッケージ : bxjspandoc 	177
I.1	準備 . . . . .	177
I.2	パッケージオプション . . . . .	177
I.3	パッケージ読込の阻止 . . . . .	178
I.4	fixltx2e パッケージ . . . . .	178
I.5	cmap パッケージ . . . . .	178
I.6	microtype パッケージ . . . . .	179
I.7	Unicode 文字変換対策 . . . . .	179
I.8	PandoLa モジュール . . . . .	180
I.9	完了 . . . . .	180

## 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。

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

本ドキュメントクラスは奥村晴彦氏および日本語 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>X 3 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。

minijs は、jsclasses に似た設定を行うパッケージです。

```
1 %<*minijs>
2 %% if jsclasses loaded, abort loading this package
3 \ifx\@jsc@uplatextrue\@undefined\else
4   \PackageInfo{minijs}{jsclasses does not need minijs, exiting}
5   \expandafter\endinput
6 \fi
7 %% "fake" jsarticle
```

```

8 \expandafter\def\csname ver@jsarticle.cls\endcsname{}
9 %</minijs>

\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。
10 %<*class>
11 %% このファイルは日本語文字を含みます。
12 %<article>\def\bxjs@clsname{bxjsarticle}
13 %<book>\def\bxjs@clsname{bxjsbook}
14 %<report>\def\bxjs@clsname{bxjsreport}
15 %<slide>\def\bxjs@clsname{bxjsslide}

\ifjsc@needsp@tch [2016-08-22] 従来 jsclasses では、pLATEX や LATEX の不都合な点に対して、クラスファ
イル内で独自に対策を施していました。しかし、2016 年以降、コミュニティ版 pLATEX が次
第に対策コードをカーネル内に取り込むようになりました。そこで、新しい pLATEX カーネ
ルと衝突しないように、日付が古い場合だけパッチをあてる場合があります。この処理に使用
するフラグを定義します。
16 \newif\ifjsc@needsp@tch
17 \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。

```

18 \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。
19 \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。

```

20 \IfFileExists{bxjscompat.sty}{%
21   \let\jsAtEndOfClass\@gobble
22   \RequirePackage{bxjscompat}%
23 }{}

```

`\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`。

```

24 \let\jsArticle=a
25 \let\jsBook=b
26 \let\jsReport=r
27 \let\jsSlide=s
28 %<article>\let\jsDocClass\jsArticle
29 %<book>\let\jsDocClass\jsBook
30 %<report>\let\jsDocClass\jsReport
31 %<slide>\let\jsDocClass\jsSlide

```

`\jsEngine` [暗黙文字トークン] エンジン (T<sub>E</sub>X の種類) の種別：`j` = pT<sub>E</sub>X 系、`x` = X<sub>T</sub>L<sub>A</sub>T<sub>E</sub>X、`p` =

pdf $\TeX$  (含 DVI モード)、1 = Lua $\TeX$ 、J = NTT j $\TeX$ 、0 = Omega 系、n = 以上の何れでもない。

```

32 \let\jsEngine=n
33 \def\bxjs@test@engine#1#2{%
34   \edef\bxjs@tmpa{\string#1}%
35   \edef\bxjs@tmpb{\meaning#1}%
36   \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
37 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
38 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
39 \bxjs@test@engine\Omegaversion{\let\jsEngine=0}
40 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
41 \bxjs@test@engine\pdftexversion{\let\jsEngine=p}
42 \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) up $\TeX$  であるか。

```

43 \newif\ifjsWithupTeX
44 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
45   \jsWithupTeXtrue
46 \fi\fi
47 \let@if@jsc@uplatex\ifjsWithupTeX

```

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

```

48 \newif\ifjsWithpTeXng
49 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}

```

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

```

50 \newif\ifjsWitheTeX
51 \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 j $\TeX$  と Omega 系。

```

52 \let\bxjs@tmpa\relax
53 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
54 \ifx O\jsEngine \def\bxjs@tmpa{Omega}\fi
55 \ifx\bxjs@tmpa\relax \expandafter\@gobble
56 \else
57   \ClassError\bxjs@clsname
58   {The engine in use (\bxjs@tmpa) is not supported}
59   {It's a fatal error. I'll quit right now.}
60   \expandafter\@firstofone
61 \fi{\endinput\@@end}

```

Lua $\TeX$  の場合、本クラス用の Lua モジュールを用意する。

```

62 \ifx l\jsEngine
63   \directlua{ bxjs = {} }
64 \fi

```

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

```

65 \ifjsWitheTeX \let\bxjs@protected\protected
66 \else \let\bxjs@protected\@empty
67 \fi

\bxjs@robust@def 無引数の頑強な命令を定義する。
68 \ifjsWitheTeX
69 \def\bxjs@robust@def{\protected\def}
70 \else
71 \def\bxjs@robust@def{\DeclareRobustCommand*}
72 \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。
73 \newif\ifjsInPdfMode
74 \@nameuse{ImposeOldLuaTeXBehavior}
75 \let\bxjs@tmpa\PackageWarningNoLine
76 \let\PackageWarningNoLine\PackageInfo % suppress warning
77 \RequirePackage{ifpdf}
78 \let\PackageWarningNoLine\bxjs@tmpa
79 \@nameuse{RevokeOldLuaTeXBehavior}
80 \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 LATEX カーネルにおいて「Unicode を表す LATEX 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 LATEX を XYLATEX または LuaTeX
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。
81 \newif\ifbxjs@TUenc
82 \def\bxjs@tmpa{TU}\edef\bxjs@tmpb{\f@encoding}
83 \ifx\bxjs@tmpa\bxjs@tmpb
84 \bxjs@TUenctrue
85 \fi

\ifbxjs@old@hook@system [スイッチ] LATEX の新しいフック管理システムが未導入であるか。
※カーネルの 2020/10/01 版で導入された。
86 \newif\ifbxjs@old@hook@system
87 \ifl@t@r{fmtversion{2020/10/01}}{\bxjs@old@hook@systemtrue}

\bxjs@cond \bxjs@cond\ifXXX……\fi{〈真〉}{〈偽〉}
TeX の if-文 (\ifXXX……〈真〉\else〈偽〉\fi) を末尾呼出形式に変換するためのマクロ。
88 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
89 #1\expandafter\@firstoftwo
90 \else\expandafter\@secondoftwo
91 \fi}

\bxjs@cslet \bxjs@cslet{〈名前 1〉}\制御綴 :
92 \def\bxjs@cslet#1{%
93 \expandafter\let\csname#1\endcsname}

```

`\bxjs@csletcs` `\bxjs@csletcs{<名前 1>}{<名前 2>}` :

```
94 \def\bxjs@csletcs#1#2{%
95   \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。

```
96 \def\bxjs@catopt#1#2{%
97   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}
```

`\bxjs@ifplus` `\@ifstar` の + 版。

```
98 \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。

```
99 \let\c@bxjs@tempcnta\@tempcnta
100 \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）。

```
101 \def\jsSetQHLlength#1#2{%
102   \begingroup
103     \bxjs@parse@qh{#2}%
104     \ifx\bxjs@tmpb\relax
105       \setlength\@tempdima{#2}%
106       \xdef\bxjs@g@tmpa{\the\@tempdima}%
107     \else \global\let\bxjs@g@tmpa\bxjs@tmpb
108     \fi
109   \endgroup
110   #1=\bxjs@g@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。

```
111 \if j\jsEngine \def\bxjs@parse@qh@units{zw,zh}
112 \else \def\bxjs@parse@qh@units{trueQ,trueH,Q,H,zw,zh}
113 \fi
114 \def\bxjs@parse@qh#1{%
115   \let\bxjs@tmpb\relax
116   \for\bxjs@tmpa:=\bxjs@parse@qh@units\do{%
117     \ifx\bxjs@tmpb\relax
118       \edef\bxjs@next{{\bxjs@tmpa}{#1}}%
119       \expandafter\bxjs@parse@qh@a\csname bxjs@unit@\bxjs@tmpa\expandafter
120         \endcsname\bxjs@next
121     \fi}}
122 \def\bxjs@parse@qh@a#1#2#3{%
```



```

123 \def\bxjs@next##1#2\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}#1}%
124 \bxjs@next#3\@nil#2\@nil\@nnil}
125 \def\bxjs@parse@qh@b#1#2#3{%
126 \ifx\@nnil#2\@nnil\else
127 \ifx#3\relax
128 \ClassError\bxjs@clsname
129 {You cannot use '\bxjs@tmpa' here}{\@ehc}%
130 \def\bxjs@tmpb{0pt}%
131 \else
132 \@tempdimb#3\relax \@tempdimb#1\@tempdimb
133 \edef\bxjs@tmpb{\the\@tempdimb}%
134 \fi
135 \fi}

```

今の段階では Q/H だけが使用可能。

```

136 \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。

```

137 \@onlypreamble\bxjs@begin@document@hook
138 \let\bxjs@begin@document@hook\empty
139 \AtBeginDocument{\bxjs@begin@document@hook}

```

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

```

140 \@onlypreamble\bxjs@post@option@hook
141 \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。

```

142 \@onlypreamble\bxjs@pre@jadriver@hook
143 \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。)

```

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

```

一時的な手続き用の制御綴。

```

146 \@onlypreamble\bxjs@tmpdo
147 \@onlypreamble\bxjs@tmpdo@a
148 \@onlypreamble\bxjs@tmpdo@b
149 \@onlypreamble\bxjs@tmpdo@c
150 \@onlypreamble\bxjs@tmpdo@d

```

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

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

```

151 \if 1\jsEngine
152 \def\bxjs@tmpdo#1{%
153 \xdef\bxjs@pre@jadriver@hook{%

```

```

154 \bxjs@pre@jadriver@hook
155 \catcode`#1=\the\catcode`#1\relax}%
156 \catcode`#1=11\relax}
157 \@tfor\bxjs@tmpa:=和西曆\do
158 {\expandafter\bxjs@tmpdo\bxjs@tmpa}
159 \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。

160 \bxjs@robust@def\jsInhibitGlue{%
161 \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。

162 \if@compatibility
163 \ClassError\bxjs@clsname
164 {Something went chaotic!\MessageBreak
165 (How come '\string\documentstyle' is there?)\MessageBreak
166 I cannot go a single step further...}
167 {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
168 then there'll still be hope....}
169 \expandafter\@firstofone
170 \else \expandafter\@gobble
171 \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。
172 \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。
173 \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。
174 %<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。
175 %<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。

---

```
176 %<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。

---

```
177 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

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

---

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

```
178 \@onlypreamble\bxjs@setpaper
179 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
180 \newif\ifbxjs@iso@bsize
181 \DeclareOption{iso-bsize}{\bxjs@iso@bsizetrue}
182 \@onlypreamble\bxjs@setpaper@bsize
183 \def\bxjs@setpaper@bsize#1{\edef\bxjs@param@paper{%
184   b#1\ifbxjs@iso@bsize paper\else j\fi}}
185 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
186 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
187 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
188 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
189 \DeclareOption{b4paper}{\bxjs@setpaper@bsize{4}}
190 \DeclareOption{b5paper}{\bxjs@setpaper@bsize{5}}
191 \DeclareOption{b6paper}{\bxjs@setpaper@bsize{6}}
192 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
193 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
194 \DeclareOption{b4j}{\bxjs@setpaper{b4j}}
195 \DeclareOption{b5j}{\bxjs@setpaper{b5j}}
196 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
```

```

197 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
198 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
199 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
200 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

geometry の用紙サイズのオプション名を全てサポートする。

```

201 \@for\bxjs@tmpa:={%
202   a0,a1,a2,c0,c1,c2,c3,c4,c5,c6,ansia,ansib,ansic,ansid,ansie%
203 }\do{\edef\bxjs@next{%
204   \noexpand\DeclareOption{\bxjs@tmpa paper}%
205   {\noexpand\bxjs@setpaper{\bxjs@tmpa paper}}%
206 }\bxjs@next}
207 \DeclareOption{screen}{\bxjs@setpaper{screen}}

```

ただし b?paper は iso-bsize の指定に従い ISO と JIS の適切な方の B 列を選択する。

```

208 \@for\bxjs@tmpa:={0,1,2,3}\do{\edef\bxjs@next{%
209   \noexpand\DeclareOption{b\bxjs@tmpa paper}%
210   {\noexpand\bxjs@setpaper@bsize{\bxjs@tmpa}}%
211 }\bxjs@next}

```

Pandoc では用紙サイズ指定について「後ろに paper を付けた名前のオプション」を指定する。これに対処するため、後ろに paper をつけた形を用意する。さらに、「用紙サイズを custom とすると何も設定しない」ようにするため custompaper というオプションを用意する。

```

212 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
213 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
214 \DeclareOption{screenpaper}{\bxjs@setpaper{screen}}
215 \DeclareOption{custompaper}{}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

216 \newif\if@landscape
217 \@landscapefalse
218 \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。

```

219 \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。

```

220 %<!slide>\@slidefalse
221 %<slide>\@slidetrue

```

---

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

---

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

```
222 \newcommand{\@ptsize}{0}
223 \def\bxjs@param@basefontsize{10pt}
224 \def\jsUnusualPtSize{-20}
```

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

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

Q 単位の長さ指定をサポートするため `\jsSetQHLength` 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、`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。

```
226 \jsSetQHLength\@tempdima{#1}%
227 \edef\bxjs@param@basefontsize{\the\@tempdima}%
228 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
229 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
230 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
231 \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。

```
232 \def\bxjs@setjbasefontsize#1{%
233 \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。

```
234 \newif\ifjsc@mag
235 \newif\ifjsc@mag@xreal
236 %\let\jsc@magscale\@undefined
237 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
238 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
```

```

239 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
240 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
241 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
242 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
243 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
244 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
245 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
246 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
247 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
248 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
249 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
250 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
251 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
252 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
253 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
254 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
255 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

256 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@@usemag}
257 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@@nomag}
258 \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。

---

```

259 \if j\jsEngine
260 \hour\time \divide\hour by 60\relax
261 \@tempcnta\hour \multiply\@tempcnta 60\relax
262 \minute\time \advance\minute-\@tempcnta
263 \DeclareOption{tombow}{%
264   \tombowtrue \tombowdatetrue
265   \setlength{\@tombowwidth}{.1\p@}%
266   \@bannertoken{%
267     \jobname\space(\number\year-\two@digits\month-\two@digits\day
268     \space\two@digits\hour:\two@digits\minute)}}%
269   \maketombowbox}
270 \DeclareOption{tombo}{%
271   \tombowtrue \tombowdatefalse
272   \setlength{\@tombowwidth}{.1\p@}%
273   \maketombowbox}

```

274 \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。

```
275 \if j\jsEngine
276 \DeclareOption{mentuke}{%
277   \tombowtrue \tombowdatefalse
278   \setlength{\@tombowwidth}{\z@}%
279   \maketombowbox}
280 \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。

```
281 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
282 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
283 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
284 \DeclareOption{onecolumn}{\@twocolumnfalse}
285 \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。

```
286 \DeclareOption{titlepage}{\@titlepagetrue}
287 \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。

```
288 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
289 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
290 %<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。

```
291 \def\eqnarray{%
292   \stepcounter{equation}%
293   \def\@currentlabel{\p@equation\theequation}%
294   \global\@eqnswtrue
```

```

295 \m@th
296 \global\@eqcnt\z@
297 \tabskip\@centering
298 \let\@eqncr
299 $$\everycr{}\halign to\displaywidth\bgroup
300 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse
301 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
302 &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
303 &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss#\egroup
304 \tabskip\z@skip
305 \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。

```

306 \DeclareOption{leqno}{\input{leqno.clo}}
307 \DeclareOption{fleqn}{\input{fleqn.clo}}
308 % fleqn 用の eqnarray 環境の再定義
309 \def\eqnarray{%
310 \stepcounter{equation}%
311 \def\@currentlabel{\p@equation\theequation}%
312 \global\@eqnswtrue\m@th
313 \global\@eqcnt\z@
314 \tabskip\mathindent
315 \let\@=\@eqncr
316 \setlength\abovedisplayskip{\topsep}%
317 \ifvmode
318 \addtolength\abovedisplayskip{\partopsep}%
319 \fi
320 \addtolength\abovedisplayskip{\parskip}%
321 \setlength\belowdisplayskip{\abovedisplayskip}%
322 \setlength\belowdisplayshortskip{\abovedisplayskip}%
323 \setlength\abovedisplayshortskip{\abovedisplayskip}%
324 $$\everycr{}\halign to\linewidth% $$
325 \bgroup
326 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse
327 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
328 &\global\@eqcnt\tw@
329 $\displaystyle{##}$\hfil \tabskip\@centering
330 &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss#\egroup
331 \tabskip\z@skip\cr
332 }}

```

■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。

```

333 % \DeclareOption{openbib}{%
334 % \AtEndOfPackage{%
335 % \renewcommand\@openbib@code{%
336 % \advance\leftmargin\bibindent

```



```

337 %      \itemindent -\bibindent
338 %      \listparindent \itemindent
339 %      \parsep \z@}%
340 %      \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：

```

341 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。
342 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
343 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
344 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
345 \define@key{bxjs}{enablejfam}[true]{%
346   \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。

```

347 \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。

```

348 \newif\ifjsDraft
349 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
350 \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。

```
351 \newif\ifbxjs@papersize
352 \bxjs@papersizetrue
353 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
354 \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。

---

```
355 \newif\if@english
356 \@englishfalse
357 \DeclareOption{english}{\@englishttrue}
```

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

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

---

BXJS では当初から `bxjsreport` クラスが用意されている。

---

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

---

BXJS クラスでは、nojslogo を既定とする。

---

```
358 \newif\if@jslogo \@jslogofalse
359 \DeclareOption{jslogo}{\@jslogotrue}
360 \DeclareOption{nojslogo}{\@jslogofalse}
```

#### ■複合設定オプション 密

**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 T<sub>E</sub>X における「長さのスケール」の逆関数を求めるもの。例えば \bxjs@invscale\dimX{1.3} は \dimX=1.3\dimX の逆の演算を行う。  
 ※局所化の \begingroup~\endgroup について、以前は \group~\egroup を使っていたが、これだと数式モード中では空のサブ数式を生み出してしまうため修正した。  
 ※元の長さが 128 pt 以上の場合でも動作するように修正した。

```
361 \mathchardef\bxjs@isc@ll=128
362 \mathchardef\bxjs@isc@sl=259
363 \def\bxjs@isc@sl@h{65539 }
364 \def\bxjs@invscale#1#2{%
365   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
366   \ifdim\@tempdima<\bxjs@isc@ll\p@
367     \@tempcnta\@tempdima \multiply\@tempcnta\@ccclvi
368     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@ccclvi
369   \else
370     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
371     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
372   \fi
373   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
374   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
375   \@tempdimb\@tempcnta\@ne
376   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
377   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
378   \@whiledim\@tempdimb<\@tempdimc\do{%
379     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
380     \advance\@tempcntb\@ne \divide\@tempcntb\@tw@
381     \ifdim #2\@tempcntb>\@tempdima
382       \advance\@tempcntb\@ne \@tempdimc=\@tempcntb\@ne
383     \else \@tempdimb=\@tempcntb\@ne \fi}%
384   \xdef\bxjs@gtmpa{\the\@tempdimb}%
385   \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  $\text{\LaTeX}$  用の既定テンプレートを用いて他形式から  $\text{\LaTeX}$  (および PDF) 形式に変換する用途に最適化した設定を与える。

```
386 \DeclareOption{pandoc}{%
387   \bxjs@apply@pandoc@opt}
388 \@onlypreamble\bxjs@apply@pandoc@opt
389 \def\bxjs@apply@pandoc@opt{%
```

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。

```
390   \g@addto@macro\bxjs@post@option@hook{%
391     \bxjs@oldfontcommandstrue
392     \setkeys{bxjs}{ja=pandoc}%
393     \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)。

```
394   \ifx\bxjs@driver@opt\undefined
395     \def\bxjs@driver@opt{dvipdfmx}%
396     \bxjs@dvi@opttrue
397   \fi
398   \global\let\bxjs@apply@pandoc@opt\relax}
```

`pandoc+` オプションは、`pandoc` と同じ設定をした上で、さらに和文パラメタの先頭に `_plus` を追加する。

```
399 \DeclareOption{pandoc+}{%
400   \g@addto@macro\bxjs@post@option@hook{%
401     \edef\jsJaParam{\bxjs@catopt{_plus}\jsJaParam}}%
402   \ExecuteOptions{pandoc}}
```

---

## ■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。

```
403 %\let\bxjs@engine@given\undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
404 %\let\bxjs@engine@opt\undefined
```

エンジン明示指定のオプションの処理。

※ 0.9pre 版の暫定仕様と異なり、エンジン名は `...latex` に限定する。`xetex` や `pdftex` は一般的な  $\text{\LaTeX}$  の慣習に従って「ドライバの指定」とみなすべきだから。

```
405 \DeclareOption{autodetect-engine}{%
406   \let\bxjs@engine@given=*\%
407 \DeclareOption{latex}{%
```

```

408 \def\bxjs@engine@opt{latex}%
409 \let\bxjs@engine@given=n}
410 \DeclareOption{platex}{%
411 \def\bxjs@engine@opt{platex}%
412 \let\bxjs@engine@given=j}
413 \DeclareOption{uplatex}{%
414 \def\bxjs@engine@opt{uplatex}%
415 \let\bxjs@engine@given=u}
416 \DeclareOption{xelatex}{%
417 \def\bxjs@engine@opt{xelatex}%
418 \let\bxjs@engine@given=x}
419 \DeclareOption{pdflatex}{%
420 \def\bxjs@engine@opt{pdflatex}%
421 \let\bxjs@engine@given=p}
422 \DeclareOption{lualatex}{%
423 \def\bxjs@engine@opt{lualatex}%
424 \let\bxjs@engine@given=l}
425 \DeclareOption{platex-ng}{%
426 \def\bxjs@engine@opt{platex-ng}%
427 \let\bxjs@engine@given=g}
428 \DeclareOption{platex-ng*}{%
429 \def\bxjs@engine@opt{platex-ng*}%
430 \let\bxjs@platexng@nodrv=t%
431 \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。

```

432 %\let\bxjs@driver@given\@undefined
433 \let\bxjs@driver@@dvimode=0
434 \let\bxjs@driver@@dvi pdfmx=1
435 \let\bxjs@driver@@pdfmode=2
436 \let\bxjs@driver@@xetex=3
437 \let\bxjs@driver@@dvips=4
438 \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。

```

439 %\let\bxjs@driver@opt\@undefined
440 \DeclareOption{dvips}{%
441 \def\bxjs@driver@opt{dvips}%
442 \let\bxjs@driver@given\bxjs@driver@@dvips}
443 \DeclareOption{dviout}{%
444 \def\bxjs@driver@opt{dviout}%
445 \let\bxjs@driver@given\bxjs@driver@@dvimode}
446 \DeclareOption{xdvi}{%
447 \def\bxjs@driver@opt{xdvi}%
448 \let\bxjs@driver@given\bxjs@driver@@dvimode}
449 \DeclareOption{dvi pdfmx}{%
450 \def\bxjs@driver@opt{dvi pdfmx}%
451 \let\bxjs@driver@given\bxjs@driver@@dvi pdfmx}

```

```

452 \DeclareOption{nodvidriver}{%
453   \def\bxjs@driver@opt{nodvidriver}%
454   \let\bxjs@driver@given\bxjs@driver@@none}
455 \DeclareOption{pdftex}{%
456   \def\bxjs@driver@opt{pdftex}%
457   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
458 \DeclareOption{luatex}{%
459   \def\bxjs@driver@opt{luatex}%
460   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
461 \DeclareOption{xetex}{%
462   \def\bxjs@driver@opt{xetex}%
463   \let\bxjs@driver@given\bxjs@driver@@xetex}

dvipdfmx-if-dvi は 2.0 版より非推奨となった。
464 \DeclareOption{dvipdfmx-if-dvi}{\bxjs@depre@opt@do{dvipdfmx-if-dvi}{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？）。

\bxjs@depre@opt 非推奨のオプションについて警告を出す。

```

\bxjs@depre@opt@do 465 \@onlypreamble\bxjs@depre@opt
466 \def\bxjs@depre@opt#1#2{%
467   \ClassWarningNoLine\bxjs@clsname
468     {The old option '#1' is DEPRECATED\MessageBreak
469     and may be abolished in future!\MessageBreak
470     You should instead write:\MessageBreak
471     \space\space #2}}
472 \@onlypreamble\bxjs@depre@opt@do
473 \def\bxjs@depre@opt@do#1#2{%
474   \bxjs@depre@opt{#1}{#2}%
475   \setkeys{bxjs}{#2}}

```

\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。

```

476 \newif\ifbxjs@bigcode \bxjs@bigcodetrue

nobigcode / bigcode オプションの定義。
477 \DeclareOption{nobigcode}{%
478   \bxjs@bigcodefalse}
479 \DeclareOption{bigcode}{%
480   \bxjs@bigcodetrue}

```

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

```
481 \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。

```
482 \DeclareOption{nooldfontcommands}{%
483   \bxjs@oldfontcommandsfalse}
484 \DeclareOption{oldfontcommands}{%
485   \bxjs@oldfontcommandstrue}
```

■JS クラスのオプションで無効なもの  ltjssclasses に倣って警告を出す。

```
486 \DeclareOption{winjis}{%
487   \ClassWarningNoLine{bxjs@clsname
488     {This class does not support `winjis' option}}%
489 }
490 \DeclareOption{mingoth}{%
491   \ClassWarningNoLine{bxjs@clsname
492     {This class does not support `mingoth' option}}%
493 }
494 \DeclareOption{jis}{%
495   \ClassWarningNoLine{bxjs@clsname
496     {This class does not support `jis' option}}%
497 }
```

■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。

```
498 \DeclareOption*{%
499   \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
500   \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。

```
501 \def\bxjs@safe@setkeys#1#2{%
502   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
503   \setkeys{#1}{#2}%
504   \let\KV@errx\bxjs@save@KV@errx}
```

`\bxjs@declare@enum@option` `\bxjs@declare@enum@option{<オプション名>}{<enum 名>}{<初期値>}`

“<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。

```
505 \@onlypreamble\bxjs@declare@enum@option
506 \def\bxjs@declare@enum@option#1#2#3{%
507   \bxjs@csletcs{bxjs@#2}{bxjs@#2@@#3}%
508   \define@key{bxjs}{#1}{%
509     \expandafter\ifx\csname bxjs@#2@@#1\endcsname\relax
```

```

510     \bxjs@error@keyval{#1}{##1}%
511     \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@@#1}%
512     \fi}}

```

`\bxjs@declare@bool@option` `\bxjs@declare@bool@option{〈オプション名〉}{〈スイッチ名〉}{〈初期値〉}`  
 “〈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。

```

513 \@onlypreamble\bxjs@declare@bool@option
514 \def\bxjs@declare@bool@option#1#2#3{%
515   \csname newif\expandafter\endcsname\csname ifbxjs@#2\endcsname
516   \@nameuse{bxjs@#2#3}%
517   \define@key{bxjs}{#1}[true]{%
518     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
519     \bxjs@error@keyval{#1}{##1}%
520     \else \@nameuse{bxjs@#2##1}%
521     \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。

```

522 \def\bxjs@set@keyval#1#2#3{%
523   \bxjs@csletcs{bxjs@next}{bxjs@kv@#1@#2}%
524   \ifx\bxjs@next\relax
525     \bxjs@error@keyval{#1}{#2}%
526     #3%
527   \else \bxjs@next
528   \fi}
529 \@onlypreamble\bxjs@error@keyval
530 \def\bxjs@error@keyval#1#2{%
531   \ClassError\bxjs{clsname
532     {Invalid value '#2' for option #1}\@ehc}

```

`\jsScale` 〔実数値マクロ〕和文スケール値。

```

533 \def\jsScale{0.924715}

```

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

```

534 %\let\bxjs@base@opt\@undefined

```

`base` オプションの処理。

```

535 \define@key{bxjs}{base}{%
536   \edef\bxjs@base@opt{#1}%
537   \bxjs@setbasefontsize{#1}}
538 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

```

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

```

539 %\let\bxjs@jbase@opt\@undefined

```

`jbase` オプションの処理。

```

540 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
541 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}

```



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

```

542 %\let\bxjs@scale@opt\@undefined

scale オプションの処理。
543 \define@key{bxjs}{scale}{%
544   \edef\bxjs@scale@opt{#1}%
545   \let\jsScale\bxjs@scale@opt}
546 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

noscale オプションの処理。
547 \DeclareOption{noscale}{\bxjs@depre@opt@do{noscale}{scale=1}}

```

`\bxjs@param@mag` `mag` オプションの値。

```

548 \let\bxjs@param@mag\relax

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

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

```

`\bxjs@jadriver` 和文ドライバの名前。

```

551 \let\bxjs@jadriver\relax
552 %\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)。
553 \define@key{bxjs}{jadriver}{%
554   \bxjs@depre@opt{jadriver}{ja=#1}\edef\bxjs@jadriver@opt{#1}}
555 \define@key{bxjs}{ja}[\relax]{%
556   \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}

```

`\jsJaFont` 和文フォント設定の名前。

```

557 \let\jsJaFont\@empty

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

```

`\jsJaParam` 和文ドライバパラメタの文字列。

```

559 \let\jsJaParam\@empty

japaram オプションの処理。
560 \define@key{bxjs}{japaram}{%
561   \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}

引数をもつ pandoc・pandoc+ オプションは、その引数を和文パラメタの指定と見なす。
562 \define@key{bxjs}{pandoc}[]{%
563   \ExecuteOptions{pandoc}%

```

```

564 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}
565 \define@key{bxjs}{pandoc+}[]{}
566 \ExecuteOptions{pandoc+}%
567 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}

```

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

```

568 \let\bxjs@magstyle@@mag=m
569 \let\bxjs@magstyle@@real=r
570 \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。

```

571 \let\bxjs@magstyle@@usemag\bxjs@magstyle@@mag
572 \let\bxjs@magstyle@@nomag\bxjs@magstyle@@real
573 \bxjs@cslet{bxjs@magstyle@@nomag*}\bxjs@magstyle@@xreal

```

`\bxjs@magstyle@@default` は既定の値を表す。

```

574 \let\bxjs@magstyle@@default\bxjs@magstyle@@usemag
575 \ifx l\jsEngine \ifnum\luatexversion>86
576 \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
577 \fi\fi
578 \ifjsWithpTeXng
579 \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
580 \fi
581 \let\bxjs@magstyle\bxjs@magstyle@@default

```

magstyle オプションの処理。

```

582 \define@key{bxjs}{magstyle}{%
583 \bxjs@csletcs{bxjs@magstyle}{bxjs@magstyle@@#1}%
584 \ifx\bxjs@magstyle\relax
585 \bxjs@error@keyval{magstyle}{#1}%
586 \let\bxjs@magstyle\bxjs@magstyle@@default
587 \fi}

```

`\bxjs@geometry` geometry オプションの指定値。

```

588 \let\bxjs@geometry@@class=c
589 \let\bxjs@geometry@@user=u
590 \bxjs@declare@enum@option{geometry}{geometry}{class}

```

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

```

591 \bxjs@declare@bool@option{fancyhdr}{fancyhdr}{true}

```

`\ifbxjs@dvi@opt` dvi オプションが指定されたか。

```

592 \newif\ifbxjs@dvi@opt

```

DVI モードのドライバとドライバ種別との対応。

```

593 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
594 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips

```

```

595 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
596 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
597 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

    dvi オプションの処理。

598 \define@key{bxjs}{dvi}{%
599   \bxjs@csletcs{bxjs@tmpa}{bxjs@dvidriver@@#1}%
600   \ifx\bxjs@tmpa\relax
601     \bxjs@error@keyval{dvi}{#1}%
602   \else

    \bxjs@driver@given を未定義にしていることに注意。

603     \def\bxjs@driver@opt{#1}%
604     \let\bxjs@driver@given\@undefined
605     \bxjs@dvi@opttrue
606   \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。
607 \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。
608 \newif\ifbxjs@force@chapterabstract
609 %<book>\bxjs@force@chapterabstracttrue

    layout オプションの処理。

610 \@namedef{bxjs@kv@layout@v1}{%
611 %<book>\bxjs@layout@buggyhmargintrue
612 %<report>\bxjs@force@chapterabstracttrue
613 }
614 \@namedef{bxjs@kv@layout@v2}{%
615 %<book>\bxjs@layout@buggyhmarginfalse
616 %<report>\bxjs@force@chapterabstractfalse
617 }
618 \define@key{bxjs}{layout}{%
619   \bxjs@set@keyval{layout}{#1}{}}

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

620 %\let\bxjs@textwidth@limit@opt\@undefined
621 \define@key{bxjs}{textwidth-limit}{%
622   \bxjs@depre@opt{textwidth-limit}{textwidth=#1zw}%
623   \edef\bxjs@textwidth@limit@opt{#1}}

\bxjs@textwidth@opt  textwidth の指定値。

624 %\let\bxjs@textwidth@opt\@undefined
625 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
626 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

```

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

```

627 %\let\bxjs@number@of@lines@opt\@undefined
628 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
629 \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。

```

630 %\let\bxjs@paragraph@mark\@undefined
631 \define@key{bxjs}{paragraph-mark}{%
632   \edef\bxjs@paragraph@mark{#1}}

```

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

```

633 \bxjs@declare@bool@option{whole-zw-lines}{whole@zw@lines}{true}

```

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

```

634 \bxjs@declare@bool@option{jaspace-cmd}{jaspace@cmd}{true}
635 \define@key{bxjs}{xkanjiskip-cmd}[true]{\setkeys{bxjs}{jaspace-cmd=#1}}

```

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

```

636 \bxjs@declare@bool@option{fix-at-cmd}{fix@at@cmd}{true}

```

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

```

637 \bxjs@declare@bool@option{hyperref-enc}{hyperref@enc}{true}

```

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

```

638 \chardef\bxjs@everyparhook@@none=0
639 \chardef\bxjs@everyparhook@@compat=1
640 \chardef\bxjs@everyparhook@@modern=2
641 \bxjs@declare@enum@option{everyparhook}{everyparhook}{%
642   \if j\jsEngine compat\else modern\fi}

```

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

```

643 \chardef\bxjs@label@section@@none=0
644 \chardef\bxjs@label@section@@compat=1
645 \chardef\bxjs@label@section@@modern=2
646 \bxjs@declare@enum@option{label-section}{label@section}{compat}

```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```

647 \bxjs@declare@bool@option{use-zw}{usezw}{true}
648 \DeclareOption{nozw}{\bxjs@depre@opt@do{nozw}{use-zw=false}}
649 \DeclareOption{zw}{\bxjs@depre@opt@do{zw}{use-zw=true}}

```

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

```

650 \bxjs@declare@bool@option{disguise-js}{disguise@js}{true}
651 \DeclareOption{nojs}{\bxjs@depre@opt@do{nojs}{disguise-js=false}}
652 \DeclareOption{js}{\bxjs@depre@opt@do{js}{disguise-js=true}}

```

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

```

653 \bxjs@declare@bool@option{precise-text}{precisetext}{false}

```

```

654 \DeclareOption{noprecisetext}{\bxjs@depre@opt@do{noprecisetext}{precise-
    text=false}}
655 \DeclareOption{precisetext}{\bxjs@depre@opt@do{precisetext}{precise-
    text=true}}

```

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

```

656 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}{true}
657 \DeclareOption{nosimplejasetup}{\bxjs@depre@opt@do{nosimplejasetup}{simple-ja-
    setup=false}}
658 \DeclareOption{simplejasetup}{\bxjs@depre@opt@do{simplejasetup}{simple-ja-
    setup=true}}

```

\ifbxjs@plautopatch [スイッチ] plautopatch の指定値。

```

659 \bxjs@declare@bool@option{plautopatch}{plautopatch}{false}
660 \g@addto@macro\bxjs@plautopatchfalse{\def\bxjs@plautopatch@given{false}}

```

---

## ■ 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。

```

661 \let\bxjs@org@removeelement\@removeelement
662 \def\@removeelement#1#2#3{%
663   \def\reserved@a{#2}%
664   \ifx\reserved@a\@empty \let#3\@empty
665   \else \bxjs@org@removeelement{#1}{#2}{#3}%
666   \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。

```

667 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
668 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
669 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
670 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
671 \ProcessOptions\relax
672 \bxjs@post@option@hook

  後処理
673 \if@slide
674   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
675 \fi
676 \if@landscape
677   \setlength\@tempdima {\paperheight}

```

```

678 \setlength\paperheight{\paperwidth}
679 \setlength\paperwidth {\@tempdima}
680 \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。

```

681 \def\bxjs@tmpdo{%
682   \def\bxjs@tmpa{\@gobble}%
683   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
684   \let\@classoptionslist\bxjs@tmpa}
685 \def\bxjs@tmpdo@a#1,{%
686   \ifx\@nil#1\relax\else
687     \bxjs@tmpdo@b#1{\@nil
688     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
689     \expandafter\bxjs@tmpdo@a
690   \fi}
691 \def\bxjs@tmpdo@b#1#{\bxjs@tmpdo@c}
692 \def\bxjs@tmpdo@c#1\@nil{%
693   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
694 \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。

```

695 \@expandtwoargs\@removeelement
696 {papersize}\@classoptionslist\@classoptionslist
697 \@expandtwoargs\@removeelement
698 {10pt}\@classoptionslist\@classoptionslist
699 \@expandtwoargs\@removeelement
700 {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

ページに`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。

```
701 \ifx\bxjs@jadriver@opt\@undefined\else
702   \let\bxjs@jadriver\bxjs@jadriver@opt
703 \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。

```
704 \let\bxjs@tmpb\jsEngine
705 \ifx j\bxjs@tmpb\ifjsWithpTeXng
706   \let\bxjs@tmpb=g
707 \fi\fi
708 \ifx j\bxjs@tmpb\ifjsWithupTeX
709   \let\bxjs@tmpb=u
710 \fi\fi
711 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
712   \let\bxjs@tmpb=n
713 \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。)


```
714 \ifx *\bxjs@engine@given
715   \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。

```
716 \ifx j\bxjs@engine@given
717   \g@addto@macro\@classoptionslist{,platex}
718 \else\ifx u\bxjs@engine@given
719   \g@addto@macro\@classoptionslist{,uplatex}
720 \fi\fi
721 \fi
722 \ifx\bxjs@engine@given\@undefined\else
723   \ifx\bxjs@engine@given\bxjs@tmpb\else
724     \ClassError\bxjs{clsname
725       {Option '\bxjs@engine@opt' used on wrong engine}}\@ehc
726   \fi
727 \fi
```

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

```
728 \ifjsWithpTeXng
729   \g@addto@macro\@classoptionslist{,uplatex}
730 \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。

```
731 \@tempwattrue
```

```

732 \ifx \bxjs@driver@given\@undefined\else
733   \ifjsInPdfMode
734     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
735       \@tempswafalse
736     \fi
737   \else\ifx x\jsEngine
738     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
739       \@tempswafalse
740     \fi
741   \else
742     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
743       \@tempswafalse
744     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
745       \@tempswafalse
746     \fi\fi
747     \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
748       \@tempswafalse
749     \fi\fi
750   \fi\fi
751 \fi
752 \if@tempswa\else
753   \ClassError\bxjs@clsname
754   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
755 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

756 \ifjsInPdfMode \@tempswafalse
757 \else\ifx x\jsEngine \@tempswafalse
758 \else\ifjsWithpTeXng \@tempswafalse
759 \else \@tempswatrue
760 \fi\fi\fi
761 \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)。

```

762 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@opt\@undefined\else
763   \ClassWarningNoLine\bxjs@clsname
764   {No driver option is given}
765 \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。)

```

766 \ifbxjs@dvi@opt
767   \edef\bxjs@next{%
768     \let\noexpand\bxjs@driver@given
769     \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
770     \noexpand\g@addto@macro\noexpand\@classoptionslist
771     {,\bxjs@driver@opt}%
772   }\bxjs@next

```



```

773 \fi
774 \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。

```

775 \ifjsWithpTeXng
776 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
777 \let\bxjs@platexng@nodrv\@undefined
778 \else\ifx t\bxjs@platexng@nodrv\else
779 \g@addto@macro\@classoptionslist{,dvipdfmx}
780 \fi\fi
781 \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。

```

782 \ifx\bxjs@driver@given\bxjs@driver@@none
783 \bxjs@papersizefalse
784 \fi

```

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

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

```

785 \def\bxjs@@minimal{minimal}
786 \ifx\bxjs@jadriver\relax
787 \ifx j\jsEngine
788 \def\bxjs@jadriver{standard}
789 \else
790 \ClassWarningNoLine\bxjs@clsname
791 {The option 'ja' is MISSING!!\MessageBreak
792 So 'ja=minimal' is assumed as fallback, but\MessageBreak
793 such implicit setting is now DEPRECATED!\MessageBreak
794 You should write 'ja=minimal' explicitly,\MessageBreak
795 if it is intended}
796 \let\bxjs@jadriver\bxjs@@minimal
797 \fi
798 \fi

```

plautopatch が真の場合はここで plautopatch を読み込む。

※この時点で既に読み込まれているパッケージは、calc、keyval、ifpdf。

※ Pandoc モードでは plautopatch の既定値を真とする。

```

799 \def\bxjs@@pandoc{pandoc}
800 \ifx\bxjs@jadriver\bxjs@@pandoc \ifx\bxjs@plautopatch@given\@undefined
801 \ifjsWithTeX
802 \bxjs@plautopatchtrue
803 \fi\fi\fi
804 \ifx j\jsEngine \ifbxjs@plautopatch
805 \RequirePackage{plautopatch}[2018/08/22]%v0.3
806 \fi\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。

```
807 \ifx\bxjs@jadriver@opt\@undefined\else
808   \ifx\bxjs@engine@given\@undefined
809     \ClassError\bxjs@clsname
810     {An engine option must be explicitly given}%
811     {When you use a Japanese-driver you must specify a correct\MessageBreak
812       engine option.\MessageBreak\@ehc}
813 \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。)

```
814 \ifx\bxjs@magstyle@@default\bxjs@magstyle@@mag\else
815   \ifx\bxjs@magstyle\bxjs@magstyle@@mag
816     \let\bxjs@magstyle\bxjs@magstyle@@default
817     \ClassError\bxjs@clsname
818     {The engine does not support 'magstyle=usemag'}%
819     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
820       The default value 'nomag*' is used instead.\MessageBreak \@ehc}
821   \fi
822 \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の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。

```
823 \ifx\bxjs@jbase@opt\@undefined\else
824   \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。

```
825   \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
826   \bxjs@invscale\@tempdima\jsScale
827   \bxjs@setbasefontsize{\@tempdima}%
828 \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。

```
829   \ifx\bxjs@scale@opt\@undefined\else
830     \ClassWarningNoLine\bxjs@clsname
831     {Redundant 'scale' option is ignored}%
832   \fi
833   \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
834   \@tempdimb=\bxjs@param@basefontsize\relax
835   \edef\jsScale{\stripopt\@tempdimb}%
836   \bxjs@invscale\@tempdima\jsScale
837   \edef\jsScale{\stripopt\@tempdima}%
838 \fi
839 \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。

```
840 \let\Cjascale\jsScale
```

---

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

```
841 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
842 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
843 \catcode\@tempcnta\active
844 \advance\@tempcnta\@ne
845 \repeat
846 \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。

```
847 \ifbxjs@disguise@js
848 %<book|report>\def\bxjs@js@clsname{jsbook}
849 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
850 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
851 \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。

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

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

```
853 \ifbxjs@oldfontcommands
854 \AtEndOfClass{\allowoldfontcommands}
855 \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  $\mathrm{pLATEX}\ 2_{\epsilon}$  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。

[2020-10-04] L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 2020-10-01 でカーネルの `\shipout` コードが拡張され `\AtBeginDvi` の実行タイミングが変化したので、この時点で発行する `\special` の中身を展開しておくようにしました。こうしないと、用紙サイズ設定を間違ってしまう (Issue #72)。

---

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。

---

```
856 \newdimen\stockwidth \newdimen\stockheight
857 \begingroup\expandafter\expandafter\expandafter\endgroup
858 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
859 % \newdimen\stockwidth \newdimen\stockheight
860 \setlength{\stockwidth}{\paperwidth}
861 \setlength{\stockheight}{\paperheight}
862 \advance \stockwidth 2in
863 \advance \stockheight 2in
864 \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。

```
865 %<slide>\def\n@baseline{13}%
866 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
867 %<!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。

```
868 \ifx\bxjs@magstyle\bxjs@magstyle@@mag
869 \jsc@magtrue
870 \else\ifx\bxjs@magstyle\bxjs@magstyle@@xreal
871 \jsc@mag@xrealtrue
872 \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` に直したので、`geometry` パッケージと共存できると思います。なお、新ドキュメントクラス側で 10pt 以外にする場合の注意：

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

---

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

```

873 \ifx\bxjs@param@mag\relax
874   \@tempdima=\bxjs@param@basefontsize
875   \advance\@tempdima.001pt \multiply\@tempdima25
876   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
877   \edef\bxjs@param@mag{\the\@tempcnta}
878 \else
879 % mag 値が直接指定された場合
880 \bxjs@gset@tempcnta{\bxjs@param@mag}
881 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
882 % 有効な mag 値の範囲は 1--32768
883 \edef\bxjs@param@mag{\the\@tempcnta}
884 \advance\@tempcnta100000
885 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
886 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
887 \edef\bxjs@param@basefontsize{\the\@tempdima}
888 \fi
889 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
890 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
891 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
892 \edef\jsc@magscale{\strip@pt\@tempdima}
893 \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。

```

894 \newdimen\jsc@mpt
895 \newdimen\jsc@mmm
896 \ifjsc@mag
897   \jsc@mpt=1\p@
898   \jsc@mmm=1mm
899   \catcode\?=9 % \p@? read as \p@
900 \else
901   \jsc@mpt=\jsc@magscale\p@
902   \jsc@mmm=\jsc@magscale mm
903   \catcode\?=11 \let\p@?\jsc@mpt
904 \fi

```

```

905 \chardef\bxjs@qmcc=\catcode`\?\relax
906 \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。

```

907 \newdimen\jsZw
908 \jsZw=10\jsc@empt \jsZw=\jsScale\jsZw
909 \ifbxjs@usezw
910 \providecommand*zw{\jsZw}
911 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

913 \ifjsc@mag@xreal
914 \RequirePackage{type1cm}
915 \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

---

```

916 \ifbxjs@TUenc
917 \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
918 \else
919 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
920 \fi
921 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
922 \let\jsc@get@external@font\get@external@font
923 \def\get@external@font{%
924 \jsc@preadjust@extract@font
925 \jsc@get@external@font}
926 \def\jsc@fstrunc#1{%
927 \edef\jsc@tmpa{\strip@pt#1}%
928 \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
929 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
930 \if#5*\else
931 \edef\jsc@tmpa{#1%
932 \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
933 \fi}
934 \def\jsc@preadjust@extract@font{%
935 \let\jsc@req@size\f@size
936 \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
937 \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
938 \let\jsc@ref@size\jsc@tmpa
939 \let\f@size\jsc@ref@size}
940 \def\execute@size@function#1{%
941 \let\jsc@cref@size\f@size
942 \let\f@size\jsc@req@size

```

```

943 \csname s@fct@#1\endcsname}
944 \let\jsc@DeclareErrorFont\DeclareErrorFont
945 \def\DeclareErrorFont#1#2#3#4#5{%
946 \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
947 \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
948 \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
949 \def\gen@sfcnt{%
950 \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
951 \empty@sfcnt}
952 \def\genb@sfcnt{%
953 \edef\mandatory@arg{%
954 \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@{}}%
955 \empty@sfcnt}
956 \ifxjs@TUenc\else
957 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
958 \fi
959 \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 960 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 961 %\def\jsc@medskip{\vspace\jsc@medskipamount}
962 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

`\jsc@smallskipamount`

```

\jsc@medskipamount 963 \newskip\jsc@smallskipamount
\jsc@bigskipamount 964 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
965 %\newskip\jsc@medskipamount
966 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
967 %\newskip\jsc@bigskipamount
968 %\jsc@bigskipamoun =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。

```

969 % \ifpapersize
970 % \setlength{\@tempdima}{\paperwidth}
971 % \setlength{\@tempdimb}{\paperheight}

```

```

972 % \iftombow
973 % \advance \@tempdima 2truein
974 % \advance \@tempdimb 2truein
975 % \fi
976 % \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
977 % \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  $\text{\def\@{\spacefactor\@m}}$  という定義（ $\text{\@m}$  は 1000）では  $\text{I watch TV\@.}$  と書くと  $\text{V}$  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、 $\text{I watch TV.\@}$  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  $\text{\{}}$  を補いました。

---

BXJS クラスでの変更点：

- $\text{fix-at-cmd}$  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  $\text{sfcode}$  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。
- 

```

978 \chardef\bxjs@periodchar=`\ .
979 \bxjs@protected\def\bxjs@SE{%
980 \ifnum\spacefactor<\@m \spacefactor\@m
981 \else \spacefactor\sfcode\bxjs@periodchar
982 \fi}
983 \ifbxjs@fix@at@cmd
984 \def\@{\bxjs@SE{}}
985 \fi

```

### 4 フォントサイズ

フォントサイズを変える命令（ $\text{\normalsize}$ ,  $\text{\small}$  など）の実際の挙動の設定は、三つの引数をとる命令  $\text{\@setfontsize}$  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

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



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

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

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

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

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

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

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

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

```

986 \def\bxjs@tmpa{\def\setfontsize##1##2##3}
987 \expandafter\bxjs@tmpa\expandafter{%
988   \setfontsize{#1}{#2}{#3}%
989 % 末尾にコードを追加
990   \expandafter\def\expandafter\size@update\expandafter{%
991     \size@update
992     \jsFontSizeChanged}%
993 }
```

`\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。

```

994 \newcommand*\jsFontSizeChanged{%
995   \jsZw=\f@size\p@
996   \jsZw=\jsScale \jsZw
997   \ifdim\parindent>\z@
998     \if@english \parindent=1em
999     \else       \parindent=1\jsZw
1000   \fi
1001   \fi\relax
1002   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
1003 \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。

```
1004 \ifjsc@mag
1005   \let\jsc@setfontsize\@setfontsize
1006 \else
1007   \def\jsc@setfontsize#1#2#3{%
1008     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
1009 % microtype 対策
1010 \ifjsWithTeX\if j\jsEngine\else
1011   \def\jsc@setfontsize#1#2#3{%
1012     \edef\bxjs@sfs@next{%
1013       \unexpanded{\@setfontsize#1}%
1014       {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
1015     }\bxjs@sfs@next}
1016 \fi\fi
1017 \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。

---

```
1018 \emergencystretch 3\jsZw
```

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

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

`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対して、しっぽ愛好家さんが次の修正を教えてくださいました。

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

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

---

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

---

```
1019 \newif\ifnarrowbaselines
```

```
1020 \if@english
```

```

1021 \narrowbaselinestru
1022 \fi
1023 \def\narrowbaselines{%
1024 \narrowbaselinestru
1025 \skip0=\abovedisplayskip
1026 \skip2=\abovedisplayshortskip
1027 \skip4=\belowdisplayskip
1028 \skip6=\belowdisplayshortskip
1029 % 一時的に警告を無効化する
1030 \let\bxjs@save@nomath\@nomath
1031 \let\@nomath\@gobble
1032 \@currsize\selectfont
1033 \let\@nomath\bxjs@save@nomath
1034 \abovedisplayskip=\skip0
1035 \abovedisplayshortskip=\skip2
1036 \belowdisplayskip=\skip4
1037 \belowdisplayshortskip=\skip6\relax}
1038 \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。

```

1039 \def\bxjs@if@narrowbaselines{%
1040 \ifnarrowbaselines\expandafter\@firstoftwo
1041 \else \expandafter\@secondoftwo
1042 \fi
1043 }

```

---

`\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。

---

```

1044 \renewcommand{\normalsize}{%
1045 \bxjs@if@narrowbaselines{%
1046 \jsc@setfontsize\normalsize\@xpt\@xipt
1047 }{%else
1048 \jsc@setfontsize\normalsize\@xpt\@n@baseline}%
1049 }%

```

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

[2003-02-16] ちょっと変えました。

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

```
1050 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1051 \abovedisplayshortskip \z@ \@plus3\p@?
1052 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1053 \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。

```
1054 \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。

```
1055 %</class>
1056 %<*class|minijs>
1057 %% initialize
1058 \normalsize
1059 %</class|minijs>
1060 %<*class>
```

`\Cht` 基準となる長さの設定をします。pL<sub>A</sub>T<sub>E</sub>X 2<sub>ε</sub> カーネル (`plfonts.dtx`) で宣言されているパラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (1zw) です。  
`\Cwd` [2017-08-31] 基準とする文字を「全角空白」(EUC コード 0xA1A1) から「漢」(JIS コード 0x3441) へ変更しました。  
`\Cvs`

---

`\Chs`

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

```
1061 \ifx\Cht\@undefined \newdimen\Cht \fi
1062 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1063 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1064 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1065 \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。

---

```
1066 \setlength\Cht{0.88\jsZw}
1067 \setlength\Cdp{0.12\jsZw}
1068 \setlength\Cwd{1\jsZw}
1069 \setlength\Cvs{\baselineskip}
1070 \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。

---

```

1071 \newcommand{\small}{%
1072   \bxjs@if@narrowbaselines{%
1073     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1074     %<kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1075   }{%else
1076     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1077     %<kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1078   }%
1079   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1080   \abovedisplayshortskip \z@ \@plus3\p@?
1081   \belowdisplayskip \abovedisplayskip
1082   \belowdisplayshortskip \belowdisplayskip
1083   \def\@listi{\leftmargin\leftmargini
1084             \topsep \z@
1085             \parsep \z@
1086             \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。

```

1087 \newcommand{\footnotesize}{%
1088   \bxjs@if@narrowbaselines{%
1089     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1090     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1091   }{%else
1092     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1093     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1094   }%
1095   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1096   \abovedisplayshortskip \z@ \@plus2\p@?
1097   \belowdisplayskip \abovedisplayskip
1098   \belowdisplayshortskip \belowdisplayskip
1099   \def\@listi{\leftmargin\leftmargini
1100             \topsep \z@
1101             \parsep \z@
1102             \itemsep \parsep}}

```

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

\LARGE [2004-11-03] \HUGE を追加。

```

\huge 1103 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1104 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 1105 \if@twocolumn
\HUGE

```

```

1106 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1107 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1108 \else
1109 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
1110 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1111 \fi
1112 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1113 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1114 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1115 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1116 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1117 \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。

```

1118 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}

```

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

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

```

1119 % \newcommand{\headfont}{\bfseries}
1120 \newcommand{\headfont}{\sffamily}
1121 % \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。

```

1122 %<!kiyou>\setlength\columnsep{2\Cwd}
1123 %<kiyou>\setlength\columnsep{28truebp}
1124 \setlength\columnseprule{\z@}

```

### ■段落

`\lineskip` 上下の行の文字が `\lineskiplimit` より接近したら、`\lineskip` より近づかないようにし  
`\normallineskip`  
`\lineskiplimit`  
`\normallineskiplimit`

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。

```
1125 \setlength\lineskip{1\jsc@empt}
1126 \setlength\normallineskip{1\jsc@empt}
1127 \setlength\lineskiplimit{1\jsc@empt}
1128 \setlength\normallineskiplimit{1\jsc@empt}
```

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

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

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

```
1129 \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。

```
1130 \setlength\parskip{\z@}
1131 \if@slide
1132   \setlength\parindent{0\p@}
1133 \else
1134   \setlength\parindent{1\Cwd}
1135 \fi
```

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

```
\@highpenalty 1136 \@lowpenalty 51
1137 \@medpenalty 151
1138 \@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。

```
1139 % \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。

```
1140 % \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。

```

1141 \@onlypreamble\bxjs@bd@pre@geometry@hook
1142 \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。

```

1143 \ifjsc@mag
1144 \mag=\bxjs@param@mag
1145 \fi
1146 \setlength{\topskip}{10\p@?}

```

`\jsSetQHLLength` のための和文単位の定義。

```

1147 \def\bxjs@unit@trueQ{0.25trueem}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1148 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw

```

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

```

1149 \@tempswafalse
1150 \def\bxjs@tmpdo{\@ifnextchar\bgroup\bxjs@tmpdo@a\remove@to@nnil}
1151 \def\bxjs@tmpdo@a#1{\edef\bxjs@tmpa{#1}%
1152   \@ifnextchar\bgroup\bxjs@tmpdo@b\remove@to@nnil}
1153 \def\bxjs@tmpdo@b#1{\edef\bxjs@tmpa{\bxjs@tmpa,#1}%
1154   \@ifnextchar\@nnil\bxjs@tmpdo@c\remove@to@nnil}
1155 \def\bxjs@tmpdo@c\@nnil{\@tempswatrue
1156   \edef\bxjs@param@paper{papersize={\bxjs@tmpa}}}
1157 \expandafter\bxjs@tmpdo\bxjs@param@paper\@nnil

```

`W,H` の形式について。

```

1158 \if@tempswa\else
1159   \def\bxjs@tmpa{\@nil,\@nil}
1160   \def\bxjs@tmpdo#1,#2,#3\@nnil{%
1161     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1162       \@tempswatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1163   \expandafter\bxjs@tmpdo\bxjs@param@paper,\@nil,\@nil\@nnil
1164 \fi

```

`W*H` の形式について。

```

1165 \if@tempswa\else
1166   \def\bxjs@tmpa{\@nil*\@nil}
1167   \def\bxjs@tmpdo#1*#2*#3\@nnil{%
1168     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1169       \@tempswatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1170   \expandafter\bxjs@tmpdo\bxjs@param@paper*\@nil*\@nil\@nnil
1171 \fi

```

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

```

1172 \edef\bxjs@layout@paper{%
1173   \ifjsc@mag truedimen,\fi
1174   \if@landscape landscape,\fi
1175   \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。

```
1176 %<*article|report>
1177 \def\bxjs@layout@base{%
1178   headheight=\topskip,footskip=0.03367\paperheight,%
1179   headsep=\footskip-\topskip,includeheadfoot,%
1180 }
1181 \edef\bxjs@layout{\bxjs@layout@base
1182   hscale=0.76,hmarginratio=1:1,%
1183   vscale=0.83,vmarginratio=1:1,%
1184 }
1185 %</article|report>
1186 %<*book>
1187 \def\bxjs@layout@base{%
1188   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1189 }
1190 \ifbxjs@layout@buggyhmargin      %---
1191 % アレ
1192 \edef\bxjs@layout{\bxjs@layout@base
1193   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1194   vscale=0.83,vmarginratio=1:1,%
1195 }
1196 \else                            %---
1197 % 非アレ
1198 \edef\bxjs@layout{\bxjs@layout@base
1199   hmargin=18\jsc@mmm,%
1200   vscale=0.83,vmarginratio=1:1,%
1201 }
1202 \fi                               %---
1203 %</book>
1204 %<*slide>
1205 \def\bxjs@layout@base{%
1206   noheadfoot,%
1207 }
1208 \edef\bxjs@layout{\bxjs@layout@base
1209   hscale=0.9,hmarginratio=1:1,%
1210   vscale=0.944,vmarginratio=1:1,%
1211 }
1212 %</slide>
```

`textwidth` オプションの設定を反映する。

```
1213 %<!*book>
1214 \ifx\bxjs@textwidth@opt\undefined\else
1215   \jsSetQHLlength\@tempdima{\bxjs@textwidth@opt}
1216   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1217 \fi
1218 %</*!book>
1219 \ifx\bxjs@number@of@lines@opt\undefined\else
1220   \bxjs@egset@tempcnta{\bxjs@number@of@lines@opt}
1221   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
```

1222 \fi

\fullwidth 〔寸法レジスタ〕 ヘッダ・フッタ領域の横幅。

1223 \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。

1224 %<\*book>

1225 \newcommand\jsTextWidthLimit{40}

1226 \@tempdima=\jsTextWidthLimit\Cwd

1227 \ifx\bxjs@textwidth@limit@opt\@undefined\else

1228 \bxjs@gset@tempcnta{\bxjs@textwidth@limit@opt}

1229 \@tempdima=\@tempcnta\Cwd

1230 \fi

1231 \ifx\bxjs@textwidth@opt\@undefined\else

1232 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}

1233 \fi

1234 \edef\bxjs@textwidth@limit{\the\@tempdima}

1235 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else

1236 \bxjs@invscale\@tempdima{\strip@pt\Cwd}

1237 \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}

1238 \fi

1239 %</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。

1240 \def\bxjs@preproc@layout{%

1241 \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mppt}

\bxjs@postproc@layout geometry の後処理。

1242 \def\bxjs@postproc@layout{%

geometry のドライバを再設定する。

1243 \ifx\bxjs@geometry@driver\relax\else

1244 \let\Gm@driver\bxjs@geometry@driver

1245 \fi

\ht\strutbox の値を元に戻す。

1246 \ht\strutbox=\bxjs@save@ht@strutbox\relax

\textwidth の値を補正する。

1247 \ifbxjs@whole@zw@lines

1248 \@tempdimb=\textwidth

1249 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi

1250 \advance\textwidth.005pt\relax

1251 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima

```

1252 \advance\@tempdimb-\textwidth
1253 \advance\oddsidemargin 0.5\@tempdimb
1254 \advance\evensidemargin 0.5\@tempdimb
1255 \fi
1256 \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。

```

1257 %<*book>
1258 \@tempdima=\bxjs@textwidth@limit\relax
1259 \ifbxjs@whole@zw@lines
1260 \advance\@tempdima.005pt\relax
1261 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1262 \fi
1263 \ifdim\textwidth>\@tempdima
1264 \textwidth=\@tempdima
1265 \addtolength\evensidemargin{\fullwidth-\textwidth}
1266 \fi
1267 %</book>

```

\textheight 関連の調整。

```

1268 \@tempdimb=\textheight
1269 \advance\textheight-\topskip
1270 \advance\textheight.005pt\relax
1271 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1272 \advance\textheight\topskip
1273 \advance\@tempdimb-\textheight
1274 \advance\topmargin0.5\@tempdimb

```

\headheight 関連の調整。

```

1275 \@tempdima=\topskip
1276 \advance\headheight\@tempdima
1277 \advance\topmargin-\@tempdima

```

marginpar 関連の調整。

```

1278 \setlength\marginparsep{\columnsep}
1279 \setlength\marginparpush{\baselineskip}
1280 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1281 -\textwidth-10\jsc@mmm-\marginparsep}
1282 \ifbxjs@whole@zw@lines
1283 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1284 \fi

```

連動する変数。

```

1285 \maxdepth=.5\topskip
1286 \stockwidth=\paperwidth
1287 \stockheight=\paperheight
1288 }

```

\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。

```
1289 \edef\jsGeometryOptions{%
1290   \bxjs@layout@paper,\bxjs@layout}
```

---

## ■`geometry` パッケージを読み込む

---

ムニャムニャ。

```
1291 \def\bxjs@geometry@name{geometry}
1292 \ifbxjs@old@hook@system
1293   \let\bxjs@apply@bd@pre@geometry@hook\AtBeginDocument
1294 \else
1295   \def\bxjs@apply@bd@pre@geometry@hook{%
1296     \AddToHook{begindocument}{\bxjs@geometry@name}}
1297 \fi
```

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

```
1298 \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。

```
1299 \ifbxjs@papersize
1300   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1301     \PassOptionsToPackage{dvipdfm}{geometry}
1302   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1303     \PassOptionsToPackage{dvipdfm}{geometry}
1304   \fi\fi
1305   \let\bxPapersizeSpecialDone=t
1306 \else
1307   \PassOptionsToPackage{driver=none}{geometry}
1308 \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。

```
1309 \bxjs@apply@bd@pre@geometry@hook
1310 \bxjs@apply@bd@pre@geometry@hook{\bxjs@bd@pre@geometry@hook}
1311 \bxjs@apply@bd@pre@geometry@hook{\ImposeOldLuaTeXBehavior}
1312 \bxjs@preproc@layout
1313 \edef\bxjs@next{%
1314   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1315 }\bxjs@next
1316 \bxjs@apply@bd@pre@geometry@hook{\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。

```
1317 \let\bxjs@geometry@driver\Gm@driver
1318 \bxjs@postproc@layout
```

geometry のドライバ自動判別に対する前処理。

```
1319 \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。

```
1320 \ifpackagelater{geometry}{2010/02/12}{\else
1321 \PackageError\bxjs@clsname
1322 {Your 'geometry' package is too old (< v5.0)}}%
1323 {\@ehc}%
1324 \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。

```
1325 \ifjsWithpTeXng
1326 \ifx\Gm@driver\empty
1327 \def\Gm@driver{pdftex}%
1328 \fi
1329 \fi}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```
1330 \def\setpagelayout{%
1331 \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{\else
1332 \ifstar{\bxjs@setpagelayout@a@one}{\bxjs@setpagelayout@a@z@}}
1333 \def\bxjs@setpagelayout@a#1#2{%
1334 \ifcase#1% modify
1335 \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1336 \or% reset(*)
1337 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1338 \or% semireset(+)
1339 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1340 \fi
1341 \bxjs@preproc@layout
1342 \edef\bxjs@next{%
1343 \noexpand\geometry{\bxjs@next}%
1344 }\bxjs@next
1345 \bxjs@postproc@layout}
```

---

## ■ geometry パッケージを読み込まない

---

geometry=user の場合の処理。

```
1346 \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。

```
1347 \g@addto@macro\bxjs@begin@document@hook{%
1348   \ifdim\textwidth=.5\maxdimen
1349     \ClassError\bxjs@clsname
1350       {Page layout is not properly set}%
1351       {\@ehd}%
1352   \fi}
1353 \def\jsUseMinimalPageLayout{%
1354   \setlength{\textwidth}{6.5in}%
1355   \setlength{\textheight}{8in}}
```

\setpagelayout はとりあえず無効にしておく。

```
1356 \let\bxjs@geometry@driver\relax
1357 \def\setpagelayout{%
1358   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1359     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1360 \def\bxjs@pagelayout@a#1{%
1361   \ClassError\bxjs@clsname
1362     {Command '\string\setpagelayout' is not supported,\MessageBreak
1363     because 'geometry' value is not 'class'}\@eha}
1364 %
1365 \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。

---

```
1366 %<*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。

```

1367 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1368 \if@slide
1369   \setlength\headheight{0\jsc@empt}
1370 \else
1371   \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1372 \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<sup>3</sup>, 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。

```

1373 %<*article|kiyou>
1374 \if@slide
1375   \setlength\footskip{0pt}
1376 \else
1377   \setlength\footskip{0.03367\paperheight}
1378   \ifdim\footskip<\baselineskip
1379     \setlength\footskip{\baselineskip}
1380   \fi
1381 \fi
1382 %</article|kiyou>
1383 %<jspf>\setlength\footskip{9\jsc@mmm}
1384 %<*book>
1385 \if@report
1386   \setlength\footskip{0.03367\paperheight}
1387   \ifdim\footskip<\baselineskip
1388     \setlength\footskip{\baselineskip}
1389   \fi
1390 \else
1391   \setlength\footskip{0pt}
1392 \fi
1393 %</book>
1394 %<*report>
1395 \setlength\footskip{0.03367\paperheight}
1396 \ifdim\footskip<\baselineskip
1397   \setlength\footskip{\baselineskip}
1398 \fi
1399 %</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)。

```

1400 %<*article>
1401 \if@slide

```

```

1402 \setlength\headsep{0\jsc@empt}
1403 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1404 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1405 \else
1406 \setlength\headsep{\footskip}
1407 \addtolength\headsep{-\topskip}
1408 \fi
1409 %</article>
1410 %<*book>
1411 \if@report
1412 \setlength\headsep{\footskip}
1413 \addtolength\headsep{-\topskip}
1414 \else
1415 \setlength\headsep{6\jsc@mmm}
1416 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1417 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1418 \fi
1419 %</book>
1420 %<*report>
1421 \setlength\headsep{\footskip}
1422 \addtolength\headsep{-\topskip}
1423 %</report>
1424 %<*jspf>
1425 \setlength\headsep{9\jsc@mmm}
1426 \addtolength\headsep{-\topskip}
1427 %</jspf>
1428 %<*kiyou>
1429 \setlength\headheight{0\jsc@empt}
1430 \setlength\headsep{0\jsc@empt}
1431 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1432 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1433 %</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。

```

1434 \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。

```

1435 \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。

```
1436 %<*article>
1437 \if@slide
1438   \setlength\fullwidth{0.9\paperwidth}
1439 \else
1440   \setlength\fullwidth{0.76\paperwidth}
1441 \fi
1442 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1443 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1444 \setlength\textwidth{\fullwidth}
1445 %</article>
1446 %<*book>
1447 \if@report
1448   \setlength\fullwidth{0.76\paperwidth}
1449 \else
1450   \setlength\fullwidth{\paperwidth}
1451   \addtolength\fullwidth{-36\jsc@mmm}
1452 \fi
1453 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1454 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1455 \setlength\textwidth{\fullwidth}
1456 \if@report \else
1457   \if@twocolumn \else
1458     \ifdim \fullwidth>40zw
1459       \setlength\textwidth{40zw}
1460     \fi
1461   \fi
1462 \fi
1463 %</book>
1464 %<*report>
1465 \setlength\fullwidth{0.76\paperwidth}
1466 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1467 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1468 \setlength\textwidth{\fullwidth}
1469 %</report>
1470 %<*jspf>
1471 \setlength\fullwidth{50zw}
1472 \addtolength\fullwidth{8\jsc@mmm}
1473 \setlength\textwidth{\fullwidth}
1474 %</jspf>
1475 %<*kiyou>
1476 \setlength\fullwidth{48zw}
```

```

1477 \addtolength\fullwidth{\columnsep}
1478 \setlength\textwidth{\fullwidth}
1479 %</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)。

```

1480 %<*article|book|report>
1481 \if@slide
1482   \setlength{\textheight}{0.95\paperheight}
1483 \else
1484   \setlength{\textheight}{0.83\paperheight}
1485 \fi
1486 \addtolength{\textheight}{-10\jsc@mp}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1487 \addtolength{\textheight}{-\headsep}
1488 \addtolength{\textheight}{-\footskip}
1489 \addtolength{\textheight}{-\topskip}
1490 \divide\textheight\baselineskip
1491 \multiply\textheight\baselineskip
1492 %</article|book|report>
1493 %<jspf>\setlength{\textheight}{51\baselineskip}
1494 %<kiyou>\setlength{\textheight}{47\baselineskip}
1495 \addtolength{\textheight}{\topskip}
1496 \addtolength{\textheight}{0.1\jsc@mp}
1497 %<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。

```

1498 \def\flushbottom{%
1499   \def\@textbottom{\vskip \z@ \@plus.1\jcs@empt}%
1500   \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` みどうしの最小の間隔です。

```

1501 \setlength\marginparsep{\columnsep}
1502 \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<sub>A</sub>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。

```

1503 \setlength{\oddsidemargin}{\paperwidth}
1504 \addtolength{\oddsidemargin}{-\fullwidth}
1505 \setlength{\oddsidemargin}{.5\oddsidemargin}
1506 \iftombow
1507   \addtolength{\oddsidemargin}{-1in}
1508 \else
1509   \addtolength{\oddsidemargin}{-\inv@mag in}
1510 \fi
1511 \setlength{\evensidemargin}{\oddsidemargin}
1512 \if@mparswitch
1513   \addtolength{\evensidemargin}{\fullwidth}
1514   \addtolength{\evensidemargin}{-\textwidth}
1515 \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。

```

1516 \setlength\marginparwidth{\paperwidth}
1517 \addtolength\marginparwidth{-\oddsidemargin}
1518 \addtolength\marginparwidth{-\inv@mag in}
1519 \addtolength\marginparwidth{-\textwidth}
1520 \addtolength\marginparwidth{-10\jcs@mmm}
1521 \addtolength\marginparwidth{-\marginparsep}
1522 \@tempdima=1zw
1523 \divide\marginparwidth\@tempdima
1524 \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、変化はないはずでした。

[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)。

```

1525 \setlength\topmargin{\paperheight}
1526 \addtolength\topmargin{-\textheight}
1527 \if@slide
1528   \addtolength\topmargin{-\headheight}
1529 \else
1530   \addtolength\topmargin{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
      \headheight (2003-06-26)
1531 \fi
1532 \addtolength\topmargin{-\headsep}
1533 \addtolength\topmargin{-\footskip}
1534 \setlength\topmargin{0.5\topmargin}
1535 %<kiyou>\setlength\topmargin{81truebp}
1536 \iftombow
1537   \addtolength\topmargin{-1in}
1538 \else
1539   \addtolength\topmargin{-\inv@mag in}
1540 \fi
1541 %</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。

---

```
1542 \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。

```
1543 \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  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$  本体で定義されていますが、ここで設定変更します。本文ページ (本文とフロートが共存するページ) とフロートだけのページで設定が異なります。ちなみに、カウンタは内部では `\c@` を名前に冠したマクロになっています。

`\c@topnumber` `topnumber` カウンタは本文ページ上部のフロートの最大数です。

[2003-08-23] ちょっと増やしました。

```
1544 \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。

1545 `\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。

1546 `\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。

1547 `\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。

1548 `\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。

1549 `\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。

1550 `\renewcommand{\floatpagefraction}{.8}`

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

1551 `\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。

1552 `\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。

1553 `\renewcommand{\dblfloatpagefraction}{.8}`

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

1554 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

1555 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`

1556 `\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` 1557 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

1558 `\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`

```

1559 \setlength\@fptop{0\p@? \@plus 1fil}
1560 \setlength\@fpsep{8\p@? \@plus 2fil}
1561 \setlength\@fpbot{0\p@? \@plus 1fil}

```

\@dblftop 段抜きフロートについての値です。

```

\@dblfpsep 1562 \setlength\@dblftop{0\p@? \@plus 1fil}
\@dblfpbot 1563 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1564 \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 pL<sub>A</sub>T<sub>E</sub>X の標準クラス 2017/02/15 に合わせて、同じ命令を追  
 \pltx@cleartoleftpage 加しました。

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

となっています。

```

1565 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1566 % \ifodd\c@page
1567 % \iftdir
1568 % \hbox{}\thispagestyle{empty}\newpage
1569 % \if@twocolumn\hbox{}\newpage\fi
1570 % \fi
1571 % \else
1572 % \ifydir
1573 % \hbox{}\thispagestyle{empty}\newpage
1574 % \if@twocolumn\hbox{}\newpage\fi
1575 % \fi
1576 % \fi\fi}
1577 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1578 % \ifodd\c@page
1579 % \ifydir
1580 % \hbox{}\thispagestyle{empty}\newpage
1581 % \if@twocolumn\hbox{}\newpage\fi
1582 % \fi
1583 % \else
1584 % \iftdir
1585 % \hbox{}\thispagestyle{empty}\newpage
1586 % \if@twocolumn\hbox{}\newpage\fi
1587 % \fi
1588 % \fi\fi}
1589 \def\pltx@cleartooddpage{\clearpage\if@twoside
1590 \ifodd\c@page\else
1591 \hbox{}\thispagestyle{empty}\newpage

```

```

1592 \if@twocolumn\hbox{}\newpage\fi
1593 \fi\fi}
1594 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1595 \ifodd\c@page
1596 \hbox{}\thispagestyle{empty}\newpage
1597 \if@twocolumn\hbox{}\newpage\fi
1598 \fi\fi}

```

---

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

```

1599 \let\pltx@cleartorightpage\pltx@cleartooddpage
1600 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

---

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

```

1601 %<*book|report>
1602 \if@openleft
1603 \let\cleardoublepage\pltx@cleartoleftpage
1604 \else\if@openright
1605 \let\cleardoublepage\pltx@cleartorightpage
1606 \fi\fi
1607 %</book|report>

```

## 7 ページスタイル

ページスタイルとして、 $\text{\LaTeX}$  2<sub>ε</sub> (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<sub>ε</sub> 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ともに動作します。たとえば左マークを `\chapter`、右マークを `\section` で変更する場合はこれにあたります。しかし、同一ページに複数の `\markboth` があると、おかしい結果になることがあります。

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

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

```
1608 % \def\ps@empty{%
1609 %   \let\mkboth\gobbletwo
1610 %   \let\@oddhead\empty
1611 %   \let\@oddfoot\empty
1612 %   \let\@evenhead\empty
1613 %   \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。

```
1614 \def\ps@plainfoot{%
1615   \let\mkboth\gobbletwo
1616   \let\@oddhead\empty
1617   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1618   \let\@evenhead\empty
1619   \let\@evenfoot\@oddfoot}
1620 \def\ps@plainhead{%
1621   \let\mkboth\gobbletwo
1622   \let\@oddfoot\empty
1623   \let\@evenfoot\empty
1624   \def\@evenhead{%
1625     \ifmparswitch \hss \fi
1626     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1627     \ifmparswitch\else \hss \fi}%
1628   \def\@oddhead{%
1629     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1630 %<book>\let\ps@plain\ps@plainhead
1631 %<!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。

```
1632 %<*article|slide>
1633 \if@twoside
1634   \def\ps@headings{%
1635     \let\@oddfoot\empty
1636     \let\@evenfoot\empty
1637     \def\@evenhead{\ifmparswitch \hss \fi
1638       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1639     }
```



```

1639     \if@mparswitch\else \hss \fi}%
1640 \def\@oddhead{%
1641     \underline{%
1642         \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1643 \let\@mkboth\markboth
1644 \def\sectionmark##1{\markboth{%
1645     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1646     ##1}}}%
1647 \def\subsectionmark##1{\markright{%
1648     \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1649     ##1}}}%
1650 }
1651 \else % if not twoside
1652 \def\ps@headings{%
1653     \let\@oddfoot\@empty
1654     \def\@oddhead{%
1655         \underline{%
1656             \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1657     \let\@mkboth\markboth
1658     \def\sectionmark##1{\markright{%
1659         \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1660         ##1}}}%
1661 \fi
1662 %</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ため、「\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。

---

```

1663 %<*book|report>
1664 \def\bxjs@maybe@autoxspacing{%
1665     \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1666 \newif\if@omit@number
1667 \def\ps@headings{%
1668     \let\@oddfoot\@empty
1669     \let\@evenfoot\@empty
1670     \def\@evenhead{%
1671         \if@mparswitch \hss \fi
1672         \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1673             \textbf{\thepage}\hfil\leftmark}}}%
1674         \if@mparswitch\else \hss \fi}%
1675     \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1676         {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1677     \let\@mkboth\markboth
1678     \def\chaptermark##1{\markboth{%
1679         \ifnum \c@secnumdepth >\@m@ne

```

```

1680     \if@mainmatter
1681     \if@omit@number\else
1682     \@chapapp\thechapter\@chappos\hskip1\jsZw
1683     \fi
1684     \fi
1685     \fi
1686     ##1}{}}%
1687 \def\sectionmark##1{\markright{%
1688   \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1689   ##1}}}%
1690 %</book|report>

```

最後は学会誌の場合です。

```

1691 %<*jspf>
1692 \def\ps@headings{%
1693   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1694   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1695   \def\@oddhead{\normalfont\hfil \@title \hfil}
1696   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1697 %</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。

```

1698 \def\ps@myheadings{%
1699   \let\@oddfoot\@empty\let\@evenfoot\@empty
1700   \def\@evenhead{%
1701     \if@mparswitch \hss \fi%
1702     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1703     \if@mparswitch\else \hss \fi}%
1704   \def\@oddhead{%
1705     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1706   \let\mkboth\@gobbletwo
1707 %<book|report> \let\chaptermark\@gobble
1708 \let\sectionmark\@gobble
1709 %<!book&!report> \let\subsectionmark\@gobble
1710 }

```

## 8 文書のマークアップ

### 8.1 表題

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

```

\date 1711 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1712 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1713 % \newcommand*{\date}[1]{\gdef\@date{#1}}

```

1714 % \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。

1715 \newcommand\*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}

1716 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1717 \renewcommand\*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}

1718 \g@addto@macro\bxjs@begin@document@hook{\bxjs@decl@subtitle}

1719 \def\bxjs@decl@subtitle{%

1720 \global\let\bxjs@decl@subtitle\relax

1721 \ifx\subtitle\@undefined

1722 \global\let\subtitle\jsSubtitle

1723 \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。

1724 \def\bxjs@annihilate@subtitle{%

1725 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi

1726 \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 1727 %<\*\jspf>

\keywords 1728 \newcommand\*{\etitle}[1]{\gdef\@etitle{#1}}

1729 \newcommand\*{\eauthor}[1]{\gdef\@eauthor{#1}}

1730 \newcommand\*{\keywords}[1]{\gdef\@keywords{#1}}

1731 \newcommand\*{\email}[1]{\gdef\authors@mail{#1}}

1732 \newcommand\*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}

1733 %</\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。

1734 \def\plainifnotempty{%

1735 \ifx \@oddhead \@empty

1736 \ifx \@oddfoot \@empty

1737 \else

1738 \thispagestyle{plainfoot}%

```

1739     \fi
1740   \else
1741     \thispagestyle{plainhead}%
1742   \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。

```

1743 %<*article|book|report|slide>
1744 \if@titlepage
1745   \newcommand{\maketitle}{%
1746     \begin{titlepage}%
1747       \let\footnotesize\small
1748       \let\footnoterule\relax
1749       \let\footnote\thanks
1750       \null\vfil
1751       \if@slide
1752         {\footnotesize \@date}%
1753         \begin{center}
1754           \mbox{} \[\[1\jsZw]
1755           \large
1756           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1757           \jsc@smallskip
1758           \@title
1759           \ifx\bxjs@subtitle\@undefined\else
1760             \par\vskip\z@
1761             {\small \bxjs@subtitle\par}
1762           \fi
1763           \jsc@smallskip
1764           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1765           \vfill
1766           {\small \@author}%
1767         \end{center}
1768       \else
1769         \vskip 60\p@?
1770         \begin{center}%
1771           {\LARGE \@title \par}%
1772           \ifx\bxjs@subtitle\@undefined\else
1773             \vskip5\p@?
1774             {\normalsize \bxjs@subtitle\par}
1775           \fi
1776           \vskip 3em%
1777           {\large
1778             \lineskip .75em
1779             \begin{tabular}[t]{c}%
1780               \@author

```

```

1781         \end{tabular}\par}%
1782         \vskip 1.5em
1783         {\large \@date \par}%
1784     \end{center}%
1785     \fi
1786     \par
1787     \@thanks\vfil\null
1788 \end{titlepage}%
1789 \setcounter{footnote}{0}%
1790 \global\let\thanks\relax
1791 \global\let\maketitle\relax
1792 \global\let\@thanks\@empty
1793 \global\let\@author\@empty
1794 \global\let\@date\@empty
1795 \global\let\@title\@empty
1796 \global\let\title\relax
1797 \global\let\author\relax
1798 \global\let\date\relax
1799 \global\let\and\relax
1800 \bxjs@annihilate@subtitle
1801 }%
1802 \else
1803 \newcommand{\maketitle}{\par
1804     \begingroup
1805         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1806         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1807         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1808             \parindent 1\jsZw\noindent
1809             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1810         \if@twocolumn
1811             \ifnum \col@number=\@ne
1812                 \@maketitle
1813             \else
1814                 \twocolumn[\@maketitle]%
1815             \fi
1816         \else
1817             \newpage
1818             \global\@topnum\z@ % Prevents figures from going at top of page.
1819             \@maketitle
1820         \fi
1821         \plainifnotempty
1822         \@thanks
1823     \endgroup
1824     \setcounter{footnote}{0}%
1825     \global\let\thanks\relax
1826     \global\let\maketitle\relax
1827     \global\let\@thanks\@empty
1828     \global\let\@author\@empty
1829     \global\let\@date\@empty

```

```

1830 \global\let\@title\@empty
1831 \global\let\title\relax
1832 \global\let\author\relax
1833 \global\let\date\relax
1834 \global\let\and\relax
1835 \bxjs@annihilate@subtitle
1836 }

```

`\@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。

```

1837 \def\@maketitle{%
1838 \newpage\null
1839 \vskip 2em
1840 \begin{center}%
1841 \let\footnote\thanks
1842 {\LARGE \@title \par}%
1843 \ifx\bxjs@subtitle\@undefined\else
1844 \vskip3\p@?
1845 {\normalsize \bxjs@subtitle\par}
1846 \fi
1847 \vskip 1.5em
1848 {\large
1849 \lineskip .5em
1850 \begin{tabular}[t]{c}%
1851 \@author
1852 \end{tabular}\par}%
1853 \vskip 1em
1854 {\large \@date}%
1855 \end{center}%
1856 \par\vskip 1.5em
1857 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1858 }
1859 \fi
1860 %</article|book|report|slide>
1861 %<*jspf>
1862 \newcommand{\maketitle}{\par
1863 \begingroup
1864 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1865 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1866 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1867 \parindent 1\jsZw\noindent
1868 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1869 \twocolumn[\@maketitle]%
1870 \plainifnotempty
1871 \@thanks
1872 \endgroup
1873 \setcounter{footnote}{0}%
1874 \global\let\thanks\relax
1875 \global\let\maketitle\relax
1876 \global\let\@thanks\@empty

```

```

1877 \global\let\@author\@empty
1878 \global\let\@date\@empty
1879 % \global\let\@title\@empty % \@title は柱に使う
1880 \global\let\title\relax
1881 \global\let\author\relax
1882 \global\let\date\relax
1883 \global\let\and\relax
1884 \ifx\authors@mail\@undefined\else{%
1885     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1886     \footnotetext[0]{\itshape\authors@mail}%
1887 } \fi
1888 \global\let\authors@mail\@undefined}
1889 \def\@maketitle{%
1890     \newpage\null
1891     \vskip 6em % used to be 2em
1892     \begin{center}
1893         \let\footnote\thanks
1894         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1895         \lineskip .5em
1896         \ifx\@author\@undefined\else
1897             \vskip 1em
1898             \begin{tabular}[t]{c}%
1899                 \@author
1900             \end{tabular}\par
1901         \fi
1902         \ifx\@etitle\@undefined\else
1903             \vskip 1em
1904             {\large \@etitle \par}%
1905         \fi
1906         \ifx\@eauthor\@undefined\else
1907             \vskip 1em
1908             \begin{tabular}[t]{c}%
1909                 \@eauthor
1910             \end{tabular}\par
1911         \fi
1912         \vskip 1em
1913         \@date
1914     \end{center}
1915     \vskip 1.5em
1916     \centerline{\box\@abstractbox}
1917     \ifx\@keywords\@undefined\else
1918         \vskip 1.5em
1919         \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1920     \fi
1921     \vskip 1.5em}
1922 %</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。

```
1923 \def\bxjs@label@sect#1{%
1924   \expandafter\ifx\csname label#1\endcsname\relax
1925     \csname the#1\endcsname
1926   \else \csname label#1\endcsname
1927   \fi}
1928 \def\@secntformat#1{\bxjs@label@sect{#1}\quad}
```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```
1929 \ifnum\bxjs@label@section=\bxjs@label@section@@compat\else
1930 \def\@secapp{\presectionname}
1931 \def\@secpos{\postsectionname}
1932 \fi
```

`\labelsection` 節番号の出力書式。

```
1933 \ifnum\bxjs@label@section=\bxjs@label@section@@modern
1934 \def\labelsection{\@secapp\thesection\@secpos}
1935 \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。

```

1936 \def\@startsection#1#2#3#4#5#6{%
1937   \if@noskipsec \leavevmode \fi
1938   \par
1939 % 見出し上の空きを \@tempskipa にセットする
1940   \@tempskipa #4\relax
1941 % \@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
1942   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1943 % 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
1944   \ifdim \@tempskipa <\z@
1945     \@tempskipa -\@tempskipa \@afterindentfalse
1946   \fi
1947   \if@nobreak
1948 %   \everypar{\everyparhook}% これは間違い
1949     \everypar{}%
1950   \else
1951     \addpenalty\@secpenalty
1952 % 次の行は削除
1953 %   \addvspace\@tempskipa
1954 % 次の \noindent まで追加
1955     \ifdim \@tempskipa >\z@
1956       \if@slide\else
1957         \null
1958         \vspace*{-\baselineskip}%
1959       \fi
1960       \vskip\@tempskipa
1961     \fi
1962   \fi
1963   \noindent
1964 % 追加終わり
1965   \@ifstar
1966     {\@ssect{#3}{#4}{#5}{#6}}%
1967     {\@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。

```

1968 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
1969 \let\bxjs@if@ceph\@firstofone
1970 \else \let\bxjs@if@ceph\@gobble
1971 \fi

1972 \def\@sect#1#2#3#4#5#6[#7]#8{%
1973 \ifnum #2>\c@secnumdepth
1974 \let\@svsec\@empty
1975 \else
1976 \refstepcounter{#1}%
1977 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1978 \fi
1979 % 見出し後の空きを \@tempskipa にセット
1980 \@tempskipa #5\relax
1981 % 条件判断の順序を入れ替えました
1982 \ifdim \@tempskipa<\z@
1983 \def\@svsechd{%
1984 #6{\hskip #3\relax
1985 \@svsec #8}%
1986 \csname #1mark\endcsname{#7}%
1987 \addcontentsline{toc}{#1}{%
1988 \ifnum #2>\c@secnumdepth \else
1989 \protect\numberline{\bxjs@label@sect{#1}}%
1990 \fi
1991 #7}}% 目次にフルネームを載せるなら #8
1992 \else
1993 \begingroup
1994 \interlinepenalty \@M % 下から移動
1995 #6{%
1996 \@hangfrom{\hskip #3\relax\@svsec}%
1997 % \interlinepenalty \@M % 上に移動
1998 #8\@@par}%
1999 \endgroup
2000 \csname #1mark\endcsname{#7}%
2001 \addcontentsline{toc}{#1}{%
2002 \ifnum #2>\c@secnumdepth \else
2003 \protect\numberline{\bxjs@label@sect{#1}}%
2004 \fi
2005 #7}}% 目次にフルネームを載せるならここは #8
2006 \fi
2007 \@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。

```

2008 \def\@xsect#1{%
2009 % 見出しの後ろの空きを \@tempskipa にセット

```

```

2010 \@tempskipa #1\relax
2011 % 条件判断の順序を変えました
2012 \ifdim \@tempskipa<\z@
2013 \nobreakfalse
2014 \global\@noskipsectrue
2015 \everypar{%
2016 \if@noskipsec
2017 \global\@noskipsecfalse
2018 {\setbox\z@\lastbox}%
2019 \clubpenalty\@M
2020 \begingroup \svsechd \endgroup
2021 \unskip
2022 \@tempskipa #1\relax
2023 \hskip -\@tempskipa
2024 \else
2025 \clubpenalty \@clubpenalty
2026 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
2027 \fi\bxjs@if@ceph\everyparhook}%
2028 \else
2029 \par \nobreak
2030 \vskip \@tempskipa
2031 \@afterheading
2032 \fi
2033 \if@slide
2034 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
2035 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
2036 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
2037 \fi
2038 \par % 2000-12-18
2039 \ignorespaces}
2040 \def\@ssect#1#2#3#4#5{%
2041 \@tempskipa #3\relax
2042 \ifdim \@tempskipa<\z@
2043 \def\@svsechd{#4{\hskip #1\relax #5}}%
2044 \else
2045 \begingroup
2046 #4{%
2047 \@hangfrom{\hskip #1}%
2048 \interlinepenalty \@M #5\@@par}%
2049 \endgroup
2050 \fi
2051 \@xsect{#3}}

```

## ■柱関係の命令

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

```

\subsectionmark 2052 \newcommand*\chaptermark[1]{}
\subsubsectionmark 2053 % \newcommand*\sectionmark[1]{}

\paragraphmark
\subparagraphmark

```

```

2054 % \newcommand*{\subsectionmark}[1]{}
2055 % \newcommand*{\subsubsectionmark}[1]{}
2056 % \newcommand*{\paragraphmark}[1]{}
2057 % \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。

```

2058 %<!book&!report>\setcounter{secnumdepth}{3}
2059 %<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 2060 \newcounter{part}
2061 %<book|report>\newcounter{chapter}
\c@subsubsection 2062 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2063 %<!book&!report>\newcounter{section}
\c@subparagraph 2064 \newcounter{subsection}[section]
2065 \newcounter{subsubsection}[subsection]
2066 \newcounter{paragraph}[subsubsection]
2067 \newcounter{subparagraph}[paragraph]

```

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

`\thechapter` カウンタを出力するコマンドには次のものがあります。

<code>\thesection</code>	<code>\arabic{COUNTER}</code>	1, 2, 3, ...
<code>\thesubsection</code>	<code>\roman{COUNTER}</code>	i, ii, iii, ...
<code>\thesubsubsection</code>	<code>\Roman{COUNTER}</code>	I, II, III, ...
<code>\theparagraph</code>	<code>\alph{COUNTER}</code>	a, b, c, ...
<code>\thesubparagraph</code>	<code>\Alph{COUNTER}</code>	A, B, C, ...
	<code>\kansuji{COUNTER}</code>	一, 二, 三, ...

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

2068 \renewcommand{\thepart}{\@Roman\c@part}
2069 %<*&!book&!report>
2070 \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2071 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2072 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2073 \else
2074 \renewcommand{\thesection}{\@arabic\c@section}
2075 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2076 \fi
2077 %</&!book&!report>
2078 %<*&book|report>
2079 \renewcommand{\thechapter}{\@arabic\c@chapter}
2080 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2081 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}

```

```

2082 %</book|report>
2083 \renewcommand{\thesubsubsection}{%
2084   \thesubsection.\@arabic\c@subsubsection}
2085 \renewcommand{\theparagraph}{%
2086   \thesubsubsection.\@arabic\c@paragraph}
2087 \renewcommand{\thesubparagraph}{%
2088   \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。

```

2089 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2090 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

[2017-03-05] `\frontmatter` と `\mainmatter` の2つの命令は、改丁または改ページした後で `\pagenumbering{...}` でノンブルを1にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが、これではノンブルをリセットする際に偶奇逆転が起こる場合があります。 `openany` かどうかに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  $\text{\LaTeX}$  の標準クラス 2017/03/05 も参照)。

```

2091 %<*book|report>
2092 \newcommand\frontmatter{%
2093   \pltx@cleartooddpage
2094   \@mainmatterfalse
2095   \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。

```

2096 \newcommand\mainmatter{%
2097   \pltx@cleartooddpage
2098   \@mainmattertrue
2099   \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。

```

2100 \newcommand\backmatter{%
2101   \if@openleft
2102     \cleardoublepage
2103   \else\if@openright
2104     \cleardoublepage
2105   \else
2106     \clearpage
2107   \fi\fi
2108   \@mainmatterfalse}
2109 %</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 \CMDDB }
\def\CMDA      [#1]#2{...} % \chapter[...]{...} の定義
\def\CMDDB     #1{...}     % \chapter*{...} の定義
```

まず `book` と `report` のクラス以外です。

```
2110 %<!*book&!report>
2111 \newcommand\part{%
2112   \if@noskipsec \leavevmode \fi
2113   \par
2114   \addvspace{4ex}%
2115   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2116   \secdef\@part\@spart}
2117 %</!*book&!report>
```

`book` および `report` クラスの場合は、少し複雑です。

```
2118 %<*book|report>
2119 \newcommand\part{%
2120   \if@openleft
2121     \cleardoublepage
2122   \else\if@openright
2123     \cleardoublepage
2124   \else
2125     \clearpage
2126   \fi\fi
2127   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2128   \if@twocolumn
2129     \onecolumn
2130     \@restonecoltrue
2131   \else
2132     \@restonecolfalse
2133   \fi
2134   \null\vfil
2135   \secdef\@part\@spart}
2136 %</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。

```
2137 %<*&book&!report>
2138 \def\@part[#1]#2{%
2139   \ifnum \c@secnumdepth >\m@ne
2140     \refstepcounter{part}%
2141     \addcontentsline{toc}{part}{%
2142       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2143   \else
2144     \addcontentsline{toc}{part}{#1}%
2145   \fi
2146   \markboth{}{}%
2147   {\parindent\z@
2148     \raggedright
2149     \interlinepenalty \@M
2150     \normalfont
2151     \ifnum \c@secnumdepth >\m@ne
2152       \Large\headfont\prepartname\thepart\postpartname
2153       \par\nobreak
2154     \fi
2155     \huge \headfont #2%
2156     \markboth{}{}\par}%
2157   \nobreak
2158   \vskip 3ex
2159   \@afterheading}
2160 %</!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。

```
2161 %<*&book|report>
2162 \def\@part[#1]#2{%
2163   \ifnum \c@secnumdepth >-2\relax
2164     \refstepcounter{part}%
2165     \addcontentsline{toc}{part}{%
2166       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2167   \else
2168     \addcontentsline{toc}{part}{#1}%
2169   \fi
2170   \markboth{}{}%
2171   {\centering
2172     \interlinepenalty \@M
2173     \normalfont
2174     \ifnum \c@secnumdepth >-2\relax
2175       \huge\headfont \prepartname\thepart\postpartname
2176       \par\vskip20\p@?
2177     \fi
2178     \Huge \headfont #2\par}%
2179   \@endpart}
2180 %</book|report>
```

\@spart 番号を付けない部です。

```

2181 %<*!book&!report>
2182 \def\@spart#1{%
2183     \parindent \z@ \raggedright
2184     \interlinepenalty \@M
2185     \normalfont
2186     \huge \headfont #1\par}%
2187 \nobreak
2188 \vskip 3ex
2189 \@afterheading}
2190 %</!book&!report>
2191 %<*book|report>
2192 \def\@spart#1{%
2193     \centering
2194     \interlinepenalty \@M
2195     \normalfont
2196     \Huge \headfont #1\par}%
2197 \@endpart}
2198 %</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。

```

2199 %<*book|report>
2200 \def\@endpart{\vfil\newpage
2201     \if@twoside
2202     \if@openleft %% added (2017/02/24)
2203         \null\thispagestyle{empty}\newpage
2204     \else\if@openright %% added (2016/12/13)
2205         \null\thispagestyle{empty}\newpage
2206     \fi\fi %% added (2016/12/13, 2017/02/24)
2207 \fi
2208 \if@restonecol
2209     \twocolumn
2210 \fi}
2211 %</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。

```

2212 %<*book|report>
2213 \newcommand{\chapter}{%
2214     \if@openleft\cleardoublepage\else
2215     \if@openright\cleardoublepage\else\clearpage\fi\fi
2216     \plainifnotempty % 元: \thispagestyle{plain}

```



```

2217 \global\@topnum\z@
2218 \if@english \afterindentfalse \else \afterindenttrue \fi
2219 \secdef
2220   {\@omit@numberfalse\@chapter}%
2221   {\@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。

```

2222 \def\@chapter[#1]#2{%
2223   \ifnum \c@secnumdepth >\m@ne
2224     \if@mainmatter
2225       \refstepcounter{chapter}%
2226       \typeout{\@chapapp\thechapter\@chappos}%
2227       \addcontentsline{toc}{chapter}%
2228         {\protect\numberline
2229 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2230         {\@chapapp\thechapter\@chappos}%
2231         #1}%
2232     \else\addcontentsline{toc}{chapter}{#1}\fi
2233   \else
2234     \addcontentsline{toc}{chapter}{#1}%
2235   \fi
2236   \chaptermark{#1}%
2237   \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
2238   \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
2239   \if@twocolumn
2240     \@topnewpage[\@makechapterhead{#2}]%
2241   \else
2242     \@makechapterhead{#2}%
2243     \@afterheading
2244   \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。

```

2245 \def\@makechapterhead#1{%
2246   \vspace*{2\Cvs}% 欧文は 50pt
2247   {\parindent \z@ \raggedright \normalfont
2248     \ifnum \c@secnumdepth >\m@ne
2249       \if@mainmatter
2250         \huge\headfont \@chapapp\thechapter\@chappos
2251         \par\nobreak
2252         \vskip \Cvs % 欧文は 20pt
2253       \fi
2254     \fi
2255     \interlinepenalty\@M
2256     \Huge \headfont #1\par\nobreak
2257     \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。

```

2258 \def\@schapter#1{%

```

```

2259 \chaptermark{#1}%
2260 \if@twocolumn
2261 \topnewpage[\@makeschapterhead{#1}]%
2262 \else
2263 \@makeschapterhead{#1}\@afterheading
2264 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2265 \def\@makeschapterhead#1{%
2266 \vspace*{2\Cvs}% 欧文は 50pt
2267 {\parindent \z@ \raggedright
2268 \normalfont
2269 \interlinepenalty\@M
2270 \Huge \headfont #1\par\nobreak
2271 \vskip 3\Cvs}} % 欧文は 40pt
2272 %</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。

```

2273 \if@twocolumn
2274 \newcommand{\section}{%
2275 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2276 \@startsection{section}{1}{\z@}%
2277 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2278 %<kiyou> {\Cvs}{0.5\Cvs}%
2279 % {\normalfont\large\headfont\@secapp}}
2280 {\normalfont\large\headfont\raggedright}}
2281 \else
2282 \newcommand{\section}{%
2283 \if@slide\clearpage\fi
2284 \@startsection{section}{1}{\z@}%
2285 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2286 {.5\Cvs \@plus.3\Cdp}% 後アキ
2287 % {\normalfont\Large\headfont\@secapp}}
2288 {\normalfont\Large\headfont\raggedright}}
2289 \fi

```

`\subsection` 同上です。

```

2290 \if@twocolumn
2291 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2292 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2293 {\normalfont\normalsize\headfont}}
2294 \else
2295 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2296 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ

```

```

2297     {.5\Cvs \@plus.3\Cdp}% 後アキ
2298     {\normalfont\large\headfont}}
2299 \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)。

```

2300 \if@twocolumn
2301   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2302     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2303     {\normalfont\normalsize\headfont}}
2304 \else
2305   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2306     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2307     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2308     {\normalfont\normalsize\headfont}}
2309 \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。

---

```

2310 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2311 \let\bxjs@org@paragraph@mark\jsParagraphMark
2312 \ifx\bxjs@paragraph@mark@empty
2313   \let\jsParagraphMark@empty
2314 \else\ifx\bxjs@paragraph@mark@undefined\else
2315   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2316 \fi\fi
2317 \let\jsJaChar@empty
2318 \if@twocolumn
2319   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2320     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2321     <jspf> {\normalfont\normalsize\headfont}}
2322     <!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2323 \else
2324   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%

```

```

2325      {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2326      {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2327 %<jspf>      {\normalfont\normalsize\headfont}}
2328 %<!jspf>      {\normalfont\normalsize\headfont\jsParagraphMark}}
2329 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2330 \if@twocolumn
2331   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2332     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2333     {\normalfont\normalsize\headfont}}
2334 \else
2335   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2336     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2337     {\normalfont\normalsize\headfont}}
2338 \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 でしたが、ここでは全角幅の 2 倍にしました。

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2339 \if@slide
2340   \setlength\leftmargini{1\jsZw}
2341 \else
2342   \if@twocolumn
2343     \setlength\leftmargini{2\jsZw}
2344   \else
2345     \setlength\leftmargini{3\jsZw}
2346   \fi
2347 \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 2348 \if@slide
\leftmarginv 2349   \setlength\leftmarginii {1\jsZw}
2350   \setlength\leftmarginiii {1\jsZw}
\leftmarginvi 2351   \setlength\leftmarginiv {1\jsZw}
2352   \setlength\leftmarginv {1\jsZw}
2353   \setlength\leftmarginvi {1\jsZw}
2354 \else
2355   \setlength\leftmarginii {2\jsZw}
2356   \setlength\leftmarginiii {2\jsZw}

```

```

2357 \setlength\leftmarginiv {2\jsZw}
2358 \setlength\leftmarginv {1\jsZw}
2359 \setlength\leftmarginvi {1\jsZw}
2360 \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。

```

2361 \setlength \labelsep {0.5\jsZw} % .5em
2362 \setlength \labelwidth{\leftmargini}
2363 \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。

```

2364 \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 2365 \@beginparpenalty -\@lowpenalty
\@itempenalty 2366 \@endparpenalty -\@lowpenalty
2367 \@itempenalty -\@lowpenalty

```

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

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

```

2368 \def\@listi{\leftmargin\leftmargini
2369 \parsep \z@
2370 \topsep 0.5\baselineskip
2371 \itemsep \z@ \relax}
2372 \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）。

```

2373 \@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 2374 \def\@listii{\leftmargin\leftmarginii
\@listiv 2375 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2376 \topsep \z@
\@listv 2377 \parsep \z@
\@listvi 2378 \itemsep\parsep}
2379 \def\@listiii{\leftmargin\leftmarginiii
2380 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2381 \topsep \z@
2382 \parsep \z@

```

```

2383 \itemsep\parsep}
2384 \def\@listiv {\leftmargin\leftmarginiv
2385             \labelwidth\leftmarginiv
2386             \advance\labelwidth-\labelsep}
2387 \def\@listv {\leftmargin\leftmarginv
2388             \labelwidth\leftmarginv
2389             \advance\labelwidth-\labelsep}
2390 \def\@listvi {\leftmargin\leftmarginvi
2391             \labelwidth\leftmarginvi
2392             \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。

```

2393 \renewcommand{\theenumi}{\@arabic\c@enumi}
2394 \renewcommand{\theenumii}{\@alph\c@enumii}
2395 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2396 \renewcommand{\theenumiv}{\@Alph\c@enumiv}

```

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

---

```

2397 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2398 \newcommand*\jsInJaParen[1]{%
2399   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2400 \newcommand{\labelenumi}{\theenumi.}
2401 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2402 \newcommand{\labelenumiii}{\theenumiii.}
2403 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2404 \renewcommand{\p@enumii}{\theenumi}
2405 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2406 \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 2407 \newcommand\labelitemi{\textbullet}
\labelitemiii 2408 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiv 2409 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemv 2410 \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。

```
2411 \newenvironment{description}{%
2412   \list{}{%
2413     \labelwidth=\leftmargin
2414     \labelsep=1\jsZw
2415     \advance \labelwidth by -\labelsep
2416     \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。

```
2417 \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 (+ titlapage)` 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。

```
2418 %<*book|report>
2419 \newenvironment{chapterabstract}{%
2420   \begin{list}{}{%
2421     \listparindent=1\jsZw
2422     \itemindent=\listparindent
```

```

2423 \rightmargin=0pt
2424 \leftmargin=5\jsZw\item[]\end{list}\vspace{\baselineskip}}
2425 %</book|report>

```

“普通の” abstract 環境の定義。

```

2426 %<*article|report|slide>
2427 \newbox\@abstractbox
2428 \if@titlepage
2429 \newenvironment{abstract}{%
2430 \titlepage
2431 \null\vfil
2432 \@beginparpenalty\@lowpenalty
2433 \begin{center}%
2434 \headfont \abstractname
2435 \@endparpenalty\@M
2436 \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。

```

2437 \par}%
2438 {\par\vfil\null\endtitlepage}
2439 \else
2440 \newenvironment{abstract}{%
2441 \if@twocolumn
2442 \ifx\maketitle\relax
2443 \section*{\abstractname}%
2444 \else
2445 \global\setbox\@abstractbox\hbox\bgroup
2446 \begin{minipage}[b]{\textwidth}
2447 \small\parindent1\jsZw
2448 \begin{center}%
2449 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2450 \end{center}%
2451 \list{}{%
2452 \listparindent\parindent
2453 \itemindent \listparindent
2454 \rightmargin \leftmargin}%
2455 \item\relax
2456 \fi
2457 \else
2458 \small
2459 \begin{center}%
2460 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2461 \end{center}%
2462 \list{}{%
2463 \listparindent\parindent
2464 \itemindent \listparindent
2465 \rightmargin \leftmargin}%
2466 \item\relax
2467 \fi}{\if@twocolumn

```



```

2468     \ifx\maketitle\relax
2469     \else
2470         \endlist\end{minipage}\egroup
2471     \fi
2472     \else
2473         \endlist
2474     \fi}
2475 \fi
2476 %</article|report|slide>
2477 %<*jspf>
2478 \newbox\@abstractbox
2479 \newenvironment{abstract}{%
2480     \global\setbox\@abstractbox\hbox\bgroup
2481     \begin{minipage}[b]{157\jsc@mmm}\sffamily Abstract}\par
2482     \small
2483     \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2484     {\end{minipage}\egroup}
2485 %</jspf>

```

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

```

2486 %<*book|report>
2487 \ifbxjs@force@chapterabstract
2488     \let\abstract\chapterabstract
2489     \let\endabstract\endchapterabstract
2490 \fi
2491 %</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。

```

2492 %<*jspf>
2493 %\newbox\@keywordsbox
2494 %\newenvironment{keywords}{%
2495 %     \global\setbox\@keywordsbox\hbox\bgroup
2496 %     \begin{minipage}[b]{1570\jsc@mmm}\sffamily Keywords:}\par
2497 %         \small\parindent0\jsZw}%
2498 %     {\end{minipage}\egroup}
2499 %</jspf>

```

## ■verse 環境

**verse** 詩のための `verse` 環境です。

```

2500 \newenvironment{verse}{%
2501     \let \\\=\@centercr
2502     \list{}{%
2503         \itemsep \z@

```

```

2504 \itemindent -2\jsZw % 元: -1.5em
2505 \listparindent\itemindent
2506 \rightmargin \z@
2507 \advance\leftmargin 2\jsZw}% 元: 1.5em
2508 \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。

```

2509 \newenvironment{quotation}{%
2510 \list{}{%
2511 \listparindent\parindent
2512 \itemindent\listparindent
2513 \rightmargin \z@}%
2514 \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。

```

2515 \newenvironment{quote}%
2516 {\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。

```

2517 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2518 \item[\hskip \labelsep{\headfont #1\ #2}]}
2519 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2520 \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。

```

2521 \newenvironment{titlepage}{%

```

```

2522 %<book>      \pltx@cleartooddpage %% 2017-02-24
2523      \if@twocolumn
2524          \@restonecoltrue\onecolumn
2525      \else
2526          \@restonecolfalse\newpage
2527      \fi
2528      \thispagestyle{empty}%
2529      \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2530      }%
2531      {\if@restonecol\twocolumn \else \newpage \fi
2532      \if@twoside\else
2533          \setcounter{page}\@ne
2534      \fi}

```

## ■付録

`\appendix` 本文と付録を分離するコマンドです。

```

2535 %<!*book&!report>
2536 \newcommand{\appendix}{\par
2537     \setcounter{section}{0}%
2538     \setcounter{subsection}{0}%
2539     \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2540         \gdef\presectionname{\appendixname}%
2541         \gdef\postsectionname{}%
2542     % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2543     \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2544     \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2545     \else
2546         \gdef\@secapp{\appendixname}%
2547         \gdef\@secpos{}%
2548         \gdef\thesection{\@Alph\c@section}%
2549     \fi}
2550 %</!*book&!report>
2551 %<*book|report>
2552 \newcommand{\appendix}{\par
2553     \setcounter{chapter}{0}%
2554     \setcounter{section}{0}%
2555     \gdef\@chapapp{\appendixname}%
2556     \gdef\@chappos{}%
2557     \gdef\thechapter{\@Alph\c@chapter}}
2558 %</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 倍の幅の空きが入ります。

2559 `\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。

2560 `\setlength\tabcolsep{6\p@?}`

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

2561 `\setlength\arrayrulewidth{.4\p@}`

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

2562 `\setlength\doublerulesep{2\p@}`

### ■tabbing 環境

`\tabbingsep` \ ' コマンドで入るアキです。

2563 `\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。

2564 `\skip\@mpfootins = \skip\footins`

### ■framebox 環境

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

`\fbxrule` `\fbx`, `\framebox` の罫線の幅です。

2565 `\setlength\fbxsep{3\p@?}`

2566 `\setlength\fbxrule{.4\p@}`

### ■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

2567 `%<!book&!report>\renewcommand \theequation {\@arabic\c@equation}`

2568 `%<*book|report>`

2569 `\@addtoreset{equation}{chapter}`

2570 `\renewcommand\theequation`

2571 `{\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}`

2572 `%</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。

2573 `% \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。

2574 `% \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。

```
2575 % \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。

```
2576 %<*!book&!report>
2577 \newcounter{figure}
2578 \renewcommand \thefigure {\@arabic\c@figure}
2579 %</!book&!report>
2580 %<*book|report>
2581 \newcounter{figure}[chapter]
2582 \renewcommand \thefigure
2583     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2584 %</book|report>
```

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

```
\ext@figure 2585 \def\fps@figure{tbp}
2586 \def\ftype@figure{1}
\fnum@figure 2587 \def\ext@figure{lof}
2588 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` \* 形式は段抜きのフロートです。

```
figure* 2589 \newenvironment{figure}%
2590     {\@float{figure}}%
2591     {\endfloat}
2592 \newenvironment{figure*}%
2593     {\@dblfloat{figure}}%
2594     {\enddblfloat}
```

### ■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。

```
2595 %<!*book&!report>
2596 \newcounter{table}
2597 \renewcommand\thetable{\@arabic\c@table}
2598 %</!*book&!report>
2599 %<*book|report>
2600 \newcounter{table}[chapter]
2601 \renewcommand \thetable
2602     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2603 %</book|report>
```

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

```
\ext@table 2604 \def\fps@table{tbp}
2605 \def\ftype@table{2}
\fnun@table 2606 \def\ext@table{lot}
2607 \def\fnun@table{\tablename\nobreak\thetable}
```

`table *` は段抜きのフロートです。

```
table* 2608 \newenvironment{table}%
2609     {\@float{table}}%
2610     {\end@float}
2611 \newenvironment{table*}%
2612     {\@dblfloat{table}}%
2613     {\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。

```
2614 \newlength\abovecaptionskip
2615 \newlength\belowcaptionskip
2616 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2617 \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。

```
2618 %<!*jspf>
2619 % \long\def\@makecaption#1#2{{\small
2620 %     \advance\leftskip10\jsc@mmm
```

```

2621 % \advance\rightskip10\jsc@mmm
2622 % \vskip\abovecaptionskip
2623 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2624 % \ifdim \wd\@tempboxa >\hsize
2625 % #1\hskip1\jsZw\relax #2\par
2626 % \else
2627 % \global \@minipagefalse
2628 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2629 % \fi
2630 % \vskip\belowcaptionskip}}
2631 \long\def\@makecaption#1#2{{\small
2632 \advance\leftskip .0628\linewidth
2633 \advance\rightskip .0628\linewidth
2634 \vskip\abovecaptionskip
2635 \sbox\@tempboxa{#1\zwspace#2}%
2636 \ifdim \wd\@tempboxa <\hsize \centering \fi
2637 #1\zwspace#2\par
2638 \vskip\belowcaptionskip}}
2639 %</!jspf>
2640 %<*jspf>
2641 \long\def\@makecaption#1#2{%
2642 \vskip\abovecaptionskip
2643 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2644 \ifdim \wd\@tempboxa >\hsize
2645 {\small\sffamily
2646 \list{#1}{%
2647 \renewcommand{\makelabel}[1]{##1\hfil}
2648 \itemsep \z@
2649 \itemindent \z@
2650 \labelsep \z@
2651 \labelwidth 11\jsc@mmm
2652 \listparindent\z@
2653 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2654 \else
2655 \global \@minipagefalse
2656 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2657 \fi
2658 \vskip\belowcaptionskip}
2659 %</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 `\scrDeclareOldFontCommand` に倣い、これらの命令を使うときには警告を発することにしました。

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

---

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

---

```
2660 \newif\if@jsc@warnoldfontcmd
2661 \@jsc@warnoldfontcmdtrue
2662 \newif\if@jsc@warnoldfontcmdexception
2663 \@jsc@warnoldfontcmdexceptionfalse
```

```
\jsc@DeclareOldFontCommand
```

```
2664 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2665   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2666   \DeclareOldFontCommand{#1}{%
2667     \bxjs@oldfontcmd{#1}#2%
2668   }{%
2669     \bxjs@oldfontcmd{#1}#3%
2670   }%
2671 }
2672 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2673   \ClassInfo\bxjs@clsname
2674   {Old font command '\string#1' is used!!\MessageBreak
2675     The first occurrence is}%
2676 }
```

---

```
\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。
```

```
2677 \newcommand*{\allowoldfontcommands}{%
2678   \@jsc@warnoldfontcmdexceptiontrue}
2679 \newcommand*{\disallowoldfontcommands}{%
2680   \@jsc@warnoldfontcmdexceptionfalse}

2681 \let\bxjs@oldfontcmd@list\@empty
2682 \def\bxjs@oldfontcmd#1{%
2683   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2684 \def\bxjs@oldfontcmd@a#1#2{%
2685   \if@jsc@warnoldfontcmdexception\else
2686     \global\@jsc@warnoldfontcmdfalse
2687     \ifx#1\relax
2688       \global\let#1=t%
2689       \jsc@warnoldfontcmd{#2}%
2690     \fi
2691   \fi}
```



```

2692 \def\bxjs@warnoldfontcmd@final{%
2693 % \par
2694 \global\let\bxjs@warnoldfontcmd@final\@empty
2695 \let\@tempa\@empty
2696 \def\do##1{%
2697   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2698     \edef\@tempa{\@tempa \space\string##1}\fi}
2699 \bxjs@oldfontcmd@list
2700 \ifx\@tempa\@empty\else
2701   \ClassWarningNoLine\bxjs@clsname
2702     {Some old font commands were used in text:\MessageBreak
2703       \space\@tempa\MessageBreak
2704       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2705       new font selection scheme called NFSS2 with several\MessageBreak
2706       new, combinable font commands. The
2707       class provides\MessageBreak
2708       the old font commands only for compatibility}
2709 \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。

```

2710 \def\bxjs@warnoldfontcmd@kick@final{%
2711   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2712 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2713 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2714 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2715 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2716 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2717 \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。

```

2718 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2719 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2720 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2721 \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`

```
2722 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2723 \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。

```
2724 \newcommand\@pnumwidth{1.55em}
2725 \newcommand\@tocrmarg{2.55em}
2726 \newcommand\@dotsep{4.5}
2727 %<!book&!report>\setcounter{tocdepth}{2}
2728 %<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)

```
2729 \newdimen\jsc@tocl@width
2730 \newcommand{\tableofcontents}{%
2731 %<*book|report>
2732 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2733 \settowidth\@tempdima{\headfont\appendixname}%
2734 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2735 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2736 \if@twocolumn
2737 \@restonecoltrue\onecolumn
2738 \else
2739 \@restonecolfalse
2740 \fi
2741 \chapter*{\contentsname}%
2742 \@mkboth{\contentsname}{}%
2743 %</book|report>
2744 %<*!book&!report>
2745 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2746 \settowidth\@tempdima{\headfont\appendixname}%
2747 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2748 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2749 \section*{\contentsname}%
2750 \@mkboth{\contentsname}{\contentsname}%
2751 %</!book&!report>
2752 \@starttoc{toc}%
2753 %<book|report> \if@restonecol\twocolumn\fi
2754 }
```

`\l@part` 部の目次です。

```
2755 \newcommand*{\l@part}[2]{%
2756 \ifnum \c@tocdepth >-2\relax
2757 %<!book&!report> \addpenalty\@secpenalty
2758 %<book|report> \addpenalty{-\@highpenalty}%
2759 \addvspace{2.25em \@plus\p@}%
2760 \begingroup
2761 \parindent \z@
2762 % \@pnumwidth should be \@tocrmarg
2763 % \rightskip \@pnumwidth
2764 \rightskip \@tocrmarg
2765 \parfillskip -\rightskip
2766 {\leavevmode
2767 \large \headfont
2768 \setlength\@lnumwidth{4\jsZw}%
2769 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2770 \nobreak
```

```

2771 %<book|report>      \global\@nbreaktrue
2772 %<book|report>      \everypar{\global\@nbreakfalse\everypar{}}%
2773      \endgroup
2774      \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)

```

2775 %<*book|report>
2776 \newcommand*{\l@chapter}[2]{%
2777   \ifnum \c@tocdepth >\m@ne
2778     \addpenalty{-\@highpenalty}%
2779     \addvspace{1.0em \@plus\p@?}
2780 %   \vskip 1.0em \@plus\p@    % book.cls では↑がこうなっている
2781   \begingroup
2782     \parindent\z@
2783 %   \rightskip\@pnumwidth
2784   \rightskip\@tocrmarg
2785   \parfillskip-\rightskip
2786   \leavevmode\headfont
2787 %   % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2788     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2789     \advance\leftskip\@lnumwidth \hskip-\leftskip
2790     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2791     \penalty\@highpenalty
2792   \endgroup
2793   \fi}
2794 %</book|report>

```

\l@section 節の目次です。

```

2795 %<*!book&!report>
2796 \newcommand*{\l@section}[2]{%
2797   \ifnum \c@tocdepth >\z@
2798     \addpenalty{\@secpenalty}%
2799     \addvspace{1.0em \@plus\p@?}%
2800   \begingroup
2801     \parindent\z@
2802 %   \rightskip\@pnumwidth
2803   \rightskip\@tocrmarg
2804   \parfillskip-\rightskip
2805   \leavevmode\headfont
2806 %   % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2807     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2808     \advance\leftskip\@lnumwidth \hskip-\leftskip
2809     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2810   \endgroup
2811   \fi}
2812 %</!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。

```
2813 %<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] ここの \jsc@tocl@width から決めるようにしてみました。(by ts)
```

```
\l@subparagraph 2814 %<!*book&!report>
2815 % \newcommand*{\l@section}{\@dottedtocline{2}{1.5em}{2.3em}}
2816 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2817 % \newcommand*{\l@paragraph}{\@dottedtocline{4}{7.0em}{4.1em}}
2818 % \newcommand*{\l@subparagraph}{\@dottedtocline{5}{10em}{5em}}
2819 %
2820 % \newcommand*{\l@section}{\@dottedtocline{2}{1zw}{3zw}}
2821 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2822 % \newcommand*{\l@paragraph}{\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2823 % \newcommand*{\l@subparagraph}{\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2824 %
2825 \newcommand*{\l@section}{%
2826     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2827     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2828 \newcommand*{\l@subsubsection}{%
2829     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2830     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2831 \newcommand*{\l@paragraph}{%
2832     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2833     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2834 \newcommand*{\l@subparagraph}{%
2835     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2836     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2837 %</*!book&!report>
2838 %<*book|report>
2839 % \newcommand*{\l@section}{\@dottedtocline{2}{3.8em}{3.2em}}
2840 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2841 % \newcommand*{\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
2842 % \newcommand*{\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
2843 \newcommand*{\l@section}{%
2844     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2845     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2846 \newcommand*{\l@subsubsection}{%
2847     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2848     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2849 \newcommand*{\l@subsubsection}{%
2850     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2851     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2852 \newcommand*{\l@paragraph}{%
2853     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2854     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
```

```

2855 \newcommand*{\l@subparagraph}{%
2856         \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2857         \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2858 %</book|report>

```

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

```

2859 \newdimen\@lnumwidth
2860 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

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

[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。

```

2861 \def\jsTocLine{\leaders\hbox{%
2862   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2863 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2864   \vskip \z@ \@plus.2\p@?
2865   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2866     \parindent #2\relax\@afterindenttrue
2867     \interlinepenalty\@M
2868     \leavevmode
2869     \@lnumwidth #3\relax
2870     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2871     {#4}\nobreak
2872     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2873       \hfil\normalfont \normalcolor #5}\par}\fi}

```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2874 \newcommand{\listoffigures}{%
2875 %<*book|report>
2876   \if@twocolumn\@restonecoltrue\onecolumn
2877   \else\@restonecolfalse\fi
2878   \chapter*{\listfigurename}%
2879   \@mkboth{\listfigurename}{}%
2880 %</book|report>
2881 %<!*book&!report>
2882   \section*{\listfigurename}%
2883   \@mkboth{\listfigurename}{\listfigurename}%

```

```

2884 %</!book&!report>
2885 \starttoc{lof}%
2886 %<book|report> \if@restonecol\twocolumn\fi
2887 }

```

`\l@figure` 図目次の項目を出力します。

```

2888 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2889 \newcommand{\listoftables}{%
2890 %<*book|report>
2891 \if@twocolumn\@restonecoltrue\onecolumn
2892 \else\@restonecolfalse\fi
2893 \chapter*{\listtablename}%
2894 \@mkboth{\listtablename}{}%
2895 %</book|report>
2896 %<*!book&!report>
2897 \section*{\listtablename}%
2898 \@mkboth{\listtablename}{\listtablename}%
2899 %</!book&!report>
2900 \starttoc{lot}%
2901 %<book|report> \if@restonecol\twocolumn\fi
2902 }

```

`\l@table` 表目次は図目次と同じです。

```

2903 \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。

```

2904 \newdimen\bibindent
2905 \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。

```

2906 \newenvironment{thebibliography}[1]{%
2907 \@jsc@warnoldfontcmdexceptiontrue
2908 \global\let\presectionname\relax
2909 \global\let\postsectionname\relax
2910 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2911 %<*kiyou>
2912 \vspace{1.5\baselineskip}
2913 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2914 \vspace{0.5\baselineskip}
2915 %</kiyou>

```

```

2916 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2917 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2918 \list{\@biblabel{\@arabic{c@enumiv}}}%
2919 {\settowidth\labelwidth{\@biblabel{#1}}%
2920 \leftmargin\labelwidth
2921 \advance\leftmargin\labelsep
2922 \@openbib@code
2923 \usecounter{enumiv}%
2924 \let\p@enumiv\@empty
2925 \renewcommand\theenumiv{\@arabic{c@enumiv}}%
2926 %<kiyou> \small
2927 \sloppy
2928 \clubpenalty4000
2929 \@clubpenalty\clubpenalty
2930 \widowpenalty4000%
2931 \sfcode`.\@m}
2932 {\def\@noitemerr
2933 {\@latex@warning{Empty 'thebibliography' environment}}}%
2934 \endlist}

```

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

```

2935 \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。

```

2936 \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。

```

2937 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

`\cite` 文献の番号を出力する部分は ltbib1.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` で取りますので、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。

```

2938 % \def\@citex[#1]#2{\leavevmode
2939 % \let\@citea\@empty
2940 % \@cite{\@for\@citeb:=#2\do
2941 % {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2942 % \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2943 % \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2944 % \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2945 % \G@refundefinedtrue
2946 % \@latex@warning
2947 % {Citation ``\@citeb' on page \thepage \space undefined}}%
2948 % {\@citeofmt{\csname b@\@citeb\endcsname}}}{#1}}
2949 % \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。

```
2950 % \DeclareRobustCommand\cite{\unskip
2951 %   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]
2952 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2953 %   , \jsInhibitGlue\ #2\fi}) }}$}
```

### 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)。

```
2954 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2955   \if@twocolumn
2956     \onecolumn\@restonecolfalse
2957   \else
2958     \clearpage\@restonecoltrue
2959   \fi
2960   \columnseprule.4pt \columnsep 2\jsZw
2961   \ifx\multicols\@undefined
2962 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}]%
2963 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2964 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2965 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2966   \else
2967     \ifdim\textwidth<\fullwidth
2968       \setlength{\evensidemargin}{\oddsidemargin}
2969       \setlength{\textwidth}{\fullwidth}
2970       \setlength{\linewidth}{\fullwidth}
2971 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}]%
2972 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2973 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2974 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2975   \else
2976 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}]%
2977 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2978 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2979 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2980   \fi
2981   \fi
2982 %<book|report>      \@mkboth{\indexname}{}%
2983 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2984   \plainifnotempty % \thispagestyle{plain}
2985   \parindent\z@
2986   \parskip\z@ \@plus .3\p@?\relax
2987   \let\item\@idxitem
2988   \raggedright
2989   \footnotesize\narrowbaselines
```

```

2990 }{
2991   \ifx\multicols\@undefined
2992     \if@restonecol\onecolumn\fi
2993   \else
2994     \end{multicols}
2995   \fi
2996   \clearpage
2997 }

```

`\@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 2998 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2999 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
3000 \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。

```

3001 \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。

```

3002 \newcommand{\seename}{\if@english see\else \rightarrow\fi}
3003 \newcommand{\alsoname}{\if@english see also\else \rightarrow\fi}

```

## 10.4 脚注

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

---

```

3004 \ifx\pltx@foot@penalty\@undefined
3005   \let\footnotes@ve=\footnote
3006   \def\footnote{\jsInhibitGlue\footnotes@ve}
3007   \let\footnotemarks@ve=\footnotemark
3008   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
3009 \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 pL<sup>A</sup>T<sub>E</sub>X の変更に従いました (Thanks: 角藤さん)。pL<sup>A</sup>T<sub>E</sub>X の日付が 2016/04/17 より新しい場合は、このパッチが不要なのであてません。

---

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

---

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

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

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

```
3010 \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。

```
3011 % \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。

```
3012 \renewcommand{\footnoterule}{%
3013   \kern-2.6\p@? \kern-.4\p@
3014   \hrule width .4\columnwidth
3015   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
3016 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

[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。

```
3017 \long\def\@footnotetext{%
3018   \insert\footins\bgroup
3019     \normalfont\footnotesize
3020     \interlinepenalty\interfootnotelinepenalty
3021     \splittopskip\footnotesep
3022     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
3023     \hsize\columnwidth \@parboxrestore
3024     \protected@edef\@currentlabel{%
```

```

3025      \csname p@footnote\endcsname\@thefnmark
3026    }%
3027    \color@begingroup
3028      \@makefnmark{%
3029        \rule{z@\footnotesep\ignorespaces}%
3030        \futurelet\jsc@next\jsc@fo@t}
3031 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@f@t
3032              \else \let\jsc@next\jsc@f@t\fi \jsc@next}
3033 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3034 \def\jsc@f@t#1{#1\jsc@@foot}
3035 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
3036   \ifx\pltx@foot@penalty\@undefined\else
3037     \ifhmode\null\fi
3038     \ifnum\pltx@foot@penalty=z@\else
3039       \penalty\pltx@foot@penalty
3040       \pltx@foot@penaltyz@
3041     \fi
3042   \fi}

```

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

```

3043 \newcommand\@makefnmark[1]{%
3044   \advance\leftskip 3\jsZw
3045   \parindent 1\jsZw
3046   \noindent
3047   \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。

```

3048 % \def\@xfootnotenext[#1]{%
3049 %   \begingroup
3050 %     \ifnum#1>z@
3051 %       \csname c@\@mpfn\endcsname #1\relax
3052 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
3053 %     \else
3054 %       \unrestored@protected@xdef\@thefnmark{}%
3055 %     \fi
3056 %   \endgroup
3057 %   \@footnotetext}

```

---

ここまでのコードは JS クラスを踏襲する。

---

## 11 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

---

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

```
3058 \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。

```
3059 \def\everyparhook{\jsInhibitGlueAtParTop}
3060 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
3061 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3062 \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す使う危険性が高いので、専用の命令 `\jsc@ig@temp` に置き換えました (Issue #54)。

---

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。

---

```
3063 \ifx j\jsEngine
3064 \def\@inhibitglue{%
3065   \futurelet\@let@token\@@inhibitglue}
3066 \begingroup
3067 \let\GDEF=\gdef
3068 \let\CATCODE=\catcode
3069 \let\ENDGROUP=\endgroup
3070 \CATCODE`k=12
3071 \CATCODE`a=12
3072 \CATCODE`n=12
3073 \CATCODE`j=12
3074 \CATCODE`i=12
3075 \CATCODE`c=12
3076 \CATCODE`h=12
3077 \CATCODE`r=12
3078 \CATCODE`t=12
```

```

3079 \CATCODE`e=12
3080 \GDEF\KANJI@CHARACTER{kanji character }
3081 \ENDGROUP
3082 \def\@@inhibitglue{%
3083   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
3084   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3085   \def\jsc@ig@temp{#1}%
3086   \ifx\jsc@ig@temp\@empty
3087     \ifnum\the\inhibitxspcode`#2=2\relax
3088       \inhibitglue
3089     \fi
3090   \fi}
3091 \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。

---

```

3092 \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。

```

3093 \def\@doendpe{%
3094   \@endpetrue
3095   \def\par{%
3096     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3097   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3098 \def\@setminipage{%
3099   \@minipagetrue
3100   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3101 }

```

`\item` 命令の直後です。

```

3102 \def\@item[#1]{%
3103   \if@noperitem
3104     \@donoperitem
3105   \else
3106     \if@inlabel
3107       \indent \par
3108     \fi
3109     \ifhmode
3110       \unskip\unskip \par
3111     \fi
3112     \if@newlist

```

```

3113     \if@nobreak
3114     \@nbitem
3115     \else
3116         \addpenalty\@beginparpenalty
3117         \addvspace\@topsep
3118         \addvspace{-\parskip}%
3119     \fi
3120 \else
3121     \addpenalty\@itempenalty
3122     \addvspace\itemsep
3123 \fi
3124 \global\@inlabeltrue
3125 \fi
3126 \everypar{%
3127     \@minipagefalse
3128     \global\@newlistfalse
3129     \if@inlabel
3130         \global\@inlabelfalse
3131         {\setbox\z@\lastbox
3132         \ifvoid\z@
3133             \kern-\itemindent
3134         \fi}%
3135     \box\@labels
3136     \penalty\z@
3137 \fi
3138 \if@nobreak
3139     \@nobreakfalse
3140     \clubpenalty \@M
3141 \else
3142     \clubpenalty \@clubpenalty
3143     \everypar{\everyparhook}%
3144 \fi
3145 \everyparhook}%
3146 \if@noitemarg
3147     \@noitemargfalse
3148     \if@nmbrlist
3149         \refstepcounter\@listctr
3150     \fi
3151 \fi
3152 \sbox\@tempboxa{\makelabel{#1}}%
3153 \global\setbox\@labels\hbox{%
3154     \unhbox\@labels
3155     \hskip \itemindent
3156     \hskip -\labelwidth
3157     \hskip -\labelsep
3158     \ifdim \wd\@tempboxa >\labelwidth
3159         \box\@tempboxa
3160 \else
3161     \hbox to\labelwidth {\unhbox\@tempboxa}%

```

```

3162 \fi
3163 \hskip \labelsep}%
3164 \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。

```

3165 \def\@afterheading{%
3166 \@nobreakefalse
3167 \everypar{%
3168 \if@nobreakefalse
3169 \clubpenalty \@M
3170 \if@afterindent \else
3171 {\setbox\z@\lastbox}%
3172 \fi
3173 \else
3174 \clubpenalty \@clubpenalty
3175 \everypar{\everyparhook}%
3176 \fi\everyparhook}}

```

---

「`\everyparhook` 用のパッチ処理」はここまで。

---

```

3178 \fi

```

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

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

---

※`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。

---

```

3179 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none\else
3180 \AtEndOfPackage{%
3181 \def\@gnewline #1{%
3182 \ifvmode
3183 \@nolnerr
3184 \else
3185 \unskip \reserved@a {\reserved@f#1}\nobreake \hfil \break \null
3186 \jsInhibitGlue \ignorespaces
3187 \fi}
3188 }
3189 \fi

```



## 12 いろいろなロゴ

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

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

---

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

※`\小`、`\上小` の制御綴は定義しない。

---

```
3190 \if@jslogo
3191   \IfFileExists{jslogo.sty}{%
3192     \RequirePackage{jslogo}%
3193   }{%
3194     \ClassWarningNoLine{bxjs@clsname}
3195       {The package 'jslogo' is not installed.\MessageBreak
3196        It is included in the recent release of\MessageBreak
3197        the 'jsclasses' bundle}
3198   }
3199 \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 `ltxclass.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。

```
3200 %\let\ltx@ifnextchar\@ifnextchar
3201 %\def\ProvidesFile#1{%
3202 %   \begingroup
3203 %     \catcode`\ 10 %
3204 %     \ifnum \endlinechar<256 %
3205 %       \ifnum \endlinechar>\m@ne
3206 %         \catcode\endlinechar 10 %
3207 %       \fi
3208 %     \fi
3209 %     \@makeother\/%
```

```

3210 %      \@makeother\&%
3211 %      \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 3212 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3213 \newcommand{\postpartname}{\if@english\else 部\fi}
3214 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3215 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3216 \newcommand{\presectionname}{}% 第
\postsectionname 3217 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3218 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3219 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3220 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3221 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3222 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3223 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3224 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3225 %<jspf>\newcommand{\figurename}{Fig.~}
3226 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3227 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3228 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3229 \newcommand{\appendixname}{\if@english \else 付録\fi}
3230 %<book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

---

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。

---

```

3231 \@onlypreamble\bxjs@decl@Seireki@cmds

```

```

3232 \@tempswafalse
3233 \if p\jsEngine \@tempwattrue \fi
3234 \if n\jsEngine \@tempwattrue \fi
3235 \bxjs@cond\if@tempswa\fi{%
3236 % 欧文 8bitTeX の場合
3237 \newif\ifjsSeireki \jsSeirekitrue
3238 \def\bxjs@decl@Seireki@cmds{%
3239   \def\西暦{\jsSeirekitrue}%
3240   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3241 \def\Seireki{\jsSeirekitrue}
3242 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3243 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3244 \def\bxjs@iai{\noexpand~}
3245 }{%
3246 \newif\if 西暦 \西暦 true
3247 \def\bxjs@decl@Seireki@cmds{%
3248   \def\西暦{\西暦 true}%
3249   \def\和暦{\西暦 false\bxjs@wareki@used}}
3250 \def\Seireki{\西暦 true}
3251 \def\Wareki{\西暦 false\bxjs@wareki@used}
3252 \def\bxjs@if@use@seireki{\bxjs@cond\if 西暦\fi}
3253 \let\bxjs@iai@empty
3254 }
3255 \bxjs@decl@Seireki@cmds
3256 \let\bxjs@unxp@firstofone \let\bxjs@onxp@firstofone
3257 \bxjs@test@engine\unexpanded{%
3258   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}

```

`\ifbxjs@bxwareki@avail` `bxwareki` パッケージが使用できるか。

```

3259 \newif\ifbxjs@bxwareki@avail
3260 \IfFileExists{bxwareki.sty}{%
3261   \RequirePackage{bxwareki}[]%
3262   \bxjs@bxwareki@availtrue}{%

```

`\bxjs@wareki@used` 和暦が非対応の場合に警告を出す。

```

3263 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\empty
3264 \else
3265   \bxjs@robust@def\bxjs@wareki@used{%
3266     \global\let\bxjs@wareki@used\empty
3267     \ClassWarning\bxjs@clsname
3268       {Wareki mode is not supported, since\MessageBreak
3269         'bxwareki' is unavailable, found}}
3270   \g@addto@macro\bxjs@begin@document@hook{%
3271     \let\bxjs@wareki@used\empty}
3272 \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。

```

3273 \ifbxjs@bxwareki@avail
3274   \let\jyear\warekiyear
3275   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3276     \newcount\heisei \heisei=\value{warekiyear}
3277   \fi

    bxwareki が使えない場合は西暦表示にする。

3278 \else
3279   \edef\jyear{\the\year \bxjs@iai}
3280 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3281 \let\bxjs@next\relax
3282 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3283   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3284 \fi\fi
3285 \edef\bxjs@today{%
3286   \if@english
3287     \ifcase\month\or
3288       January\or February\or March\or April\or May\or June\or
3289       July\or August\or September\or October\or November\or December\fi
3290     \space\number\day, \number\year
3291   \else
3292     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3293     \else \noexpand\bxjs@if@use@seireki
3294     \fi {%
3295       \number\year\bxjs@iai\bxjs@unxp{年}%
3296       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3297       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3298     }\bxjs@next}%
3299   \fi}
3300 \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。

```

3301 \g@addto@macro\bxjs@begin@document@hook{%
3302   \ifx\bbl@jpn@maybekansuji\@undefined\else
3303     \bxjs@decl@Seireki@cmds
3304     \g@addto@macro\datejapanese{%
3305       \let\today\bxjs@today}%
3306   \fi}

```

---

■ハイフネーション例外 TeX のハイフネーションルールの補足です (ペンディング: english)


```

3307 \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。

```
3308 %<slide>\pagestyle{empty}%
3309 %<article|report>\pagestyle{plain}%
3310 %<book>\pagestyle{headings}%
3311 \pagenumbering{arabic}
3312 \if@twocolumn
3313   \twocolumn
3314   \sloppy
3315   \flushbottom
3316 \else
3317   \onecolumn
3318   \raggedbottom
3319 \fi
3320 %<*slide>
3321 \renewcommand\familydefault{\sfdefault}
3322 \raggedright
3323 %</slide>
```

■BXJS 独自の追加処理 

---

フックを実行する。

```
3324 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3325 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

---

```
3326 %</class>
```

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。

```
3327 %<*drv>
```

## 付録 B 和文ドライバ : minimal

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

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

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

### B.1 補助マクロ

```
3328 %<*minimal>
```

```
3329 %% 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.
```

```
3330 \def\DeclareJaTextFontCommand#1#2{%
```

```
3331   \DeclareRobustCommand#1[1]{%
```

```
3332     \relax
```

```
3333     \ifmmode \expandafter\nfss@text \fi
```

```
3334     {#2##1}}%
```

```
3335 }
```

```
\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。
```

```
3336 \def\DeclareJaMathFontCommand#1#2{%
```

```
3337   \DeclareRobustCommand#1[1]{%
```

```
3338     \relax
```

```
3339     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3340     \nfss@text{\fontfamily\familydefault
```

```
3341       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3342       #2##1}}%
```

```

3343 }%
3344 }

```

`\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。

```

3345 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3346 \@onlypreamble\bxjs@if@sf@default
3347 \def\bxjs@if@sf@default#1{%
3348   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3349   \g@addto@macro\bxjs@begin@document@hook{%
3350     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3351 }

```

`\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。

```

3352 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3353 \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。

```

3354 \def\jsLetHeadChar#1#2{%
3355   \begingroup
3356     \escapechar=`\ %
3357     \let\bxjs@tmpa={% brace-match-hack
3358       \bxjs@let@hchar@exp#2}%
3359   \endgroup
3360   \let#1\bxjs@g@tmpa}
3361 \def\bxjs@let@hchar@exp{%
3362   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3363 \def\bxjs@let@hchar@exp@a{%
3364   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3365   \bxjs@let@hchar@out\let\relax
3366   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3367   \bxjs@let@hchar@out\let\space%
3368   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3369   \bxjs@let@hchar@out\let\@backslashchar
3370   }{\bxjs@let@hchar@exp@b}}}%
3371 \def\bxjs@let@hchar@exp@b#1{%
3372   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3373 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3374   %\message{<#1#2>}%
3375   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```



```

3376 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3377 \bxjs@let@hchar@out\let\relax
3378 }{%else
3379 \expandafter\bxjs@let@hchar@exp
3380 }%
3381 }{%else
3382 \bxjs@let@hchar@chr#1%
3383 }}
3384 \def\bxjs@let@hchar@chr#1{%
3385 \bxjs@let@hchar@out\def{{#1}}}}
3386 \def\bxjs@let@hchar@out#1#2{%
3387 \global#1\bxjs@g@tmpa#2\relax
3388 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3389 \chardef\bxjs@let@hchar@csta=128
3390 \chardef\bxjs@let@hchar@cstb=192
3391 \chardef\bxjs@let@hchar@cstc=224
3392 \chardef\bxjs@let@hchar@cstd=240
3393 \chardef\bxjs@let@hchar@cste=248
3394 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3395 \def\bxjs@let@hchar@chr@ue#1{%
3396 \@tempcnta=#1\relax
3397 %\message{\the\@tempcnta}%
3398 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3399 \bxjs@let@hchar@chr@ue@a#1%
3400 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3401 \bxjs@let@hchar@out\let\relax
3402 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3403 \bxjs@let@hchar@chr@ue@b
3404 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3405 \bxjs@let@hchar@chr@ue@c
3406 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3407 \bxjs@let@hchar@chr@ue@d
3408 }{%else
3409 \bxjs@let@hchar@out\let\relax
3410 }}}}}
3411 \def\bxjs@let@hchar@chr@ue@a#1{%
3412 \bxjs@let@hchar@out\def{{#1}}}}
3413 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3414 \bxjs@let@hchar@out\def{{#1#2}}}}
3415 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3416 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3417 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3418 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

## B.2 (u)pTeX 用の設定

```
3419 \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。

```
3420 \def\bxjs@let@hchar@chr@pp#1#2{%
3421   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3422 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3423   %\message{(\meaning#3:\meaning#4)}%
3424   \bxjs@cond\if#1k\fi{%
3425     \bxjs@let@hchar@out\def{#{#4}}%
3426   }{%else
3427     \bxjs@let@hchar@chr@ue#3#4%
3428   }}
3429 \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。

```
3430 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3431 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3432 \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。

```
3433 \@onlypreamble\bxjs@declarefontshape
3434 \ifjsWithupTeX
3435 \def\bxjs@declarefontshape{%
3436   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3437   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3438   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3439   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3440 }
3441 \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。

```
3442 \else
3443 \def\bxjs@declarefontshape{%
3444   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3445   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3446   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3447 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3448 }
3449 \def\bxjs@sizereference{jis}
3450 \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。

```

3451 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3452   \def\bxjs@tmpb{#5}}
3453 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3454 \else
3455 \expandafter\expandafter\expandafter\bxjs@next
3456 \expandafter\string\the\jfont\relax
3457 \fi
3458 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3459                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3460   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3461    \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。

```

3462 \begingroup
3463 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3464 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3465 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3466 % 幅が丁度 10pt なら補正は不要
3467 \ifdim\wd\z@=10pt
3468   \global\let\bxjs@scale\jsScale
3469 \else
3470 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3471   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3472   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3473   \bxjs@invscale\@tempdima\bxjs@tmpa
3474   \xdef\bxjs@scale{\strip@pt\@tempdima}
3475 \fi
3476 \endgroup
3477 %\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。

```

3478 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3479 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3480 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3481 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3482 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
3483 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
3484 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
3485 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
3486 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
3487 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
3488 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
3489 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
3490 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
3491 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
3492 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
3493 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
3494 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{ }
3495 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
3496 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
3497 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
3498 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
3499 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
3500 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
3501 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
3502 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
3503 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
3504 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
3505 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
3506 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
3507 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
3508 \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。

```

3509 \@ifl@t@r\fmtversion{2020/10/01}
3510   {\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
3511 \ifjsc@needspace@tch % --- for 2020-02-02 or older BEGIN
3512 \ifx\@rmfamilyhook\@undefined % old
3513 \DeclareRobustCommand\rmfamily
3514   {\not@math@alphabet\rmfamily\mathrm
3515    \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3516 \DeclareRobustCommand\sffamily
3517   {\not@math@alphabet\sffamily\mathsf
3518    \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3519 \DeclareRobustCommand\ttfamily
3520   {\not@math@alphabet\ttfamily\mathtt
3521    \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3522 \g@addto@macro\bxjs@begin@document@hook{%
3523   \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3524     % my definitions above should have been overwritten, recover it!
3525     % \selectfont is executed twice but I don't care about speed...
3526     \expandafter\g@addto@macro\csname rmfamily \endcsname

```

```

3527     {\kanjifamily\mcdefault\selectfont}%
3528     \expandafter\g@addto@macro\csname sffamily \endcsname
3529     {\kanjifamily\gtdefault\selectfont}%
3530     \expandafter\g@addto@macro\csname ttfamily \endcsname
3531     {\kanjifamily\gtdefault\selectfont}%
3532     \fi}
3533 \else                                     % 2020-02-02
3534 \g@addto@macro\@rmfamilyhook
3535     {\prepare@family@series@update@kanji{mc}\mcdefault}
3536 \g@addto@macro\@sffamilyhook
3537     {\prepare@family@series@update@kanji{gt}\gtdefault}
3538 \g@addto@macro\@ttfamilyhook
3539     {\prepare@family@series@update@kanji{gt}\gtdefault}
3540 \fi
3541 \else % --- for 2020-02-02 or older END & for 2020-10-01 BEGIN
3542 \AddToHook{rmfamily}%
3543     {\prepare@family@series@update@kanji{mc}\mcdefault}
3544 \AddToHook{sffamily}%
3545     {\prepare@family@series@update@kanji{gt}\gtdefault}
3546 \AddToHook{ttfamily}%
3547     {\prepare@family@series@update@kanji{gt}\gtdefault}
3548 \fi % --- for 2020-10-01 END
3549 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3550 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3551 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3552 \fi
3553 \bxjs@if@sf@default{%
3554     \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3555 \selectfont

```

`\bxjs@parse@qh` の処理は p<sub>T</sub><sub>E</sub>X 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）。

```

3556 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3557 \let\bxjs@parse@qh@a\undefined
3558 \let\bxjs@parse@qh@b\undefined

```

#### ■パラメタの設定

```

3559 \prebreakpenalty\jis"2147=10000
3560 \postbreakpenalty\jis"2148=10000
3561 \prebreakpenalty\jis"2149=10000
3562 \inhibitxspcode`!=1
3563 \inhibitxspcode`〒=2
3564 \xspcode`+=3
3565 \xspcode`\%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3566 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%

```

```
3567 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
```

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

```
3568 \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?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。

```
3569 \begingroup
```

```
3570 \catcode`\!=0
```

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

```
3571 \gdef\bxjs@ptex@dir{%
```

```
3572 !iftdir t%
```

```
3573 !else!ifydir y%
```

```
3574 !else ?%
```

```
3575 !fi!fi}
```

新版の pTeX で脚注番号の周囲の空きが過大になる現象への対処。

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

```
3576 % 古い \@makefnmark の定義
```

```
3577 \long\def\bxjs@tmpa{\hbox{%
```

```
3578 !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
```

```
3579 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
```

```
3580 \ifx\@makefnmark\bxjs@tmpa
```

```
3581 \long\gdef\@makefnmark{%
```

```
3582 !ifydir \hbox{}\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}%
```

```
3583 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
```

```
3584 \fi
```

エスケープ文字の変更はここまで。

```
3585 \endgroup
```

■minijs パッケージのブロック やっておく。

```
3586 \@namedef{ver@minijs.sty}{}
```

### B.3 pdfTeX 用の処理

```
3587 \else\ifx p\jsEngine
```

```
3588 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

```
3589 \@onlypreamble\bxjs@cjk@loaded
```

```
3590 \def\bxjs@cjk@loaded{%
```

```
3591 \def\@footnotemark{%
```

```
3592 \leavevmode
```

```
3593 \ifhmode
```

```

3594 \edef\x@sf{\the\spacefactor}%
3595 \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3596 \unkern\unkern
3597 \ifdim\lastskip>\z@ \unskip \fi
3598 \fi\fi
3599 \nobreak
3600 \fi
3601 \@makefnmark
3602 \ifhmode \spacefactor\x@sf \fi
3603 \relax}%
3604 \let\bxjs@cjk@loaded\relax
3605 }
3606 \g@addto@macro\bxjs@begin@document@hook{%
3607 \ifpackageloaded{CJK}{%
3608 \bxjs@cjk@loaded
3609 }{}%
3610 }

```

## B.4 X<sub>3</sub>TEX 用の処理

```

3611 \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。

```

3612 \def\bxjs@let@hchar@chr#1{%
3613 \tempcnta`#1\relax \divide\tempcnta"800\relax
3614 \bxjs@cond\ifnum\tempcnta=27 \fi{%
3615 \bxjs@let@hchar@chr@xe
3616 }{\bxjs@let@hchar@out\def{{#1}}}}
3617 \def\bxjs@let@hchar@chr@xe#1{%
3618 \lccode`0=`#1\relax
3619 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3620 \onlypreamble\bxjs@do@precisetext
3621 \ifx\XeTeXgenerateactualtext\undefined\else
3622 \def\bxjs@do@precisetext{%
3623 \XeTeXgenerateactualtext=\@ne}
3624 \fi

```

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

```

3625 \onlypreamble\bxjs@do@simplejasetup
3626 \def\bxjs@do@simplejasetup{%
3627 \ifnum\XeTeXinterchartokenstate>\z@
3628 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3629 \jsSimpleJaSetup
3630 \ClassInfo\bxjs@clsname
3631 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3632 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```
3633 \newcommand*{\jsSimpleJaSetup}{%
3634   \XeTeXlinebreaklocale "ja"\relax
3635   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3636   \XeTeXlinebreakpenalty=0\relax}
```

## B.5 後処理（エンジン共通）

```
3637 \fi\fi\fi
```

`simplejasetup` オプションの処理。

```
3638 \ifx\bxjs@do@simplejasetup\@undefined\else
3639   \g@addto@macro\bxjs@begin@document@hook{%
3640     \ifbxjs@simplejasetup
3641       \bxjs@do@simplejasetup
3642     \fi}
3643 \fi

3644 \ifbxjs@precisetext
3645   \ifx\bxjs@do@precisetext\@undefined
3646     \ClassWarning\bxjs@clsname
3647       {The current engine does not support the\MessageBreak
3648        'precise-text' option\@gobble}
3649   \else
3650     \bxjs@do@precisetext
3651   \fi
3652 \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。

```
3653 \g@addto@macro\bxjs@begin@document@hook{%
3654   \ifx\jsInhibitGlueAtParTop\@empty
3655     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3656     \ifx\everyparhook\bxjs@tmpa
3657       \let\everyparhook\@empty
3658     \fi
3659   \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。

```
3660 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@modern

3661   まず \everypar を “乗っ取る” 処理を行う。
3662   \let\bxjs@everypar\everypar
3663   \newtoks\everypar
3664   \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。



```

3664 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3665 \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。

```

3666 \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。

```

3667 \@onlypreamble\bxjs@adjust@fancyhdr
3668 \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。

```

3669 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3670 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3671 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3672 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3673 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3674 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3675 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3676 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3677 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3678 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3679 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3680 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3681 \def\bxjs@tmpa{\rm\thepage\strut}%
3682 \def\bxjs@tmpb{\thepage\strut}%
3683 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3684 \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。

```

3685 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth
3686 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3687 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3688 }\bxjs@tmpa
3689 \fi\fi
3690 \PackageInfo\bxjs@clsname
3691 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3692 \def\bxjs@pagestyle@hook{%
3693 \ifpackageloaded{fancyhdr}{%

```

```

3694 \bxjs@adjust@fancyhdr
3695 \global\let\bxjs@adjust@fancyhdr\relax
3696 }{}

```

\pagestyle にフックを入れ込む。

```

3697 \let\bxjs@org@pagestyle\pagestyle
3698 \def\pagestyle{%
3699 \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。

```

3700 \g@addto@macro\bxjs@begin@document@hook{%
3701 \bxjs@pagestyle@hook
3702 \global\let\bxjs@pagestyle@hook\relax}

```

## ■和文空白命令

```

3703 \ifbxjs@jaspace@cmd

```

\jaenspace 半角幅の水平空き。

```

3704 \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。

```

3705 \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。

```

3706 \def\_ {\zwspace}

```

\jaspace jlreq クラスと互換の命令。

```

3707 \DeclareRobustCommand*{\jaspace}[1]{%
3708 \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3709 \ClassError\bxjs@clsname
3710 {Unknown jaspac: #1}{\@eha}%
3711 \else
3712 \csname bxjs@jaspace@@#1\endcsname
3713 \fi}
3714 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3715 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3716 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}

```

```

3717 \fi

```

終わり。

```

3718 \fi

```

以上で終わり。

```

3719 %</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/\gttfamily`
- `\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

#### C.1 準備

まず `minimal` ドライバを読み込む。

```
3720 %<*standard>
3721 %% このファイルは日本語文字を含みます
3722 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3723 \bxjs@simplejasetupfalse
```

#### C.2 和文ドライバパラメタ

`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@jismniv` 2004JIS 字形を優先させるか。

```
3724 \newif\ifbxjs@jp@jismniv

jis2004 オプションの処理。
3725 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismnivtrue
3726 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismnivfalse
3727 \define@key{bxjsStd}{jis2004}[true]{%
3728   \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。

```
3729 \newif\ifbxjs@jp@units

units オプションの処理。
```

```

3730 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3731 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3732 \define@key{bxjsStd}{units}[true]{%
3733   \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。

```

3734 \let\bxjs@jp@font\@empty

font オプションの処理。

3735 \define@key{bxjsStd}{font}{%
3736   \edef\bxjs@jp@font{#1}}

```

`\ifbxjs@jp@strong@cmd` `\strong` 命令を補填するか。

```

3737 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue

strong-cmd オプションの処理。

3738 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
3739 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
3740 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}

```

実際の `japaram` の値を適用する。

```

3741 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3742 \expandafter\bxjs@next\expandafter{\jsJaParam}

```

### C.3 共通処理 (1)

■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。

```

3743 \@onlypreamble\bxjs@apply@mmiv
3744 \def\bxjs@apply@mmiv{%
3745   \g@addto@macro\@classoptionslist{,jis2004}
3746 % \@ifpackagewith 判定への対策
3747   \PassOptionsToPackage{jis2004}{otf}
3748   \global\let\bxjs@apply@mmiv\relax}
3749 \ifbxjs@jp@jismmiv \bxjs@apply@mmiv \fi

```

■和文用単位のサポート エンジンが (u)pTeX の場合は `units` を無効にする。

```

3750 \if j\jsEngine
3751   \bxjs@jp@unitsfalse
3752 \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。

```

3753 \ifbxjs@jp@units
3754   \IfFileExists{bxcalc.sty}{%
3755     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3756     \ifx\usepTeXunits\@undefined

```

```

3757 \PackageWarningNoLine\bxjs@clsname
3758 {Cannot support pTeX units (zw etc.), since\MessageBreak
3759 the package 'bxcalc' is too old}%
3760 \bxjs@jp@unitsfalse
3761 \else \usepTeXunits
3762 \fi
3763 }{%else
3764 \PackageWarningNoLine\bxjs@clsname
3765 {Cannot support pTeX units (zw etc.), since\MessageBreak
3766 the package 'bxcalc' is unavailable}%
3767 \bxjs@jp@unitsfalse
3768 }
3769 \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。

```

3770 \ifbxjs@jp@units
3771 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3772 \let\bxjs@parse@qh@a\@undefined
3773 \let\bxjs@parse@qh@b\@undefined
3774 \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。

```

3775 \ifbxjs@jp@units
3776 \def\bxjs@let@lenexpr#1#2{%
3777 \edef#1{#2}%
3778 \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3779 \else
3780 \def\bxjs@let@lenexpr{\edef}
3781 \fi

```

## ■\strong 命令の補填

\strong fontspec で提供される \strong 命令と strongenv 環境を全てのエンジンで使えるよう strongenv にする。

※

```

3782 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3783 \ifx\strong\@undefined\ifx\strongenv\@undefined
3784 \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3785 \DeclareTextFontCommand{\strong}{\strongenv}%

```

fontspec と互換の \strongfontdeclare 命令も提供する。ただし、BX.JS での \strong は多段階には対応しないので、引数のコンマ区切りリストのうちの先頭の項目だけが意味をもつ。既定の設定は \bfseries (太字) である。

※\strongfontdeclare は試験的機能とする。

```

3786 \@ifdefinable{\strongfontdeclare}{\bxjs@protected\def\strongfontdeclare#1{%
3787 \bxjs@strong@font@declare@a#1,\@nil}}%

```

```

3788 \def\bxjs@strong@font@declare@a#1,#2\@nil{\def\bxjs@strong@font{#1}}%
3789 \def\bxjs@strong@font{\bfseries}%
3790 \fi\fi
3791 }\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  $\epsilon$ -TeX 拡張が使えるか  
 検査する。

```

3792 \ifjsWithTeX

```

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

\jQ \jQ と \jH はともに 0.25 mm に等しい。

```

\jH 3793 \@tempdima=0.25mm
3794 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3795 \let\jH\jQ

```

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

```

\trueH 3796 \ifjsc@mag
3797 \@tempdimb=\jsBaseFontSize\relax
3798 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3799 \@tempdima=2.5mm
3800 \bxjs@invscale\@tempdima\bxjs@tmpa
3801 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3802 \@tempdima=10pt
3803 \bxjs@invscale\@tempdima\bxjs@tmpa
3804 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3805 \else \let\trueQ\jQ \let\bxjs@truept\p@
3806 \fi
3807 \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。

```

3808 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3809 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3810 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3811 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3812 \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。

```

3813 \def\jafontsize#1#2{%
3814 \begingroup
3815 \bxjs@jafontsize@a{#1}%
3816 \@tempdimb\jsInverseScale\@tempdima
3817 \bxjs@jafontsize@a{#2}%

```

```

3818 \xdef\bxjs@g@tmpa{%
3819 \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3820 \endgroup\bxjs@g@tmpa}
3821 \def\bxjs@jafontsize@a#1{%
3822 \bxjs@parse@qh{#1}%
3823 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3824 \@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。

```
3825 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```

3826 \newcommand*\setkanjiskip[1]{%
3827 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3828 \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3829 \newcommand*\getkanjiskip{%
3830 \bxjs@kanjiskip}

```

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

```
3831 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

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

```

\bxjs@disable@kanjiskip 3832 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3833 \bxjs@kanjiskip@enabledtrue
3834 \bxjs@reset@kanjiskip}
3835 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3836 \bxjs@kanjiskip@enabledfalse
3837 \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。

```

3838 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3839 \ifbxjs@kanjiskip@enabled
3840 \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3841 \else \@tempskipa\z@
3842 \fi
3843 \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 3844 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3845 \newcommand*\setxkanjiskip[1]{%
```

```

3846 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3847 \bxjs@reset@xkanjiskip}

```

```
\bxjs@enable@xkanjiskip 3848 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3849 \bxjs@xkanjiskip}
```

```
\bxjs@reset@xkanjiskip
```

```

3850 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
3851 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3852   \bxjs@xkanjiskip@enabledtrue
3853   \bxjs@reset@xkanjiskip}
3854 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3855   \bxjs@xkanjiskip@enabledfalse
3856   \bxjs@reset@xkanjiskip}
3857 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3858   \ifbxjs@xkanjiskip@enabled
3859     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3860   \else \@tempskipa\z@
3861   \fi
3862   \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。

```

3863 \g@addto@macro\jsResetDimen{%
3864   \bxjs@reset@kanjiskip
3865   \bxjs@reset@xkanjiskip}
3866 \let\bxjs@apply@kanjiskip\relax
3867 \let\bxjs@apply@xkanjiskip\relax

```

■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、T<sub>E</sub>X 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。

\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。

```

3868 \@onlypreamble\bxjs@adjust@jafont
3869 \def\bxjs@adjust@jafont#1{%
3870   \ifx\jsJaFont\bxjs@auto
3871     \bxjs@get@kanjiEmbed
3872     \ifx\bxjs@jaEmbed\relax
3873       \let\bxjs@tmpa\@empty
3874     \else
3875       \let\bxjs@tmpa\bxjs@jaEmbed
3876       \ifx\bxjs@jaVariant\bxjs@hziv
3877         \bxjs@apply@mmiv
3878       \fi
3879     \fi
3880   \else
3881     \let\bxjs@tmpa\jsJaFont
3882   \fi
3883   \if f#1\ifx\bxjs@tmpa\bxjs@noEmbed
3884     \ClassWarningNoLine\bxjs@clsname
3885       {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak

```



```

3886      not available on the current situation}%
3887      \let\bxjs@tmpa\@empty
3888      \fi\fi
3889  }
3890  \def\bxjs@@auto{auto}
3891  \def\bxjs@@noEmbed{noEmbed}
3892  \def\bxjs@@hziv{-04}

```

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

※古い版の updmap では kanjiEmbed・kanjiVariant であった。

```

3893 \let\bxjs@jaEmbed\relax
3894 \let\bxjs@jaVariant\relax

```

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

```

3895 \@onlypreamble\bxjs@get@kanjiEmbed
3896 \def\bxjs@get@kanjiEmbed{%
3897   \begingroup\setbox\z@=\hbox{%
3898     \global\let\bxjs@tmpdo\@empty
3899     \def\bxjs@next##1##2##3{%
3900       \def##1###1##3 ###2\@nil###3\@nnil{%
3901         \ifx$###1$\gdef##2{###2}\fi}%
3902       \g@addto@macro\bxjs@tmpdo{%
3903         \expandafter##1\bxjs@tmpa\@nil##3 \@nil\@nnil}}%
3904       \bxjs@next\bxjs@tmpdo@a\bxjs@g@tmpa{kanjiEmbed}%
3905       \bxjs@next\bxjs@tmpdo@b\bxjs@g@tmpa{jaEmbed}%
3906       \bxjs@next\bxjs@tmpdo@c\bxjs@g@tmpb{kanjiVariant}%
3907       \bxjs@next\bxjs@tmpdo@d\bxjs@g@tmpb{jaVariant}%
3908   }%
3909   \global\let\bxjs@g@tmpa\relax
3910   \global\let\bxjs@g@tmpb\relax
3911   \endlinechar\m@ne
3912   \let\do\@makeother\dospecials
3913   \catcode32=10 \catcode12=10 %form-feed
3914   \let\bxjs@tmpa\@empty
3915   \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3916   \ifeof\@inputcheck\else
3917     \read\@inputcheck to\bxjs@tmpa
3918     \closein\@inputcheck
3919   \fi
3920   \ifx\bxjs@tmpa\@empty\else
3921     \openin\@inputcheck="\bxjs@tmpa"\relax
3922     \@tempswatrue
3923     \loop\if@tempswa
3924       \read\@inputcheck to\bxjs@tmpa
3925       \bxjs@tmpdo
3926       \ifeof\@inputcheck \@tempswafalse \fi
3927     \repeat
3928   \fi

```

```

3929 } \endgroup
3930 \let\bxjs@jaEmbed\bxjs@g@tmpa
3931 \let\bxjs@jaVariant\bxjs@g@tmpb
3932 }

```

`\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。

```

3933 \@onlypreamble\bxjs@resolve@jafont@paren
3934 \def\bxjs@resolve@jafont@paren#1{%
3935   \def\bxjs@tmpb{\let#1}%
3936   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\@nnil#1}
3937 \@onlypreamble\bxjs@resolve@jafont@paren@a
3938 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\@nnil#5{%
3939   \ifx\relax#4\relax \bxjs@tmpb#5%
3940   \else
3941     \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3942     \bxjs@tmpb\bxjs@tmpa
3943   \fi}

```

■和文として出力 「欧文扱い」となっている文字を和文として出力するための機能。

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

```

3944 \newcommand*\jachar[1]{%
3945   \begingroup
3946     \jsLetHeadChar\bxjs@tmpa{#1}%
3947     \ifx\bxjs@tmpa\relax
3948       \ClassWarningNoLine\bxjs@clsname
3949       {Illegal argument given to \string\jachar}%
3950     \else
3951       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3952     \fi
3953   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3954 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3955 \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。

```

3956 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3957 \@onlypreamble\bxjs@fix@hyperref@unicode
3958 \def\bxjs@fix@hyperref@unicode#1{%

```

```

3959 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3960 \@namedef{KV@Hyp@bxjs/hook}##1{%
3961   \KV@Hyp@unicode{##1}%
3962   \def\KV@Hyp@unicode####1{%
3963     \expandafter\ifx\csname if##1\expandafter\endcsname
3964       \csname if####1\endcsname\else
3965       \ClassWarningNoLine{bxjs@clsname
3966         {Blcoked hyperref option 'unicode=####1'}}%
3967     \fi
3968   }%
3969 }%
3970 }

```

`\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。

```

3971 \@onlypreamble\jsCheckHyperrefUnicode
3972 \let\jsCheckHyperrefUnicode\@empty
3973 \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。

```

3974 \@onlypreamble\bxjs@check@hyperref@unicode
3975 \def\bxjs@check@hyperref@unicode#1{%
3976   \g@addto@macro\jsCheckHyperrefUnicode{%
3977     \@tempwafalse
3978     \begingroup
3979       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3980         \aftergroup\@tempwattrue \fi
3981       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3982         \csname if#1\endcsname
3983         \aftergroup\@tempwattrue \fi
3984     \endgroup
3985     \if@tempwa\else
3986       \ClassError{bxjs@clsname
3987         {The value of hyperref 'unicode' key is not suitable\MessageBreak
3988           for the present engine (must be #1)}}%
3989       {\@ehc}%
3990     \fi}}

```

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

```

3991 \@onlypreamble\bxjs@urgent@special
3992 \@onlypreamble\bxjs@urgent@special@a

```

LaTeX カーネルの新フック管理が導入済かを調べる。未導入の古い版である場合。

```

3993 \ifbxjs@old@hook@system
3994 \def\bxjs@urgent@special#1{%
3995   \AtBeginDvi{\special{#1}}%
3996   \g@addto@macro\bxjs@begin@document@hook{%
3997     \ifpackageloaded{atbegshi}{%
3998       \begingroup
3999       \toks\z@{\special{#1}}%

```

```

4000      \toks\tw@\expandafter{\AtBegShi@HookFirst}%
4001      \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw}%
4002      \endgroup
4003      }{}%
4004      }%
4005  }

```

導入済の場合。

※自分が先行する必要がある対象のパッケージを適宜追加する。

※pxjahyper パッケージの処理と合わせる。

```

4006 \else
4007   \def\bxjs@urgent@special#1{%
4008     \bxjs@urgent@special@a
4009     \AddToHook{shipout/firstpage}[pxjahyper/enc]{\special{#1}}
4010   \def\bxjs@urgent@special@a{%
4011     \DeclareHookRule{shipout/firstpage}{pxjahyper/enc}{<}{hyperref}%
4012     \global\let\bxjs@urgent@special@a\relax}
4013 \fi

```

## C.4 pTeX 用設定

```

4014 \if j\jsEngine

```

### ■ 共通命令の実装

```

4015 \def\bxjs@apply@kanjiskip{%
4016   \kanjiskip\@tempskipa}
4017 \def\bxjs@apply@xkanjiskip{%
4018   \xkanjiskip\@tempskipa}

```

\jaJaChar のサブマクロ。

```

4019 \def\bxjs@jachar#1{%
4020   \bxjs@jachar@a#1...\@nil}
4021 \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。

```

4022   \ifx.#2#1%

```

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

```

4023   \else\ifx.#3%
4024     \@tempcnta`#1 \multiply\@tempcnta64
4025     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
4026     \bxjs@jachar@b
4027   \else\ifx.#4%
4028     \@tempcnta`#1 \multiply\@tempcnta64
4029     \advance\@tempcnta`#2 \multiply\@tempcnta64
4030     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
4031     \bxjs@jachar@b
4032   \else

```

```

4033 \tempcnta`#1 \multiply\tempcnta64
4034 \advance\tempcnta`#2 \multiply\tempcnta64
4035 \advance\tempcnta`#3 \multiply\tempcnta64
4036 \advance\tempcnta`#4 \advance\tempcnta-"3C82080
4037 \bxjs@jachar@b
4038 \fi\fi\fi}

```

符号値が \tempcnta の和文文字を出力する処理。

```

4039 \ifjsWithupTeX
4040 \def\bxjs@jachar@b{\kchar\tempcnta}
4041 \else
4042 \def\bxjs@jachar@b{%
4043 \ifx\bxUInt\@undefined\else
4044 \bxUInt{\tempcnta}%
4045 \fi}
4046 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

4047 \ifbxjs@jaspace@cmd
4048 \def\jathinspace{\hskip\xkanjiskip}
4049 \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。

```

4050 \ifbxjs@jp@jismmiv
4051 \PassOptionsToPackage{prefer2004jis}{pxchfon}
4052 \PassOptionsToPackage{prefer2004jis}{pxbabel}
4053 \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。

```

4054 \let\bxjs@jafont@paren\@firstofone
4055 \let\bxjs@tmpa\jsJaFont
4056 \ifx\bxjs@tmpa\bxjs@@auto
4057 \let\bxjs@tmpa\@empty
4058 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
4059 \def\bxjs@tmpa{noembed}
4060 \fi\fi
4061 \bxjs@resolve@jafont@paren\bxjs@tmpa
4062 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4063 \ifx\bxjs@tmpa\@empty\else
4064 \edef\bxjs@next{%
4065 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4066 }\bxjs@next
4067 \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……。)

```

4068 \begingroup
4069   \global\let\bxjs@g@tmpa\relax
4070   \catcode`\|=0 \catcode`\|=12
4071   |def|bxjs@tmpdo@a#1|@nil{%
4072     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4073   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4074     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4075   |catcode`\|=0 \catcode`\|=12
4076   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4077     \ifx$#2$\else
4078       \xdef\bxjs@g@tmpa{%
4079         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4080     \fi}
4081 \@firstofone{%
4082   \catcode10=12 \endlinechar\m@ne
4083   \let\do\@makeother \dospecials \catcode32=10
4084   \openin\@inputcheck=otf.sty\relax
4085   \@tempwattrue
4086   \loop\if@tempwa
4087     \ifeof\@inputcheck \@tempwafalse \fi
4088     \if@tempwa
4089       \read\@inputcheck to\bxjs@next
4090       \expandafter\bxjs@tmpdo\bxjs@next\@nil
4091     \fi
4092   \repeat
4093   \closein\@inputcheck
4094 \endgroup}
4095 \bxjs@g@tmpa

```

■hyperref 対策 unicode にしてはいけない。

```

4096 \ifbxjs@hyperref@enc
4097   \bxjs@check@hyperref@unicode{false}

```

暫定的なナニカ。

```

4098   \ifjsWithupTeX\ifbxjs@old@hook@system\else
4099     \IfFileExists{pxjahyper-uni.def}{%
4100       \AddToHook{package/after/hyperref}{\input{pxjahyper-uni.def}}
4101     }{}
4102   \fi\fi
4103 \fi

```

tounicode special 命令を出力する。

```

4104 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4105   \else\ifjsWithpTeXng T\else F\fi\fi T%
4106   \IfFileExists{pxjahyper-enc.sty}{\@tempswatrue}{\@tempswafalse}
4107   \if@tempswa
4108     \RequirePackage{pxjahyper-enc}[2020/10/05]%v0.6
4109     \ifbxjs@bigcode\else \suppressbigcode \fi
4110   \else
4111     \ifnum\jis"2121="A1A1 %euc
4112       \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4113     \else\ifnum\jis"2121="8140 %sjis
4114       \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4115     \else\ifnum\jis"2121="3000 %uptex
4116       \ifbxjs@bigcode
4117         \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4118         \PassOptionsToPackage{bigcode}{pxjahyper}
4119       \else
4120         \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4121         \PassOptionsToPackage{nobigcode}{pxjahyper}
4122       \fi
4123     \fi\fi\fi
4124     \let\bxToUnicodeSpecialDone=t
4125     \fi
4126 \fi

```

■和文数式ファミリ 和文数式ファミリは既定で有効とする。すなわち `enablejfam=false` 以外の場合は `@enablejfam` を真にする。

```

4127 \ifx f\bxjs@enablejfam\else
4128   \@enablejfamtrue
4129 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

4130 \if@enablejfam
4131   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4132   \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4133   \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4134   \jfam\symmincho
4135   \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4136   \g@addto@macro\bxjs@begin@document@hook{%
4137     \ifx\reDeclareMathAlphabet\@undefined\else
4138       \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4139       \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4140       \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4141     \fi}
4142 \fi

```

## C.5 pdf<sub>T</sub>E<sub>X</sub> 用設定 : CJK + bxcjkatype

```

4143 \else\if p\jsEngine

```

■`bxckjatype` 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 `bxckjatype` 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 `bxckjatype` の側で行われる。

※ Pandoc モードでは `autotilde` を指定しない。

```
4144 \bxjs@adjust@jafont{f}
4145 \let\bxjs@jafont@paren\@firstofone
4146 \bxjs@resolve@jafont@paren\bxjs@tmpa
4147 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4148 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4149 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4150   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4151 \fi
4152 \edef\bxjs@next{%
4153   \noexpand\RequirePackage[\bxjs@tmpa]{bxckjatype}[2013/10/15]% v0.2c
4154 }\bxjs@next
4155 \bxjs@cjk@loaded
```

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

※取りあえず固定はしない。

```
4156 \ifbxjs@hyperref@enc
4157   \PassOptionsToPackage{unicode}{hyperref}
4158 \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。

※ `bxckjatype` 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。

```
4159 \ifx\bxckjatypeHyperrefPatchDone\@undefined
4160 \begingroup
4161   \CJK@input{UTF8.bdg}
4162 \endgroup
4163 \g@addto@macro\pdfstringdefPreHook{%
4164   \@nameuse{CJK@UTF8Binding}%
4165 }
4166 \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。

```
4167 \ifx\bxckjatypeHyperrefPatchDone\@undefined
4168 \g@addto@macro\pdfstringdefPreHook{%
4169   \ifx~\bxjs@CJKtilde
4170     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4171     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4172     \let~\@empty
4173   \fi
4174 }
```



```

4175 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4176 \def\bxjs@@tildecmd{~}
4177 \def\bxjs@LetUnexpandableSpace#1{%
4178   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4179     \bxjs@org@LetUnexpandableSpace#1%
4180   \fi}
4181 \fi

```

### ■ 共通命令の実装

```

4182 \newskip\jsKanjiSkip
4183 \newskip\jsXKanjiSkip
4184 \ifx\CJKecglue\@undefined
4185   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4186 \fi
4187 \let\autospacing\bxjs@enable@kanjiskip
4188 \let\noautospacing\bxjs@disable@kanjiskip
4189 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4190 \def\bxjs@apply@kanjiskip{%
4191   \jsKanjiSkip\@tempskipa
4192   \let\CJKglue\bxjs@CJKglue}
4193 \let\autoxspacing\bxjs@enable@xkanjiskip
4194 \let\noautoxspacing\bxjs@disable@xkanjiskip
4195 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4196 \def\bxjs@apply@xkanjiskip{%
4197   \jsXKanjiSkip\@tempskipa
4198   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4199 \def\bxjs@jachar#1{%
4200   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4201 \ifbxjs@jaspace@cmd
4202   \protected\def\jathinspace{\CJKecglue}
4203 \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。

```

4204 \ifx t\bxjs@enablejfam
4205   \ClassWarningNoLine\bxjs@clsname
4206     {You cannot use 'enablejfam=true', since the\MessageBreak
4207       CJK package does not support Japanese math}
4208 \fi

```

## C.6 Xe<sub>La</sub>TeX 用設定：xeCJK + zxjatype

```

4209 \else\if x\jsEngine

```

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

```

4210 \RequirePackage{zxjatype}
4211 \PassOptionsToPackage{no-math}{fontspec}%!
4212 \PassOptionsToPackage{xetex}{graphicx}%!
4213 \PassOptionsToPackage{xetex}{graphics}%!
4214 \ifx\zxJaFamilyName\@undefined
4215   \ClassError\bxjs@clsname
4216   {xeCJK or zxjatype is too old}\@ehc
4217 \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。

```

4218 \bxjs@adjust@jafont{f}
4219 \let\bxjs@jafont@paren\gobble
4220 \bxjs@resolve@jafont@paren\bxjs@tmpa
4221 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4222 \ifx\bxjs@tmpa\@empty
4223   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4224   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4225 \else
4226   \edef\bxjs@next{%
4227     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4228   }\bxjs@next
4229 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

$\text{\XeTeX}$  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、 $\text{\XeTeX}$  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、「 $\text{\XeTeX}$  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。

```

4230 \ifnum\stricmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4231   \ifbxjs@hyperref@enc
4232     \PassOptionsToPackage{unicode}{hyperref}
4233   \fi
4234 \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。

```
4235 \AtEndOfPackage{%
4236 \def\@gnewline #1{%
4237   \ifvmode \@nolnerr
4238   \else
4239     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4240     \nobreak \hskip-1sp\hskip1sp\relax
4241     \ignorespaces
4242   \fi}
4243 }
```

#### ■ 共通命令の実装

```
4244 \newskip\jsKanjiSkip
4245 \newskip\jsXKanjiSkip
4246 \ifx\CJKecglue\@undefined
4247   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4248 \fi
4249 \let\autospacing\bxjs@enable@kanjiskip
4250 \let\noautospacing\bxjs@disable@kanjiskip
4251 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4252 \def\bxjs@apply@kanjiskip{%
4253   \jsKanjiSkip\@tempskipa
4254   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4255 \let\autoxspacing\bxjs@enable@xkanjiskip
4256 \let\noautoxspacing\bxjs@disable@xkanjiskip
4257 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4258 \def\bxjs@apply@xkanjiskip{%
4259   \jsXKanjiSkip\@tempskipa
4260   \xeCJKsetup{CJKecglue={\bxjs@CJKecglue}}}
```

\mcfamily、\gtfamily は本来は zxjatype の方で定義すべきであろうが、現状は暫定的にここで定義する。

```
4261 \ifx\mcfamily\@undefined
4262   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4263   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4264 \fi
```

\jachar のサブマクロの実装。

```
4265 \def\bxjs@jachar#1{%
4266   \xeCJKDeclareCharClass{CJK}{`#1}\relax
4267   #1}
```

\jathinspace の実装。

```
4268 \ifbxjs@jaspace@cmd
4269   \protected\def\jathinspace{\CJKecglue}
4270 \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。

```
4271 \ifx t\bxjs@enablejfam
4272   \@enablejfamtrue
4273 \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。

```
4274 \if@enablejfam
4275   \xeCJKsetup{CJKmath=true}
4276 \fi
```

## C.7 LuaTeX 用設定 : LuaTeX-ja

```
4277 \else\if 1\jsEngine
```

■LuaTeX-ja パッケージの読込 `luatexja` とともに `luatexja-fontspec` パッケージを読み込む。

`luatexja` は自前の `\zw` (これは実際の現在和文フォントに基づく値を返す) を定義するので、`\zw` の定義を消しておく。なお、レイアウト定義の「全角幅」は「規定」に基づく `\jsZw` であることに注意が必要。

※ 1.0b 版から「graphics パッケージに `pdftex` オプションを渡す」処理を行っていたが、1.4 版で廃止された。

```
4278 \let\zw\@undefined
4279 \RequirePackage{luatexja}
4280 \edef\bxjs@next{%
4281   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4282 }\bxjs@next
```

フォント代替の明示的定義。

```
4283 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{-}
4284 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
4285 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
4286 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{-}
4287 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
4288 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
4289 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
4290 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
4291 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
4292 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
4293 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
4294 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
4295 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
4296 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
4297 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{-}
4298 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{-}
4299 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
```

```

4300 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4301 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4302 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4303 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4304 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4305 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4306 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4307 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4308 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4309 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4310 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4311 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4312 \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。

```

4313 \bxjs@adjust@jafont{t}
4314 \ifx\bxjs@tmpa\bxjs@@noEmbed
4315   \def\bxjs@tmpa{noembed}
4316 \fi
4317 \let\bxjs@jafont@paren\@gobble
4318 \bxjs@resolve@jafont@paren\bxjs@tmpa
4319 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4320 \ifx\bxjs@tmpa\@empty
4321   \defaultjfontfeatures{ Kerning=Off }
4322   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4323   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4324 \else
4325   \edef\bxjs@next{%
4326     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4327   }\bxjs@next
4328 \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。

```

4329 \@ifpackagelater{luatexja}{2016/03/31}{ }{%else
4330 \DeclareRobustCommand\rmfamily
4331 { \not@math@alphabet\rmfamily\mathrm
4332   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4333 \DeclareRobustCommand\sffamily
4334 { \not@math@alphabet\sffamily\mathsf
4335   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4336 \DeclareRobustCommand\ttfamily
4337 { \not@math@alphabet\ttfamily\mathtt

```

```

4338 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4339 }
4340 \long\def\jttdefault{\gtdefault}
4341 \unless\ifx\@ltj@match@familytrue\@undefined
4342 \@ltj@match@familytrue
4343 \fi
4344 \g@addto@macro\bxjs@begin@document@hook{%
4345 \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4346 \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4347 \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4348 \bxjs@if@sf@default{%
4349 \renewcommand\kanjifamilydefault{\gtdefault}}

```

#### ■和文パラメタの設定

```

4350 % 次の3つは既定値の通り
4351 \ltjsetparameter{prebreakpenalty={`',10000}}
4352 \ltjsetparameter{postbreakpenalty={`",10000}}
4353 \ltjsetparameter{prebreakpenalty={`",10000}}
4354 \ltjsetparameter{jaxspmode={`!,1}}
4355 \ltjsetparameter{jaxspmode={`〒,2}}
4356 \ltjsetparameter{alxspmode={`+,3}}
4357 \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-j` 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。

```

4358 \ifx\ltjfakeparbegin\@undefined
4359 \protected\def\ltjfakeparbegin{%
4360 \ifhmode
4361 \relax\directlua{%
4362 \lua{texja.jfmglue.create_beginpar_node()}}
4363 \fi}
4364 \fi

```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```

4365 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none
4366 \begingroup
4367 \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4368 \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4369 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
4370 \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4371 \directlua{
4372 local function patchcmd(cs, code, from, to)
4373 tex.sprint(code:gsub(from:gsub("%W", "%\\%0"), "%0"..to)
4374 :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."")
4375 end

```

```

4376 patchcmd(\@xsect, [[\meaning\xsect]],
4377 \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4378 patchcmd(\@item, [[\meaning@item]],
4379 \bxjs@tmpa, \@ltjfakeparbegin)}
4380 \endgroup
4381 \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。

```

4382 \ifbxjs@hyperref@enc
4383 \PassOptionsToPackage{unicode}{hyperref}
4384 \bxjs@check@hyperref@unicode{true}
4385 \fi

```

■共通命令の実装

```

4386 \protected\def\autospacing{%
4387 \ltjsetparameter{autospacing=true}}
4388 \protected\def\noautospacing{%
4389 \ltjsetparameter{autospacing=false}}
4390 \protected\def\autoxspacing{%
4391 \ltjsetparameter{autoxspacing=true}}
4392 \protected\def\noautoxspacing{%
4393 \ltjsetparameter{autoxspacing=false}}
4394 \def\bxjs@apply@kanjiskip{%
4395 \ltjsetparameter{kanjiskip={\@tempskipa}}}
4396 \def\bxjs@apply@xkanjiskip{%
4397 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4398 \def\bxjs@jachar#1{%
4399 \ltjjachar`#1\relax}

```

\jathinspace の実装。

```

4400 \ifbxjs@jaspace@cmd
4401 \protected\def\jathinspace{%
4402 \hskip\ltjgetparameter{xkanjiskip}\relax}
4403 \fi

```

■和文数式ファミリ LuaTeX-japan 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。

```

4404 \ifx f\bxjs@enablejfam
4405 \ClassWarningNoLine\bxjs@clsname
4406 {You cannot use 'enablejfam=false', since the\MessageBreak
4407 LuaTeX-japan always provides Japanese math families}
4408 \fi

```

## C.8 共通処理 (2)

```
4409 \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 4410 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4411 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4412 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4413 \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。

```
4414 \ifx\mathmc\undefined
4415 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4416 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4417 \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)。

```
4418 \ifbxjs@jaspace@cmd
4419 \bxjs@protected\def\bxjs@choice@jathinspace{%
4420 \relax\ifmmode \mskip\medmuskip
4421 \else \jathinspace\ignorespaces
4422 \fi}
4423 \jsAtEndOfClass{%
4424 \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4425 \else \def\>\protect\bxjs@choice@jathinspace}%
4426 \fi}
4427 \fi
```

### ■和文・和欧文間空白の初期値

```
4428 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4429 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4430 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4431 \fi
```

以上で終わり。

```
4432 %</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。

```
4433 %<*modern>
4434 \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。

```
4435 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4436 \def\encodingdefault{T1}%
4437 \input{t1enc.def}%
4438 \fontencoding\encodingdefault\selectfont
4439 \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。

```
4440 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4441 \renewcommand{\rmdefault}{lmr}
4442 \renewcommand{\sfdefault}{lmss}
4443 \renewcommand{\ttdefault}{lmtt}
4444 \fi
```

大型演算子用の数式フォントの設定。

※`amsfonts` パッケージと同等にする。

```
4445 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4446   <-7.5>cmex7<7.5-8.5>cmex8%
4447   <8.5-9.5>cmex9<9.5->cmex10}{}%
4448 \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。
4449 \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。

```
4450 \ifx\@IncludeInRelease\@undefined
4451 \RequirePackage{fixltx2e}
4452 \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。

```
4453 \RequirePackage{bxjsjkat}
```

## D.4 完了

おしまい。

```
4454 %</modern>
```

## 付録 E 和文ドライバ : pandoc

「Pandoc モード」で使用される和文ドライバ。standard ドライバの機能を継承するが、「Pandoc の既定の latex テンプレート」が使われることを前提として、それと BXJS の設定を整合させるための措置を加えている。

### E.1 準備

standard ドライバの設定を引き継ぐ。

```
4455 %<*pandoc>
4456 \input{bxjsja-standard.def}

bxjspandoc パッケージを読み込む。
4457 \RequirePackage{bxjspandoc}

 $\varepsilon$ -TeX ではない場合に警告を出す。
※近い将来に  $\varepsilon$ -TeX 拡張を必須にする予定。
4458 \ifjsWitheTeX\else
4459   \ClassWarningNoLine{bxjs@clsname}
4460   {!!!!!!! WARNING !!!!!!!\MessageBreak
4461     This engine does not support e-TeX extension!\MessageBreak
4462     Some feature might not work properly}
4463 \fi
```

`\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。)

```
4464 \@onlypreamble\bxjs@endpreamble@hook
4465 \let\bxjs@endpreamble@hook\empty

パッケージ読込。
4466 \RequirePackage{iftex}[2013/04/04]% v0.2
4467 \ifjsWitheTeX
4468   \RequirePackage{etoolbox}[2010/08/21]% v2.0
4469   \AtEndPreamble{\bxjs@endpreamble@hook}
4470   \RequirePackage{filehook}[2011/10/12]% v0.5d
4471 \fi
```

### E.2 和文ドライバパラメタ

keyval のファミリーは `bxjsPan` とする。

`\ifbxjs@jp@fix@strong` 重要要素を補正するか。

```
4472 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue

fix-strong オプションの処理。
4473 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
```

```

4474 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
4475 \define@key{bxjsPan}{fix-strong}[true]{%
4476   \bxjs@set@keyval{fixstrong}{#1}{}}

```

`\ifbxjs@jp@fix@code` インラインコード要素を補正するか。

```

4477 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue

```

`fix-code` オプションの処理。

```

4478 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
4479 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
4480 \define@key{bxjsPan}{fix-code}[true]{%
4481   \bxjs@set@keyval{fixcode}{#1}{}}

```

`\bxjs@jp@strong` 重要要素に適用される書体変更の種類。

```

4482 \chardef\bxjs@jp@strong=0

```

`strong` オプションの処理。

```

4483 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
4484 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
4485 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }
4486 \define@key{bxjsPan}{strong}{%
4487   \bxjs@set@keyval{strong}{#1}{}}

```

`\ifbxjs@jp@or@indent` プレアンブルでのレイアウト上書きを許可するか。既定値は真。

```

\ifbxjs@jp@or@secnumdepth 4488 \newif\ifbxjs@jp@or@indent \bxjs@jp@or@indenttrue

```

```

4489 \newif\ifbxjs@jp@or@secnumdepth \bxjs@jp@or@secnumdepthtrue

```

```

\ifbxjs@jp@or@block@heading

```

```

4490 \newif\ifbxjs@jp@or@block@heading \bxjs@jp@or@block@headingtrue

```

クラスで `pandoc+` が指定された場合、内部和文パラメタ `_plus` が和文ドライバに渡される。この場合、レイアウト上書きを禁止する。

※ `_plus` は必ずパラメタ列の先頭にあるので、個別のパラメタ設定の方が常に優先される。

```

4491 \define@key{bxjsPan}{_plus}[]{}
4492   \bxjs@jp@or@indentfalse
4493   \bxjs@jp@or@secnumdepthfalse
4494   \bxjs@jp@or@block@headingfalse}

```

レイアウト上書き許可オプション (`or-indent`・`or-secnumdepth`・`or-block-heading`) の処理。

```

4495 \let\bxjs@kv@orindent@true\bxjs@jp@or@indenttrue
4496 \let\bxjs@kv@orindent@false\bxjs@jp@or@indentfalse
4497 \define@key{bxjsPan}{or-indent}[true]{%
4498   \bxjs@set@keyval{orindent}{#1}{}}
4499 \let\bxjs@kv@orsecnumdepth@true\bxjs@jp@or@secnumdepthtrue
4500 \let\bxjs@kv@orsecnumdepth@false\bxjs@jp@or@secnumdepthfalse
4501 \define@key{bxjsPan}{or-secnumdepth}[true]{%
4502   \bxjs@set@keyval{orsecnumdepth}{#1}{}}
4503 \let\bxjs@kv@orblockheading@true\bxjs@jp@or@block@headingtrue
4504 \let\bxjs@kv@orblockheading@false\bxjs@jp@or@block@headingfalse
4505 \define@key{bxjsPan}{or-block-heading}[true]{%

```

```
4506 \bxjs@set@keyval{blockheading}{#1}{}}
```

実際の japaram の値を適用する。

```
4507 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}
```

```
4508 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

### E.3 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。

```
4509 \@onlypreamble\bxjs@set@dupload@proc
```

```
4510 \def\bxjs@set@dupload@proc#1{%
```

```
4511 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
4512 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
4513 \def\bxjs@set@dupload@proc@a#1{%
```

```
4514 \@onlypreamble#1\def#1##1}
```

```
4515 \def\bxjs@unset@dupload@proc#1{%
```

```
4516 \expandafter\let\csname bxjs@dlp/#1\endcsname\@undefined}
```

`\@if@options` `\@if@options` の再定義。

```
4517 \@onlypreamble\bxjs@org@if@options
```

```
4518 \let\bxjs@org@if@options\@if@options
```

```
4519 \@onlypreamble\bxjs@org@reset@options
```

```
4520 \let\bxjs@org@reset@options\relax
```

```
4521 \def\@if@options#1#2#3{%
```

```
4522 \let\bxjs@next\@secondoftwo
```

```
4523 \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
```

```
4524 \ifx\bxjs@tmpa\bxjs@tmpb
```

```
4525 \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
```

```
4526 \let\bxjs@next\@firstoftwo \fi
```

```
4527 \fi
```

```
4528 \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
```

```
4529 \g@addto@macro\bxjs@begin@document@hook{%
```

```
4530 \let\@if@options\bxjs@org@if@options}
```

```
4531 \@onlypreamble\bxjs@do@dupload@proc
```

```
4532 \def\bxjs@do@dupload@proc#1#2#3{%
```

```
4533 \ifx\bxjs@org@reset@options\relax
```

```
4534 \let\bxjs@org@reset@options\@reset@options
```

```
4535 \fi
```

```
4536 \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
```

```
4537 \def\@reset@options{%
```

```
4538 \let\@reset@options\bxjs@org@reset@options
```

```
4539 \@reset@options
```

```

4540 \bxjs@next{#3}}%
4541 \@firstoftwo}

```

## E.4 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。

```

4542 \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。

```

4543 \let\bxjs@babel@options\relax

```

■Polyglossia について つまり  $X_{\text{D}}\text{T}_{\text{E}}\text{X}$ ・ $\text{LuaT}_{\text{E}}\text{X}$  の場合。

※この場合 `etoolbox` が使用可能になっている。

```

4544 \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。

```

4545 \pandocSkipLoadPackage{polyglossia}
4546 \bxjs@set@dupload@proc{polyglossia.sty}{%
4547 \bxjs@unset@dupload@proc{polyglossia.sty}%
4548 \ClassWarning\bxjs@clsname
4549 {Package polyglossia is required}%
4550 \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。

```

4551 \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。

```

4552 \ifblank{##2}{%
4553 \ClassWarning\bxjs@clsname
4554 {Main language is invalid, thus the fallback\MessageBreak
4555 definitions will be employed}%
4556 \bxjs@pandoc@polyglossia@ja

```

言語名が `japanese` だった場合も同様。

```

4557 }{\ifstrequal{##2}{japanese}{%
4558 \ClassWarning\bxjs@clsname
4559 {Main language is japanese, thus the fallback\MessageBreak
4560 definitions will be employed}%
4561 \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。

```
4562   }{%else
4563     \ClassWarning{bxjs@clsname
4564       {Main language is valid (##2),\MessageBreak
4565         thus polyglossia will be loaded}%
4566     \csundef{ver@polyglossia.sty}%
4567     \edef{bxjs@next}{%
4568       \noexpand\RequirePackage[ $\backslash$ bxjs@polyglossia@options]{polyglossia}[]%
4569     } $\backslash$ bxjs@next
4570     \setmainlanguage[##1]{##2}%
4571   }}}
```

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。

```
4572 \g@addto@macro{bxjs@endpreamble@hook}{%
4573   \ifx{bxjs@polyglossia@options}\relax
4574     \csundef{ver@polyglossia.sty}%
4575   \fi}
```

$\backslash$ 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  $\backslash$ setotherlanguage(s) を定義する。

※現在では Polyglossia の日本語用の定義ファイル (gloss-japanese.ldf) が存在するので、本来なら普通に処理できるはずであるが、現状の定義ファイルはアレなので回避したい。

```
4576 \@onlypreamble{bxjs@pandoc@polyglossia@ja
4577 \def{bxjs@pandoc@polyglossia@ja}{%
4578   \renewcommand*{setmainlanguage}[2] [] {}%
4579   \newcommand*{setotherlanguage}[2] [] {}%
4580   \ifblank{##2}{}{%else
4581     \cslet{##2}\@empty \cslet{end##2}\@empty
4582     \cslet{text##2}\@firstofone}}%
4583   \newcommand*{setotherlanguages}[2] [] {}%
4584   \@for{bxjs@tmpa:}={##2}\do{%
4585     \setotherlangauge{bxjs@tmpa}}}%

```

Polyglossia の読込済マークは外れるようにしておく。

```
4586   \let{bxjs@polyglossia@options}\relax}%
4587 \fi
```

■Babel について Xe<sub>Λ</sub>TeX 以外の場合。

※ Pandoc 2.6 版において、Lua<sub>Λ</sub>TeX 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。

```
4588 \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。

```
4589 \pandocSkipLoadPackage{babel}
4590 \bxjs@set@dupload@proc{babel.sty}{%
```

```

4591 \bxjs@unset@dupload@proc{babel.sty}%
4592 \ClassWarning\bxjs@clsname
4593 {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。

```

4594 \@tempswafalse \let\bxjs@babel@options\@empty
4595 \def\bxjs@tmpb{main=}%
4596 \@for\bxjs@tmpa:=#1\do{%
4597   \ifx\bxjs@tmpa\bxjs@tmpb \@tempswatru
4598   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4599   \fi}%
4600 \if@tempswa
4601   \ClassWarning\bxjs@clsname
4602   {Main language is invalid, so fallback\MessageBreak
4603     definitions will be employed}%
4604   \bxjs@pandoc@polyglossia@ja

```

ない場合は、本来の babel の処理を実行する。

```

4605 \else
4606   \ClassWarning\bxjs@clsname
4607   {Main language is valid,\MessageBreak
4608     thus babel will be loaded}%
4609   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4610   \RequirePackage[#1]{babel}[]%
4611 \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。

```

4612 \g@addto@macro\bxjs@endpreamble@hook{%
4613   \ifx\bxjs@babel@options\relax
4614   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4615   \fi}

```

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

```

4616 \@onlypreamble\bxjs@pandoc@babel@ja
4617 \def\bxjs@pandoc@babel@ja{%
4618   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4619   \edef\bxjs@next{%
4620     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4621   }\bxjs@next
4622   \if j\jsEngine
4623     \RequirePackage[main=japanese]{pxbabel}[]%
4624   \else
4625     \RequirePackage{bxorigcapt}[]%
4626   \fi}

```

lang 対策はこれで終わり。

```

4627 \fi

```

## E.5 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。

```
4628 \bxjs@set@dupload@proc{geometry.sty}{%
4629   \setpagelayout*{#1}}
```

## E.6 CJKmainfont 変数

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

```
4630 \if l\jsEngine
4631   \pandocSkipLoadPackage{xeCJK}
4632   \providecommand*{\setCJKmainfont}{\setmainfont}
4633 \fi
```

## E.7 Option clash 対策

xeCJK パッケージについて。

※xeCJK はクラス内で既に読み込まれているので、space は（意図通りに）無効になる。

※ v2.8～v2.9.2 の間。

```
4634 \if x\jsEngine
4635   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4636     ,space}
4637 \fi
```

## E.8 レイアウト上書き禁止

レイアウト上書き禁止の実装は etoolbox の機能を使う。

```
4638 \ifjsWithTeX
4639 \onlypreamble\bxjs@info@or@ban
4640 \def\bxjs@info@or@ban#1{%
4641   \PackageInfo\bxjs@clsname
4642   {Freeze layout on '#1',\MessageBreak detected}}
```

■indent について indent 変数を指定しない場合に「段落表現形式をインデント方式に変更する」動作を抑止する。

```
4643 \unless\ifbxjs@jp@or@indent
4644   \bxjs@info@or@ban{indent}
```

parskip がある場合はそれを読み込もうとするため、parskip の読込をブロックする。

```
4645 \IfFileExists{parskip.sty}{%
4646   \pandocSkipLoadPackage{parskip}%
```



`\parskip` がない場合はパラメタを変更しようとするため、該当のパラメタを復帰させる。

```
4647 }{%else
4648   \eappto\bxjs@endpreamble@hook{%
4649     \parindent=\the\parindent\relax
4650     \parskip=\the\parskip\relax}}
4651 \fi
4652 %
4653 % \paragraph{secnumdepth について}
4654 % |secnumdepth| の値を決めるのは numbersections 変数
4655 % (|-N|/|--number-sections| オプションに連動する)
4656 % や secnumdepth 変数であるが、何れにしても |secnumdepth|
4657 % の値は書き換えられる。
4658 % そのため、|secnumdepth| を復帰させる。
4659 \ifbxjs@jp@or@secnumdepth\else
4660   \bxjs@info@or@ban{secnumdepth}
4661   \eappto\bxjs@endpreamble@hook{%
4662     \c@secnumdepth=\the\c@secnumdepth\relax}
4663 \fi
4664 %
4665 % \paragraph{block-heading について}
4666 \ifbxjs@jp@or@block@heading\else
4667   \let\bxjs@frozen@paragraph\paragraph
4668   \let\bxjs@frozen@subparagraph\subparagraph
4669   \bxjs@info@or@ban{block-heading}
4670   \appto\bxjs@endpreamble@hook{%
4671     \let\oldparagraph\undefined
4672     \let\paragraph\bxjs@frozen@paragraph
4673     \let\subparagraph\bxjs@frozen@subparagraph}
4674 \fi
4675 \fi
```

## E.9 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。

```
4676 \g@addto@macro\bxjs@begin@document@hook{%
4677   \@tempwafalse
```

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。

```

4678 \ifx\oldparagraph\@undefined\else
4679 \@tempwattrue
4680 \fi

```

\paragraph が番号付きの場合は、マーク変更が必要である。

```

4681 \ifnum\c@secnumdepth>3
4682 \@tempwattrue
4683 \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。

```

4684 \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4685 \let\jsParagraphMark\@empty
4686 \fi\fi}

```

## E.10 全角空白文字

```

4687 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4688 \catcode"3000=\active
4689 \begingroup \catcode`\!=7
4690 \protected\gdef!!!!3000{\zwspace}
4691 \endgroup
4692 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4693 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4694 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4695 \fi\fi

```

## E.11 hyperref 対策

hyperref の unicode オプションの固定を行う。

```

4696 \if j\jsEngine
4697 \bxjs@fix@hyperref@unicode{false}
4698 \else
4699 \bxjs@fix@hyperref@unicode{true}
4700 \fi

```

## E.12 Pandoc 要素に対する和文用の補正

■重要要素 重要 (Strong) 要素に対する L<sup>A</sup>T<sub>E</sub>X 出力は \textbf となるが、代わりに \strong を使いたいため、\textbf を書き換えてしまう (うわぁ)。

```

4701 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
4702 \let\orgtextbf\textbf
4703 \DeclareRobustCommand\bxjstextbf[1]{%
4704 \begingroup
4705 \let\textbf\orgtextbf
4706 \strong{#1}%
4707 \endgroup}%
4708 \g@addto@macro\bxjs@begin@document@hook{%

```

```

4709 \let\textbf\bxjstextbf}
4710 \fi\fi

\strong の書体を設定する。
4711 \jsAtEndOfClass{%
4712 \ifx\strongfontdeclare\@undefined\else
4713 \ifcase\bxjs@jp@strong
4714 \or \strongfontdeclare{\sffamily}%
4715 \or \strongfontdeclare{\sffamily\bfseries}%
4716 \fi
4717 \fi}

```

■インラインコード要素 インラインコード (Code) 要素に対する L<sup>A</sup>T<sub>E</sub>X 出力は `\texttt` となる。`\texttt` の両端に欧文ゴーストが入るようにする。

```

4718 \ifbxjs@jp@fix@code
4719 \chardef\bxjs@eghost@c=23
4720 \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4721 \else\ifx l\jsEngine \ltjsetparameter{alxspmode={\bxjs@eghost@c,3}}
4722 \else\ifx x\jsEngine %no-op
4723 \else \let\bxjs@eghost@c\@undefined
4724 \fi\fi\fi
4725 \ifx\bxjs@eghost@c\@undefined\else
4726 \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4727 \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4728 \let\orgtexttt\texttt
4729 \DeclareRobustCommand\bxjstexttt[1]{%
4730 \ifmmode \nfss@text{\ttfamily #1}%
4731 \else
4732 \ifvmode \leavevmode \fi
4733 \bxjs@pan@eghost\null
4734 \begingroup \ttfamily #1\endgroup
4735 \null\bxjs@pan@eghost
4736 \fi}
4737 \g@addto@macro\bxjs@begin@document@hook{%
4738 \let\texttt\bxjstexttt}

```

さらに `\verb` の外側にも欧文ゴーストが入るようにする。

※`bxjaghost` の実装を参考にした。

```

4739 \expandafter\def\expandafter\verb\expandafter{%
4740 \expandafter\bxjs@pan@eghost\verb}
4741 \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4742 \fi
4743 \fi

```

## E.13 ifPDFTeX スイッチ

Pandoc モードでは Pandoc の既定テンプレートを（無理やり）(u)p<sub>T</sub>E<sub>X</sub> に対応させることを目的にしている。

旧版のテンプレートでは`ifxetex` と `ifluatex` パッケージを読み込んだ上で「`XYTeX` でも `LuaTeX` でもないものは `pdfTeX`」という前提の動作をしていた。よって、`(u)pTeX` に対応させる際には「`pdfTeX` 用の処理が実行される」ことを前提にすればよかった。

ところが、Pandoc の 2.12 版では `iftex` パッケージが導入されて「`pdfTeX` の判定を直接 `\ifPDFTeX` で行う」ように改修された。このため、`(u)pTeX` での実行でどのコードが実行されるかを予測することが困難になってしまった。

これに対処するため、「文書ファイルのプレアンブル実行中に限って `\ifPDFTeX` が（実際とは異なり）真になるようにする」という細工を施すことで、従来通り「`pdfTeX` 用の処理が実行される」前提が維持されるようにする。

```
4744 \if j\jsEngine
```

`\bxjs@check@frontier` `\bxjs@check@frontier\CS` は現在のパッケージ読込ネストレベルが丁度 1 であるときのみ `\CS` を実行する。

```
4745 \def\bxjs@check@frontier{%
4746   \expandafter\bxjs@check@frontier@a\@currnamestack\noindent...\@nil}
4747 \def\bxjs@check@frontier@a#1#2#3#4#5\@nil#6{%
4748   \ifx\noindent#4#6\fi}
```

`\bxjs@unforge@ifPDFTeX` `\ifPDFTeX` を偽（正しい値）にする。

```
4749 \@onlypreamble\bxjs@unforge@ifPDFTeX
4750 \def\bxjs@unforge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iffalse}}
```

`\bxjs@forge@ifPDFTeX` `\ifPDFTeX` を真（偽装した値）にする。

```
4751 \@onlypreamble\bxjs@forge@ifPDFTeX
4752 \def\bxjs@forge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iftrue}}
```

`\bxjs@unload@forge@ifPDFTeX` `\ifPDFTeX` に対する細工を無効化する。

```
4753 \def\bxjs@unload@forge@ifPDFTeX{%
4754   \bxjs@unforge@ifPDFTeX
4755   \global\let\bxjs@check@frontier\@gobble}
```

プレアンブル開始時は `\ifPDFTeX` は真で、終了時に偽装を無効化する。`filehook` のフックで「パッケージ読込中は偽装を解除する」ことを実現している。

```
4756 \jsAtEndOfClass{\bxjs@forge@ifPDFTeX}
4757 \ifjsWithTeX
4758   \AtBeginOfEveryFile{\bxjs@check@frontier\bxjs@unforge@ifPDFTeX}%
4759   \AtEndOfEveryFile{\bxjs@check@frontier\bxjs@forge@ifPDFTeX}%
4760   \g@addto@macro\bxjs@endpreamble@hook{\bxjs@unload@forge@ifPDFTeX}
4761 \else
4762   \g@addto@macro\bxjs@begin@document@hook{\bxjs@unload@forge@ifPDFTeX}
4763 \fi
4764 \fi
```

## E.14 完了

おしまい。

```
4765 %</pandoc>
```

和文ドライバ実装はここまで。

```
4766 %</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。

```
4767 %<*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

```
4768 %<*compat>
```

```
4769 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
4770 \let\bxac@engine=n
```

```
4771 \def\bxac@do#1#2{%
```

```
4772   \edef\bxac@tmpa{\string#1}%
```

```
4773   \edef\bxac@tmpb{\meaning#1}%
```

```
4774   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
4775 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
4776 \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。

```
4777 \ifx\jsAtEndOfClass\@undefined
```

```
4778   \let\bxac@delayed@if@bxjs\@firstofone
```

```
4779 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
4780 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4781 \newif@ifbxac@in@old@behavior
```

```
4782 \let\ImposeOldLuaTeXBehavior\relax
```

```
4783 \let\RevokeOldLuaTeXBehavior\relax
```

## G.2 XeTeX 部分

```
4784 \ifx x\bxac@engine
      XeTeX 文字クラスのムニャムニャ。
4785 \@onlypreamble\bxac@adjust@charclass
4786 \bxac@delayed@if@bxjs{%
4787   \@ifpackageloaded{xeCJK}{-}{-}%else
4788   \ifx\xe@alloc@intercharclass\@undefined\else
4789     \ifnum\xe@alloc@intercharclass=\z@
4790       \PackageInfo\bxac@pkgname
4791       {Setting up interchar class for CJK...\@gobble}%
4792       \InputIfFileExists{load-unicode-xetex-classes.tex}{-}%
4793       \xe@alloc@intercharclass=3
4794     }{-}%else
4795     \PackageWarning\bxac@pkgname
4796     {Cannot find file 'load-unicode-xetex-classes.tex'%
4797     \@gobble}%
4798   }%
4799   \fi\fi
4800   \ifnum\XeTeXcharclass"3041=\z@
4801     \PackageInfo\bxac@pkgname
4802     {Adjusting interchar class for CJK...\@gobble}%
4803     \@for\bxac@x:={%
4804       3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4805       3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4806       30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4807       31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4808       31FF%
4809     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4810   \fi
4811 }%
4812 }
      以上。
4813 \fi
```

## G.3 LuaTeX 部分

```
4814 \ifx l\bxac@engine
      ムニャムニャ。
4815 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4816   \chardef\pdftexversion=200
4817   \def\pdftexrevision{0}
4818   \let\pdftexbanner\luatexbanner
4819 \fi\fi
\ImposeOldLuaTeXBehavior ムニャムニャ。
\RevokeOldLuaTeXBehavior 4820 \begingroup\expandafter\expandafter\expandafter\endgroup
4821 \expandafter\ifx\csname outputmode\endcsname\relax\else
```

```

4822 \def\bxac@ob@list{%
4823   \do{\let}\pdfoutput{\outputmode}%
4824   \do{\let}\pdfpagewidth{\pagewidth}%
4825   \do{\let}\pdfpageheight{\pageheight}%
4826   \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4827   \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}%
4828 \def\bxac@ob@do#1#2{\begingroup
4829   \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4830 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4831   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4832   \else \let#3#1\relax \let#1@undefined
4833   \fi}
4834 \protected\def\ImposeOldLuaTeXBehavior{%
4835   \unless\ifbxac@in@old@behavior
4836     \bxac@in@old@behaviortrue
4837     \let\do\bxac@ob@do \bxac@ob@list
4838   \fi}
4839 \protected\def\RevokeOldLuaTeXBehavior{%
4840   \ifbxac@in@old@behavior
4841     \bxac@in@old@behaviorfalse
4842     \let\do\bxac@ob@do \bxac@ob@list
4843   \fi}
4844 \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。

```

4845 \ifnum\luatexversion>64 \directlua{
4846   local function range(cs, ce, cc, ff)
4847     if ff or not tex.getcatcode(cs) == cc then
4848       local setcc = tex.setcatcode
4849       for c = cs, ce do setcc(c, cc) end
4850     end
4851   end
4852   range(0x3400, 0x4DB5, 11, false)
4853   range(0x4DB5, 0x4DBF, 11, true)
4854   range(0x4E00, 0x9FCC, 11, false)
4855   range(0x9FCD, 0x9FFF, 11, true)
4856   range(0xAC00, 0xD7A3, 11, false)
4857   range(0x20000, 0x2A6D6, 11, false)
4858   range(0x2A6D7, 0x2A6FF, 11, true)
4859   range(0x2A700, 0x2B734, 11, false)
4860   range(0x2B735, 0x2B73F, 11, true)
4861   range(0x2B740, 0x2B81D, 11, false)
4862   range(0x2B81E, 0x2B81F, 11, true)
4863   range(0x2B820, 0x2CEA1, 11, false)
4864   range(0x2CEA2, 0x2FFFD, 11, true)
4865 } \fi

```

以上。

```

4866 \fi

```

## G.4 完了

おしまい。

```
4867 %</compat>
```

## 付録 H 補助パッケージ : bxjscjkat

modern ドライバ用の和文カテゴリを適用する。

### H.1 準備

```
4868 %<*cjkat>
4869 \def\bxjx@pkgname{bxjscjkat}
4870 \newcount\bxjx@canta
4871 \@onlypreamble\bxjx@tmpdo
4872 \@onlypreamble\bxjx@tmpdo@a
4873 \@onlypreamble\bxjx@tmpdo@b
```

\bxjx@engine エンジンの種別。

```
4874 \let\bxjx@engine=n
4875 \def\bxjx@tmpdo#1#2{%
4876   \edef\bxjx@tmpa{\string#1}%
4877   \edef\bxjx@tmpb{\meaning#1}%
4878   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4879 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4880 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4881 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4882 \bxjx@tmpdo\pdfTeXversion{\let\bxjx@engine=p}
4883 \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。

```
4884 \def\bxjx@tmpdo#1#2{%
4885   \if#1\bxjx@engine
4886     \@ifpackageloaded{#2}{\}%else
4887     \PackageError\bxjx@pkgname
4888       {Package '#2' must be loaded}%
4889       {Package loading is aborted.\MessageBreak\@ehc}%
4890     \endinput}
4891   \fi}
4892 \bxjx@tmpdo{p}{bxckjatype}
4893 \bxjx@tmpdo{x}{xeCJK}
4894 \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。

```
4895 \ifx\TextOrMath\@undefined
4896   \RequirePackage{fixltx2e}
```



4897 \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-j<sub>a</sub> と（ほぼ）等価なものに変更する。

※ LuaT<sub>E</sub>X-j<sub>a</sub> との相違点：A830、A960、1B000。

```
4898 \if u\bxjx@engine
4899 \@for\bxjx@tmpa:={%
4900 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4901 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4902 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4903 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4904 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4905 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4906 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4907 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4908 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4909 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4910 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4911 FE70,FFF0,%
4912 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4913 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4914 10480,104B0,10500,10530,10600,10800,10840,10860,%
4915 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4916 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4917 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4918 11200,11280,112B0,11300,11400,11480,11580,11600,%
4919 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4920 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4921 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4922 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4923 1EE00,1F000,1F030,1FOA0,1F300,1F600,1F650,1F680,%
4924 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4925 00C0%
4926 }\do{%
4927 \@tempcnta="\bxjx@tmpa\relax
4928 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4929 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4930 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4931 \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-j<sub>a</sub> 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。

```

4932 \onlypreamble\bxjx@grkcyr@list
4933 \def\bxjx@grkcyr@list{%
4934 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4935 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4936 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4937 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4938 \do{0395}{LGR}{\textEpsilon}{E}%       % GR. C. L. EPSILON
4939 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4940 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
4941 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4942 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4943 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4944 \do{039B}{LGR}{\textLambda}{\Lambda}%    % GR. C. L. LAMDA
4945 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4946 \do{039D}{LGR}{\textNu}{N}%             % GR. C. L. NU
4947 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
4948 \do{039F}{LGR}{\textOmicron}{O}%      % GR. C. L. OMICRON
4949 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
4950 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
4951 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4952 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
4953 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4954 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
4955 \do{03A7}{LGR}{\textChi}{X}%           % GR. C. L. CHI
4956 \do{03A8}{LGR}{\textPsi}{\Psi}%         % GR. C. L. PSI
4957 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA
4958 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
4959 \do{03B2}{LGR}{\textbeta}{\beta}%       % GR. S. L. BETA
4960 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA
4961 \do{03B4}{LGR}{\textdelta}{\delta}%     % GR. S. L. DELTA
4962 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
4963 \do{03B6}{LGR}{\textzeta}{\zeta}%       % GR. S. L. ZETA
4964 \do{03B7}{LGR}{\texteta}{\eta}%          % GR. S. L. ETA
4965 \do{03B8}{LGR}{\texttheta}{\theta}%     % GR. S. L. THETA
4966 \do{03B9}{LGR}{\textiota}{\iota}%       % GR. S. L. IOTA
4967 \do{03BA}{LGR}{\textkappa}{\kappa}%     % GR. S. L. KAPPA
4968 \do{03BB}{LGR}{\textlambda}{\lambda}%   % GR. S. L. LAMDA
4969 \do{03BC}{LGR}{\textmu}{\mu}%           % GR. S. L. MU
4970 \do{03BD}{LGR}{\textnu}{\nu}%          % GR. S. L. NU
4971 \do{03BE}{LGR}{\textxi}{\xi}%          % GR. S. L. XI
4972 \do{03BF}{LGR}{\textomicron}{o}%          % GR. S. L. OMICRON

```

4973	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4974	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4975	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4976	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4977	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4978	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4979	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4980	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4981	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4982	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4983	\do{0401}{T2A}{\CYRYO}{\O}	% CY. C. L. IO
4984	\do{0410}{T2A}{\CYRA}{\A}	% CY. C. L. A
4985	\do{0411}{T2A}{\CYRB}{\B}	% CY. C. L. BE
4986	\do{0412}{T2A}{\CYRV}{\V}	% CY. C. L. VE
4987	\do{0413}{T2A}{\CYRG}{\G}	% CY. C. L. GHE
4988	\do{0414}{T2A}{\CYRD}{\D}	% CY. C. L. DE
4989	\do{0415}{T2A}{\CYRE}{\E}	% CY. C. L. IE
4990	\do{0416}{T2A}{\CYRZH}{\ZH}	% CY. C. L. ZHE
4991	\do{0417}{T2A}{\CYRZ}{\Z}	% CY. C. L. ZE
4992	\do{0418}{T2A}{\CYRI}{\I}	% CY. C. L. I
4993	\do{0419}{T2A}{\CYRISHRT}{\I}	% CY. C. L. SHORT I
4994	\do{041A}{T2A}{\CYRK}{\K}	% CY. C. L. KA
4995	\do{041B}{T2A}{\CYRL}{\L}	% CY. C. L. EL
4996	\do{041C}{T2A}{\CYRM}{\M}	% CY. C. L. EM
4997	\do{041D}{T2A}{\CYRN}{\N}	% CY. C. L. EN
4998	\do{041E}{T2A}{\CYRO}{\O}	% CY. C. L. O
4999	\do{041F}{T2A}{\CYRP}{\P}	% CY. C. L. PE
5000	\do{0420}{T2A}{\CYRR}{\R}	% CY. C. L. ER
5001	\do{0421}{T2A}{\CYRS}{\S}	% CY. C. L. ES
5002	\do{0422}{T2A}{\CYRT}{\T}	% CY. C. L. TE
5003	\do{0423}{T2A}{\CYRU}{\U}	% CY. C. L. U
5004	\do{0424}{T2A}{\CYRF}{\F}	% CY. C. L. EF
5005	\do{0425}{T2A}{\CYRH}{\H}	% CY. C. L. HA
5006	\do{0426}{T2A}{\CYRC}{\C}	% CY. C. L. TSE
5007	\do{0427}{T2A}{\CYRCH}{\CH}	% CY. C. L. CHE
5008	\do{0428}{T2A}{\CYRSH}{\SH}	% CY. C. L. SHA
5009	\do{0429}{T2A}{\CYRSHCH}{\SHCH}	% CY. C. L. SHCHA
5010	\do{042A}{T2A}{\CYRHRDSN}{\HRDSN}	% CY. C. L. HARD SIGN
5011	\do{042B}{T2A}{\CYRERY}{\ERY}	% CY. C. L. YERU
5012	\do{042C}{T2A}{\CYRSFTSN}{\SFTSN}	% CY. C. L. SOFT SIGN
5013	\do{042D}{T2A}{\CYREREV}{\EREV}	% CY. C. L. E
5014	\do{042E}{T2A}{\CYRYU}{\YU}	% CY. C. L. YU
5015	\do{042F}{T2A}{\CYRYA}{\YA}	% CY. C. L. YA
5016	\do{0430}{T2A}{\cyra}{\a}	% CY. S. L. A
5017	\do{0431}{T2A}{\cyrb}{\b}	% CY. S. L. BE
5018	\do{0432}{T2A}{\cyrv}{\v}	% CY. S. L. VE
5019	\do{0433}{T2A}{\cyrg}{\g}	% CY. S. L. GHE
5020	\do{0434}{T2A}{\cyrd}{\d}	% CY. S. L. DE
5021	\do{0435}{T2A}{\cyre}{\e}	% CY. S. L. IE

5022 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE  
 5023 \do{0437}{T2A}{\cyrz}{}% % CY. S. L. ZE  
 5024 \do{0438}{T2A}{\cyri}{}% % CY. S. L. I  
 5025 \do{0439}{T2A}{\cyrishrt}{}% % CY. S. L. SHORT I  
 5026 \do{043A}{T2A}{\cyrk}{}% % CY. S. L. KA  
 5027 \do{043B}{T2A}{\cyr1}{}% % CY. S. L. EL  
 5028 \do{043C}{T2A}{\cyrn}{}% % CY. S. L. EM  
 5029 \do{043D}{T2A}{\cyrn}{}% % CY. S. L. EN  
 5030 \do{043E}{T2A}{\cyro}{}% % CY. S. L. O  
 5031 \do{043F}{T2A}{\cyrp}{}% % CY. S. L. PE  
 5032 \do{0440}{T2A}{\cyr{r}}{}% % CY. S. L. ER  
 5033 \do{0441}{T2A}{\cyr{s}}{}% % CY. S. L. ES  
 5034 \do{0442}{T2A}{\cyr{t}}{}% % CY. S. L. TE  
 5035 \do{0443}{T2A}{\cyr{u}}{}% % CY. S. L. U  
 5036 \do{0444}{T2A}{\cyr{f}}{}% % CY. S. L. EF  
 5037 \do{0445}{T2A}{\cyr{h}}{}% % CY. S. L. HA  
 5038 \do{0446}{T2A}{\cyr{c}}{}% % CY. S. L. TSE  
 5039 \do{0447}{T2A}{\cyr{ch}}{}% % CY. S. L. CHE  
 5040 \do{0448}{T2A}{\cyr{sh}}{}% % CY. S. L. SHA  
 5041 \do{0449}{T2A}{\cyr{shch}}{}% % CY. S. L. SHCHA  
 5042 \do{044A}{T2A}{\cyr{hrdsn}}{}% % CY. S. L. HARD SIGN  
 5043 \do{044B}{T2A}{\cyr{rery}}{}% % CY. S. L. YERU  
 5044 \do{044C}{T2A}{\cyr{sftsn}}{}% % CY. S. L. SOFT SIGN  
 5045 \do{044D}{T2A}{\cyr{rerev}}{}% % CY. S. L. E  
 5046 \do{044E}{T2A}{\cyr{yu}}{}% % CY. S. L. YU  
 5047 \do{044F}{T2A}{\cyr{ya}}{}% % CY. S. L. YA  
 5048 \do{0451}{T2A}{\cyr{yo}}{}% % CY. S. L. IO  
 5049 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL  
 5050 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS  
 5051 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN  
 5052 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN  
 5053 \do{00B4}{TS1}{\textasciicute}{}% % ACUTE ACCENT  
 5054 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN  
 5055 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN  
 5056 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN  
 5057 }

\mathdegree 面倒なので補っておく。

5058 \providecommand\*\mathdegree{{}\textcircled{}}\circ}

\ifbxjx@gcc@cjk [スイッチ]「特定 CJK 曖昧文字」を和文扱いにするか。

5059 \newif\ifbxjx@gcc@cjk

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

5060 \newcommand\*\greekasCJK{%

5061 \bxjx@gcc@cjktrue}

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

5062 \newcommand\*\nogreekasCJK{%

```
5063 \bxjx@gcc@cjklfalse}
```

`\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。

```
5064 \def\bxjx@tmpdo#1\relax{%
5065   \def\bxjx@fake@grk##1##2{%
5066     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
5067   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
5068     \ifx\##1\%
5069       \bxjx@canta##4\divide\bxjx@canta\@cclvi
5070       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
5071       \mathchar\bxjx@canta
5072     \else ##3\fi}
5073 }\expandafter\bxjx@tmpdo\string\mathchar\relax
```

#### ■pdfTeX・upTeX の場合

```
5074 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0
```

- `\[bxjx@KC/〈符号値〉]` : その文字が「特定曖昧 CJK 文字」に該当する場合に定義済になる。

まず `inputenc` を読み込んで入力エンコーディングを `utf8` に変更する。

※「既定 UTF-8 化」後の  $\text{\LaTeX}$  においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```
5075 \@ifpackageloaded{inputenc}{\fi}{%else
5076   \RequirePackage[utf8]{inputenc}}
5077 \def\bxjx@tmpa{utf8}
5078 \ifx\bxjx@tmpa\inputencdoingname
5079   \PackageWarningNoLine\bxjx@pkgname
5080     {Input encoding changed to utf8}%
5081   \inputencoding{utf8}%
5082 \fi
```

upTeX の場合に、「特定曖昧 CJK 文字」を含むブロックの和文カテゴリコードを変更する。

```
5083 \if u\bxjx@engine
5084 \kcatcode"0370=15
5085 \kcatcode"0400=15
5086 \kcatcode"0500=15
5087 \fi
```

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

```

5088 \def\bxjx@tmpdo#1{%
5089   \@tempcnta="#1\relax
5090   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
5091 \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。

```

5092   \ifx\\#5\\%
5093     \def\bxjx@tmpa{\@inmathwarn#4}%

```

$\langle\text{数式 LICR}\rangle$  が英字である場合は  $\backslash\text{bxjx@fake@grk}$  で出力する。大文字なら  $\backslash\text{Pi}$ 、小文字なら  $\backslash\text{pi}$  を基準文字にする。

```

5094   \else\ifcat A\noexpand#5%
5095     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5096       {\ifnum\uccode`#5=`#5\noexpand\text{Pi}\else\noexpand\text{pi}\fi}}%

```

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

```

5097   \else \def\bxjx@tmpa{#5}%
5098   \fi\fi
5099   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
5100   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}

```

以降はエンジン種別で分岐する。up $\text{T}_{\text{E}}\text{X}$  の場合。

```

5101 \if u\bxjx@engine
5102 \def\bxjx@tmpdo@b#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。  
 $\text{LICR}$  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  $\text{LICR}$  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  $\text{kchardef}$  を使用し、その制御綴として  $\backslash\text{bxjx@KC}/\dots$  を流用している。

```

5103   \kchardef#1=\@tempcnta
5104   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
5105   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdf $\text{T}_{\text{E}}\text{X}$  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  $\text{bxcjkjatype}$  の命令）。 $\backslash\text{bxjx@KC}/\dots$  は使わないが定義済にする必要がある。

```

5106 \else\if p\bxjx@engine
5107 \def\bxjx@tmpdo@b#1#2#3#4#5{%
5108   \mathchardef#1=\@tempcnta
5109   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
5110   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
5111 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

5112 \let\do\bxjx@tmpdo \bxjx@grkcyr@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。

```

5113 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
5114 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
5115 \@onlypreamble\bxjx@DeclareUnicodeCharacter
5116 \def\bxjx@DeclareUnicodeCharacter#1#2{%
5117   \count@="#1\relax
5118   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
5119     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
5120   \else
5121     \wlog{ \space\space skipped defining Unicode char U+#1}%
5122   \fi}

```

`\bxjx@ja@or@not` `\bxjx@ja@or@not{〈和文用定義〉}{〈対象 fontenc〉}{〈LICR〉}` : `\[no]greekasCJK` の状態に応じて和文または欧文で文字を出力する。

```

5123 \def\bxjx@ja@or@not#1#2#3{%

```

`\greekasCJK` の場合は、無条件に `〈和文用定義〉` を実行する。

```

5124   \ifbxjx@gcc@cjkk #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。

```

5125   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
5126   \else \UseTextSymbol{#2}{#3}%
5127   \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。

```

5128 \begingroup
5129 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
5130 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
5131   \noexpand\bxjx@swap@DUC@cmd
5132   \the\toks@
5133   \noexpand\bxjx@swap@DUC@cmd}}
5134 \endgroup\next
5135 \def\bxjx@swap@DUC@cmd{%
5136   \let\bxjx@tmpa\DeclareUnicodeCharacter
5137   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
5138   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
5139   \let\bxjx@tmpa\relax}

```

以上。

#### ■ $\mathrm{X\TeX}$ ・ $\mathrm{Lua\TeX}$ の場合

```

5140 \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。

```

5141 \def\bxjx@tmpdo#1{%
5142   \bxjx@cmta="#1\relax
5143   \begingroup
5144     \lccode`~=\bxjx@cmta

```

```

5145 \lowercase{\endgroup
5146 \bxjx@tmpdo@a{~}}{#1}}
5147 \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、up $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  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。

```

5148 \ifx\\#5\\\let\bxjx@tmpa\relax
5149 \else\ifcat A\noexpand#5%
5150 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5151 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
5152 \else \def\bxjx@tmpa{#5}%
5153 \fi\fi
5154 \ifx\bxjx@tmpa\relax\else
5155 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
5156 \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。

```

5157 \mathchardef\bxjx@tmpa="119
5158 \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  の場合は、Lua $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -ja 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。

```

5159 \if l\bxjx@engine
5160 \protected\def\greekasCJK{%
5161 \bxjx@gcc@cjktrue
5162 \ltjsetparameter{jacharrange={+2, +8}}}
5163 \protected\def\nogreekasCJK{%
5164 \bxjx@gcc@cjkfalse
5165 \ltjsetparameter{jacharrange={-2, -8}}}
5166 \fi

```

X $\mathrm{E}_{\mathrm{L}}\mathrm{A}_{\mathrm{T}}\mathrm{E}_{\mathrm{X}}$  の場合、xeCJK は X $\mathrm{E}_{\mathrm{L}}\mathrm{A}_{\mathrm{T}}\mathrm{E}_{\mathrm{X}}$  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。

```

5167 \if x\bxjx@engine
5168 \let\bxjx@gcc@cjk@list\@empty
5169 \def\do#1#2#3#4{%
5170 \edef\bxjx@gcc@cjk@list{\bxjx@gcc@cjk@list
5171 \noexpand\XeTeXcharclass"#1\bxjx@cmta}}
5172 \bxjx@grkcyr@list
5173 \protected\def\greekasCJK{%
5174 \bxjx@gcc@cjktrue
5175 \bxjx@cmta=\@ne \bxjx@gcc@cjk@list}
5176 \protected\def\nogreekasCJK{%
5177 \bxjx@gcc@cjkfalse
5178 \bxjx@cmta=\z@ \bxjx@gcc@cjk@list}
5179 \fi

```



以上。

```
5180 \fi\fi
```

## H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5181 \nogreekasCJK
```

## H.5 完了

おしまい。

```
5182 %</cjkat>
```

## 付録 I 補助パッケージ : bxjspanc

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

```
5183 %<*ancpandoc>
```

```
5184 %% このファイルは日本語文字を含みます.
```

```
5185 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```
5186 \let\bxjsp@engine=n
```

```
5187 \@onlypreamble\bxjsp@do
```

```
5188 \def\bxjsp@do#1#2{%
```

```
5189   \edef\bxjsp@tmpa{\string#1}%
```

```
5190   \edef\bxjsp@tmpb{\meaning#1}%
```

```
5191   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
```

```
5192 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
```

```
5193 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
```

```
5194 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
```

```
5195 \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。

```
5196 \newif\ifbxjsp@english
5197 \DeclareOption{english}{\bxjsp@englishttrue}

    オプション定義はおしまい。

5198 \ProcessOptions*
```

### 1.3 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。

```
5199 \@onlypreamble\pandocSkipLoadFile
5200 \newcommand*\pandocSkipLoadFile[1]{%
5201   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5202 \def\bxjsp@skip@load@file@a#1#2{%
5203   \ifx#1\relax
5204     \def#1{2001/01/01}%
5205     \PackageInfo{bxjsp@pkgname
5206       {File '#2' marked as loaded\@gobble}}%
5207   \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。

```
5208 \@onlypreamble\pandocSkipLoadPackage
5209 \newcommand*\pandocSkipLoadPackage[1]{%
5210   \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。

```
5211 \ifx\@IncludeInRelease\@undefined\else
5212   \pandocSkipLoadPackage{fixltx2e}
5213 \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。)

```
5214 \if j\bxjsp@engine
5215   \pandocSkipLoadPackage{cmap}
5216 \fi
```

## I.6 microtype パッケージ

警告が多すぎなので消す。

```
5217 \if j\bxjsp@engine \else
5218   \PassOptionsToPackage{verbose=silent}{microtype}
5219 \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。

```
5220 \if j\bxjsp@engine
5221   \pandocSkipLoadPackage{microtype}
5222   \newcommand*\UseMicrotypeSet[2][]{\fi}
5223 \fi
```

## I.7 Unicode 文字変換対策

Pandoc で  $\text{\LaTeX}$  形式に書き出す場合は、元データ中の一部の Unicode 文字を「 $\text{\LaTeX}$  の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

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

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、「 $\backslash \text{ldots}$  を『 $\cdots$ 』に戻す」という処置を行う。

`\pandocLdots`   Pandoc 用の  $\backslash \text{ldots}$  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  $\backslash \text{ldots} \{ \}$  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  $\backslash \text{ldots}$  を全て  $\cdots$  に戻す動作に変更した。

```
5224 \DeclareRobustCommand{\pandocLdots}{\%
5225   \relax\ifmmode \expandafter\bxjsp@org@ldots
5226   \else \expandafter\bxjsp@ja@ellipsis
5227   \fi}
5228 \def\bxjsp@ja@ellipsis{\cdots}
5229 \let\bxjsp@org@ldots\ldots
```

$\backslash \text{ldots}$  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。

```
5230 \AtBeginDocument{\%
5231   \let\bxjsp@org@ldots\ldots
5232   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}
```

`\ldots` の直後の文字が非英字の場合、Pandoc は「`\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  $\text{Xe}\text{L}\text{a}\text{T}\text{E}\text{X}$ ・ $\text{L}\text{u}\text{a}\text{T}\text{E}\text{X}$  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。

```
5233 \chardef\bxjsp@cc@other=12
5234 \@onlypreamble\bxjsp@makeother@range
5235 \def\bxjsp@makeother@range#1#2{%
5236   \@tempcnta"#1\relax \@tempcntb"#2\relax
5237   \loop\ifnum\@tempcnta<\@tempcntb
5238     \catcode\@tempcnta\bxjsp@cc@other
5239     \advance\@tempcnta\@ne
5240   \repeat}
5241 \ifnum0\if x\bxjsp@engine1\fi\if l\bxjsp@engine1\fi>0
5242   \catcode"1F23B=\bxjsp@cc@other
5243   \bxjsp@makeother@range{9FCD}{A000}
5244   \bxjsp@makeother@range{1B002}{1B170}
5245   \bxjsp@makeother@range{2B820}{2EBF0}
5246 \fi
```

## 1.8 PandoLa モジュール

インストール済であれば読み込む。

```
5247 \IfFileExists{bxpandola.sty}{%
5248   \RequirePackage{bxpandola}\relax
5249   \PackageInfo\bxjsp@pkgname
5250   {PandoLa module is loaded\@gobble}
5251 }{}
```

## 1.9 完了

おしまい。

```
5252 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
5253 %</anc>
```