

BXjscls パッケージ (BXJS 文書クラス集) ソースコード説明書

八登崇之 (Takayuki YATO; aka. “ZR”)





v1.9 [2018/04/19]

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

目次

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

10	相互参照	95
10.1	目次の類	95
10.2	参考文献	101
10.3	索引	102
10.4	脚注	104
11	段落の頭へのグルー挿入禁止	106
12	いろいろなロゴ	110
13	amsmath との衝突の回避	111
14	初期設定	111
付録 A	和文ドライバの仕様 罫	116
付録 B	和文ドライバ : minimal 罫	117
B.1	補助マクロ	117
B.2	(u)pTeX 用の設定	119
B.3	pdfTeX 用の処理	123
B.4	X _Y TeX 用の処理	124
B.5	後処理 (エンジン共通)	125
付録 C	和文ドライバ : standard 罫	127
C.1	共通処理 (1)	128
C.2	pTeX 用設定	136
C.3	pdfTeX 用設定 : CJK + bxcjkatype	139
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	141
C.5	LuaTeX 用設定 : LuaTeX-ja	143
C.6	共通処理 (2)	146
付録 D	和文ドライバ : modern 罫	147
D.1	フォント設定	147
D.2	fixltx2e 読込	148
D.3	和文カテゴリコード	148
D.4	完了	148
付録 E	和文ドライバ : pandoc 罫	148
E.1	dupload システム	148
E.2	lang 変数	149
E.3	geometry 変数	150
E.4	CJKmainfont 変数	150
E.5	paragraph のマーク	150

E.6	全角空白文字	151
E.7	完了	151
付録 F	補助パッケージ一覧 	151
付録 G	補助パッケージ : bxjscompat 	152
G.1	準備	152
G.2	X _Y TeX 部分	152
G.3	LuaTeX 部分	153
G.4	完了	155
付録 H	補助パッケージ : bxjscjkat 	155
H.1	準備	155
H.2	和文カテゴリコードの設定	156
H.3	ギリシャ・キリル文字の扱い	157
H.4	初期設定	164
H.5	完了	164
付録 I	補助パッケージ : bxjspandoc 	164
I.1	準備	164
I.2	パッケージ読込の阻止	165
I.3	fixltx2e パッケージ	165
I.4	cmap パッケージ	165
I.5	microtype パッケージ	166
I.6	Unicode 文字変換対策	166
I.7	PandoLa モジュール	167
I.8	完了	167

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><article></code>	<code>bxjsarticle.cls</code>	短いレポート (章なし)
<code><report></code>	<code>bxjsreport.cls</code>	長いレポート (章あり)
<code><book></code>	<code>bxjsbook.cls</code>	書籍用
<code><slide></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「p_{La}TeX 2_ε 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^AT_EX3 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_EX 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_EX 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^AT_EX 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><article></code>	<code>jsarticle.cls</code>	論文・レポート用
<code><book></code>	<code>jsbook.cls</code>	書籍用
<code><report></code>	<code>jsreport.cls</code>	レポート用
<code><jspf></code>	<code>jspf.cls</code>	某学会誌用
<code><kiyou></code>	<code>kiyou.cls</code>	某紀要用

以下では実際のコードに即して説明します。

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

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

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

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

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

■BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため `calc` を読み込む。

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

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

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

クラスの本体ではこの他に geometry パッケージが読み込まれる。

TODO: 依存パッケージの情報。

互換性のための補助パッケージを読み込む。

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

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

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

\jsEngine [暗黙文字トークン] エンジン (T_EX の種類) の種別：j = pT_EX 系、x = X_YT_EX、p = pdfT_EX (含 DVI モード)、l = LuaT_EX、J = NTT jT_EX、O = Omega 系、n = 以上の何れでもない。

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

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

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

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

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

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

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

非サポートのエンジンの場合は強制終了させる。

※ NTT jTeX と Omega 系。

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

LuaTeX の場合、本クラス用の Lua モジュールを用意する。

```
53 \ifx l\jsEngine
54   \directlua{ bxjs = {} }
55 \fi
```

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

```
56 \ifjsWitheTeX \let\bxjs@protected\protected
57 \else \let\bxjs@protected\@empty
58 \fi
```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

```
59 \ifjsWitheTeX
60   \def\bxjs@robust@def{\protected\def}
61 \else
62   \def\bxjs@robust@def{\DeclareRobustCommand*}
63 \fi
```

`\ifjsInPdfMode` [スイッチ] pdfTeX / LuaTeX が PDF モードで動作しているか。

※ LuaTeX 0.8x 版でのプリミティブ名変更に対応。

```
64 \newif\ifjsInPdfMode
65 \@nameuse{ImposeOldLuaTeXBehavior}
66 \let\bxjs@tmpa\PackageWarningNoLine
67 \let\PackageWarningNoLine\PackageInfo % suppress warning
68 \RequirePackage{ifpdf}
69 \let\PackageWarningNoLine\bxjs@tmpa
70 \@nameuse{RevokeOldLuaTeXBehavior}
71 \let\ifjsInPdfMode\ifpdf
```

`\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。

```
72 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
```

```

73 #1\expandafter\@firstoftwo
74 \else\expandafter\@secondoftwo
75 \fi}

\bxjs@cslet \bxjs@cslet{<名前 1>}\制御綴 :
76 \def\bxjs@cslet#1{%
77 \expandafter\let\csname#1\endcsname}

\bxjs@csletcs \bxjs@csletcs{<名前 1>}{<名前 2>} :
78 \def\bxjs@csletcs#1#2{%
79 \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。
80 \def\bxjs@catopt#1#2{%
81 #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

\bxjs@ifplus \@ifstar の + 版。
82 \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。
83 \let\c@bxjs@tempcnta\@tempcnta
84 \def\bxjs@gset@tempcnta{\setcounter{bxjs@tempcnta}}

\jsSetQHLength \jsSetQHLength\CS{<長さ式>} : \setlength の変種で、通常の calc の長さ式の代わ
りに、「Q/H/trueQ/trueH/zw/zh の単位付きの実数」が記述できる（この場合は式は使え
ない）。
85 \def\jsSetQHLength#1#2{%
86 \begin{group}
87 \bxjs@parse@qh{#2}%
88 \ifx\bxjs@tmpb\relax
89 \setlength\@tempdima{#2}%
90 \xdef\bxjs@g@tmpa{\the\@tempdima}%
91 \else \global\let\bxjs@g@tmpa\bxjs@tmpb
92 \fi
93 \end{group}
94 #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)pLATEX 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。
95 \if j\jsEngine \def\bxjs@parse@qh@units{zw,zh}
96 \else \def\bxjs@parse@qh@units{trueQ,trueH,Q,H,zw,zh}
97 \fi
98 \def\bxjs@parse@qh#1{%

```

```

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

```

今の段階では Q/H だけが使用可能。

```

120 \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。

```

121 \@onlypreamble\bxjs@begin@document@hook
122 \let\bxjs@begin@document@hook\@empty
123 \AtBeginDocument{\bxjs@begin@document@hook}

```

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

```

124 \@onlypreamble\bxjs@post@option@hook
125 \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。

```

126 \@onlypreamble\bxjs@pre@jadriver@hook
127 \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。)

```

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

```

一時的な手続き用の制御綴。

```

130 \@onlypreamble\bxjs@tmpdo
131 \@onlypreamble\bxjs@tmpdo@a
132 \@onlypreamble\bxjs@tmpdo@b
133 \@onlypreamble\bxjs@tmpdo@c
134 \@onlypreamble\bxjs@tmpdo@d

```


Lua_T_EX 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_A_T_EX 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。

```
135 \if 1\jsEngine
136 \def\bxjs@tmpdo#1{%
137   \xdef\bxjs@pre@jadriver@hook{%
138     \bxjs@pre@jadriver@hook
139     \catcode`#1=\the\catcode`#1\relax}%
140   \catcode`#1=11\relax}
141 \@tfor\bxjs@tmpa:=和西暦\do
142   {\expandafter\bxjs@tmpdo\bxjs@tmpa}
143 \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。

```
144 \bxjs@robust@def\jsInhibitGlue{%
145   \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。

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

```
156 \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。

```
157 \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。

```
158 %<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。

```
159 %<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。

```
160 %<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。

```
161 \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 1m^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 1.5m^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、 \LaTeX 2_ϵ の `b5paper` は $250\text{mm} \times 176\text{mm}$ ですが、 $\text{p}\text{\LaTeX 2}_\epsilon$ の `b5paper` は $257\text{mm} \times 182\text{mm}$ になっています。ここでは $\text{p}\text{\LaTeX 2}_\epsilon$ にならって JIS に従いました。

デフォルトは `a4paper` です。

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

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

```
162 \@onlypreamble\bxjs@setpaper
163 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
164 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
165 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
166 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
167 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
168 \DeclareOption{b4paper}{\bxjs@setpaper{{257truemm}{364truemm}}}
169 \DeclareOption{b5paper}{\bxjs@setpaper{{182truemm}{257truemm}}}
170 \DeclareOption{b6paper}{\bxjs@setpaper{{128truemm}{182truemm}}}
```

```

171 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
172 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
173 \DeclareOption{b4j}{\bxjs@setpaper{{257truemm}{364truemm}}}
174 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
175 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
176 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}

```

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

```

177 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
178 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
179 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
180 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
181 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

182 \newif\if@landscape
183 \@landscapefalse
184 \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。

```

185 \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。

```

186 %<!slide>\@slidefalse
187 %<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。

```
188 \newcommand{\@ptsize}{0}
189 \def\bxjs@param@basefontsize{10pt}
190 \def\jsUnusualPtSize{-20}
```

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

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

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

※クラスオプションのトークン列の中に展開可能なトークンがある場合、 \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。

```
192 \jsSetQHLlength\@tempdima{#1}%
193 \edef\bxjs@param@basefontsize{\the\@tempdima}%
194 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
195 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
196 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
197 \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi
```

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

```
198 \newif\ifjsc@mag
199 \newif\ifjsc@mag@xreal
200 %\let\jsc@magscale\@undefined
201 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
202 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
203 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
204 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
205 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
206 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
207 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
208 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
209 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
210 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
211 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
212 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
213 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
214 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
215 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
216 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
217 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
218 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
219 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}
```

JS クラス互換の `magstyle` 設定オプション。

```
220 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
```

```

221 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
222 \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_{La}T_EX 2_ε 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_{La}T_EX 2_ε 本体で宣言されています。

取りあえず、p_TE_X 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。

```

223 \if j\jsEngine
224 \hour\time \divide\hour by 60\relax
225 \@tempcnta\hour \multiply\@tempcnta 60\relax
226 \minute\time \advance\minute-\@tempcnta
227 \DeclareOption{tombow}{%
228   \tombowtrue \tombowdatetrue
229   \setlength{\@tombowwidth}{.1\p}%
230   \@bannertoken{%
231     \jobname\space(\number\year-\two@digits\month-\two@digits\day
232     \space\two@digits\hour:\two@digits\minute)}}%
233   \maketombowbox}
234 \DeclareOption{tombo}{%
235   \tombowtrue \tombowdatefalse
236   \setlength{\@tombowwidth}{.1\p}%
237   \maketombowbox}
238 \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。

```

239 \if j\jsEngine
240 \DeclareOption{mentuke}{%
241   \tombowtrue \tombowdatefalse
242   \setlength{\@tombowwidth}{\z@}%
243   \maketombowbox}
244 \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。

```

245 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
246 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
247 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 twocolumn で二段組になります。

```

248 \DeclareOption{onecolumn}{\@twocolumnfalse}
249 \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。

```

250 \DeclareOption{titlepage}{\@titlepagetrue}
251 \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 L^AT_EX 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。

```

252 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
253 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
254 %<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 L^AT_EX 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。

```

255 \def\eqnarray{%
256   \stepcounter{equation}%
257   \def\@currentlabel{\p@equation\theequation}%
258   \global\@eqnswtrue
259   \m@th
260   \global\@eqcnt\z@
261   \tabskip\@centering
262   \let\\\@eqnocr
263   $$$\everycr{}\halign to\displaywidth\bgroup
264     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
265     &\global\@eqcnt\@ne \hfil$\displaystyle{##}\hfil
266     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
267     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
268     \tabskip\z@skip
269   \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。

```

270 \DeclareOption{leqno}{\input{leqno.clo}}
271 \DeclareOption{fleqn}{\input{fleqn.clo}}
272 % fleqn 用の eqnarray 環境の再定義
273 \def\eqnarray{%
274   \stepcounter{equation}%

```

```

275 \def\@currentlabel{\p@equation\theequation}%
276 \global\@eqnswtrue\m@th
277 \global\@eqcnt\z@
278 \tabskip\mathindent
279 \let\=\@eqnocr
280 \setlength\abovedisplayskip{\topsep}%
281 \ifvmode
282 \addtolength\abovedisplayskip{\partopsep}%
283 \fi
284 \addtolength\abovedisplayskip{\parskip}%
285 \setlength\belowdisplayskip{\abovedisplayskip}%
286 \setlength\belowdisplayshortskip{\abovedisplayskip}%
287 \setlength\abovedisplayshortskip{\abovedisplayskip}%
288 $$\everycr{}\halign to\linewidth% $$
289 \bgroup
290 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
291 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
292 &\global\@eqcnt\tw@
293 $\displaystyle{##}$\hfil \tabskip\@centering
294 &\global\@eqcnt\thr@@ \hbext@\z@\bgroup\hss##\egroup
295 \tabskip\z@skip\cr
296 }}

```

■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。

```

297 % \DeclareOption{openbib}{%
298 % \AtEndOfPackage{%
299 % \renewcommand\@openbib@code{%
300 % \advance\leftmargin\bibindent
301 % \itemindent -\bibindent
302 % \listparindent \itemindent
303 % \parsep \z@}%
304 % \renewcommand\newblock{\par}}}

```

■数式フォントとして和文フォントを登録しないオプション 数式中では 16 通りのフォントしか使えません。AMSFonTS や mathpmtx 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：

```

305 %\let\bxjs@enablejfam\@undefined

```

enablejfam オプションの処理。

```
306 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
307 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
308 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\@undefined}
309 \define@key{bxjs}{enablejfam}[true]{%
310   \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。

```
311 \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「数式和文ファミリー」が有効と無効の選択は異なるし、またそもそも有効と無効の一方しか選択できない場合もある。

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

[2016-07-13] \ifdraft を定義するのをやめました。

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

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

```
312 \let\ifjsDraft\iffalse
313 \@onlypreamble\bxjs@draft
314 \def\bxjs@draft#1{%
315   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
316 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
317 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
318 \g@addto@macro\bxjs@begin@document@hook{%
319   \expandafter\ifx\csname ifdraft\endcsname\relax
320     \expandafter\let\csname ifdraft\expandafter\endcsname
321     \csname ifjsDraft\endcsname
322   \fi}
```

■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 などでも出力出来るようになる) が使えます。

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

BXJS クラスではここは和文ドライバの管轄。

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

BXJS クラスでは geometry パッケージがこの処理を行う。

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

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

```
323 \newif\ifbxjs@papersize
324 \bxjs@papersizetrue
325 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
326 \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。

```
327 \newif\if@english
328 \@englishfalse
329 \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^AT_EX 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。

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

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

※元の長さが 128pt 以上の場合でも動作するように修正した。

```

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

```

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

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

```

358 \DeclareOption{pandoc}{%

```

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

※実際の和文ドライバ・エンジン設定より優先される。

```

359   \g@addto@macro\bxjs@post@option@hook{%
360     \bxjs@oldfontcommandstrue
361     \setkeys{bxjs}{ja=pandoc}%
362     \let\bxjs@engine@given=*}%

```

ドライバオプションを `dvi=dvipdfmx` 相当に変更する。

※これは実際のドライバ設定で上書きできる（オプション宣言順に注意）。

TODO: できない気がする…。

```
363 \def\bxjs@driver@opt{dvipdfmx}%
364 \bxjs@dvi@opttrue}
```

■エンジン・ドライバオプション

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

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

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
366 %\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 \LaTeX の慣習に従って「ドライバの指定」とみなすべきだから。

```
367 \DeclareOption{autodetect-engine}{%
368   \let\bxjs@engine@given=*}
369 \DeclareOption{latex}{%
370   \def\bxjs@engine@opt{latex}%
371   \let\bxjs@engine@given=n}
372 \DeclareOption{platex}{%
373   \def\bxjs@engine@opt{platex}%
374   \let\bxjs@engine@given=j}
375 \DeclareOption{uplatex}{%
376   \def\bxjs@engine@opt{uplatex}%
377   \let\bxjs@engine@given=u}
378 \DeclareOption{xelatex}{%
379   \def\bxjs@engine@opt{xelatex}%
380   \let\bxjs@engine@given=x}
381 \DeclareOption{pdflatex}{%
382   \def\bxjs@engine@opt{pdflatex}%
383   \let\bxjs@engine@given=p}
384 \DeclareOption{lualatex}{%
385   \def\bxjs@engine@opt{lualatex}%
386   \let\bxjs@engine@given=l}
387 \DeclareOption{platex-ng}{%
388   \def\bxjs@engine@opt{platex-ng}%
389   \let\bxjs@engine@given=g}
390 \DeclareOption{platex-ng*}{%
391   \def\bxjs@engine@opt{platex-ng*}%
392   \let\bxjs@platexng@nodrv=t%
393   \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。

```

394 %\let\bxjs@driver@given\@undefined
395 \let\bxjs@driver@@dvimode=0
396 \let\bxjs@driver@@dvipdfmx=1
397 \let\bxjs@driver@@pdfmode=2
398 \let\bxjs@driver@@xetex=3
399 \let\bxjs@driver@@dvips=4
400 \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。

```

401 %\let\bxjs@driver@opt\@undefined

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

```

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

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

```

426 \DeclareOption{dvipdfmx-if-dvi}{%
427   \setkeys{bxjs}{dvi=dvipdfmx}}

```

■その他の BXJS 独自オプション

TODO: 互換用オプションを分離する (2.0 版で?)。

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

```
428 \newif\ifbxjs@bigcode \bxjs@bigcodefalse
```

その上で、「 $\mathrm{T}_\mathrm{E}\mathrm{X}$ 環境がある程度新しければ利用可能であろう」と判断し `bxjs@bigcode` の既定値を真とする。具体的な判断基準として、「 $\mathrm{T}_\mathrm{E}\mathrm{X}$ のバージョンが 3.14159265 (2014 年 1 月) 以上であるか」を採用する。

```
429 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
430 \def\bxjs@tmpb#1 #2#3\@nil{%
431   \ifx1#2\bxjs@bigcodetrue \fi}
432 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil
```

`nobigcode` / `bigcode` オプションの定義。

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

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

```
437 \newif\ifbxjs@oldfontcommands
```

`nooldfontcommands`、`oldfontcommands` オプションの定義。

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

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

■keyval 型のオプション

その他のオプションは `keyval` の機構を用いて処理する。

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

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

※ネスト不可。

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

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

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

```
449 \@onlypreamble\bxjs@declare@enum@option
450 \def\bxjs@declare@enum@option#1#2{%
```

```

451 \define@key{bxjs}{#1}{%
452   \expandafter\ifx\csname bxjs@#2@@#1\endcsname\relax
453   \bxjs@error@keyval{#1}{##1}%
454   \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@@#1}%
455   \fi}}

```

`\bxjs@declare@bool@option` `\bxjs@declare@bool@option{<オプション名>}{<スイッチ名>}`

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

```

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

```

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

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

```

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

```

`\jsScale` 〔実数値マクロ〕 和文スケール値。

```

474 \def\jsScale{0.924715}

```

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

```

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

```

`base` オプションの処理。

```

476 \define@key{bxjs}{base}{%
477   \edef\bxjs@base@opt{#1}%
478   \bxjs@setbasefontsize{#1}}
479 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

```

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

```

480 %\let\bxjs@jbase@opt\@undefined

```

`jbase` オプションの処理。

```

481 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
482 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}

```

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

```

483 %\let\bxjs@scale@opt\@undefined

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

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

```

`\bxjs@param@mag` `mag` オプションの値。

```

489 \let\bxjs@param@mag\relax

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

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

```

`\bxjs@jadriver` 和文ドライバの名前。

```

492 \let\bxjs@jadriver\relax
493 %\let\bxjs@jadriver@opt\@undefined

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

```

`\jsJaFont` 和文フォント設定の名前。

```

497 \let\jsJaFont\@empty

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

```

`\jsJaParam` 和文ドライバパラメタの文字列。

```

499 \let\jsJaParam\@empty

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

```

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

```

501 \let\bxjs@magstyle@mag=m
502 \let\bxjs@magstyle@real=r
503 \let\bxjs@magstyle@xreal=x

```

(新しい素敵な名前。)

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

```
504 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
505 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
506 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

`\bxjs@magstyle@default` は既定の値を表す。

```
507 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
508 \ifx l\jsEngine \ifnum\luatexversion>86
509   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
510 \fi\fi
511 \ifjsWithpTeXng
512   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
513 \fi
514 \let\bxjs@magstyle\bxjs@magstyle@default
```

`magstyle` オプションの処理。

```
515 \define@key{bxjs}{magstyle}{%
516   \expandafter\let\expandafter\bxjs@magstyle\csname
517     bxjs@magstyle@#1\endcsname
518   \ifx\bxjs@magstyle\relax
519     \ClassError\bxjs{clsname
520       {Invalid value '#1' for option magstyle}}\@ehc
521     \let\bxjs@magstyle\bxjs@magstyle@default
522   \fi}
```

`\bxjs@geometry` `geometry` オプションの値。

```
523 \let\bxjs@geometry@class=c
524 \let\bxjs@geometry@user=u
525 \let\bxjs@geometry\bxjs@geometry@class
```

`geometry` オプションの処理。

```
526 \define@key{bxjs}{geometry}{%
527   \expandafter\let\expandafter\bxjs@geometry\csname
528     bxjs@geometry@#1\endcsname
529   \ifx\bxjs@geometry\relax
530     \ClassError\bxjs{clsname
531       {Invalid value '#1' for option geometry}}\@ehc
532     \let\bxjs@geometry\bxjs@geometry@class
533   \fi}
```

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

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

`fancyhdr` オプションの処理。

```
535 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
536 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
537 \define@key{bxjs}{fancyhdr}[true]{%
```



```

538 \bxjs@set@keyval{fancyhdr}{#1}{}

\ifbxjs@dvi@opt dvi オプションが指定されたか。
539 \newif\ifbxjs@dvi@opt

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

dvi オプションの処理。
545 \define@key{bxjs}{dvi}{%
546 \expandafter\let\expandafter\bxjs@tmpa\csname
547 bxjs@dvidriver@@#1\endcsname
548 \ifx\bxjs@tmpa\relax
549 \ClassError\bxjs{clsname
550 {Invalid value '#1' for option dvi}}\@ehc
551 \else

\bxjs@driver@given を未定義にしていることに注意。
552 \def\bxjs@driver@opt{#1}%
553 \let\bxjs@driver@given\undefined
554 \bxjs@dvi@opttrue
555 \fi}

\ifbxjs@layout@buggyhmargin [スイッチ] bxjsbook の左右マージンがアレか。
※ layout が v1 の場合はアレになる。
556 \newif\ifbxjs@layout@buggyhmargin

\ifbxjs@force@chapterabstract [スイッチ] abstract 環境を chapterabstract にするか。
※ bxjsbook では常に真。bxjsreport では layout が v1 の場合に真になる。
557 \newif\ifbxjs@force@chapterabstract
558 %<book>\bxjs@force@chapterabstracttrue

layout オプションの処理。
559 \@namedef{bxjs@kv@layout@v1}{%
560 %<book>\bxjs@layout@buggyhmargintrue
561 %<report>\bxjs@force@chapterabstracttrue
562 }
563 \@namedef{bxjs@kv@layout@v2}{%
564 %<book>\bxjs@layout@buggyhmarginfalse
565 %<report>\bxjs@force@chapterabstractfalse
566 }
567 \define@key{bxjs}{layout}{%
568 \bxjs@set@keyval{layout}{#1}{}

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

598 \chardef\bxjs@everyparhook@none=0
599 \chardef\bxjs@everyparhook@compat=1
600 \chardef\bxjs@everyparhook@modern=2
601 \if j\jsEngine

```

```

602 \let\bxjs@everyparhook\bxjs@everyparhook@compat
603 \else
604 \let\bxjs@everyparhook\bxjs@everyparhook@modern
605 \fi
606 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
607 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
608 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
609 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}

```

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

```

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

```

`\ifbxjs@usezw` [スイッチ] use-zw の指定値。

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

■ オプションの実行

L^AT_EX の実装では、クラスやパッケージのオプションのトークン列の中に { } が含まれると正常に処理ができない。これに対処する為 `\@removeelement` の実装に少し手を加える

(仕様は変わらない)。

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

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

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

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

後処理

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

■ グローバルオプションの整理

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

```
654 \def\bxjs@tmpdo{%
655   \def\bxjs@tmpa{\@gobble}%
656   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
657   \let\@classoptionslist\bxjs@tmpa}
658 \def\bxjs@tmpdo@a#1,{%
659   \ifx\@nil#1\relax\else
660     \bxjs@tmpdo@b#1{\@nil
661       \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
662       \expandafter\bxjs@tmpdo@a
663     \fi}
664 \def\bxjs@tmpdo@b#1#\bxjs@tmpdo@c}
665 \def\bxjs@tmpdo@c#1\@nil{%
```

```

666 \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
667 \bxjs@tmpdo

```

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

```

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

```

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

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

このコードを削除。

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

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

```

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

```

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

```

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

```

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

```

687 \ifx *\bxjs@engine@given
688   \let\bxjs@engine@given\bxjs@tmpb

```

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

```

689   \ifx j\bxjs@engine@given
690     \g@addto@macro\@classoptionslist{,latex}
691   \else\ifx u\bxjs@engine@given
692     \g@addto@macro\@classoptionslist{,uplatex}
693   \fi\fi
694 \fi
695 \ifx\bxjs@engine@given\@undefined\else
696   \ifx\bxjs@engine@given\bxjs@tmpb\else
697     \ClassError\bxjs@clsname
698       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
699   \fi
700 \fi


```

エンジンが pT_EX-ng の場合、グローバルオプションに `uplatex` を追加する。

```

701 \ifjsWithpTeXng
702   \g@addto@macro\@classoptionslist{,uplatex}
703 \fi

```

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

```

704 \@tempswatrue
705 \ifx \bxjs@driver@given\@undefined\else
706   \ifjsInPdfMode
707     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
708       \@tempswafalse
709     \fi
710   \else\ifx x\jsEngine
711     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
712       \@tempswafalse
713     \fi
714   \else
715     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
716       \@tempswafalse
717     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
718       \@tempswafalse
719     \fi\fi
720   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
721     \@tempswafalse
722   \fi\fi
723 \fi\fi
724 \fi
725 \if@tempswa\else
726   \ClassError\bxjs@clsname
727     {Option '\bxjs@driver@opt' used on wrong engine}\@ehc

```

728 \fi

DVI 出力のエンジンである場合の追加処理。

```
729 \ifjsInPdfMode \@tempswafalse
730 \else\ifx x\jsEngine \@tempswafalse
731 \else\ifjsWithpTeXng \@tempswafalse
732 \else \@tempswatru
733 \fi\fi\fi
734 \if@tempswa
```

ドライバオプションがない場合は警告を出す。

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

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

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

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

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

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

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

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

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

```
758 \def\bxjs@@minimal{minimal}
759 \ifx\bxjs@jadriver\relax
```

```

760 \ifx j\jsEngine
761   \def\bxjs@jadriver{standard}
762 \else
763   \let\bxjs@jadriver\bxjs@@minimal
764 \fi
765 \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。

```

766 \ifx\bxjs@jadriver@opt\@undefined\else
767   \ifx\bxjs@engine@given\@undefined
768     \ClassError\bxjs@clsname
769       {An engine option must be explicitly given}%
770       {When you use a Japanese-driver you must specify a correct\MessageBreak
771         engine option.\MessageBreak\@ehc}
772 \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。)

```

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

```

base、jbase、scale の値を用いて和文スケール値を解決する。

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

```

782 \ifx\bxjs@jbase@opt\@undefined\else
783   \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。

```

784     \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
785     \bxjs@invscale\@tempdima\jsScale
786     \bxjs@setbasefontsize{\@tempdima}%
787 \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。

```

788   \ifx\bxjs@scale@opt\@undefined\else
789     \ClassWarningNoLine\bxjs@clsname
790       {Redundant 'scale' option is ignored}%
791   \fi
792   \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%

```



```

793 \tempdimb=\bxjs@param@basefontsize\relax
794 \edef\jsScale{\strip@pt\tempdimb}%
795 \bxjs@invscale\tempdima\jsScale
796 \edef\jsScale{\strip@pt\tempdima}%
797 \fi
798 \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。

```

799 \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。）

```

800 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
801 \tempcnta="80 \loop \ifnum\tempcnta<"100
802 \catcode\tempcnta\active
803 \advance\tempcnta\@ne
804 \repeat
805 \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。

```

806 \ifbxjs@disguise@js
807 %<book|report>\def\bxjs@js@clsname{jsbook}
808 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
809 \@namedef{ver@bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
810 \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。

```

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

```

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

```

812 \ifbxjs@oldfontcommands
813 \AtEndOfClass{\allowoldfontcommands}
814 \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。

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。

```
815 \newdimen\stockwidth \newdimen\stockheight
816 \begingroup\expandafter\expandafter\expandafter\endgroup
817 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
818 % \newdimen\stockwidth \newdimen\stockheight
819 \setlength{\stockwidth}{\paperwidth}
820 \setlength{\stockheight}{\paperheight}
821 \advance \stockwidth 2in
822 \advance \stockheight 2in
823 \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。

```
824 %<slide>\def\n@baseline{13}%
825 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
826 %<!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。

```
827 \ifx\bxjs@magstyle\bxjs@magstyle@mag
828 \jsc@magtrue
829 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
830 \jsc@mag@xrealtrue
831 \fi\fi
```

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

[2008-12-26] $1000 / \mag$ に相当する `\inv@mag` を定義しました。`truein` を使っていたところを `\inv@mag in` に直しましたので、`geometry` パッケージと共存できると思います。

なお、新ドキュメントクラス側で 10pt 以外にする場合の注意：

- geometry 側でオプション `true`dimen 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。

```
832 \ifx\bxjs@param@mag\relax
833   \@tempdima=\bxjs@param@basefontsize
834   \advance\@tempdima.001pt \multiply\@tempdima25
835   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
836   \edef\bxjs@param@mag{\the\@tempcnta}
837 \else
838 % mag 値が直接指定された場合
839   \bxjs@gset@tempcnta{\bxjs@param@mag}
840   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
841 % 有効な mag 値の範囲は 1--32768
842   \edef\bxjs@param@mag{\the\@tempcnta}
843   \advance\@tempcnta100000
844   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
845   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
846   \edef\bxjs@param@basefontsize{\the\@tempdima}
847 \fi
848 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
849 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
850 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
851 \edef\jsc@magscale{\strip@pt\@tempdima}
852 \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。

```
853 \newdimen\jsc@mpt
854 \newdimen\jsc@mmm
855 \ifjsc@mag
856   \jsc@mpt=1\p@
857   \jsc@mmm=1mm
858   \catcode\?=9 % \p@? read as \p@
859 \else
860   \jsc@mpt=\jsc@magscale\p@
861   \jsc@mmm=\jsc@magscale mm
```

```

862 \catcode\?=11 \let\p@?\jsc@mpt
863 \fi
864 \chardef\bxjs@qmc=\catcode\?\relax
865 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode\?=12\relax}

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

    nozw 非指定時は \zw を \jsZw と同義にする。

866 \newdimen\jsZw
867 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
868 \ifbxjs@usezw
869 \providecommand*\zw{\jsZw}
870 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

872 \ifjsc@mag@xreal
873 \RequirePackage{type1cm}
874 \let\jsc@invscale\bxjs@invscale

ムニャムニャムニャ……。

```

```

875 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
876 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
877 \let\jsc@get@external@font\get@external@font
878 \def\get@external@font{%
879   \jsc@preadjust@extract@font
880   \jsc@get@external@font}
881 \def\jsc@fstrunc#1{%
882   \edef\jsc@tmpa{\strip@pt#1}%
883   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
884 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
885   \if#5*\else
886     \edef\jsc@tmpa{#1%
887       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
888   \fi}
889 \def\jsc@preadjust@extract@font{%
890   \let\jsc@req@size\fontsize
891   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
892   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
893   \let\jsc@ref@size\jsc@tmpa
894   \let\fontsize\jsc@ref@size}
895 \def\execute@size@function#1{%
896   \let\jsc@c@ref@size\fontsize
897   \let\fontsize\jsc@req@size
898   \csname s@fct@#1\endcsname}
899 \let\jsc@DeclareErrorFont\DeclareErrorFont

```

```

900 \def\DeclareErrorFont#1#2#3#4#5{%
901   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
902   \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
903   \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
904 \def\gen@sfcnt{%
905   \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
906   \empty@sfcnt}
907 \def\genb@sfcnt{%
908   \edef\mandatory@arg{%
909     \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@0}%
910   \empty@sfcnt}
911 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
912 \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 913 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 914 %\def\jsc@medskip{\vspace\jsc@medskipamount}
915 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 916 \newskip\jsc@smallskipamount
\jsc@bigskipamount 917 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
918 %\newskip\jsc@medskipamount
919 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
920 %\newskip\jsc@bigskipamount
921 %\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] トンボオプションが指定されているとき「だけ」`\stockwidth`, `\stockheight` が定義されています。

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

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

```

922 % \ifpapersize
923 %   \setlength{\@tempdima}{\paperwidth}
924 %   \setlength{\@tempdimb}{\paperheight}
925 %   \iftombow
926 %     \advance \@tempdima 2truein
927 %     \advance \@tempdimb 2truein
928 %   \fi

```

```

929 % \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
930 % \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、 \LaTeX の `\def\@{\spacefactor\@m}` という定義 ($\@m$ は 1000) では `I watch TV\@.` と書くと `V` とピリオドのベアカーニングが効かなくなります。そこで、次のような定義に直し、`I watch TV.\@` と書くことにします。

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

BXJS クラスでの変更点：

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

```

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

```

4 フォントサイズ

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

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

のようにして行います。これは

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

という意味です。ただし、処理を速くするため、以下では 10 と同義の \LaTeX 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、 \LaTeX 本体で定義されてい

ます。

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

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

`\kanjiskip` は $\text{pLATEX}_2\epsilon$ で `0pt plus .4pt minus .5pt` に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

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

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

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

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

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

`\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。

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

`\jsResetDimen` ユーザ定義用のフック。

```
956 \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。

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

```
971 \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 (`\@currsiz` 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。

```
972 \newif\ifnarrowbaselines
973 \if@english
974   \narrowbaselinestrue
975 \fi
976 \def\narrowbaselines{%
```



```

977 \narrowbaselinestruе
978 \skip0=\abovedisplayskip
979 \skip2=\abovedisplayshortskip
980 \skip4=\belowdisplayskip
981 \skip6=\belowdisplayshortskip
982 % 一時的に警告を無効化する
983 \let\bxjs@save@nomath\@nomath
984 \let\@nomath\@gobble
985 \@currsize\selectfont
986 \let\@nomath\bxjs@save@nomath
987 \abovedisplayskip=\skip0
988 \abovedisplayshortskip=\skip2
989 \belowdisplayskip=\skip4
990 \belowdisplayshortskip=\skip6\relax}
991 \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。

```

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

```

\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。

```

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

```

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_EX Q & A 52569 から始まる議論について逡巡していましたが、結局、微調節してみることにしました。

```
1003 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1004 \abovedisplayshortskip \z@ \@plus3\p@?
1005 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1006 \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。

```
1007 \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。

```
1008 \normalsize
```

`\Cht` 基準となる長さの設定をします。pL_AT_EX 2_ε 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_EX 系以外では未定義なのでここで定義する。

```
1009 \ifx\Cht\undefined \newdimen\Cht \fi
1010 \ifx\Cdp\undefined \newdimen\Cdp \fi
1011 \ifx\Cwd\undefined \newdimen\Cwd \fi
1012 \ifx\Cvs\undefined \newdimen\Cvs \fi
1013 \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。

```
1014 \setlength\Cht{0.88\jsZw}
1015 \setlength\Cdp{0.12\jsZw}
1016 \setlength\Cwd{1\jsZw}
1017 \setlength\Cvs{\baselineskip}
1018 \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 ± 2 , 2 ± 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。

```
1019 \newcommand{\small}{%
1020   \bxjs@if@narrowbaselines{%
1021     %<!kiyou> \jsc@setfontsize\small\@ixpt{11}%
```

```

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

```

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

```

1035 \newcommand{\footnotesize}{%
1036 \bxjs@if@narrowbaselines{%
1037 %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
1038 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
1039 }{%else
1040 %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
1041 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1042 }%
1043 \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1044 \abovedisplayshortskip \z@ \@plus2\p@?
1045 \belowdisplayskip \abovedisplayskip
1046 \belowdisplayshortskip \belowdisplayskip
1047 \def\@listi{\leftmargin\leftmargini
1048 \topsep \z@
1049 \parsep \z@
1050 \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 1051 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1052 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 1053 \if@twocolumn
\HUGE 1054 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1055 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1056 \else
1057 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
1058 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1059 \fi
1060 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}

```

```

1061 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1062 \newcommand{\LARGE}{\jsc@setfontsize\LARGE@xvipt{25}}
1063 \newcommand{\huge}{\jsc@setfontsize\huge@xxpt{28}}
1064 \newcommand{\Huge}{\jsc@setfontsize\Huge@xxvpt{33}}
1065 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

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

```

1066 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}

```

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

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

```

1067 % \newcommand{\headfont}{\bfseries}
1068 \newcommand{\headfont}{\sffamily}
1069 % \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。

```

1070 %<kiyou>\setlength\columnsep{2\Cwd}
1071 %<kiyou>\setlength\columnsep{28truebp}
1072 \setlength\columnseprule{\z@}

```

■段落

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

```

\lineskiplimit 1073 \setlength\lineskip{1\jsc@empt}
\normallineskiplimit 1074 \setlength\normallineskip{1\jsc@empt}
1075 \setlength\lineskiplimit{1\jsc@empt}
1076 \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。

```
1077 \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。

```
1078 \setlength\parskip{\z@}
```

```
1079 \if@slide
```

```
1080 \setlength\parindent{0\p@}
```

```
1081 \else
```

```
1082 \setlength\parindent{1\Cwd}
```

```
1083 \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 1084 \@lowpenalty 51
```

```
1085 \@medpenalty 151
```

```
1086 \@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。

```
1087 % \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。

```
1088 % \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。

```
1089 \@onlypreamble\bxjs@bd@pre@geometry@hook
```

```
1090 \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。

```
1091 \ifjsc@mag
```

```
1092 \mag=\bxjs@param@mag
```

```

1093 \fi
1094 \setlength{\topskip}{10\p@?}

\jsSetQHLength のための和文単位の定義。
1095 \def\bxjs@unit@trueQ{0.25truemm}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1096 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw

\bxjs@param@paper が長さ指定({W}{H})の場合、geometry の形式(papersize={W,H})
に変換する。
1097 \def\bxjs@tmpdo{\futurelet\bxjs@tmpa\bxjs@tmpdo@a}
1098 \def\bxjs@tmpdo@a{%
1099   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@tmpdo@b
1100   \else \expandafter\bxjs@tmpdo@c \fi}
1101 \def\bxjs@tmpdo@b#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1102 \def\bxjs@tmpdo@c#1\@nil{\bxjs@tmpdo@d#1,,\@nil}
1103 \def\bxjs@tmpdo@d#1,#2,#3\@nil{%
1104   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1105 \expandafter\bxjs@tmpdo\bxjs@param@paper\@empty\@empty\@nil

```

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

```

1106 \edef\bxjs@layout@paper{%
1107   \ifjsc@mag truedimen,\fi
1108   \if@landscape landscape,\fi
1109   \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。

```

1110 %<article|report>
1111 \def\bxjs@layout@base{%
1112   headheight=\topskip,footskip=0.03367\paperheight,%
1113   headsep=\footskip-\topskip,includeheadfoot,%
1114 }
1115 \edef\bxjs@layout{\bxjs@layout@base
1116   hscale=0.76,hmarginratio=1:1,%
1117   vscale=0.83,vmarginratio=1:1,%
1118 }
1119 %</article|report>
1120 %<book>
1121 \def\bxjs@layout@base{%
1122   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1123 }
1124 \ifbxjs@layout@buggyhmargin %---
1125 % アレ
1126 \edef\bxjs@layout{\bxjs@layout@base
1127   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1128   vscale=0.83,vmarginratio=1:1,%
1129 }
1130 \else %---
1131 % 非アレ
1132 \edef\bxjs@layout{\bxjs@layout@base

```

```

1133 hmargin=18\jsc@mmm,%
1134 vscale=0.83,vmarginratio=1:1,%
1135 }
1136 \fi %---
1137 %</book>
1138 %<*slide>
1139 \def\bxjs@layout@base{%
1140 noheadfoot,%
1141 }
1142 \edef\bxjs@layout{\bxjs@layout@base
1143 hscale=0.9,hmarginratio=1:1,%
1144 vscale=0.944,vmarginratio=1:1,%
1145 }
1146 %</slide>

```

textwidth オプションの設定を反映する。

```

1147 %<*!book>
1148 \ifx\bxjs@textwidth@opt\undefined\else
1149 \jsSetQHLlength\@tempdima{\bxjs@textwidth@opt}
1150 \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1151 \fi
1152 %</!book>
1153 \ifx\bxjs@number@of@lines@opt\undefined\else
1154 \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1155 \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1156 \fi

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1157 \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。

```

1158 %<*book>
1159 \newcommand\jsTextWidthLimit{40}
1160 \@tempdima=\jsTextWidthLimit\Cwd
1161 \ifx\bxjs@textwidth@limit@opt\undefined\else
1162 \bxjs@gset@tempcnta{\bxjs@textwidth@limit@opt}
1163 \@tempdima=\@tempcnta\Cwd
1164 \fi
1165 \ifx\bxjs@textwidth@opt\undefined\else
1166 \jsSetQHLlength\@tempdima{\bxjs@textwidth@opt}
1167 \fi
1168 \edef\bxjs@textwidth@limit{\the\@tempdima}
1169 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1170 \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1171 \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1172 \fi
1173 %</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。

```
1174 \def\bxjs@preproc@layout{%
1175   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mpt}
```

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

```
1176 \def\bxjs@postproc@layout{%
```

geometry のドライバを再設定する。

```
1177   \ifx\bxjs@geometry@driver\relax\else
1178     \let\Gm@driver\bxjs@geometry@driver
1179   \fi
```

`\ht\strutbox` の値を元に戻す。

```
1180   \ht\strutbox=\bxjs@save@ht@strutbox\relax
```

`\textwidth` の値を補正する。

```
1181   \ifbxjs@whole@zw@lines
1182     \@tempdimb=\textwidth
1183     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1184     \advance\textwidth.005pt\relax
1185     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1186     \advance\@tempdimb-\textwidth
1187     \advance\oddsidemargin 0.5\@tempdimb
1188     \advance\evensidemargin 0.5\@tempdimb
1189   \fi
1190   \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。

```
1191 %<*book>
1192   \@tempdima=\bxjs@textwidth@limit\relax
1193   \ifbxjs@whole@zw@lines
1194     \advance\@tempdima.005pt\relax
1195     \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1196   \fi
1197   \ifdim\textwidth>\@tempdima
1198     \textwidth=\@tempdima
1199     \addtolength\evensidemargin{\fullwidth-\textwidth}
1200   \fi
1201 %</book>
```

`\textheight` 関連の調整。

```
1202   \@tempdimb=\textheight
1203   \advance\textheight-\topskip
1204   \advance\textheight.005pt\relax
1205   \divide\textheight\baselineskip \multiply\textheight\baselineskip
```



```

1206 \advance\textheight\topskip
1207 \advance\@tempdima-\textheight
1208 \advance\topmargin0.5\@tempdima

\headheight 関連の調整。
1209 \@tempdima=\topskip
1210 \advance\headheight\@tempdima
1211 \advance\topmargin-\@tempdima

marginpar 関連の調整。
1212 \setlength\marginparsep{\columnsep}
1213 \setlength\marginparpush{\baselineskip}
1214 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1215     -\textwidth-10\jsc@mmm-\marginparsep}
1216 \ifbxjs@whole@zw@lines
1217     \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1218 \fi

連動する変数。
1219 \maxdepth=.5\topskip
1220 \stockwidth=\paperwidth
1221 \stockheight=\paperheight
1222 }

```

`\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。

```

1223 \edef\jsGeometryOptions{%
1224     \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージを読み込む

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

```

1225 \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。

```

1226 \ifbxjs@papersize
1227     \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1228         \PassOptionsToPackage{dvipdfm}{geometry}
1229     \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1230         \PassOptionsToPackage{dvipdfm}{geometry}
1231     \fi\fi
1232     \let\bxPapersizeSpecialDone=t
1233 \else
1234     \PassOptionsToPackage{driver=none}{geometry}

```

1235 \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。

```
1236 \AtBeginDocument{\bxjs@bd@pre@geometry@hook}
1237 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1238 \bxjs@preproc@layout
1239 \edef\bxjs@next{%
1240   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1241 }\bxjs@next
1242 \AtBeginDocument{\RevokeOldLuaTeXBehavior}
```

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

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

```
1243 \let\bxjs@geometry@driver\Gm@driver
1244 \bxjs@postproc@layout
```

geometry のドライバ自動判別に対する前処理。

```
1245 \g@addto@macro\bxjs@bd@pre@geometry@hook{%
```

geometry の 4.x 版に関する対策。

※ geometry の 4.x 版の使用は BXJS 1.2 版より非推奨とし、2.0 版でサポートを廃止する予定。

```
1246   \@ifpackagelater{geometry}{2010/02/12}{\fi}%else
1247   \PackageWarningNoLine\bxjs@clsname
1248     {The 'geometry' package installed\MessageBreak
1249     is too old (< v5.0)}%
1250   \if x\jsEngine \ifnum\mag=\@m%else
1251     \def\bxjs@Gm@driver{pdftex}
1252     \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1253     \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1254     \fi\fi}%
```

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

```
1255   \ifjsWithpTeXng
1256     \ifx\Gm@driver\@empty
1257       \def\Gm@driver{pdftex}%
1258     \fi
1259   \fi}
```

\setpagelayout ページレイアウト設定のためのユーザ命令。

```
1260 \def\setpagelayout{%
1261   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{\fi}%else
1262   \@ifstar{\bxjs@setpagelayout@a@one@}{\bxjs@setpagelayout@a@z@}}
1263 \def\bxjs@setpagelayout@a#1#2{%
1264   \ifcase#1% modify
1265     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1266   \or% reset(*)
```

```

1267 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1268 \or% semireset(+)
1269 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1270 \fi
1271 \bxjs@preproc@layout
1272 \edef\bxjs@next{%
1273 \noexpand\geometry{\bxjs@next}%
1274 }\bxjs@next
1275 \bxjs@postproc@layout}

```

■ geometry パッケージを読み込まない

geometry=user の場合の処理。

```
1276 \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。

```

1277 \g@addto@macro\bxjs@begin@document@hook{%
1278 \ifdim\textwidth=.5\maxdimen
1279 \ClassError\bxjs@clsname
1280 {Page layout is not properly set}%
1281 {\@ehd}%
1282 \fi}
1283 \def\jsUseMinimalPageLayout{%
1284 \setlength{\textwidth}{6.5in}%
1285 \setlength{\textheight}{8in}}

```

\setpagelayout はとりあえず無効にしておく。

```

1286 \let\bxjs@geometry@driver\relax
1287 \def\setpagelayout{%
1288 \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1289 \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1290 \def\bxjs@pagelayout@a#1{%
1291 \ClassError\bxjs@clsname
1292 {Command '\string\setpagelayout' is not supported,\MessageBreak
1293 because 'geometry' value is not 'class'}\@eha}
1294 %
1295 \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。

```
1296 %<*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 ですが、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。

```
1297 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1298 \if@slide
1299   \setlength\headheight{0\jsc@empt}
1300 \else
1301   \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1302 \fi
```

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

```
1303 %<*article|kiyou>
1304 \if@slide
1305   \setlength\footskip{0pt}
1306 \else
1307   \setlength\footskip{0.03367\paperheight}
1308   \ifdim\footskip<\baselineskip
1309     \setlength\footskip{\baselineskip}
1310   \fi
1311 \fi
1312 %</article|kiyou>
1313 %<jspf>\setlength\footskip{9\jsc@mmm}
1314 %<*book>
1315 \if@report
1316   \setlength\footskip{0.03367\paperheight}
1317   \ifdim\footskip<\baselineskip
1318     \setlength\footskip{\baselineskip}
```

```

1319 \fi
1320 \else
1321 \setlength\footskip{0pt}
1322 \fi
1323 %</book>
1324 %<*report>
1325 \setlength\footskip{0.03367\paperheight}
1326 \ifdim\footskip<\baselineskip
1327 \setlength\footskip{\baselineskip}
1328 \fi
1329 %</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)。

```

1330 %<*article>
1331 \if@slide
1332 \setlength\headsep{0\jsc@empt}
1333 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1334 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1335 \else
1336 \setlength\headsep{\footskip}
1337 \addtolength\headsep{-\topskip}
1338 \fi
1339 %</article>
1340 %<*book>
1341 \if@report
1342 \setlength\headsep{\footskip}
1343 \addtolength\headsep{-\topskip}
1344 \else
1345 \setlength\headsep{6\jsc@mmm}
1346 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1347 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1348 \fi
1349 %</book>
1350 %<*report>
1351 \setlength\headsep{\footskip}
1352 \addtolength\headsep{-\topskip}
1353 %</report>
1354 %<*jspf>
1355 \setlength\headsep{9\jsc@mmm}
1356 \addtolength\headsep{-\topskip}
1357 %</jspf>
1358 %<*kiyou>
1359 \setlength\headheight{0\jsc@empt}
1360 \setlength\headsep{0\jsc@empt}

```

```

1361 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1362 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1363 %</kiyou>

```

\maxdepth \maxdepth は本文最下行の最大の深さで、plain TeX や L^AT_EX 2.09 では 4pt に固定でした。L^AT_EX 2e 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。

```

1364 \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。

```

1365 \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。

```

1366 %< *article>
1367 \if@slide
1368   \setlength\fullwidth{0.9\paperwidth}
1369 \else
1370   \setlength\fullwidth{0.76\paperwidth}
1371 \fi
1372 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1373 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1374 \setlength\textwidth{\fullwidth}
1375 %< /article>
1376 %< *book>
1377 \if@report
1378   \setlength\fullwidth{0.76\paperwidth}
1379 \else
1380   \setlength\fullwidth{\paperwidth}
1381   \addtolength\fullwidth{-36\jsc@mmm}
1382 \fi
1383 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1384 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1385 \setlength\textwidth{\fullwidth}
1386 \if@report \else

```

```

1387 \if@twocolumn \else
1388 \ifdim \fullwidth>40zw
1389 \setlength\textwidth{40zw}
1390 \fi
1391 \fi
1392 \fi
1393 %</book>
1394 %<*report>
1395 \setlength\fullwidth{0.76\paperwidth}
1396 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1397 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1398 \setlength\textwidth{\fullwidth}
1399 %</report>
1400 %<*jspf>
1401 \setlength\fullwidth{50zw}
1402 \addtolength\fullwidth{8\jsc@mmm}
1403 \setlength\textwidth{\fullwidth}
1404 %</jspf>
1405 %<*kiyou>
1406 \setlength\fullwidth{48zw}
1407 \addtolength\fullwidth{\columnsep}
1408 \setlength\textwidth{\fullwidth}
1409 %</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つたので、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)。

```

1410 %<*article|book|report>
1411 \if@slide
1412 \setlength{\textheight}{0.95\paperheight}
1413 \else
1414 \setlength{\textheight}{0.83\paperheight}
1415 \fi
1416 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
\headheight (2003-06-26)

```

```

1417 \addtolength{\textheight}{-\headsep}
1418 \addtolength{\textheight}{-\footskip}
1419 \addtolength{\textheight}{-\topskip}
1420 \divide\textheight\baselineskip
1421 \multiply\textheight\baselineskip
1422 %</article|book|report>
1423 %<jspf>\setlength{\textheight}{51\baselineskip}
1424 %<kiyou>\setlength{\textheight}{47\baselineskip}
1425 \addtolength{\textheight}{\topskip}
1426 \addtolength{\textheight}{0.1\jsc@mpt}
1427 %<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^AT_EX 2_ε 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。

```

1428 \def\flushbottom{%
1429   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1430   \let\@texttop\relax}

```

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

```

1431 \setlength\marginparsep{\columnsep}
1432 \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_EX 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^AT_EX 2_ε (`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。

```

1433 \setlength{\oddsidemargin}{\paperwidth}
1434 \addtolength{\oddsidemargin}{-\fullwidth}
1435 \setlength{\oddsidemargin}{.5\oddsidemargin}
1436 \iftombow
1437   \addtolength{\oddsidemargin}{-1in}
1438 \else
1439   \addtolength{\oddsidemargin}{-\inv@mag in}
1440 \fi
1441 \setlength{\evensidemargin}{\oddsidemargin}
1442 \if@mparswitch
1443   \addtolength{\evensidemargin}{\fullwidth}
1444   \addtolength{\evensidemargin}{-\textwidth}
1445 \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。

```
1446 \setlength\marginparwidth{\paperwidth}
1447 \addtolength\marginparwidth{-\oddsidemargin}
1448 \addtolength\marginparwidth{-\inv@mag in}
1449 \addtolength\marginparwidth{-\textwidth}
1450 \addtolength\marginparwidth{-10\jsc@mmm}
1451 \addtolength\marginparwidth{-\marginparsep}
1452 \@tempdima=1zw
1453 \divide\marginparwidth\@tempdima
1454 \multiply\marginparwidth\@tempdima
```

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

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

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

```
1455 \setlength\topmargin{\paperheight}
1456 \addtolength\topmargin{-\textheight}
1457 \if@slide
1458   \addtolength\topmargin{-\headheight}
1459 \else
1460   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
      \headheight (2003-06-26)
1461 \fi
1462 \addtolength\topmargin{-\headsep}
1463 \addtolength\topmargin{-\footskip}
1464 \setlength\topmargin{0.5\topmargin}
1465 %<kiyou>\setlength\topmargin{81truebp}
1466 \iftombow
1467   \addtolength\topmargin{-1in}
1468 \else
1469   \addtolength\topmargin{-\inv@mag in}
1470 \fi
1471 %</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。

```
1472 \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。

```
1473 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

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

```
1474 \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。

```
1475 \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。

```
1476 \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。

```
1477 \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。

```
1478 \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。

```
1479 \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。

```
1480 \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。

```
1481 \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。

```
1482 \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。

```
1483 \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。

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

```
1485 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1486 \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 1487 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1488 \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 1489 \setlength\@fptop{0\p@? \@plus 1fil}
```

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

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

`\@dblfpsep` 段抜きフロートについての値です。

```
\@dblfpsep 1492 \setlength\@dblfpsep{0\p@? \@plus 1fil}
```

```
\@dblfpbot 1493 \setlength\@dblfpsep{8\p@? \@plus 2fil}
```

```
1494 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

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

`\pltx@cleartooddpage`

`\pltx@cleartoevenpage`

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

となっています。

```
1495 %\def\pltx@cleartorightpage{\clearpage\if@twoside
```

```
1496 % \ifodd\c@page
```

```
1497 % \iftdir
```

```
1498 % \hbox{}\thispagestyle{empty}\newpage
```

```
1499 % \if@twocolumn\hbox{}\newpage\fi
```

```

1500 % \fi
1501 % \else
1502 % \ifdir
1503 % \hbox{}\thispagestyle{empty}\newpage
1504 % \if@twocolumn\hbox{}\newpage\fi
1505 % \fi
1506 % \fi\fi}
1507 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1508 % \ifodd\c@page
1509 % \ifdir
1510 % \hbox{}\thispagestyle{empty}\newpage
1511 % \if@twocolumn\hbox{}\newpage\fi
1512 % \fi
1513 % \else
1514 % \iftdir
1515 % \hbox{}\thispagestyle{empty}\newpage
1516 % \if@twocolumn\hbox{}\newpage\fi
1517 % \fi
1518 % \fi\fi}
1519 \def\pltx@cleartooddpage{\clearpage\if@twoside
1520 \ifodd\c@page\else
1521 \hbox{}\thispagestyle{empty}\newpage
1522 \if@twocolumn\hbox{}\newpage\fi
1523 \fi\fi}
1524 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1525 \ifodd\c@page
1526 \hbox{}\thispagestyle{empty}\newpage
1527 \if@twocolumn\hbox{}\newpage\fi
1528 \fi\fi}

```

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

```

1529 \let\pltx@cleartorightpage\pltx@cleartooddpage
1530 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

`\cleardoublepage` [2017-02-24] コミュニティ版 pL^AT_EX 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。

```

1531 %<*book|report>
1532 \if@openleft
1533 \let\cleardoublepage\pltx@cleartoleftpage
1534 \else\if@openright
1535 \let\cleardoublepage\pltx@cleartorightpage
1536 \fi\fi
1537 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

`\@evenhead` `\@oddhead`, `\@oddfoot`, `\@evenhead`, `\@evenfoot` は偶数・奇数ページの柱 (ヘッダ, フッタ) を出力する命令です。これらは `\fullwidth` 幅の `\hbox` の中で呼び出されます。
`\@evenfoot` `\ps@...` の中で定義しておきます。

`\@oddfoot` 柱の内容は、`\chapter` が呼び出す `\chaptermark{何々}`, `\section` が呼び出す `\sectionmark{何々}` で設定します。柱を扱う命令には次のものがあります。

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

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

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

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

```
1538 % \def\ps@empty{%
1539 %   \let\mkboth\gobbletwo
1540 %   \let\oddhead\empty
1541 %   \let\oddfoot\empty
1542 %   \let\evenhead\empty
1543 %   \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。

```
1544 \def\ps@plainfoot{%
1545   \let\mkboth\gobbletwo
1546   \let\oddhead\empty
1547   \def\oddfoot{\normalfont\hfil\thepage\hfil}%
1548   \let\evenhead\empty
```

```

1549 \let\@evenfoot\@oddfoot}
1550 \def\ps@plainhead{%
1551 \let\@mkboth\@gobbletwo
1552 \let\@oddfoot\@empty
1553 \let\@evenfoot\@empty
1554 \def\@evenhead{%
1555 \ifmparswitch \hss \fi
1556 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1557 \ifmparswitch\else \hss \fi}%
1558 \def\@oddhead{%
1559 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1560 %<book>\let\ps@plain\ps@plainhead
1561 %<!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。

```

1562 %<*article|slide>
1563 \if@twoside
1564 \def\ps@headings{%
1565 \let\@oddfoot\@empty
1566 \let\@evenfoot\@empty
1567 \def\@evenhead{\ifmparswitch \hss \fi
1568 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1569 \ifmparswitch\else \hss \fi}%
1570 \def\@oddhead{%
1571 \underline{%
1572 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1573 \let\@mkboth\markboth
1574 \def\sectionmark##1{\markboth{%
1575 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1576 ##1}{}}%
1577 \def\subsectionmark##1{\markright{%
1578 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1579 ##1}}%
1580 }
1581 \else % if not twoside
1582 \def\ps@headings{%
1583 \let\@oddfoot\@empty
1584 \def\@oddhead{%
1585 \underline{%
1586 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1587 \let\@mkboth\markboth
1588 \def\sectionmark##1{\markright{%
1589 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1590 ##1}}%
1591 \fi
1592 %</article|slide>

```

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

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

```
1593 %<*book|report>
1594 \def\bxjs@maybe@autoxspacing{%
1595   \ifx\autoxspacing\undefined\else \autoxspacing \fi}
1596 \newif\if@omit@number
1597 \def\ps@headings{%
1598   \let\@oddfoot\@empty
1599   \let\@evenfoot\@empty
1600   \def\@evenhead{%
1601     \if@mparswitch \hss \fi
1602     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1603       \textbf{\thepage}\hfil\leftmark}}}%
1604     \if@mparswitch\else \hss \fi}%
1605   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1606     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1607   \let\@mkboth\markboth
1608   \def\chaptermark##1{\markboth{%
1609     \ifnum \c@secnumdepth >\m@ne
1610       \if@mainmatter
1611         \if@omit@number\else
1612           \@chapapp\thechapter\@chappos\hskip1\jsZw
1613         \fi
1614       \fi
1615     \fi
1616     ##1}{}}}%
1617   \def\sectionmark##1{\markright{%
1618     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1619     ##1}}}%
1620 %</book|report>
```

最後は学会誌の場合です。

```
1621 %<*jspf>
1622 \def\ps@headings{%
1623   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1624   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1625   \def\@oddhead{\normalfont\hfil \@title \hfil}
1626   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1627 %</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。

```

1628 \def\ps@myheadings{%
1629   \let\@oddfont\@empty\let\@evenfont\@empty
1630   \def\@evenhead{%
1631     \if@mparswitch \hss \fi%
1632     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1633     \if@mparswitch\else \hss \fi}%
1634   \def\@oddhead{%
1635     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1636   \let\@mkboth\@gobbletwo
1637 %<book|report> \let\chaptermark\@gobble
1638 \let\sectionmark\@gobble
1639 %<!book&!report> \let\subsectionmark\@gobble
1640 }

```

8 文書のマークアップ

8.1 表題

`\title` これらは L^AT_EX 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 1641 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1642 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1643 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1644 % \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。

```

1645 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1646 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1647 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1648 \AtBeginDocument{\bxjs@decl@subtitle}
1649 \def\bxjs@decl@subtitle{%
1650   \global\let\bxjs@decl@subtitle\relax
1651   \ifx\subtitle\@undefined
1652     \global\let\subtitle\jsSubtitle
1653   \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。

```

1654 \def\bxjs@annihilate@subtitle{%

```



```

1655 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1656 \global\let\jsSubtitle\relax}

```

`\etitle` 某学会誌スタイルで使う英語のタイトル, 英語の著者名, キーワード, メールアドレスです。

```

\author 1657 %<*jspf>
1658 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
\keywords 1659 \newcommand*{\author}[1]{\gdef\@author{#1}}
1660 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1661 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1662 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1663 %</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。

```

1664 \def\plainifnotempty{%
1665   \ifx \@oddhead \@empty
1666     \ifx \@oddfoot \@empty
1667       \else
1668         \thispagestyle{plainfoot}%
1669       \fi
1670     \else
1671       \thispagestyle{plainhead}%
1672     \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。

```

1673 %<*article|book|report|slide>
1674 \if@titlepage
1675   \newcommand{\maketitle}{%
1676     \begin{titlepage}%
1677       \let\footnotesize\small
1678       \let\footnoterule\relax
1679       \let\footnote\thanks
1680       \null\vfil
1681       \if@slide
1682         {\footnotesize \@date}%
1683       \begin{center}
1684         \mbox{} \ \ [1\jsZw]
1685         \large

```

```

1686         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1687         \jsc@smallskip
1688         \@title
1689         \ifx\bxjs@subtitle\@undefined\else
1690             \par\vskip\z@
1691             {\small \bxjs@subtitle\par}
1692         \fi
1693         \jsc@smallskip
1694         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1695         \vfill
1696         {\small \@author}%
1697     \end{center}
1698 \else
1699 \vskip 60\p@?
1700 \begin{center}%
1701     {\LARGE \@title \par}%
1702     \ifx\bxjs@subtitle\@undefined\else
1703         \vskip5\p@?
1704         {\normalsize \bxjs@subtitle\par}
1705     \fi
1706     \vskip 3em%
1707     {\large
1708         \lineskip .75em
1709         \begin{tabular}[t]{c}%
1710             \@author
1711         \end{tabular}\par}%
1712     \vskip 1.5em
1713     {\large \@date \par}%
1714 \end{center}%
1715 \fi
1716 \par
1717 \@thanks\vfil\null
1718 \end{titlepage}%
1719 \setcounter{footnote}{0}%
1720 \global\let\thanks\relax
1721 \global\let\maketitle\relax
1722 \global\let\@thanks\@empty
1723 \global\let\@author\@empty
1724 \global\let\@date\@empty
1725 \global\let\@title\@empty
1726 \global\let\title\relax
1727 \global\let\author\relax
1728 \global\let\date\relax
1729 \global\let\and\relax
1730 \bxjs@annihilate@subtitle
1731 }%
1732 \else
1733 \newcommand{\maketitle}{\par
1734 \begingroup

```

```

1735 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1736 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1737 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1738 \parindent 1\jsZw\noindent
1739 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1740 \if@twocolumn
1741 \ifnum \col@number=\@ne
1742 \maketitle
1743 \else
1744 \twocolumn[\maketitle]%
1745 \fi
1746 \else
1747 \newpage
1748 \global\@topnum\z@ % Prevents figures from going at top of page.
1749 \maketitle
1750 \fi
1751 \plainifnotempty
1752 \@thanks
1753 \endgroup
1754 \setcounter{footnote}{0}%
1755 \global\let\thanks\relax
1756 \global\let\maketitle\relax
1757 \global\let\@thanks\@empty
1758 \global\let\@author\@empty
1759 \global\let\@date\@empty
1760 \global\let\@title\@empty
1761 \global\let\title\relax
1762 \global\let\author\relax
1763 \global\let\date\relax
1764 \global\let\and\relax
1765 \bxjs@annihilate@subtitle
1766 }

```

\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。

```

1767 \def\maketitle{%
1768 \newpage\null
1769 \vskip 2em
1770 \begin{center}%
1771 \let\footnote\thanks
1772 {\LARGE \@title \par}%
1773 \ifx\bxjs@subtitle\@undefined\else
1774 \vskip3\p@?
1775 {\normalsize \bxjs@subtitle\par}
1776 \fi
1777 \vskip 1.5em
1778 {\large
1779 \lineskip .5em
1780 \begin{tabular}[t]{c}%
1781 \author

```

```

1782         \end{tabular}\par}%
1783     \vskip 1em
1784     {\large \@date}%
1785     \end{center}%
1786     \par\vskip 1.5em
1787 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1788 }
1789 \fi
1790 %</article|book|report|slide>
1791 %<*jspf>
1792 \newcommand{\maketitle}{\par
1793     \begin{group}
1794         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1795         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1796         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1797             \parindent 1\jsZw\noindent
1798             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1799         \twocolumn[\@maketitle]%
1800         \plainifnotempty
1801         \@thanks
1802     \end{group}
1803     \setcounter{footnote}{0}%
1804     \global\let\thanks\relax
1805     \global\let\maketitle\relax
1806     \global\let\@thanks\@empty
1807     \global\let\@author\@empty
1808     \global\let\@date\@empty
1809 % \global\let\@title\@empty % \@title は柱に使う
1810     \global\let\title\relax
1811     \global\let\author\relax
1812     \global\let\date\relax
1813     \global\let\and\relax
1814     \ifx\authors@mail\@undefined\else{%
1815         \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1816         \footnotetext[0]{\itshape\authors@mail}%
1817     }\fi
1818     \global\let\authors@mail\@undefined}
1819 \def\@maketitle{%
1820     \newpage\null
1821     \vskip 6em % used to be 2em
1822     \begin{center}
1823         \let\footnote\thanks
1824         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1825         \lineskip .5em
1826         \ifx\@author\@undefined\else
1827             \vskip 1em
1828             \begin{tabular}[t]{c}%
1829                 \@author
1830             \end{tabular}\par

```

```

1831 \fi
1832 \ifx\@etitle\@undefined\else
1833 \vskip 1em
1834 {\large \@etitle \par}%
1835 \fi
1836 \ifx\@eauthor\@undefined\else
1837 \vskip 1em
1838 \begin{tabular}[t]{c}%
1839 \@eauthor
1840 \end{tabular}\par
1841 \fi
1842 \vskip 1em
1843 \@date
1844 \end{center}
1845 \vskip 1.5em
1846 \centerline{\box\@abstractbox}
1847 \ifx\@keywords\@undefined\else
1848 \vskip 1.5em
1849 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1850 \fi
1851 \vskip 1.5em}
1852 %</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。

```

1853 \def\bxjs@label@sect#1{%
1854 \expandafter\ifx\csname label#1\endcsname\relax
1855 \csname the#1\endcsname
1856 \else \csname label#1\endcsname
1857 \fi}
1858 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1859 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1860 \def\@secapp{\presectionname}
1861 \def\@secpos{\postsectionname}
1862 \fi

```

`\labelsection` 節番号の出力書式。

```

1863 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1864 \def\labelsection{\@secapp\thesection\@secpos}

```

■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トに 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、完全に行送りがない \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。

```
1866 \def\startsection#1#2#3#4#5#6{%
1867   \if@noskipsec \leavevmode \fi
1868   \par
1869 % 見出し上の空きを \@tempskipa にセットする
1870   \@tempskipa #4\relax
1871 % \@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
1872   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1873 % 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
1874   \ifdim \@tempskipa <\z@
1875     \@tempskipa -\@tempskipa \@afterindentfalse
1876   \fi
1877   \if@nobreak
1878 %   \everypar{\everyparhook}% これは間違い
1879     \everypar{}%
1880   \else
1881     \addpenalty\@secpenalty
1882 % 次の行は削除
```

```

1883 % \addvspace\@tempskipa
1884 % 次の \noindent まで追加
1885 \ifdim \@tempskipa >\z@
1886 \if@slide\else
1887 \null
1888 \vspace*{-\baselineskip}%
1889 \fi
1890 \vskip\@tempskipa
1891 \fi
1892 \fi
1893 \noindent
1894 % 追加終わり
1895 \@ifstar
1896 {\@ssect{#3}{#4}{#5}{#6}}%
1897 {\@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、多少変えています。 \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。

```

1898 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1899 \let\bxjs@if@ceph\@firstofone
1900 \else \let\bxjs@if@ceph\@gobble
1901 \fi

1902 \def\@sect#1#2#3#4#5#6[#7]#8{%
1903 \ifnum #2>\c@secnumdepth
1904 \let\@svsec\@empty
1905 \else
1906 \refstepcounter{#1}%
1907 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1908 \fi
1909 % 見出し後の空きを \@tempskipa にセット
1910 \@tempskipa #5\relax
1911 % 条件判断の順序を入れ替えました
1912 \ifdim \@tempskipa<\z@
1913 \def\@svsechd{%
1914 #6{\hskip #3\relax
1915 \@svsec #8}%
1916 \csname #1mark\endcsname{#7}}%
1917 \addcontentsline{toc}{#1}{%
1918 \ifnum #2>\c@secnumdepth \else
1919 \protect\numberline{\bxjs@label@sect{#1}}%
1920 \fi
1921 #7}}% 目次にフルネームを載せるなら #8
1922 \else

```

```

1923 \begingroup
1924 \interlinepenalty \@M % 下から移動
1925 #6{%
1926 \hangfrom{\hskip #3\relax\@svsec}%
1927 % \interlinepenalty \@M % 上に移動
1928 #8\@@par}%
1929 \endgroup
1930 \csname #1mark\endcsname{#7}%
1931 \addcontentsline{toc}{#1}{%
1932 \ifnum #2>\c@secnumdepth \else
1933 \protect\numberline{\bxjs@label@sect{#1}}%
1934 \fi
1935 #7}% 目次にフルネームを載せるならここは #8
1936 \fi
1937 \@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。

```

1938 \def\@xsect#1{%
1939 % 見出しの後ろの空きを \@tempkipa にセット
1940 \@tempkipa #1\relax
1941 % 条件判断の順序を変えました
1942 \ifdim \@tempkipa<\z@
1943 \@nobreakfalse
1944 \global\@noskipsecttrue
1945 \everypar{%
1946 \if@noskipsec
1947 \global\@noskipsecfalse
1948 {\setbox\z@\lastbox}%
1949 \clubpenalty\@M
1950 \begingroup \@svsechd \endgroup
1951 \unskip
1952 \@tempkipa #1\relax
1953 \hskip -\@tempkipa
1954 \else
1955 \clubpenalty \@clubpenalty
1956 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1957 \fi\bxjs@if@ceph\everyparhook}%
1958 \else
1959 \par \nobreak
1960 \vskip \@tempkipa
1961 \@afterheading
1962 \fi
1963 \if@slide
1964 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1965 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt

```



```

1966 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1967 \fi
1968 \par % 2000-12-18
1969 \ignorespaces}
1970 \def\@ssect#1#2#3#4#5{%
1971 \@tempskipa #3\relax
1972 \ifdim \@tempskipa<\z@
1973 \def\@svsechd{#4{\hskip #1\relax #5}}%
1974 \else
1975 \begingroup
1976 #4{%
1977 \@hangfrom{\hskip #1}%
1978 \interlinepenalty \@M #5\@@par}%
1979 \endgroup
1980 \fi
1981 \@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^AT_EX 本体で
`\sectionmark` 定義済みです。

```

\subsectionmark 1982 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1983 % \newcommand*\sectionmark[1]{}
1984 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1985 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1986 % \newcommand*\paragraphmark[1]{}
1987 % \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。

```

1988 %<!book&!report>\setcounter{secnumdepth}{3}
1989 %<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 1990 \newcounter{part}
\c@subsubsection 1991 %<book|report>\newcounter{chapter}
1992 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1993 %<!book&!report>\newcounter{section}
\c@subparagraph 1994 \newcounter{subsection}[section]
1995 \newcounter{subsubsection}[subsection]
1996 \newcounter{paragraph}[subsubsection]
1997 \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。

`\thesection`

`\thesubsection`

`\thesubsubsection`

`\theparagraph`

`\thesubparagraph`

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

1998 \renewcommand{\thepart}{\@Roman\c@part}
1999 %<*!book&!report>
2000 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2001 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2002 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2003 \else
2004 \renewcommand{\thesection}{\@arabic\c@section}
2005 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2006 \fi
2007 %</!book&!report>
2008 %<*book|report>
2009 \renewcommand{\thechapter}{\@arabic\c@chapter}
2010 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2011 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2012 %</book|report>
2013 \renewcommand{\thesubsubsection}{\%
2014   \thesubsection.\@arabic\c@subsubsection}
2015 \renewcommand{\theparagraph}{\%
2016   \thesubsubsection.\@arabic\c@paragraph}
2017 \renewcommand{\thesubparagraph}{\%
2018   \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。

```

2019 %<book|report>\newcommand{\@chapapp}{\prechaptername}

```

```

2020 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

[2017-03-05] `\frontmatter` と `\mainmatter` の2つの命令は, 改丁または改ページした後で `\pagenumbering{...}` でノンブルを1にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが, これではノンブルをリセットする際に偶奇逆転が起こる場合がありました。 `openany` かどうかに関らず奇数ページまで繰るように修正することで, 問題を解消しました。実は, L^AT_EX の標準クラスでは1998年に修正され

ていた問題です（コミュニティ版 pL^AT_EX の標準クラス 2017/03/05 も参照）。

```
2021 %<*book|report>
2022 \newcommand\frontmatter{%
2023   \pltx@cleartooddpage
2024   \@mainmatterfalse
2025   \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。

```
2026 \newcommand\mainmatter{%
2027   \pltx@cleartooddpage
2028   \@mainmattertrue
2029   \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。

```
2030 \newcommand\backmatter{%
2031   \if@openleft
2032     \cleardoublepage
2033   \else\if@openright
2034     \cleardoublepage
2035   \else
2036     \clearpage
2037   \fi\fi
2038   \@mainmatterfalse}
2039 %</book|report>
```

■部

`\part` 新しい部を始めます。

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

`\secdef{星なし}{星あり}`

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

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

まず `book` と `report` のクラス以外です。

```
2040 %<*&!book&!report>
2041 \newcommand\part{%
2042   \if@noskipsec \leavevmode \fi
2043   \par
2044   \addvspace{4ex}%
2045   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2046   \secdef\@part\@spart}
2047 %</!book&!report>
```

book および report クラスの場合は、少し複雑です。

```
2048 %<*book|report>
2049 \newcommand\part{%
2050   \if@openleft
2051     \cleardoublepage
2052   \else\if@openright
2053     \cleardoublepage
2054   \else
2055     \clearpage
2056   \fi\fi
2057   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2058   \if@twocolumn
2059     \onecolumn
2060     \@restonecoltrue
2061   \else
2062     \@restonecolfalse
2063   \fi
2064   \null\vfil
2065   \secdef\@part\@spart}
2066 %</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。

```
2067 %<!*book&!report>
2068 \def\@part[#1]#2{%
2069   \ifnum \c@secnumdepth >\m@ne
2070     \refstepcounter{part}%
2071     \addcontentsline{toc}{part}{%
2072       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2073   \else
2074     \addcontentsline{toc}{part}{#1}%
2075   \fi
2076   \markboth{}{}%
2077   {\parindent\z@
2078     \raggedright
2079     \interlinepenalty \@M
2080     \normalfont
2081     \ifnum \c@secnumdepth >\m@ne
2082       \Large\headfont\prepartname\thepart\postpartname
2083       \par\nobreak
2084     \fi
2085     \huge \headfont #2%
2086     \markboth{}{}\par}%
2087   \nobreak
2088   \vskip 3ex
2089   \@afterheading}
2090 %</!*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。

```
2091 %<*book|report>
2092 \def\@part[#1]#2{%
2093   \ifnum \c@secnumdepth >-2\relax
2094     \refstepcounter{part}%
2095     \addcontentsline{toc}{part}{%
2096       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2097   \else
2098     \addcontentsline{toc}{part}{#1}%
2099   \fi
2100   \markboth{}{}%
2101   {\centering
2102     \interlinepenalty \@M
2103     \normalfont
2104     \ifnum \c@secnumdepth >-2\relax
2105       \huge\headfont \prepartname\thepart\postpartname
2106       \par\vskip20\p@?
2107     \fi
2108     \Huge \headfont #2\par}%
2109   \@endpart}
2110 %</book|report>
```

\@spart 番号を付けない部です。

```
2111 %<*!book&!report>
2112 \def\@spart#1{%
2113   \parindent \z@ \raggedright
2114   \interlinepenalty \@M
2115   \normalfont
2116   \huge \headfont #1\par}%
2117   \nobreak
2118   \vskip 3ex
2119   \@afterheading}
2120 %</!book&!report>
2121 %<*book|report>
2122 \def\@spart#1{%
2123   \centering
2124   \interlinepenalty \@M
2125   \normalfont
2126   \Huge \headfont #1\par}%
2127   \@endpart}
2128 %</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^AT_EX では classes.dtx v1.4b (2000/05/19) で修正されています。

```
2129 %<*book|report>
```

```

2130 \def\@endpart{\vfil\newpage
2131   \if@twoside
2132     \if@openleft %% added (2017/02/24)
2133       \null\thispagestyle{empty}\newpage
2134     \else\if@openright %% added (2016/12/13)
2135       \null\thispagestyle{empty}\newpage
2136     \fi\fi %% added (2016/12/13, 2017/02/24)
2137   \fi
2138   \if@restonecol
2139     \twocolumn
2140   \fi}
2141 %</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。

```

2142 %<*book|report>
2143 \newcommand{\chapter}{%
2144   \if@openleft\cleardoublepage\else
2145     \if@openright\cleardoublepage\else\clearpage\fi\fi
2146   \plainifnotempty % 元: \thispagestyle{plain}
2147   \global\@topnum\z@
2148   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2149   \secdef
2150     {\@omit@numberfalse\@chapter}%
2151     {\@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。

```

2152 \def\@chapter[#1]#2{%
2153   \ifnum \c@secnumdepth >\m@ne
2154     \if@mainmatter
2155       \refstepcounter{chapter}%
2156       \typeout{\@chapapp\thechapter\@chappos}%
2157       \addcontentsline{toc}{chapter}%
2158         {\protect\numberline
2159 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2160         {\@chapapp\thechapter\@chappos}%
2161         #1}%
2162     \else\addcontentsline{toc}{chapter}{#1}\fi
2163   \else
2164     \addcontentsline{toc}{chapter}{#1}%
2165   \fi
2166   \chaptermark{#1}%
2167   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}%
2168   \addtocontents{lot}{\protect\addvspace{10\jsc@empt}}%
2169   \if@twocolumn

```

```

2170 \topnewpage[\@makechapterhead{#2}]\%
2171 \else
2172 \makechapterhead{#2}\%
2173 \afterheading
2174 \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。

```

2175 \def\@makechapterhead#1{%
2176 \vspace*{2\Cvs}\% 欧文は 50pt
2177 {\parindent \z@ \raggedright \normalfont
2178 \ifnum \c@secnumdepth >\m@ne
2179 \if@mainmatter
2180 \huge\headfont \@chapapp\thechapter\@chappos
2181 \par\nobreak
2182 \vskip \Cvs \% 欧文は 20pt
2183 \fi
2184 \fi
2185 \interlinepenalty\@M
2186 \Huge \headfont #1\par\nobreak
2187 \vskip 3\Cvs}} \% 欧文は 40pt

```

`\@chapter` `\chapter*{...}` コマンドの本体です。`\chaptermark` を補いました。

```

2188 \def\@chapter#1{%
2189 \chaptermark{#1}\%
2190 \if@twocolumn
2191 \topnewpage[\@makeschapterhead{#1}]\%
2192 \else
2193 \makeschapterhead{#1}\@afterheading
2194 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2195 \def\@makeschapterhead#1{%
2196 \vspace*{2\Cvs}\% 欧文は 50pt
2197 {\parindent \z@ \raggedright
2198 \normalfont
2199 \interlinepenalty\@M
2200 \Huge \headfont #1\par\nobreak
2201 \vskip 3\Cvs}} \% 欧文は 40pt
2202 %</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。

```

2203 \if@twocolumn
2204 \newcommand{\section}{\%
2205 %<jspf>\ifx\maketitle\relax\else\maketitle\fi

```

```

2206 \startsection{section}{1}{\z@}%
2207 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2208 %<kiyou> {\Cvs}{0.5\Cvs}%
2209 % {\normalfont\large\headfont\@secapp}}
2210 {\normalfont\large\headfont\raggedright}}
2211 \else
2212 \newcommand{\section}{%
2213 \if@slide\clearpage\fi
2214 \startsection{section}{1}{\z@}%
2215 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2216 {.5\Cvs \@plus.3\Cdp}% 後アキ
2217 % {\normalfont\Large\headfont\@secapp}}
2218 {\normalfont\Large\headfont\raggedright}}
2219 \fi

```

`\subsection` 同上です。

```

2220 \if@twocolumn
2221 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2222 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2223 {\normalfont\normalsize\headfont}}
2224 \else
2225 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2226 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2227 {.5\Cvs \@plus.3\Cdp}% 後アキ
2228 {\normalfont\large\headfont}}
2229 \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)。

```

2230 \if@twocolumn
2231 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2232 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2233 {\normalfont\normalsize\headfont}}
2234 \else
2235 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2236 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2237 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2238 {\normalfont\normalsize\headfont}}
2239 \fi

```

`\paragraph` 見出しの後ろで改行されません。

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

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

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

※ BXJS クラスでは、1.1 版 [2016-02-14] から `\jsParagraphMark` をサポートしている。
 段落のマーク (■) が必ず和文フォントで出力されるようにする。
`\jsJaChar` は standard 和文ドライバが読み込まれた場合は `\jachar` と同義になるが、
 それ以外は何もしない。

```

2240 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2241 \let\bxjs@paragraph@mark\jsParagraphMark
2242 \ifx\bxjs@paragraph@mark\@empty
2243   \let\jsParagraphMark\@empty
2244 \else\ifx\bxjs@paragraph@mark\undefined\else
2245   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2246 \fi\fi
2247 \let\jsJaChar\@empty
2248 \if@twocolumn
2249   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2250     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2251     <jspf> {\normalfont\normalsize\headfont}}
2252   <!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2253 \else
2254   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2255     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2256     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2257     <jspf> {\normalfont\normalsize\headfont}}
2258   <!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2259 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2260 \if@twocolumn
2261   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2262     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2263     {\normalfont\normalsize\headfont}}
2264 \else
2265   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2266     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2267     {\normalfont\normalsize\headfont}}
2268 \fi

```

8.3 リスト環境

第 k レベルのリストの初期化するのが `\@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。

```
2269 \if@slide
2270   \setlength\leftmargini{1\jsZw}
2271 \else
2272   \if@twocolumn
2273     \setlength\leftmargini{2\jsZw}
2274   \else
2275     \setlength\leftmargini{3\jsZw}
2276   \fi
2277 \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 2278 \if@slide
\leftmarginv 2279   \setlength\leftmarginii {1\jsZw}
2280   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2281   \setlength\leftmarginiv {1\jsZw}
2282   \setlength\leftmarginv {1\jsZw}
2283   \setlength\leftmarginvi {1\jsZw}
2284 \else
2285   \setlength\leftmarginii {2\jsZw}
2286   \setlength\leftmarginiii{2\jsZw}
2287   \setlength\leftmarginiv {2\jsZw}
2288   \setlength\leftmarginv {1\jsZw}
2289   \setlength\leftmarginvi {1\jsZw}
2290 \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。

```
2291 \setlength \labelsep {0.5\jsZw} % .5em
2292 \setlength \labelwidth{\leftmargini}
2293 \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。

```
2294 \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 2295 \@beginparpenalty -\@lowpenalty
\@itempenalty 2296 \@endparpenalty -\@lowpenalty
2297 \@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。

```
2298 \def\@listi{\leftmargin\leftmargini
2299   \parsep \z@
2300   \topsep 0.5\baselineskip
2301   \itemsep \z@ \relax}
2302 \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）。

```
2303 \@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 2304 \def\@listii{\leftmargin\leftmarginii
\@listiv 2305   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2306   \topsep \z@
\@listv 2307   \parsep \z@
\@listvi 2308   \itemsep\parsep}
2309 \def\@listiii{\leftmargin\leftmarginiii
2310   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2311   \topsep \z@
2312   \parsep \z@
2313   \itemsep\parsep}
2314 \def\@listiv {\leftmargin\leftmarginiv
2315   \labelwidth\leftmarginiv
2316   \advance\labelwidth-\labelsep}
2317 \def\@listv {\leftmargin\leftmarginv
2318   \labelwidth\leftmarginv
2319   \advance\labelwidth-\labelsep}
2320 \def\@listvi {\leftmargin\leftmarginvi
2321   \labelwidth\leftmarginvi
2322   \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^AT_EX 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。

```
2323 \renewcommand{\theenumi}{\@arabic\c@enumi}
2324 \renewcommand{\theenumii}{\@alph\c@enumii}
2325 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2326 \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。

```
2327 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2328 \newcommand*\jsInJaParen[1]{%
2329   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2330 \newcommand\labelenumi{\theenumi.}
2331 \newcommand\labelenumii{\jsInJaParen{ (\theenumii) }}
2332 \newcommand\labelenumiii{\theenumiii.}
2333 \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
`\p@enumiii` 式です。これも第 2 レベルは和文用かっこにしました。

```
\p@enumiv 2334 \renewcommand\p@enumii{\theenumi}
2335 \renewcommand\p@enumiii{\theenumi\jsInhibitGlue (\theenumii) }
2336 \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 2337 \newcommand\labelitemi{\textbullet}
2338 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 2339 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2340 \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してしまいます。これを解決した新しい `description` の実装です。

```
2341 \newenvironment{description}{%
2342   \list{}{%
2343     \labelwidth=\leftmargin
2344     \labelsep=1\jsZw
2345     \advance \labelwidth by -\labelsep
2346     \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。

```
2347 \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。

```

2348 %<*book|report>
2349 \newenvironment{chapterabstract}{%
2350   \begin{list}{}{%
2351     \listparindent=1\jsZw
2352     \itemindent=\listparindent
2353     \rightmargin=0pt
2354     \leftmargin=5\jsZw}\item[]{}{\end{list}\vspace{\baselineskip}}
2355 %</book|report>

```

“普通の” abstract 環境の定義。

```

2356 %<*article|report|slide>
2357 \newbox\@abstractbox
2358 \if@titlepage
2359   \newenvironment{abstract}{%
2360     \titlepage
2361     \null\vfil
2362     \@beginparpenalty\@lowpenalty
2363     \begin{center}%
2364       \headfont \abstractname
2365       \@endparpenalty\@M
2366     \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。

```

2367   \par}%
2368   {\par\vfil\null\endtitlepage}
2369 \else
2370   \newenvironment{abstract}{%
2371     \if@twocolumn
2372       \ifx\maketitle\relax
2373         \section*{\abstractname}%
2374       \else
2375         \global\setbox\@abstractbox\hbox\bgroup

```

```

2376 \begin{minipage}[b]{\textwidth}
2377 \small\parindent1\jsZw
2378 \begin{center}%
2379 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2380 \end{center}%
2381 \list{}{%
2382 \listparindent\parindent
2383 \itemindent \listparindent
2384 \rightmargin \leftmargin}%
2385 \item\relax
2386 \fi
2387 \else
2388 \small
2389 \begin{center}%
2390 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2391 \end{center}%
2392 \list{}{%
2393 \listparindent\parindent
2394 \itemindent \listparindent
2395 \rightmargin \leftmargin}%
2396 \item\relax
2397 \fi}{\if@twocolumn
2398 \ifx\maketitle\relax
2399 \else
2400 \endlist\end{minipage}\egroup
2401 \fi
2402 \else
2403 \endlist
2404 \fi}
2405 \fi
2406 %</article|report|slide>
2407 %<*jspf>
2408 \newbox\@abstractbox
2409 \newenvironment{abstract}{%
2410 \global\setbox\@abstractbox\hbox\bgroup
2411 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2412 \small
2413 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2414 {\end{minipage}\egroup}
2415 %</jspf>

```

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

```

2416 %<*book|report>
2417 \ifbxjs@force@chapterabstract
2418 \let\abstract\chapterabstract
2419 \let\endabstract\endchapterabstract
2420 \fi
2421 %</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。

```
2422 %<*jspf>
2423 %\newbox\@keywordsbox
2424 %\newenvironment{keywords}{%
2425 % \global\setbox\@keywordsbox\hbox\bgroup
2426 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2427 % \small\parindent0\jsZw}%
2428 % {\end{minipage}\egroup}
2429 %</jspf>
```

■verse 環境

verse 詩のための `verse` 環境です。

```
2430 \newenvironment{verse}{%
2431 \let \=\@centercr
2432 \list{}{%
2433 \itemsep \z@
2434 \itemindent -2\jsZw % 元: -1.5em
2435 \listparindent\itemindent
2436 \rightmargin \z@
2437 \advance\leftmargin 2\jsZw}% 元: 1.5em
2438 \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。

```
2439 \newenvironment{quotation}{%
2440 \list{}{%
2441 \listparindent\parindent
2442 \itemindent\listparindent
2443 \rightmargin \z@}%
2444 \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。

```
2445 \newenvironment{quote}%
2446 {\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。

```
2447 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2448   \item[\hskip \labelsep{\headfont #1\ #2}]}
2449 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2450   \item[\hskip \labelsep{\headfont #1\ #2 (#3)}]}
```

`titlepage` タイトルを独立のページに出力するのに使われます。

[2017-02-24] コミュニティ版 pL^AT_EX の標準クラス 2017/02/15 に合わせて、book クラスでタイトルを必ず奇数ページに送るようにしました。といっても、横組クラスしかありませんでしたので、従来の挙動は何も変わっていません。また、book 以外の場合のページ番号のリセットもコミュニティ版 pL^AT_EX 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。

```
2451 \newenvironment{titlepage}{%
2452   %<book>   \pltx@cleartooddpage %% 2017-02-24
2453   \if@twocolumn
2454     \@restonecoltrue\onecolumn
2455   \else
2456     \@restonecolfalse\newpage
2457   \fi
2458   \thispagestyle{empty}%
2459   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2460   24
2461 }%
2462 {\if@restonecol\twocolumn \else \newpage \fi
2463   \if@twoside\else
2464     \setcounter{page}\@ne
2465   \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2465 %<*&book&!report>
2466 \newcommand{\appendix}{\par
2467   \setcounter{section}{0}%
2468   \setcounter{subsection}{0}%
2469   \ifnum\bxjs@label@section=\bxjs@label@section@compat
2470     \gdef\presectionname{\appendixname}%
2471     \gdef\postsectionname{}}%
2472 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
```



```

2473 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2474 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2475 \else
2476 \gdef\@secapp{\appendixname}%
2477 \gdef\@secpos{}%
2478 \gdef\thesection{\@Alph\c@section}%
2479 \fi}
2480 %</!book&!report>
2481 %<*book|report>
2482 \newcommand{\appendix}{\par
2483 \setcounter{chapter}{0}%
2484 \setcounter{section}{0}%
2485 \gdef\@chapapp{\appendixname}%
2486 \gdef\@chappos{}%
2487 \gdef\thechapter{\@Alph\c@chapter}}
2488 %</book|report>

```

8.4 パラメータの設定

■array と tabular 環境

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

```
2489 \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。

```
2490 \setlength\tabcolsep{6\p@?}
```

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

```
2491 \setlength\arrayrulewidth{.4\p@}
```

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

```
2492 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2493 \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。

```
2494 \skip\@mpfootins = \skip\footins
```

■framebox 環境

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

`\fboxrule` `\fbox`, `\framebox` の罫線の幅です。

```
2495 \setlength\fboxsep{3\p@?}
2496 \setlength\fboxrule{.4\p@}
```

■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

```
2497 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2498 %<*book|report>
2499 \@addtoreset{equation}{chapter}
2500 \renewcommand\theequation
2501 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2502 %</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。

```
2503 % \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。

```
2504 % \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。

```
2505 % \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。

```
2506 %<*!book&!report>
2507 \newcounter{figure}
2508 \renewcommand \thefigure {\@arabic\c@figure}
```

```

2509 %</*!book&!report>
2510 %<*book|report>
2511 \newcounter{figure}[chapter]
2512 \renewcommand \thefigure
2513     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2514 %</book|report>

```

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

```

\ext@figure 2515 \def\fps@figure{tbp}
2516 \def\ftype@figure{1}
\fnun@figure 2517 \def\ext@figure{lof}
2518 \def\fnun@figure{\figurename\nobreak\thefigure}

```

figure * 形式は段抜きのフロートです。

```

figure* 2519 \newenvironment{figure}%
2520     {\@float{figure}}%
2521     {\end@float}
2522 \newenvironment{figure*}%
2523     {\@dblfloat{figure}}%
2524     {\end@dblfloat}

```

■table 環境

\c@table 表番号カウンタと表番号を出力するコマンドです。アスキー版では \thechapter. が
\thetable \thechapter{} になっていますが、ここではオリジナルのままにしています。

```

2525 %</*!book&!report>
2526 \newcounter{table}
2527 \renewcommand\thetable{\@arabic\c@table}
2528 %</*!book&!report>
2529 %<*book|report>
2530 \newcounter{table}[chapter]
2531 \renewcommand \thetable
2532     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2533 %</book|report>

```

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

```

\ext@table 2534 \def\fps@table{tbp}
2535 \def\ftype@table{2}
\fnun@table 2536 \def\ext@table{lot}
2537 \def\fnun@table{\tablename\nobreak\thetable}

```

table * は段抜きのフロートです。

```

table* 2538 \newenvironment{table}%
2539     {\@float{table}}%
2540     {\end@float}
2541 \newenvironment{table*}%

```

```

2542          {\@dblfloat{table}}}%
2543          {\end@dblfloat}

```

8.6 キャプション

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

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

```

2544 \newlength\abovecaptionskip
2545 \newlength\belowcaptionskip
2546 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2547 \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。

```

2548 %<*.jspf>
2549 % \long\def\@makecaption#1#2{{\small
2550 %   \advance\leftskip10\jsc@mmm
2551 %   \advance\rightskip10\jsc@mmm
2552 %   \vskip\abovecaptionskip
2553 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2554 %   \ifdim \wd\@tempboxa >\hsize
2555 %     #1\hskip1\jsZw\relax #2\par
2556 %   \else
2557 %     \global \@minipagefalse
2558 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2559 %   \fi
2560 %   \vskip\belowcaptionskip}}
2561 \long\def\@makecaption#1#2{{\small
2562   \advance\leftskip .0628\linewidth
2563   \advance\rightskip .0628\linewidth
2564   \vskip\abovecaptionskip
2565   \sbox\@tempboxa{#1\zwspace#2}%
2566   \ifdim \wd\@tempboxa <\hsize \centering \fi
2567   #1\zwspace#2\par
2568   \vskip\belowcaptionskip}}
2569 %</!.jspf>
2570 %<*.jspf>
2571 \long\def\@makecaption#1#2{%
2572   \vskip\abovecaptionskip
2573   \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2574   \ifdim \wd\@tempboxa >\hsize
2575     {\small\sffamily

```

```

2576 \list{#1}{%
2577 \renewcommand{\makelabel}[1]{##1\hfil}
2578 \itemsep \z@
2579 \itemindent \z@
2580 \labelsep \z@
2581 \labelwidth 11\jsc@mmm
2582 \listparindent\z@
2583 \leftmargin 11\jsc@mmm\item\relax #2\endlist}
2584 \else
2585 \global \@minipagefalse
2586 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2587 \fi
2588 \vskip\belowcaptionskip}
2589 %</jspf>

```

9 フォントコマンド

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

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

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

```

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

```

```

2590 \newif\if@jsc@warnoldfontcmd
2591 \@jsc@warnoldfontcmdtrue
2592 \newif\if@jsc@warnoldfontcmdexception
2593 \@jsc@warnoldfontcmdexceptionfalse

```

`\jsc@DeclareOldFontCommand`

```

2594 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2595 \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2596 \DeclareOldFontCommand{#1}{%
2597 \bxjs@oldfontcmd{#1}#2%
2598 }{%
2599 \bxjs@oldfontcmd{#1}#3%
2600 }%
2601 }
2602 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%

```

```

2603 \ClassInfo\bxjs@clsname
2604 {Old font command '\string#1' is used!!\MessageBreak
2605   The first occurrence is}%
2606 }

```

`\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。

```

2607 \newcommand*{\allowoldfontcommands}{%
2608   \@jsc@warnoldfontcmdexceptiontrue}
2609 \newcommand*{\disallowoldfontcommands}{%
2610   \@jsc@warnoldfontcmdexceptionfalse}

2611 \let\bxjs@oldfontcmd@list\@empty
2612 \def\bxjs@oldfontcmd#1{%
2613   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2614 \def\bxjs@oldfontcmd@a#1#2{%
2615   \if@jsc@warnoldfontcmdexception\else
2616     \global\@jsc@warnoldfontcmdfalse
2617     \ifx#1\relax
2618       \global\let#1=t%
2619       \jsc@warnoldfontcmd{#2}%
2620     \fi
2621   \fi}
2622 \def\bxjs@warnoldfontcmd@final{%
2623 % \par
2624 \global\let\bxjs@warnoldfontcmd@final\@empty
2625 \let\@tempa\@empty
2626 \def\do##1{%
2627   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2628     \edef\@tempa{\@tempa \space\string##1}\fi}
2629 \bxjs@oldfontcmd@list
2630 \ifx\@tempa\@empty\else
2631   \ClassWarningNoLine\bxjs@clsname
2632   {Some old font commands were used in text:\MessageBreak
2633     \space\@tempa\MessageBreak
2634     You should note, that since 1994 LaTeX2e provides a\MessageBreak
2635     new font selection scheme called NFSS2 with several\MessageBreak
2636     new, combinable font commands. The
2637     class provides\MessageBreak
2638     the old font commands only for compatibility}
2639 \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。

```

2640 \def\bxjs@warnoldfontcmd@kick@final{%
2641   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2642 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2643 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2644 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2645 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2646 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2647 \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。

```

2648 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2649 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2650 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2651 \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 2652 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2653 \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。

```
2654 \newcommand\@pnumwidth{1.55em}
2655 \newcommand\@tocrmarg{2.55em}
2656 \newcommand\@dotsep{4.5}
2657 %<!book&!report>\setcounter{tocdepth}{2}
2658 %<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)

```
2659 \newdimen\jsc@tocl@width
2660 \newcommand{\tableofcontents}{%
2661 %<*book|report>
2662   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2663   \settowidth\@tempdima{\headfont\appendixname}%
2664   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2665   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2666   \if@twocolumn
2667     \@restonecoltrue\onecolumn
2668   \else
2669     \@restonecolfalse
2670   \fi
2671   \chapter*{\contentsname}%
2672   \@mkboth{\contentsname}{}%
2673 %</book|report>
2674 %<*&!book&!report>
```



```

2675 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2676 \settowidth\@tempdima{\headfont\appendixname}%
2677 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2678 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2679 \section*{\contentsname}%
2680 \@mkboth{\contentsname}{\contentsname}%
2681 %<!/book&!report>
2682 \@starttoc{toc}%
2683 %<book|report> \if@restonecol\twocolumn\fi
2684 }

```

\l@part 部の目次です。

```

2685 \newcommand*{\l@part}[2]{%
2686 \ifnum \c@tocdepth >-2\relax
2687 %<!book&!report> \addpenalty\@secpenalty
2688 %<book|report> \addpenalty{-\@highpenalty}%
2689 \addvspace{2.25em \@plus\p@}%
2690 \begingroup
2691 \parindent \z@
2692 % \@pnumwidth should be \@tocrmarg
2693 % \rightskip \@pnumwidth
2694 \rightskip \@tocrmarg
2695 \parfillskip -\rightskip
2696 {\leavevmode
2697 \large \headfont
2698 \setlength\@lnumwidth{4\jsZw}%
2699 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2700 \nobreak
2701 %<book|report> \global\@nobreaktrue
2702 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2703 \endgroup
2704 \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)

```

2705 %<*book|report>
2706 \newcommand*{\l@chapter}[2]{%
2707 \ifnum \c@tocdepth >\m@ne
2708 \addpenalty{-\@highpenalty}%
2709 \addvspace{1.0em \@plus\p@}
2710 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2711 \begingroup
2712 \parindent\z@
2713 % \rightskip\@pnumwidth
2714 \rightskip\@tocrmarg
2715 \parfillskip-\rightskip
2716 \leavevmode\headfont
2717 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi

```

```

2718      \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2719      \advance\leftskip\@lnumwidth \hskip-\leftskip
2720      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2721      \penalty\@highpenalty
2722      \endgroup
2723      \fi}
2724 %</book|report>

```

\l@section 節の目次です。

```

2725 %<*&book&!report>
2726 \newcommand*{\l@section}[2]{%
2727   \ifnum \c@tocdepth >\z@
2728     \addpenalty{\@secpenalty}%
2729     \addvspace{1.0em \@plus\p@}%
2730     \begingroup
2731     \parindent\z@
2732 %     \rightskip\@pnumwidth
2733     \rightskip\@tocrmarg
2734     \parfillskip-\rightskip
2735     \leavevmode\headfont
2736 %     % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2737     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2738     \advance\leftskip\@lnumwidth \hskip-\leftskip
2739     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2740     \endgroup
2741     \fi}
2742 %</!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。

```

2743 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

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

\l@subsubsection しれません。

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

```

\l@subparagraph 2744 %<*&book&!report>
2745 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2746 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2747 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2748 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2749 %
2750 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2751 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2752 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2753 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2754 %
2755 \newcommand*{\l@subsection}{%
2756   \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw

```

```

2757         \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2758 \newcommand*\l@subsubsection}{%
2759         \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2760         \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2761 \newcommand*\l@paragraph}{%
2762         \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2763         \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2764 \newcommand*\l@subparagraph}{%
2765         \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2766         \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2767 %</!book&!report>
2768 %<*book|report>
2769 % \newcommand*\l@section}{\@dottedtocline{2}{3.8em}{3.2em}}
2770 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2771 % \newcommand*\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
2772 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
2773 \newcommand*\l@section}{%
2774         \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2775         \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2776 \newcommand*\l@subsubsection}{%
2777         \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2778         \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2779 \newcommand*\l@subsubsection}{%
2780         \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2781         \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2782 \newcommand*\l@paragraph}{%
2783         \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2784         \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2785 \newcommand*\l@subparagraph}{%
2786         \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2787         \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2788 %</book|report>

```

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

```

2789 \newdimen\@lnumwidth
2790 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

\@dottedtocline L^AT_EX 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。

```

2791 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2792   \vskip \z@ \@plus.2\p@?
2793   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2794     \parindent #2\relax\@afterindenttrue
2795     \interlinepenalty\@M
2796     \leavevmode

```

```

2797 \lnumwidth #3\relax
2798 \advance\leftskip \lnumwidth \null\nobreak\hskip -\leftskip
2799 {#4}\nobreak
2800 \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2801 mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2802 \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2803 \newcommand{\listoffigures}{%
2804 %<*book|report>
2805 \if@twocolumn\@restonecoltrue\onecolumn
2806 \else\@restonecolfalse\fi
2807 \chapter*{\listfigurename}%
2808 \@mkboth{\listfigurename}{}%
2809 %</book|report>
2810 %<*&!book&!report>
2811 \section*{\listfigurename}%
2812 \@mkboth{\listfigurename}{\listfigurename}%
2813 %</&!book&!report>
2814 \@starttoc{lof}%
2815 %<book|report> \if@restonecol\twocolumn\fi
2816 }

```

`\l@figure` 図目次の項目を出力します。

```

2817 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsw}{3.683\jsw}}

```

`\listoftables` 表目次を出力します。

```

2818 \newcommand{\listoftables}{%
2819 %<*book|report>
2820 \if@twocolumn\@restonecoltrue\onecolumn
2821 \else\@restonecolfalse\fi
2822 \chapter*{\listtablename}%
2823 \@mkboth{\listtablename}{}%
2824 %</book|report>
2825 %<*&!book&!report>
2826 \section*{\listtablename}%
2827 \@mkboth{\listtablename}{\listtablename}%
2828 %</&!book&!report>
2829 \@starttoc{lot}%
2830 %<book|report> \if@restonecol\twocolumn\fi
2831 }

```

`\l@table` 表目次は図目次と同じです。

```

2832 \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。

```
2833 \newdimen\bibindent
2834 \setlength\bibindent{2\jsZw}
```

`thebibliography` 参考文献リストを出力します。

[2016-07-16] L^AT_EX 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。

```
2835 \newenvironment{thebibliography}[1]{%
2836   \@jsc@warnoldfontcmdexceptiontrue
2837   \global\let\presectionname\relax
2838   \global\let\postsectionname\relax
2839   %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2840   %<*kiyou>
2841   \vspace{1.5\baselineskip}
2842   \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2843   \vspace{0.5\baselineskip}
2844   %</kiyou>
2845   %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2846   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2847   \list{\@biblabel{\@arabic\c@enumiv}}%
2848     {\settowidth\labelwidth{\@biblabel{#1}}%
2849     \leftmargin\labelwidth
2850     \advance\leftmargin\labelsep
2851     \@openbib@code
2852     \usecounter{enumiv}%
2853     \let\p@enumiv\@empty
2854     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2855   %<kiyou> \small
2856   \sloppy
2857   \clubpenalty4000
2858   \@clubpenalty\clubpenalty
2859   \widowpenalty4000%
2860   \sfcode`\.\@m}
2861   {\def\@noitemerr
2862     {\@latex@warning{Empty `thebibliography' environment}}}%
2863   \endlist}
```

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

```
2864 \newcommand{\newblock}{\hspace .11em\@plus.33em\@minus.07em}
```

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

```
2865 \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。

```
2866 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

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

```
2867 % \def\@citex[#1]#2{\leavevmode
2868 %   \let\@citea\@empty
2869 %   \@cite{\@for\@citeb:=#2\do
2870 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m }%
2871 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2872 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2873 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2874 %     \G@refundefinedtrue
2875 %     \@latex@warning
2876 %       {Citation `'\@citeb' on page \thepage \space undefined}}%
2877 %     {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2878 % \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。

```
2879 % \DeclareRobustCommand\cite{\unskip
2880 %   \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}]
2881 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2882 %   , \jsInhibitGlue\ #2\fi}})}$}
```

10.3 索引

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

```
2883 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2884   \if@twocolumn
2885     \onecolumn\@restonecolfalse
2886   \else
2887     \clearpage\@restonecoltrue
2888   \fi
2889   \columnseprule.4pt \columnsep 2\jsZw
2890   \ifx\multicols\@undefined
2891 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}]%
2892 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2893 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2894 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2895   \else
```

```

2896 \ifdim\textwidth<\fullwidth
2897 \setlength{\evensidemargin}{\oddsidemargin}
2898 \setlength{\textwidth}{\fullwidth}
2899 \setlength{\linewidth}{\fullwidth}
2900 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}]%
2901 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]%
2902 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2903 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]%
2904 \else
2905 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}]%
2906 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]%
2907 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2908 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]%
2909 \fi
2910 \fi
2911 %<book|report> \@mkboth{\indexname}{}%
2912 %<!book&!report> \@mkboth{\indexname}{\indexname}%
2913 \plainifnotempty % \thispagestyle{plain}
2914 \parindent\z@
2915 \parskip\z@ \@plus .3\p@?\relax
2916 \let\item\@idxitem
2917 \raggedright
2918 \footnotesize\narrowbaselines
2919 }{
2920 \ifx\multicols\@undefined
2921 \if@restonecol\onecolumn\fi
2922 \else
2923 \end{multicols}
2924 \fi
2925 \clearpage
2926 }

```

`\@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 2927 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2928 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2929 \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。

```

2930 \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。

```

2931 \newcommand\seename{\if@english see\else →\fi}
2932 \newcommand\alsoname{\if@english see also\else →\fi}

```

10.4 脚注

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

```
2933 \ifx\pltx@foot@penalty\@undefined
2934   \let\footnotes@ve=\footnote
2935   \def\footnote{\jsInhibitGlue\footnotes@ve}
2936   \let\footnotemarks@ve=\footnotemark
2937   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2938 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

[2013-04-23] 新しい pT_EX 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^AT_EX の変更に従いました (Thanks: 角藤さん)。pL^AT_EX 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_EX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
2939 \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。

```
2940 % \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。


```

2941 \renewcommand{\footnoterule}{%
2942   \kern-2.6\p@? \kern-.4\p@
2943   \hrule width .4\columnwidth
2944   \kern 2.6\p@?}

```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```

2945 %<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_EX and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9)

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

[2016-09-08] コミュニティ版 p_ΛT_EX のバグ修正に追随しました。

[2016-11-29] 古い p_ΛT_EX 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。

```

2946 \long\def\@footnotetext{%
2947   \insert\footins\bgroup
2948     \normalfont\footnotesize
2949     \interlinepenalty\interfootnotelinepenalty
2950     \splittopskip\footnotesep
2951     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2952     \hsize\columnwidth \@parboxrestore
2953     \protected@edef\@currentlabel{%
2954       \csname p@footnote\endcsname\@thefnmark
2955     }%
2956     \color@begingroup
2957       \@makefnmark
2958       \rule{z@\footnotesep\ignorespaces}%
2959       \futurelet\jsc@next\jsc@fo@t}
2960 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@f@t
2961               \else \let\jsc@next\jsc@f@t\fi \jsc@next}
2962 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
2963 \def\jsc@f@t#1{#1\jsc@@foot}
2964 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
2965   \ifx\pltx@foot@penalty\undefined\else
2966     \ifhmode\null\fi
2967     \ifnum\pltx@foot@penalty=z@\else
2968       \penalty\pltx@foot@penalty
2969       \pltx@foot@penalty\z@
2970     \fi
2971   \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。

```

2972 \newcommand\@makefnmark[1]{%
2973   \advance\leftskip 3\jsZw

```

```

2974 \parindent 1\jsZw
2975 \noindent
2976 \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。

```

2977 % \def\@xfootnotenext[#1]{%
2978 %   \begingroup
2979 %     \ifnum#1>\z@
2980 %       \csname c@\@mpfn\endcsname #1\relax
2981 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2982 %     \else
2983 %       \unrestored@protected@xdef\@thefnmark{}%
2984 %     \fi
2985 %   \endgroup
2986 %   \@footnotetext}

```

ここまでのコードは JS クラスを踏襲する。

11 段落の頭へのグルー挿入禁止

段落頭のかぎカッコなどを見かけ 1 字半下げから全角 1 字下げに直します。

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

```

2987 \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。

```

2988 \def\everyparhook{\jsInhibitGlueAtParTop}
2989 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2990 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
2991 \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] すぐ上の変更に `\@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る危険性が高いので、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。

```
2992 \ifx j\jsEngine
2993 \def\@inhibitglue{%
2994   \futurelet\@let@token\@inhibitglue}
2995 \begingroup
2996 \let\GDEF=\gdef
2997 \let\CATCODE=\catcode
2998 \let\ENDGROUP=\endgroup
2999 \CATCODE`k=12
3000 \CATCODE`a=12
3001 \CATCODE`n=12
3002 \CATCODE`j=12
3003 \CATCODE`i=12
3004 \CATCODE`c=12
3005 \CATCODE`h=12
3006 \CATCODE`r=12
3007 \CATCODE`t=12
3008 \CATCODE`e=12
3009 \GDEF\KANJI@CHARACTER{kanji character }
3010 \ENDGROUP
3011 \def\@inhibitglue{%
3012   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3013   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3014   \def\jsc@ig@temp{#1}%
3015   \ifx\jsc@ig@temp\@empty
3016     \ifnum\the\inhibitxspcode`#2=2\relax
3017       \inhibitglue
3018     \fi
3019   \fi}
3020 \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。

```
3021 \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。

```
3022 \def\@doendpe{%
3023   \@endpetrue
3024   \def\par{%
3025     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3026   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] minipage 環境にも対策します。

```
3027 \def\@setminipage{%
3028   \@minipagetrue
3029   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3030 }
```

\item 命令の直後です。

```
3031 \def\@item[#1]{%
3032   \if@noparitem
3033     \@donoparitem
3034   \else
3035     \if@inlabel
3036       \indent \par
3037     \fi
3038     \ifhmode
3039       \unskip\unskip \par
3040     \fi
3041     \if@newlist
3042       \if@nobreak
3043         \@nbitem
3044       \else
3045         \addpenalty\@beginparpenalty
3046         \addvspace\@topsep
3047         \addvspace{-\parskip}%
3048       \fi
3049     \else
3050       \addpenalty\@itempenalty
3051       \addvspace\itemsep
3052     \fi
3053     \global\@inlabeltrue
3054   \fi
3055   \everypar{%
3056     \@minipagefalse
3057     \global\@newlistfalse
3058     \if@inlabel
3059       \global\@inlabelfalse
3060     {\setbox\z@\lastbox
3061     \ifvoid\z@
3062       \kern-\itemindent
3063     \fi}%
3064     \box\@labels
3065     \penalty\z@
```

```

3066 \fi
3067 \if@nobreak
3068 \clubpenalty \clubpenaltyfalse
3069 \clubpenalty \@M
3070 \else
3071 \clubpenalty \@clubpenalty
3072 \everypar{\everyparhook}%
3073 \fi
3074 \everyparhook}%
3075 \if@noitemarg
3076 \noitemargfalse
3077 \if@nmbrlist
3078 \refstepcounter\listctr
3079 \fi
3080 \fi
3081 \sbox\@tempboxa{\makelabel{#1}}%
3082 \global\setbox\@labels\hbox{%
3083 \unhbox\@labels
3084 \hskip \itemindent
3085 \hskip -\labelwidth
3086 \hskip -\labelsep
3087 \ifdim \wd\@tempboxa >\labelwidth
3088 \box\@tempboxa
3089 \else
3090 \hbox to\labelwidth {\unhbox\@tempboxa}%
3091 \fi
3092 \hskip \labelsep}%
3093 \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。

```

3094 \def\@afterheading{%
3095 \@nobreaktrue
3096 \everypar{%
3097 \if@nobreak
3098 \clubpenalty \clubpenaltyfalse
3099 \clubpenalty \@M
3100 \if@afterindent \else
3101 {\setbox\z@\lastbox}%
3102 \fi
3103 \else
3104 \clubpenalty \@clubpenalty
3105 \everypar{\everyparhook}%
3106 \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

```

3107 \fi

```

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

※ `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。

```
3108 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3109 \AtEndOfPackage{%
3110 \def\@gnewline #1{%
3111   \ifvmode
3112     \@nolnerr
3113   \else
3114     \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
3115     \jsInhibitGlue \ignorespaces
3116   \fi}
3117 }
3118 \fi
```

12 いろいろなロゴ

$\text{L}\text{A}\text{T}\text{E}\text{X}$ 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。

```
3119 \if@jslogo
3120   \IfFileExists{jslogo.sty}{%
3121     \RequirePackage{jslogo}%
3122   }{%
3123     \ClassWarningNoLine\bxjs@clsname
3124       {The package 'jslogo' is not installed.\MessageBreak
3125        It is included in the recent release of\MessageBreak
3126        the 'jsclasses' bundle}
3127   }
3128 \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^AT_EX の `\ProvidesFile` `\ProvidesFile` で悪さをする例が F_TE_X で報告されています。これを避けるための tDB さんのフィックスを挿入しておきます。副作用がありましたらお知らせください。

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

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

```
3129 %\let\ltx@ifnextchar\@ifnextchar
3130 %\def\ProvidesFile#1{%
3131 % \begingroup
3132 % \catcode`\ 10 %
3133 % \ifnum \endlinechar<256 %
3134 % \ifnum \endlinechar>\m@ne
3135 % \catcode\endlinechar 10 %
3136 % \fi
3137 % \fi
3138 % \@makeoother\/%
3139 % \@makeoother\&%
3140 % \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]}
```

14 初期設定

■いろいろな語

```
\prepartname
\postpartname 3141 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3142 \newcommand{\postpartname}{\if@english\else 部\fi}
3143 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3144 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3145 \newcommand{\presectionname}{}% 第
\postsectionname 3146 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3147 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
3148 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
\listtablename 3149 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3150 \newcommand{\refname}{\if@english References\else 参考文献\fi}
3151 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
\indexname 3152 \newcommand{\indexname}{\if@english Index\else 索引\fi}
```

```

\figurename
\tablename 3153 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3154 %<jspf>\newcommand{\figurename}{Fig.~}
3155 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3156 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3157 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3158 \newcommand{\appendixname}{\if@english \else 付録\fi}
3159 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

■今日の日付 L^AT_EX 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。その場合、和暦表記は 1970 年から平成の終わりまでの期間でのみサポートする。
 ※新元号が公表されたら対応する。

```

3160 \onlypreamble\bxjs@decl@Seireki@cmds
3161 \@tempswafalse
3162 \if p\jsEngine \@tempswattrue \fi
3163 \if n\jsEngine \@tempswattrue \fi
3164 \if@tempswa \expandafter\@firstoftwo
3165 \else \expandafter\@secondoftwo
3166 \fi
3167 {%
3168 % 欧文 8bitTeX の場合
3169 \newif\ifjsSeireki \jsSeirekitrue
3170 \def\bxjs@decl@Seireki@cmds{%
3171 \def\西暦{\jsSeirekitrue}%
3172 \def\和暦{\jsSeirekifalse}}
3173 \def\Seireki{\jsSeirekitrue}
3174 \def\Wareki{\jsSeirekifalse}
3175 \def\bxjs@if@use@seireki{%
3176 \ifjsSeireki \expandafter\@firstoftwo
3177 \else \expandafter\@secondoftwo \fi}
3178 }{%
3179 \newif\if 西暦 \西暦 true
3180 \def\bxjs@decl@Seireki@cmds{%
3181 \def\西暦{\西暦 true}%
3182 \def\和暦{\西暦 false}}
3183 \def\Seireki{\西暦 true}
3184 \def\Wareki{\西暦 false}
3185 \def\bxjs@if@use@seireki{%
3186 \if 西暦 \expandafter\@firstoftwo

```



```

3187 \else \expandafter\@secondoftwo \fi}
3188 }
3189 \bxjs@decl@Seireki@cmds
3190 \let\bxjs@unxp\@firstofone \let\bxjs@onxp\@firstofone
3191 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded
3192 \def\bxjs@onxp{\unexpanded\expandafter}}
3193 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3194 \def\bxjs@iai{\noexpand~}
3195 \else \def\bxjs@iai{}
3196 \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。

※一般の年数の整数レジスタは \bxjs@jayear で、\heisei はこれの別名になる。

```

3197 \newcount\bxjs@jayear
3198 \let\bxjs@gengo\@empty \bxjs@jayear\year
3199 \@tempcnta=\the\year\two@digits\month\two@digits\day\relax
3200 \ifnum\@tempcnta<19700101
3201 \else\ifnum\@tempcnta<19890108
3202 \def\bxjs@gengo{昭和}\advance\bxjs@jayear-1925\relax
3203 \let\heisei\bxjs@jayear
3204 \else\ifnum\@tempcnta<20190501
3205 \def\bxjs@gengo{平成}\advance\bxjs@jayear-1988\relax
3206 \let\heisei\bxjs@jayear
3207 \else

```

現在が新元号である場合。bxwareki とかムニャムニャとかを試してみる。

```

3208 \IfFileExists{bxwareki.sty}{\IfFileExists{bxwareki2019.def}{%
3209 \RequirePackage{bxwareki}[]%
3210 \let\bxjs@gengo\WarekiGengo
3211 }{}{}%
3212 \if l\jsEngine \ifx\bxjs@gengo\@empty
3213 \toks@{}
3214 \directlua{pcall(function()
3215 tex.toks[0] = bxjs.get_new_gengo()
3216 end)}}
3217 \edef\bxjs@gengo{\the\toks@}
3218 \fi\fi

```

新元号が既知の場合にのみ、和暦の年を求める。

```

3219 \ifx\bxjs@gengo\@empty\else \advance\bxjs@jayear-2018\relax \fi
3220 \fi\fi\fi
3221 \edef\jayear{%
3222 \ifx\bxjs@gengo\@empty \the\bxjs@jayear\bxjs@iai
3223 \else \bxjs@onxp{\bxjs@gengo}%
3224 \ifnum\bxjs@jayear=\@ne \bxjs@unxp{元}%
3225 \else \bxjs@iai\the\bxjs@jayear\bxjs@iai
3226 \fi
3227 \fi}

```

`\today` 英語、西暦、和暦で場合分けをする。

```
3228 \edef\bxjs@today{%
3229   \if@english
3230     \ifcase\month\or
3231       January\or February\or March\or April\or May\or June\or
3232       July\or August\or September\or October\or November\or December\fi
3233     \space\number\day, \number\year
3234   \else
3235     \noexpand\bxjs@if@use@seireki{%
3236       \number\year\bxjs@iai\bxjs@unxp{年}%
3237       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3238       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3239     }{%
3240       \bxjs@onxp{\jyear}\bxjs@unxp{年}%
3241       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3242       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3243     }%
3244   \fi}
3245 \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。

```
3246 \g@addto@macro\bxjs@begin@document@hook{%
3247   \ifx\bb1@jpn@Seirekitrue\@undefined\else
3248     \bxjs@decl@Seireki@cmds
3249     \g@addto@macro\datejapanese{%
3250       \let\today\bxjs@today}%
3251   \fi}
```

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

```
3252 \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。

```
3253 %<slide>\pagestyle{empty}%
3254 %<article|report>\pagestyle{plain}%
3255 %<book>\pagestyle{headings}%
3256 \pagenumbering{arabic}
3257 \if@twocolumn
3258   \twocolumn
3259   \sloppy
3260   \flushbottom
3261 \else
3262   \onecolumn
```

```
3263 \raggedbottom
3264 \fi
3265 %<*slide>
3266 \renewcommand\familydefault{\sfdefault}
3267 \raggedright
3268 %</slide>
```

■ BXJS 独自の追加処理

フックを実行する。

```
3269 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3270 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

```
3271 %</cls>
```

以上です。

付録 A 和文ドライバの仕様

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

- `\jsDocClass` [文字トークンの `let`] 文書クラスの種類を示し、次のいずれかと一致する (`\if` で判定可能)。
 - `\jsArticle` `bxjsarticle` クラス
 - `\jsBook` `bxjsbook` クラス
 - `\jsReport` `bxjsreport` クラス
 - `\jsSlide` `bxjsslide` クラス
- `\jsEngine` [文字トークンの `let`] 使用されているエンジンの種別。 (`\if` で判定可能)。
 - `p` `pdfTeX` (DVI モードも含む)
 - `l` `LuaTeX` (＃)
 - `x` `XYTeX`
 - `j` `pTeX` または `upTeX`
 - `n` 以上の何れでもない
- `\ifjsWithupTeX` [スイッチ] 使用されているエンジンが `upTeX` であるか。
- `\ifjsWitheTeX` [スイッチ] 使用されているエンジンが ϵ -`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。

```
3272 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

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

B.1 補助マクロ

```
3273 %<*minimal>
```

```
3274 %% 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.
```

```
3275 \def\DeclareJaTextFontCommand#1#2{%
```

```
3276   \DeclareRobustCommand#1[1]{%
```

```
3277     \relax
```

```
3278     \ifmmode \expandafter\nfss@text \fi
```

```
3279     {#2##1}}%
```

```
3280 }
```

```
\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。
```

```
3281 \def\DeclareJaMathFontCommand#1#2{%
```

```
3282   \DeclareRobustCommand#1[1]{%
```

```
3283     \relax
```

```
3284     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3285     \nfss@text{\fontfamily\familydefault
```

```
3286       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3287       #2##1}}%
```

```

3288 }%
3289 }

```

`\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。

```

3290 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3291 \@onlypreamble\bxjs@if@sf@default
3292 \def\bxjs@if@sf@default#1{%
3293   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3294   \g@addto@macro\bxjs@begin@document@hook{%
3295     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3296 }

```

`\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。

```

3297 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3298 \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。

```

3299 \def\jsLetHeadChar#1#2{%
3300   \begingroup
3301     \escapechar=`\ %
3302     \let\bxjs@tmpa={% brace-match-hack
3303       \bxjs@let@hchar@exp#2}%
3304   \endgroup
3305   \let#1\bxjs@g@tmpa}
3306 \def\bxjs@let@hchar@exp{%
3307   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3308 \def\bxjs@let@hchar@exp@a{%
3309   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
3310     \bxjs@let@hchar@out\let\relax
3311   }\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi{% 空白
3312     \bxjs@let@hchar@out\let\space%
3313   }\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
3314     \bxjs@let@hchar@out\let\@backslashchar
3315   }\bxjs@let@hchar@exp@b}}}%
3316 \def\bxjs@let@hchar@exp@b#1{%
3317   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3318 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3319   %\message{<#1#2>}%
3320   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴

```

```

3321 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3322 \bxjs@let@hchar@out\let\relax
3323 }{%else
3324 \expandafter\bxjs@let@hchar@exp
3325 }%
3326 }{%else
3327 \bxjs@let@hchar@chr#1%
3328 }}
3329 \def\bxjs@let@hchar@chr#1{%
3330 \bxjs@let@hchar@out\def{{#1}}}%
3331 \def\bxjs@let@hchar@out#1#2{%
3332 \global#1\bxjs@g@tmpa#2\relax
3333 \toks@{bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3334 \chardef\bxjs@let@hchar@csta=128
3335 \chardef\bxjs@let@hchar@cstb=192
3336 \chardef\bxjs@let@hchar@cstc=224
3337 \chardef\bxjs@let@hchar@cstd=240
3338 \chardef\bxjs@let@hchar@cste=248
3339 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3340 \def\bxjs@let@hchar@chr@ue#1{%
3341 \@tempcnta=`#1\relax
3342 %\message{\the\@tempcnta}%
3343 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3344 \bxjs@let@hchar@chr@ue@a#1%
3345 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3346 \bxjs@let@hchar@out\let\relax
3347 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3348 \bxjs@let@hchar@chr@ue@b
3349 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3350 \bxjs@let@hchar@chr@ue@c
3351 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3352 \bxjs@let@hchar@chr@ue@d
3353 }{%else
3354 \bxjs@let@hchar@out\let\relax
3355 }}}}}}
3356 \def\bxjs@let@hchar@chr@ue@a#1{%
3357 \bxjs@let@hchar@out\def{{#1}}}%
3358 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3359 \bxjs@let@hchar@out\def{{#1#2}}}%
3360 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3361 \bxjs@let@hchar@out\def{{#1#2#3}}}%
3362 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3363 \bxjs@let@hchar@out\def{{#1#2#3#4}}}%

```

B.2 (u)pTeX 用の設定

```
3364 \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。

```
3365 \def\bxjs@let@hchar@chr@pp#1#2{%
3366   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3367 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3368   \%message{(\meaning#3:\meaning#4)}%
3369   \bxjs@cond\if#1k\fi{%
3370     \bxjs@let@hchar@out\def{#{#4}}%
3371   }{%else
3372     \bxjs@let@hchar@chr@ue#3#4%
3373   }%
3374 \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。

```
3375 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3376 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3377 \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。

```
3378 \@onlypreamble\bxjs@declarefontshape
3379 \ifjsWithupTeX
3380 \def\bxjs@declarefontshape{%
3381   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3382   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3383   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3384   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3385 }
3386 \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。

```
3387 \else
3388 \def\bxjs@declarefontshape{%
3389   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3390   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3391   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```



```

3392 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3393 }
3394 \def\bxjs@sizereference{jis}
3395 \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。

```

3396 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3397   \def\bxjs@tmpb{#5}}
3398 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3399 \else
3400 \expandafter\expandafter\expandafter\bxjs@next
3401 \expandafter\string\the\jfont\relax
3402 \fi
3403 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3404                 \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3405   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3406    \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。

```

3407 \begingroup
3408 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3409 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3410 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3411 % 幅が丁度 10pt なら補正は不要
3412 \ifdim\wd\z@=10pt
3413   \global\let\bxjs@scale\jsScale
3414 \else
3415 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3416   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3417   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3418   \bxjs@invscale\@tempdima\bxjs@tmpa
3419   \xdef\bxjs@scale{\strip@pt\@tempdima}
3420 \fi
3421 \endgroup
3422 %\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。

```

3423 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3424 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3425 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3426 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3427 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3428 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3429 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3430 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3431 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3432 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3433 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3434 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3435 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3436 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3437 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}

```

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

```

3438 \DeclareRobustCommand\rmfamily
3439 {\not@math@alphabet\rmfamily\mathrm
3440 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3441 \DeclareRobustCommand\sffamily
3442 {\not@math@alphabet\sffamily\mathsf
3443 \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3444 \DeclareRobustCommand\ttfamily
3445 {\not@math@alphabet\ttfamily\mathtt
3446 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3447 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3448 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3449 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3450 \fi
3451 \bxjs@if@sf@default{%
3452 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3453 \selectfont

```

`\bxjs@parse@qh` の処理は p_TE_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）。

```

3454 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3455 \let\bxjs@parse@qh@a\@undefined
3456 \let\bxjs@parse@qh@b\@undefined

```

■パラメタの設定

```

3457 \prebreakpenalty\jis"2147=10000
3458 \postbreakpenalty\jis"2148=10000
3459 \prebreakpenalty\jis"2149=10000
3460 \inhibitxspcode`!=1
3461 \inhibitxspcode`〒=2
3462 \xspcode`+=3
3463 \xspcode`\%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3464 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3465   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3466 \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。

```

3467 \begingroup
3468 \catcode`\!=0

```

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

```

3469 \gdef\bxjs@ptex@dir{%
3470   !iftdir t%
3471   !else!ifydir y%
3472   !else ?%
3473   !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。

```

3474 % 古い \@makefnmark の定義
3475 \long\def\bxjs@tmpa{\hbox{%
3476   !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
3477   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3478 \ifx\@makefnmark\bxjs@tmpa
3479 \long\gdef\@makefnmark{%
3480   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3481   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3482 \fi
3483 \endgroup

```

B.3 pdfTeX 用の処理

```

3484 \else\ifx p\jsEngine
3485 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3486 \@onlypreamble\bxjs@cjk@loaded
3487 \def\bxjs@cjk@loaded{%
3488   \def\@footnotemark{%
3489     \leavevmode
3490     \ifhmode
3491       \edef\@x@sf{\the\spacefactor}%
3492       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3493         \unkern\unkern
3494         \ifdim\lastskip>\z@ \unskip \fi

```

```

3495     \fi\fi
3496     \nobreak
3497     \fi
3498     \@makefnmark
3499     \ifhmode \spacefactor\@x@sf \fi
3500     \relax}%
3501     \let\bxjs@cjk@loaded\relax
3502 }
3503 \g@addto@macro\bxjs@begin@document@hook{%
3504   \@ifpackageloaded{CJK}{%
3505     \bxjs@cjk@loaded
3506   }{}%
3507 }

```

B.4 X_YTeX 用の処理

```

3508 \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。

```

3509 \def\bxjs@let@hchar@chr#1{%
3510   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3511   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3512     \bxjs@let@hchar@chr@xe
3513   }\bxjs@let@hchar@out\def{{#1}}}}
3514 \def\bxjs@let@hchar@chr@xe#1{%
3515   \lccode`0=`#1\relax
3516   \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3517 \@onlypreamble\bxjs@do@precisetext
3518 \ifx\XeTeXgenerateactualtext\@undefined\else
3519   \def\bxjs@do@precisetext{%
3520     \XeTeXgenerateactualtext=\@ne}
3521 \fi

```

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

```

3522 \@onlypreamble\bxjs@do@simplejasetup
3523 \def\bxjs@do@simplejasetup{%
3524   \ifnum\XeTeXinterchartokenstate>\z@
3525   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3526     \jsSimpleJaSetup
3527     \ClassInfo\bxjs@clsname
3528     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3529   \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

3530 \newcommand*{\jsSimpleJaSetup}{%
3531   \XeTeXlinebreaklocale "ja"\relax
3532   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3533   \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理（エンジン共通）

```
3534 \fi\fi\fi
```

simplejasetup オプションの処理。

```
3535 \ifx\bxjs@do@simplejasetup\undefined\else
3536   \g@addto@macro\bxjs@begin@document@hook{%
3537     \ifbxjs@simplejasetup
3538       \bxjs@do@simplejasetup
3539     \fi}
3540 \fi
```

precisetext オプションの処理。

```
3541 \ifbxjs@precisetext
3542   \ifx\bxjs@do@precisetext\undefined
3543     \ClassWarning\bxjs@clsname
3544       {The current engine does not support the\MessageBreak
3545        'precise-text' option\@gobble}
3546   \else
3547     \bxjs@do@precisetext
3548   \fi
3549 \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。

```
3550 \g@addto@macro\bxjs@begin@document@hook{%
3551   \ifx\jsInhibitGlueAtParTop\empty
3552     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3553     \ifx\everyparhook\bxjs@tmpa
3554       \let\everyparhook\empty
3555     \fi
3556   \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。

```
3557 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern
```

まず `\everypar` を“乗っ取る” 処理を行う。

```
3558 \let\bxjs@everypar\everypar
3559 \newtoks\everypar
3560 \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。

```
3561 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3562 \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。

3563 `\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。

3564 `\@onlypreamble\bxjs@adjust@fancyhdr`

3565 `\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。

3566 `\def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%`

3567 `\def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%`

3568 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

3569 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

3570 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

3571 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

3572 `\def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%`

3573 `\def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%`

3574 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

3575 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

3576 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

3577 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

3578 `\def\bxjs@tmpa{\rm\thepage\strut}%`

3579 `\def\bxjs@tmpb{\thepage\strut}%`

3580 `\ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi`

3581 `\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。

3582 `\ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth`

3583 `\setlength{\@tempdima}{\fullwidth-\textwidth}%`

3584 `\edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%`

3585 `}\bxjs@tmpa`

3586 `\fi\fi`

3587 `\PackageInfo\bxjs{clsname`

3588 `{Patch to fancyhdr is applied\@gobble}}`

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

3589 `\def\bxjs@pagestyle@hook{%`

3590 `\@ifpackageloaded{fancyhdr}{%`

3591 `\bxjs@adjust@fancyhdr`

3592 `\global\let\bxjs@adjust@fancyhdr\relax`

3593 `}{}}`

`\pagestyle` にフックを入れ込む。

3594 `\let\bxjs@org@pagestyle\pagestyle`

```

3595 \def\pagestyle{%
3596   \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。
3597 \g@addto@macro\bxjs@begin@document@hook{%
3598   \bxjs@pagestyle@hook
3599   \global\let\bxjs@pagestyle@hook\relax}

```

■和文空白命令

```

3600 \ifbxjs@jaspace@cmd

\jaenspace 半角幅の水平空き。
3601 \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。
3602 \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。
3603 \def\_ {\zwspace}

\jaspace jlreq クラスと互換の命令。
3604 \DeclareRobustCommand*\jaspace[1]{%
3605   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3606   \ClassError\bxjs@clsname
3607   {Unknown jaspac: #1}{\@eha}%
3608   \else
3609   \csname bxjs@jaspace@@#1\endcsname
3610   \fi}
3611 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3612 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3613 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}

3614 \fi

終わり。
3615 \fi

以上で終わり。
3616 %</minimal>

```

付録 C 和文ドライバ : standard

標準のドライバ。

- \rmfamily/\sffamily/\ttfamily での和文ファミリー連動
- \mcfamily/\gtfamily

- `\textmc/\textgt`
- `\zw`
- `\jQ/\jH`
- `\trueQ/\trueH/\ascQ`
- `\setkanjiskip/\getkanjiskip`
- `\setxkanjiskip/\getxkanjiskip`
- `\autospacing/\noautospacing`
- `\autoxspacing/\noautoxspacing`

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

C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
3617 %<*standard>
3618 %% このファイルは日本語文字を含みます
3619 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3620 \bxjs@simplejasetupfalse
```

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

`\ifbxjs@jp@jismmiv` 2004JIS 字形を優先させるか。

```
3621 \newif\ifbxjs@jp@jismmiv
```

`jis2004` オプションの処理。

```
3622 \bxjs@cslet{bxjs@kv{jis2004@true}}\bxjs@jp@jismmivtrue
3623 \bxjs@cslet{bxjs@kv{jis2004@false}}\bxjs@jp@jismmivfalse
3624 \define@key{bxjsStd}{jis2004}[true]{%
3625   \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。

```
3626 \newif\ifbxjs@jp@units
```

`units` オプションの処理。

```
3627 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3628 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3629 \define@key{bxjsStd}{units}[true]{%
3630   \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。


```
3631 \let\bxjs@jp@font\@empty
```

font オプションの処理。

```
3632 \define@key{bxjsStd}{font}{%
```

```
3633 \edef\bxjs@jp@font{#1}}
```

実際の japaram の値を適用する。

```
3634 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
```

```
3635 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

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

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

```
3636 \ifbxjs@jp@jismmiv
```

```
3637 \g@addto@macro\@classoptionslist{,jis2004}
```

```
3638 % \@ifpackagewith 判定への対策
```

```
3639 \PassOptionsToPackage{jis2004}{otf}
```

```
3640 \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。

```
3641 \if j\jsEngine
```

```
3642 \bxjs@jp@unitsfalse
```

```
3643 \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。

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

```
3645 \IfFileExists{bxcalc.sty}{%
```

```
3646 \RequirePackage{bxcalc}[2018/01/28]%v1.0a
```

```
3647 \ifx\usepTeXunits\undefined
```

```
3648 \PackageWarningNoLine\bxjs@clsname
```

```
3649 {Cannot support pTeX units (zw etc.), since\MessageBreak
```

```
3650 the package 'bxcalc' is too old}%
```

```
3651 \bxjs@jp@unitsfalse
```

```
3652 \else \usepTeXunits
```

```
3653 \fi
```

```
3654 }{%else
```

```
3655 \PackageWarningNoLine\bxjs@clsname
```

```
3656 {Cannot support pTeX units (zw etc.), since\MessageBreak
```

```
3657 the package 'bxcalc' is unavailable}%
```

```
3658 \bxjs@jp@unitsfalse
```

```
3659 }
```

```
3660 \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。

```
3661 \ifbxjs@jp@units
```

```
3662 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
```

```

3663 \let\bxjs@parse@qh@a\@undefined
3664 \let\bxjs@parse@qh@b\@undefined
3665 \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。

```

3666 \ifbxjs@jp@units
3667   \def\bxjs@let@lenexpr#1#2{%
3668     \edef#1{#2}%
3669     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3670 \else
3671   \def\bxjs@let@lenexpr{\edef}
3672 \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 ε -TeX 拡張が使えるか検査する。

```

3673 \ifjsWitheTeX

```

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

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

```

\jH 3674   \@tempdima=0.25mm
3675   \protected\edef\jq{\dimexpr\the\@tempdima\relax}
3676   \let\jH\jq

```

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

```

\trueH 3677 \ifjsc@mag
3678   \@tempdimb=\jsBaseFontSize\relax
3679   \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3680   \@tempdima=2.5mm
3681   \bxjs@invscale\@tempdima\bxjs@tmpa
3682   \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3683   \@tempdima=10pt
3684   \bxjs@invscale\@tempdima\bxjs@tmpa
3685   \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3686   \else \let\trueQ\jq \let\bxjs@truept\p@
3687   \fi
3688   \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}` 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。

```

3689   \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3690   \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3691   \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3692   \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3693 \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。

```

3694 \def\jafontsize#1#2{%
3695   \begingroup
3696     \bxjs@jafontsize@a{#1}%
3697     \@tempdimb\jsInverseScale\@tempdima
3698     \bxjs@jafontsize@a{#2}%
3699     \xdef\bxjs@g@tmpa{%
3700       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3701   \endgroup\bxjs@g@tmpa}
3702 \def\bxjs@jafontsize@a#1{%
3703   \bxjs@parse@qh{#1}%
3704   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3705   \@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。

```

3706 \def\bxjs@kanjiskip{0pt}

```

`\setkanjiskip` 和文間空白の量を設定する。

```

3707 \newcommand*\setkanjiskip[1]{%
3708   \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3709   \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3710 \newcommand*\getkanjiskip{%
3711   \bxjs@kanjiskip}

```

`\ifbxjs@kanjiskip@enabled` 和文間空白の挿入が有効か。ただし p_T_EX 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。

```

3712 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue

```

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

```

\bxjs@disable@kanjiskip 3713 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3714   \bxjs@kanjiskip@enabledtrue
3715   \bxjs@reset@kanjiskip}
3716 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3717   \bxjs@kanjiskip@enabledfalse
3718   \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。

```

3719 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3720   \ifbxjs@kanjiskip@enabled
3721     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3722   \else \@tempskipa\z@
3723   \fi
3724   \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 3725 \def\bxjs@xkanjiskip{Opt}
\getxkanjiskip 3726 \newcommand*\setxkanjiskip[1]{%
\ifbxjs@xkanjiskip@enabled 3727 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
3728 \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3729 \newcommand*\getxkanjiskip{%
\bxjs@disable@xkanjiskip 3730 \bxjs@xkanjiskip}
3731 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3732 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3733 \bxjs@xkanjiskip@enabledtrue
3734 \bxjs@reset@xkanjiskip}
3735 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3736 \bxjs@xkanjiskip@enabledfalse
3737 \bxjs@reset@xkanjiskip}
3738 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3739 \ifbxjs@xkanjiskip@enabled
3740 \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3741 \else \@tempskipa\z@
3742 \fi
3743 \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。

```
3744 \g@addto@macro\jsResetDimen{%
3745 \bxjs@reset@kanjiskip
3746 \bxjs@reset@xkanjiskip}
3747 \let\bxjs@apply@kanjiskip\relax
3748 \let\bxjs@apply@xkanjiskip\relax
```

■和文フォント指定の扱い

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

```
3749 \@onlypreamble\bxjs@adjust@jafont
3750 \def\bxjs@adjust@jafont#1{%
3751 \ifx\jsJaFont\bxjs@@auto
3752 \bxjs@get@kanjiEmbed
3753 \ifx\bxjs@kanjiEmbed\relax
3754 \let\bxjs@tmpa\@empty
3755 \else
3756 \let\bxjs@tmpa\bxjs@kanjiEmbed
3757 \fi
3758 \else
3759 \let\bxjs@tmpa\jsJaFont
3760 \fi
3761 \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3762 \ClassWarningNoLine\bxjs@clsname
```

```

3763      {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3764      not available on the current situation}%
3765      \let\bxjs@tmpa\@empty
3766      \fi\fi
3767  }
3768  \def\bxjs@@auto{auto}
3769  \def\bxjs@@noEmbed{noEmbed}

```

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

```

3770 \let\bxjs@kanjiEmbed\relax

```

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

```

3771 \@onlypreamble\bxjs@get@kanjiEmbed
3772 \def\bxjs@get@kanjiEmbed{%
3773   \begingroup\setbox\z@=\hbox{%
3774     \global\let\bxjs@g@tmpa\relax
3775     \endlinechar\m@ne
3776     \let\do\@makeother\dospecials
3777     \catcode32=10 \catcode12=10 %form-feed
3778     \let\bxjs@tmpa\@empty
3779     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3780     \ifeof\@inputcheck\else
3781       \read\@inputcheck to\bxjs@tmpa
3782       \closein\@inputcheck
3783     \fi
3784     \ifx\bxjs@tmpa\@empty\else
3785       \openin\@inputcheck="\bxjs@tmpa"\relax
3786       \@tempswatrue
3787       \loop\if@tempswa
3788         \read\@inputcheck to\bxjs@tmpa
3789         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3790         \ifx\bxjs@tmpb\relax\else
3791           \global\let\bxjs@g@tmpa\bxjs@tmpb
3792           \@tempswafalse
3793         \fi
3794         \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3795         \ifx\bxjs@tmpb\relax\else
3796           \global\let\bxjs@g@tmpa\bxjs@tmpb
3797           \@tempswafalse
3798         \fi
3799         \ifeof\@inputcheck \@tempswafalse \fi
3800       \repeat
3801     \fi
3802   }\endgroup
3803   \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3804 }
3805 \@onlypreamble\bxjs@get@ke@a
3806 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%

```

```

3807 \ifx$#1$\def\bxjs@tmpb{#2}%
3808 \else \let\bxjs@tmpb\relax
3809 \fi}
3810 \@onlypreamble\bxjs@get@ke@b
3811 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3812 \ifx$#1$\def\bxjs@tmpb{#2}%
3813 \else \let\bxjs@tmpb\relax
3814 \fi}

```

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

```

3815 \newcommand*\jachar[1]{%
3816 \begingroup
    \jsLetHeadChar で先頭の “文字” を拾ってそれを \bxjs@jachar に渡す。

```

```

3817 \jsLetHeadChar\bxjs@tmpa{#1}%
3818 \ifx\bxjs@tmpa\relax
3819 \ClassWarningNoLine\bxjs@clsname
3820 {Illegal argument given to \string\jachar}%
3821 \else
3822 \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3823 \fi
3824 \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3825 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3826 \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。

```

3827 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3828 \@onlypreamble\bxjs@fix@hyperref@unicode
3829 \def\bxjs@fix@hyperref@unicode#1{%
3830 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3831 \@namedef{KV@Hyp@bxjs/hook}##1{%
3832 \KV@Hyp@unicode{##1}%
3833 \def\KV@Hyp@unicode####1{%
3834 \expandafter\ifx\csname if##1\expandafter\endcsname
3835 \csname if####1\endcsname\else
3836 \ClassWarningNoLine\bxjs@clsname
3837 {Blcoked hyperref option 'unicode=####1'}%
3838 \fi
3839 }%
3840 }%
3841 }

```

`\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。

```
3842 \@onlypreamble\jsCheckHyperrefUnicode
3843 \let\jsCheckHyperrefUnicode\@empty
3844 \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。

```
3845 \@onlypreamble\bxjs@check@hyperref@unicode
3846 \def\bxjs@check@hyperref@unicode#1{%
3847   \g@addto@macro\jsCheckHyperrefUnicode{%
3848     \@tempwafalse
3849     \begingroup
3850       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3851         \aftergroup\@tempwattrue \fi
3852       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3853         \csname if#1\endcsname
3854       \aftergroup\@tempwattrue \fi
3855     \endgroup
3856     \if@tempwa\else
3857       \ClassError\bxjs@clsname
3858       {The value of hyperref 'unicode' key is not suitable\MessageBreak
3859        for the present engine (must be #1)}%
3860       {\@ehc}%
3861     \fi}}
```

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

```
3862 \@onlypreamble\bxjs@urgent@special
3863 \def\bxjs@urgent@special#1{%
3864   \AtBeginDvi{\special{#1}}%
3865   \g@addto@macro\bxjs@begin@document@hook{%
3866     \ifpackageloaded{atbegshi}{%
3867       \begingroup
3868         \toks\z@{\special{#1}}%
3869         \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3870         \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3871       \endgroup
3872     }{}%
3873   }%
3874 }
```

`\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。

```
3875 \@onlypreamble\bxjs@resolve@jafont@paren
3876 \def\bxjs@resolve@jafont@paren#1{%
3877   \def\bxjs@tmpb{\let#1}%
3878   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3879 \@onlypreamble\bxjs@resolve@jafont@paren@a
3880 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3881   \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3882   \else
```

```

3883 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3884 \bxjs@tmpb\bxjs@tmpa
3885 \fi}

```

C.2 pTeX 用設定

```
3886 \if j\jsEngine
```

■ 共通命令の実装

```

3887 \def\bxjs@apply@kanjiskip{%
3888 \kanjiskip\@tempskipa}
3889 \def\bxjs@apply@xkanjiskip{%
3890 \xkanjiskip\@tempskipa}

```

\jaJaChar のサブマクロ。

```

3891 \def\bxjs@jachar#1{%
3892 \bxjs@jachar@a#1...\@nil}
3893 \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。

```
3894 \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。

```

3895 \else\ifx.#3%
3896 \@tempcnta`#1 \multiply\@tempcnta64
3897 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3898 \bxjs@jachar@b
3899 \else\ifx.#4%
3900 \@tempcnta`#1 \multiply\@tempcnta64
3901 \advance\@tempcnta`#2 \multiply\@tempcnta64
3902 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3903 \bxjs@jachar@b
3904 \else
3905 \@tempcnta`#1 \multiply\@tempcnta64
3906 \advance\@tempcnta`#2 \multiply\@tempcnta64
3907 \advance\@tempcnta`#3 \multiply\@tempcnta64
3908 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3909 \bxjs@jachar@b
3910 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3911 \ifjsWithupTeX
3912 \def\bxjs@jachar@b{\kchar\@tempcnta}
3913 \else
3914 \def\bxjs@jachar@b{%
3915 \ifx\bxUInt\@undefined\else
3916 \bxUInt{\@tempcnta}%
3917 \fi}
3918 \fi

```


和欧文間空白の命令 `\jathinspace` の実装。

```
3919 \ifbxjs@jaspace@cmd
3920   \def\jathinspace{\hskip\xkanjiskip}
3921 \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。

```
3922 \ifbxjs@jp@jismmiv
3923   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3924   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3925 \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。

```
3926 \let\bxjs@jafont@paren\@firstofone
3927 \let\bxjs@tmpa\jsJaFont
3928 \ifx\bxjs@tmpa\bxjs@@auto
3929   \let\bxjs@tmpa\@empty
3930 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3931   \def\bxjs@tmpa{noembed}
3932 \fi\fi
3933 \bxjs@resolve@jafont@paren\bxjs@tmpa
3934 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3935 \ifx\bxjs@tmpa\@empty\else
3936   \edef\bxjs@next{%
3937     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3938   }\bxjs@next
3939 \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……。)

```
3940 \begingroup
3941   \global\let\bxjs@g@tmpa\relax
3942   \catcode`\|=0 \catcode`\|=12
3943   |def|bxjs@tmpdo#1|@nil{%
3944     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
3945   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
3946     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
3947   |catcode`\|=0 \catcode`\|=12
3948   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
3949     \ifx$#2$\else
```

```

3950      \xdef\bxjs@g@tmpa{%
3951      \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3952      \fi}
3953 \firstofone{%
3954   \catcode10=12 \endlinechar\m@ne
3955   \let\do\@makeother \dospecials \catcode32=10
3956   \openin\@inputcheck=otf.sty\relax
3957   \@tempswatrue
3958   \loop\if@tempswa
3959     \ifeof\@inputcheck \@tempswafalse \fi
3960     \if@tempswa
3961       \read\@inputcheck to\bxjs@next
3962       \expandafter\bxjs@tmpdo\bxjs@next\@nil
3963     \fi
3964   \repeat
3965   \closein\@inputcheck
3966 \endgroup}
3967 \bxjs@g@tmpa

```

■hyperref 対策 unicode にはいけない。

```

3968 \ifbxjs@hyperref@enc
3969   \bxjs@check@hyperref@unicode{false}
3970 \fi

```

tounicode special 命令を出力する。

```

3971 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3972   \else\ifjsWithpTeXng T\else F\fi\fi T%
3973   \ifnum\jis"2121="A1A1 %euc
3974     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3975   \else\ifnum\jis"2121="8140 %sjis
3976     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3977   \else\ifnum\jis"2121="3000 %uptex
3978     \ifbxjs@bigcode
3979       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3980       \PassOptionsToPackage{bigcode}{pxjahyper}
3981     \else
3982       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3983     \fi
3984   \fi\fi\fi
3985   \let\bxToUnicodeSpecialDone=t
3986 \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。

```

3987 \ifx f\bxjs@enablejfam\else
3988   \@enablejfamtrue
3989 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

3990 \if@enablejfam
3991   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3992   \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3993   \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3994   \jfam\symmincho
3995   \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3996   \g@addto@macro\bxjs@begin@document@hook{%
3997     \ifx\reDeclareMathAlphabet\undefined\else
3998       \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3999       \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4000       \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4001     \fi}
4002 \fi

```

C.3 pdfTeX 用設定 : CJK + bxcjkatype

```

4003 \else\if p\jsEngine

```

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

※ Pandoc モードでは autotilde を指定しない。

```

4004 \bxjs@adjust@jafont{f}
4005 \let\bxjs@jafont@paren\@firstofone
4006 \bxjs@resolve@jafont@paren\bxjs@tmpa
4007 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4008 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4009 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4010   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4011 \fi
4012 \edef\bxjs@next{%
4013   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4014 }\bxjs@next
4015 \bxjs@cjk@loaded

```

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

※取りあえず固定はしない。

```

4016 \ifbxjs@hyperref@enc
4017   \PassOptionsToPackage{unicode}{hyperref}
4018 \fi

```

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

※ bxcjkatype を whole 付きで使っていることが前提。

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

```

4019 \ifx\bxCJKtypeHyperrefPatchDone\@undefined
4020 \begingroup
4021   \CJK@input{UTF8.bdg}
4022 \endgroup
4023 \g@addto@macro\pdfstringdefPreHook{%
4024   \@nameuse{CJK@UTF8Binding}%
4025 }
4026 \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。

```

4027 \ifx\bxCJKtypeHyperrefPatchDone\@undefined
4028 \g@addto@macro\pdfstringdefPreHook{%
4029   \ifx~\bxjs@@CJKtilde
4030     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4031     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4032     \let~\@empty
4033   \fi
4034 }
4035 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4036 \def\bxjs@@tildecmd{~}
4037 \def\bxjs@LetUnexpandableSpace#1{%
4038   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4039     \bxjs@org@LetUnexpandableSpace#1%
4040   \fi}
4041 \fi

```

■ 共通命令の実装

```

4042 \newskip\jsKanjiSkip
4043 \newskip\jsXKanjiSkip
4044 \ifx\CJKecglue\@undefined
4045   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4046 \fi
4047 \let\autospacing\bxjs@enable@kanjiskip
4048 \let\noautospacing\bxjs@disable@kanjiskip
4049 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4050 \def\bxjs@apply@kanjiskip{%
4051   \jsKanjiSkip\@tempskipa
4052   \let\CJKglue\bxjs@CJKglue}
4053 \let\autoxspacing\bxjs@enable@xkanjiskip
4054 \let\noautoxspacing\bxjs@disable@xkanjiskip
4055 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4056 \def\bxjs@apply@xkanjiskip{%
4057   \jsXKanjiSkip\@tempskipa
4058   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4059 \def\bxjs@jachar#1{%
4060   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4061 \ifbxjs@jaspace@cmd
4062   \protected\def\jathinspace{\CJKecglue}
4063 \fi

```

■和文数式ファミリー 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。

```

4064 \ifx t\bxjs@enablejfam
4065   \ClassWarningNoLine\bxjs@clsname
4066     {You cannot use 'enablejfam=true', since the\MessageBreak
4067       CJK package does not support Japanese math}
4068 \fi

```

C.4 X_YTeX 用設定 : xeCJK + zxjatype

```

4069 \else\if x\jsEngine

```

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

```

4070 \RequirePackage{zxjatype}
4071 \PassOptionsToPackage{no-math}{fontspec}%!
4072 \PassOptionsToPackage{xetex}{graphicx}%!
4073 \PassOptionsToPackage{xetex}{graphics}%!
4074 \ifx\zxJaFamilyName\@undefined
4075   \ClassError\bxjs@clsname
4076     {xeCJK or zxjatype is too old}\@ehc
4077 \fi

```

■和文フォント定義 `\jsJaFont` が指定された場合は、その値をオプションとして `zxjafont` を読み込む。非指定の場合は IPAex フォントを使用する。

```

4078 \bxjs@adjust@jafont{f}
4079 \let\bxjs@jafont@paren\@gobble
4080 \bxjs@resolve@jafont@paren\bxjs@tmpa
4081 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4082 \ifx\bxjs@tmpa\@empty
4083   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
4084   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
4085 \else
4086   \edef\bxjs@next{%
4087     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4088   }\bxjs@next
4089 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X_YTeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、X_YTeX 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、「Xe_{La}TeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```
4090 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4091   \ifbxjs@hyperref@enc
4092     \PassOptionsToPackage{unicode}{hyperref}
4093   \fi
4094 \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。

```
4095 \AtEndOfPackage{%
4096 \def\@gnewline #1{%
4097   \ifvmode \@nolnerr
4098   \else
4099     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4100     \nobreak \hskip-1sp\hskip1sp\relax
4101     \ignorespaces
4102   \fi}
4103 }
```

■共通命令の実装

```
4104 \newskip\jsKanjiSkip
4105 \newskip\jsXKanjiSkip
4106 \ifx\CJKecglue\@undefined
4107   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4108 \fi
4109 \let\autospacing\bxjs@enable@kanjiskip
4110 \let\noautospacing\bxjs@disable@kanjiskip
4111 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4112 \def\bxjs@apply@kanjiskip{%
4113   \jsKanjiSkip\@tempskipa
4114   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4115 \let\autoxspacing\bxjs@enable@xkanjiskip
4116 \let\noautoxspacing\bxjs@disable@xkanjiskip
4117 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4118 \def\bxjs@apply@xkanjiskip{%
4119   \jsXKanjiSkip\@tempskipa
4120   \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。

```
4121 \ifx\mcfamily\@undefined
4122   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4123   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4124 \fi
```

`\jachar` のサブマクロの実装。

```
4125 \def\bxjs@jachar#1{%
4126   \xeCJKDeclareCharClass{CJK}{`#1}\relax
4127   #1}
```

`\jathinspace` の実装。

```
4128 \ifbxjs@jaspace@cmd
4129   \protected\def\jathinspace{\CJKecglue}
4130 \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。

```
4131 \ifx t\bxjs@enablejfam
4132   \@enablejfamtrue
4133 \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。

```
4134 \if@enablejfam
4135   \xeCJKsetup{CJKmath=true}
4136 \fi
```

C.5 Lua_T_EX 用設定 : Lua_T_EX-ja

```
4137 \else\if 1\jsEngine
```

■Lua_T_EX-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。

```
4138 \let\zw\@undefined
4139 \RequirePackage{luatexja}
4140 \edef\bxjs@next{%
4141   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4142 }\bxjs@next
```

■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、`luatexja-preset` パッケージの `ipaex`

オプション (IPAex フォント使用) と等価な設定を用いる (luatexja-preset は読み込まない)。

```
4143 \bxjs@adjust@jafont{t}
4144 \ifx\bxjs@tmpa\bxjs@noEmbed
4145   \def\bxjs@tmpa{noembed}
4146 \fi
4147 \let\bxjs@jafont@paren@gobble
4148 \bxjs@resolve@jafont@paren\bxjs@tmpa
4149 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4150 \ifx\bxjs@tmpa\empty
4151   \defaultfontfeatures{ Kerning=Off }
4152   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
4153   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
4154 \else
4155   \edef\bxjs@next{%
4156     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4157   }\bxjs@next
4158 \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。

```
4159 \DeclareRobustCommand\rmfamily
4160   {\not@math@alphabet\rmfamily\mathrm
4161     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4162 \DeclareRobustCommand\sffamily
4163   {\not@math@alphabet\sffamily\mathsf
4164     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4165 \DeclareRobustCommand\ttfamily
4166   {\not@math@alphabet\ttfamily\mathtt
4167     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4168 \g@addto@macro\bxjs@begin@document@hook{%
4169   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4170   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4171   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4172 \bxjs@if@sf@default{%
4173   \renewcommand\kanjifamilydefault{\gtdefault}}
```

■和文パラメタの設定

```
4174 % 次の3つは既定値の通り
4175 %\ltjsetparameter{prebreakpenalty}={`,10000}
4176 %\ltjsetparameter{postbreakpenalty}={`“,10000}
4177 %\ltjsetparameter{prebreakpenalty}={`”,10000}
4178 \ltjsetparameter{jaxspmde}={`!,1}
4179 \ltjsetparameter{jaxspmde}={`〒,2}
4180 \ltjsetparameter{alxspmde}={`+,3}
4181 \ltjsetparameter{alxspmde}={`\%,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 Lua_T_EX-ja で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合に備えて同等のものを用意する。

```
4182 \ifx\ltjfakeparbegin\@undefined
4183   \protected\def\ltjfakeparbegin{%
4184     \ifhmode
4185       \relax\directlua{%
4186         luatexja.jfmglue.create_beginpar_node()}}
4187   \fi}
4188 \fi
```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```
4189 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4190 \begingroup
4191   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4192   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}
4193   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4194     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}\fi
4195     \directlua{
4196       local function patchcmd(cs, code, from, to)
4197         tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4198           :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1)..")")
4199       end
4200       patchcmd(\@xsect, [[\meaning\xsect]],
4201         \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4202       patchcmd(\@item, [[\meaning\@item]],
4203         \bxjs@tmpa, \@{\ltjfakeparbegin})}
4204 \endgroup
4205 \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。

```
4206 \ifbxjs@hyperref@enc
4207   \PassOptionsToPackage{unicode}{hyperref}
4208   \bxjs@check@hyperref@unicode{true}
4209 \fi
```

■共通命令の実装

```
4210 \protected\def\autospacing{%
4211   \ltjsetparameter{autospacing=true}}
4212 \protected\def\noautospacing{%
4213   \ltjsetparameter{autospacing=false}}
4214 \protected\def\autoxspacing{%
4215   \ltjsetparameter{autoxspacing=true}}
4216 \protected\def\noautoxspacing{%
4217   \ltjsetparameter{autoxspacing=false}}
4218 \def\bxjs@apply@kanjiskip{%
```

```

4219 \ltjsetparameter{kanjiskip={\@tempskipa}}
4220 \def\bxjs@apply@xkanjiskip{%
4221 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

\jachar のサブマクロの実装。

4222 \def\bxjs@jachar#1{%
4223 \ltjjachar`#1\relax}

\jathinspace の実装。

4224 \ifbxjs@jaspace@cmd
4225 \protected\def\jathinspace{%
4226 \hskip\ltjgetparameter{xkanjiskip}\relax}
4227 \fi

```

■和文数式ファミリー Lua_T_EX-j_a では和文数式ファミリーは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablej_fam は常に真になる。

```

4228 \ifx f\bxjs@enablejfam
4229 \ClassWarningNoLine\bxjs@clsname
4230 {You cannot use 'enablejfam=false', since the\MessageBreak
4231 LuaTEX-ja always provides Japanese math families}
4232 \fi

```

C.6 共通処理 (2)

```

4233 \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 4234 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4235 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4236 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4237 \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。

```

4238 \ifx\mathmc\@undefined
4239 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4240 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4241 \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)。

```

4242 \ifbxjs@jaspace@cmd
4243 \bxjs@protected\def\bxjs@choice@jathinspace{%
4244 \relax\ifmmode \mskip\medmuskip

```

```

4245 \else \jathinspace\ignorespaces
4246 \fi}
4247 \jsAtEndOfClass{%
4248 \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4249 \else \def\>{\protect\bxjs@choice@jathinspace}%
4250 \fi}
4251 \fi

```

■和文・和欧文間空白の初期値

```

4252 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4253 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4254 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4255 \fi

```

以上で終わり。

```

4256 %</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。

```

4257 %<*modern>
4258 \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。

```

4259 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4260 \def\encodingdefault{T1}%
4261 \input{t1enc.def}%
4262 \fontencoding\encodingdefault\selectfont
4263 \fi

```

基本フォントを Latin Modern フォントファミリーに変更する。

※以下は `\usepackage[noamth]{lmodern}` と同じ。ユーザは後で `lmodern` を好きなオプションを付けて読み込むことができる。

```

4264 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4265 \renewcommand{\rmdefault}{lmr}
4266 \renewcommand{\sfdefault}{lms}
4267 \renewcommand{\ttdefault}{lmtt}
4268 \fi

```

大型演算子用の数式フォントの設定。

※ `amsmath` パッケージと同等にする。

```

4269 \DeclareFontShape{OMX}{cmex}{m}{n}{%

```

```

4270 <-7.5>cmex7<7.5-8.5>cmex8%
4271 <8.5-9.5>cmex9<9.5->cmex10}{}%
4272 \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。
4273 \def\cmex@opt{10}

```

D.2 fixltx2e 読込

※ fixltx2e 廃止前の L^AT_EX カーネルの場合。

```

4274 \ifx\@IncludeInRelease\@undefined
4275 \RequirePackage{fixltx2e}
4276 \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。

```

4277 \RequirePackage{bxjscjkat}

```

D.4 完了

おしまい。

```

4278 %</modern>

```

付録 E 和文ドライバ : pandoc

Pandoc 用の何か。

standard ドライバの設定を引き継ぐ。

```

4279 %<*pandoc>
4280 \input{bxjsja-standard.def}
4281 \RequirePackage{bxjspandoc}

```

E.1 dupload システム

パッケージが重複して読み込まれたときに “option clash” の検査をスキップする。この時に何らかのコードを実行させることができる。

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

```

4282 \@onlypreamble\bxjs@set@dupload@proc
4283 \def\bxjs@set@dupload@proc#1{%
4284 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}

```

```

4285 \@onlypreamble\bxjs@set@dupload@proc@a
4286 \def\bxjs@set@dupload@proc@a#1{%
4287   \@onlypreamble#1\def#1##1}

```

\@if@ptions \@if@ptions の再定義。

```

4288 \@onlypreamble\bxjs@org@if@ptions
4289 \let\bxjs@org@if@ptions\@if@ptions
4290 \newif\ifbxjs@dlp
4291 \def\@if@ptions#1#2#3{%
4292   \bxjs@dlpfalse
4293   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentt}%
4294   \ifx\bxjs@tmpa\bxjs@tmpb
4295     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4296       \bxjs@dlptrue \fi
4297   \fi
4298   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
4299   \else \expandafter\bxjs@org@if@ptions
4300   \fi {#1}-{#2}-{#3}}
4301 \g@addto@macro\bxjs@begin@document@hook{%
4302   \let\@if@ptions\bxjs@org@if@ptions}
4303 \@onlypreamble\bxjs@do@dupload@proc
4304 \def\bxjs@do@dupload@proc#1#2#3{%
4305   \csname bxjs@dlp/#2.#1\endcsname{#3}%
4306   \@firstoftwo}

```

E.2 lang 変数

lang=ja という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐため、とりあえず両パッケージを無効化しておく。

```

4307 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0

```

Polyglossia について。

```

4308 \pandocSkipLoadPackage{polyglossia}
4309 \bxjs@set@dupload@proc{polyglossia.sty}{%
4310   \ClassWarning\bxjs@clsname
4311   {Loading of polyglossia is blocked}}
4312 \ifx\setmainlanguage\@undefined
4313 \newcommand*\setmainlanguage[2][]{%
4314 \newcommand*\setotherlanguage[2][]{%
4315   \ifcat_#2_\else
4316     \expandafter\let\csname #2\endcsname\@empty
4317     \expandafter\let\csname end#2\endcsname\@empty
4318     \expandafter\let\csname text#2\endcsname\@firstofone
4319   \fi}
4320 \newcommand*\setotherlanguages[2][]{%
4321   \@for\bxjs@tmpa:={#2}\do{%
4322     \setotherlangauge{\bxjs@tmpa}}}

```

```
4323 \fi
4324 \else
```

Babel について。

```
4325 \pandocSkipLoadPackage{babel}
4326 \bxjs@set@dupload@proc{babel.sty}{%
4327   \ClassWarning\bxjs@clsname
4328     {Loading of babel is blocked}}
4329 \let\foreignlanguage\@secondoftwo
4330 \let\otherlanguage\@gobble
4331 \let\endotherlanguage\@empty
4332 \fi
```

E.3 geometry 変数

`geometry` を “再度読み込んだ” 場合に、そのパラメタで `\setpagelayout*` が呼ばれるようにする。

```
4333 \bxjs@set@dupload@proc{geometry.sty}{%
4334   \setpagelayout*{#1}}
```

E.4 CJKmainfont 変数

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

```
4335 \if 1\jsEngine
4336   \pandocSkipLoadPackage{xeCJK}
4337   \providecommand*\setCJKmainfont{\setmainfont}
4338 \fi
```

E.5 paragraph のマーク

BXJS クラスでは `\paragraph` の見出しの前に `\jsParagraphMark` で指定したマークが付加され、既定ではこれは “■” である。しかし、この規定は `\paragraph` が本来のレイアウトを保っている、すなわち「行内見出しである」「節番号が付かない」ことが前提になっていると考えられる。Pandoc はこの規定を変更することがある（特に既定で `\paragraph` を別行見出しに再定義する）ため、変更された場合は `\jsParagraphMark` の既定値を空にする。

Pandoc がプレアンブルで行う再定義の結果を調べるため、`begin-document` フックを利用する。

```
4339 \g@addto@macro\bxjs@begin@document@hook{%
4340   \@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。

```
4341   \ifx\oldparagraph\@undefined\else
```

```
4342 \tempwattrue
4343 \fi
```

\paragraph が番号付きの場合は、マーク変更が必要である。

```
4344 \ifnum\c@secnumdepth>3
4345 \tempwattrue
4346 \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。

```
4347 \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4348 \let\jsParagraphMark\empty
4349 \fi\fi}
```

E.6 全角空白文字

```
4350 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4351 \catcode"3000=\active
4352 \begingroup \catcode`\!=7
4353 \protected\gdef!!!3000{\zwspace}
4354 \endgroup
4355 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4356 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4357 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4358 \fi\fi
```

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

```
4359 \if j\jsEngine
4360 \bxjs@fix@hyperref@unicode{false}
4361 \else
4362 \bxjs@fix@hyperref@unicode{true}
4363 \fi
```

E.7 完了

おしまい。

```
4364 %</pandoc>
```

和文ドライバ実装はここまで。

```
4365 %</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。

4366 %<*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

4367 %<*compat>

4368 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4369 \let\bxac@engine=n

4370 \def\bxac@do#1#2{%

4371 \edef\bxac@tmpa{\string#1}%

4372 \edef\bxac@tmpb{\meaning#1}%

4373 \ifx\bxac@tmpa\bxac@tmpb #2\fi}

4374 \bxac@do\XeTeXversion{\let\bxac@engine=x}

4375 \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。

4376 \ifx\jsAtEndOfClass\@undefined

4377 \let\bxac@delayed@if@bxjs\@firstofone

4378 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass

4379 \fi

\ImposeOldLuaTeXBehavior ムニャムニャ。

\RevokeOldLuaTeXBehavior 4380 \newif@ifbxac@in@old@behavior

4381 \let\ImposeOldLuaTeXBehavior\relax

4382 \let\RevokeOldLuaTeXBehavior\relax

G.2 X_YTEX 部分

4383 \ifx x\bxac@engine

X_YTEX 文字クラスのムニャムニャ。

4384 \@onlypreamble\bxac@adjust@charclass

4385 \bxac@delayed@if@bxjs{%

4386 \@ifpackageloaded{xeCJK}{-}{%else

4387 \ifx\xe@alloc@intercharclass\@undefined\else

4388 \ifnum\xe@alloc@intercharclass=\z@

4389 \PackageInfo\bxac@pkgname

4390 {Setting up interchar class for CJK...\@gobble}%

4391 \InputIfFileExists{load-unicode-xetex-classes.tex}{%

4392 \xe@alloc@intercharclass=3


```

4393     }{%else
4394         \PackageWarning\bxac@pkgname
4395             {Cannot find file 'load-unicode-xetex-classes.tex'%
4396             \@gobble}%
4397     }%
4398 \fi\fi
4399 \ifnum\XeTeXcharclass"3041=\z@
4400     \PackageInfo\bxac@pkgname
4401         {Adjusting interchar class for CJK...\@gobble}%
4402     \@for\bxac@x={%
4403         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4404         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4405         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4406         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4407         31FF%
4408     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4409 \fi
4410 }%
4411 }

```

以上。

```

4412 \fi

```

G.3 LuaTeX 部分

```

4413 \ifx l\bxac@engine

```

ムニヤムニヤ。

```

4414 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4415     \chardef\pdfTeXversion=200
4416     \def\pdfTeXrevision{0}
4417     \let\pdfTeXbanner\luatexbanner
4418 \fi\fi

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 4419 \begingroup\expandafter\expandafter\expandafter\endgroup
4420 \expandafter\ifx\cename outputmode\endcsname\relax\else
4421 \def\bxac@ob@list{%
4422     \do{\let}\pdfoutput{\outputmode}%
4423     \do{\let}\pdfpagewidth{\pagewidth}%
4424     \do{\let}\pdfpageheight{\pageheight}%
4425     \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4426     \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}}
4427 \def\bxac@ob@do#1#2{\begingroup
4428     \expandafter\bxac@ob@do@a\cename bxac@\string#2\endcsname{#1}#2}
4429 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4430     \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4431     \else \let#3#1\relax \let#1\@undefined
4432     \fi}
4433 \protected\def\ImposeOldLuaTeXBehavior{%
4434     \unless\ifbxac@in@old@behavior

```

```

4435 \bxac@in@old@behaviortrue
4436 \let\do\bxac@ob@do \bxac@ob@list
4437 \fi}
4438 \protected\def\RevokeOldLuaTeXBehavior{%
4439 \ifbxac@in@old@behavior
4440 \bxac@in@old@behaviorfalse
4441 \let\do\bxac@ob@do \bxac@ob@list
4442 \fi}
4443 \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。

```

4444 \ifnum\luatexversion>64 \directlua{
4445   local function range(cs, ce, cc, ff)
4446     if ff or not tex.getcatcode(cs) == cc then
4447       local setcc = tex.setcatcode
4448       for c = cs, ce do setcc(c, cc) end
4449     end
4450   end
4451   range(0x3400, 0x4DB5, 11, false)
4452   range(0x4DB5, 0x4DBF, 11, true)
4453   range(0x4E00, 0x9FCC, 11, false)
4454   range(0x9FCD, 0x9FFF, 11, true)
4455   range(0xAC00, 0xD7A3, 11, false)
4456   range(0x20000, 0x2A6D6, 11, false)
4457   range(0x2A6D7, 0x2A6FF, 11, true)
4458   range(0x2A700, 0x2B734, 11, false)
4459   range(0x2B735, 0x2B73F, 11, true)
4460   range(0x2B740, 0x2B81D, 11, false)
4461   range(0x2B81E, 0x2B81F, 11, true)
4462   range(0x2B820, 0x2CEA1, 11, false)
4463   range(0x2CEA2, 0x2FFFD, 11, true)
4464 } \fi

```

新元号のムニャムニャ。

```

4465 \begingroup
4466 \catcode\~ = 12 \let\0\relax \let\1\relax \let\2\relax
4467 \directlua{
4468   if not bxjs then bxjs = {} end
4469   function bxjs.get_new_gengo()
4470     local lc_time = nil
4471     local ok, ret = pcall(function()
4472       assert(os.type == "unix")
4473       lc_time = os.setlocale(nil, "time")
4474       assert(os.setlocale("ja_JP.utf8", "time") or
4475         os.setlocale("ja_JP.UTF-8", "time"))
4476       local heisei = "\229\185\179\230\136\144"
4477       local kanji = "[\228-\233][\128-\191][\128-\191]"
4478       local gh = os.date("\037EC", 1500000000)
4479       local gn = os.date("\037EC", 1600000000)

```

```

4480      assert(gh == heisei and gn ~= heisei and
4481             gn:match("^"..kanji.."kanji.."..$"))
4482      return gn
4483    end)
4484    os.setlocale(lc_time, "time")
4485    return ok and ret or ""
4486  end}
4487 \endgroup

```

以上。

```
4488 \fi
```

G.4 完了

おしまい。

```
4489 %</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

```

4490 %<*cjkat>
4491 \def\bxjx@pkgname{bxjscjkat}
4492 \newcount\bxjx@canta
4493 \@onlypreamble\bxjx@tmpdo
4494 \@onlypreamble\bxjx@tmpdo@a
4495 \@onlypreamble\bxjx@tmpdo@b

```

\bxjx@engine エンジンの種別。

```

4496 \let\bxjx@engine=n
4497 \def\bxjx@tmpdo#1#2{%
4498   \edef\bxjx@tmpa{\string#1}%
4499   \edef\bxjx@tmpb{\meaning#1}%
4500   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4501 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4502 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4503 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4504 \bxjx@tmpdo\pdfTeXversion{\let\bxjx@engine=p}
4505 \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。

```

4506 \def\bxjx@tmpdo#1#2{%
4507   \if#1\bxjx@engine
4508     \@ifpackageloaded{#2}{-}{%else

```

```

4509 \PackageError\bxjx@pkgname
4510 {Package '#2' must be loaded}%
4511 {Package loading is aborted.\MessageBreak\@ehc}%
4512 \endinput}
4513 \fi}
4514 \bxjx@tmpdo{p}{bxCJKtype}
4515 \bxjx@tmpdo{x}{xeCJK}
4516 \bxjx@tmpdo{l}{luatexja}

```

古い L^AT_EX の場合、`\TextOrMath` は `fixltx2e` パッケージで提供される。

```

4517 \ifx\TextOrMath\@undefined
4518 \RequirePackage{fixltx2e}
4519 \fi

```

H.2 和文カテゴリコードの設定

upL^AT_EX の場合、和文カテゴリコードの設定を LuaT_EX-ja と（ほぼ）等価なものに変更する。

※ LuaT_EX-ja との相違点：A830、A960、1B000。

```

4520 \if u\bxjx@engine
4521 \@for\bxjx@tmpa:={%
4522 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4523 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4524 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4525 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4526 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4527 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4528 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4529 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4530 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4531 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4532 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4533 FE70,FFF0,%
4534 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4535 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4536 10480,104B0,10500,10530,10600,10800,10840,10860,%
4537 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4538 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4539 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4540 11200,11280,112B0,11300,11400,11480,11580,11600,%
4541 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4542 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4543 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4544 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4545 1EE00,1F000,1F030,1F0A0,1F300,1F600,1F650,1F680,%
4546 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4547 00C0%
4548 }\do{%

```

```

4549 \@tempcnta="\bxjx@tmpa\relax
4550 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4551 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4552 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4553 \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（LuaT_EX-j_a 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。

```

4554 \@onlypreamble\bxjx@grkcyr@list
4555 \def\bxjx@grkcyr@list{%
4556 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4557 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4558 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4559 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4560 \do{0395}{LGR}{\textEpsilon}{E}%         % GR. C. L. EPSILON
4561 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4562 \do{0397}{LGR}{\textEta}{H}%           % GR. C. L. ETA
4563 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4564 \do{0399}{LGR}{\textIota}{I}%          % GR. C. L. IOTA
4565 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4566 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4567 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4568 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
4569 \do{039E}{LGR}{\textXi}{\Xi}%          % GR. C. L. XI
4570 \do{039F}{LGR}{\textOmicron}{O}%      % GR. C. L. OMICRON
4571 \do{03A0}{LGR}{\textPi}{\Pi}%          % GR. C. L. PI
4572 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
4573 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4574 \do{03A4}{LGR}{\textTau}{T}%           % GR. C. L. TAU
4575 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4576 \do{03A6}{LGR}{\textPhi}{\Phi}%        % GR. C. L. PHI
4577 \do{03A7}{LGR}{\textChi}{X}%           % GR. C. L. CHI
4578 \do{03A8}{LGR}{\textPsi}{\Psi}%        % GR. C. L. PSI
4579 \do{03A9}{LGR}{\textOmega}{\Omega}%    % GR. C. L. OMEGA
4580 \do{03B1}{LGR}{\textalpha}{\alpha}%    % GR. S. L. ALPHA
4581 \do{03B2}{LGR}{\textbeta}{\beta}%      % GR. S. L. BETA
4582 \do{03B3}{LGR}{\textgamma}{\gamma}%    % GR. S. L. GAMMA

```

4583 \do{03B4}{LGR}{\textdelta}{\delta}% % GR. S. L. DELTA
 4584 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
 4585 \do{03B6}{LGR}{\textzeta}{\zeta}% % GR. S. L. ZETA
 4586 \do{03B7}{LGR}{\texteta}{\eta}% % GR. S. L. ETA
 4587 \do{03B8}{LGR}{\texttheta}{\theta}% % GR. S. L. THETA
 4588 \do{03B9}{LGR}{\textiota}{\iota}% % GR. S. L. IOTA
 4589 \do{03BA}{LGR}{\textkappa}{\kappa}% % GR. S. L. KAPPA
 4590 \do{03BB}{LGR}{\textlambda}{\lambda}% % GR. S. L. LAMDA
 4591 \do{03BC}{LGR}{\textmu}{\mu}% % GR. S. L. MU
 4592 \do{03BD}{LGR}{\textnu}{\nu}% % GR. S. L. NU
 4593 \do{03BE}{LGR}{\textxi}{\xi}% % GR. S. L. XI
 4594 \do{03BF}{LGR}{\textomicron}{\o}% % GR. S. L. OMICRON
 4595 \do{03C0}{LGR}{\textpi}{\pi}% % GR. S. L. PI
 4596 \do{03C1}{LGR}{\textrho}{\rho}% % GR. S. L. RHO
 4597 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
 4598 \do{03C3}{LGR}{\textsigma}{\sigma}% % GR. S. L. SIGMA
 4599 \do{03C4}{LGR}{\texttau}{\tau}% % GR. S. L. TAU
 4600 \do{03C5}{LGR}{\textupsilon}{\upsilon}% % GR. S. L. UPSILON
 4601 \do{03C6}{LGR}{\textphi}{\phi}% % GR. S. L. PHI
 4602 \do{03C7}{LGR}{\textchi}{\chi}% % GR. S. L. CHI
 4603 \do{03C8}{LGR}{\textpsi}{\psi}% % GR. S. L. PSI
 4604 \do{03C9}{LGR}{\textomega}{\omega}% % GR. S. L. OMEGA
 4605 \do{0401}{T2A}{\CYRYO}{\}% % CY. C. L. IO
 4606 \do{0410}{T2A}{\CYRA}{\}% % CY. C. L. A
 4607 \do{0411}{T2A}{\CYRB}{\}% % CY. C. L. BE
 4608 \do{0412}{T2A}{\CYRV}{\}% % CY. C. L. VE
 4609 \do{0413}{T2A}{\CYRG}{\}% % CY. C. L. GHE
 4610 \do{0414}{T2A}{\CYRD}{\}% % CY. C. L. DE
 4611 \do{0415}{T2A}{\CYRE}{\}% % CY. C. L. IE
 4612 \do{0416}{T2A}{\CYRZH}{\}% % CY. C. L. ZHE
 4613 \do{0417}{T2A}{\CYRZ}{\}% % CY. C. L. ZE
 4614 \do{0418}{T2A}{\CYRI}{\}% % CY. C. L. I
 4615 \do{0419}{T2A}{\CYRISHRT}{\}% % CY. C. L. SHORT I
 4616 \do{041A}{T2A}{\CYRK}{\}% % CY. C. L. KA
 4617 \do{041B}{T2A}{\CYRL}{\}% % CY. C. L. EL
 4618 \do{041C}{T2A}{\CYRM}{\}% % CY. C. L. EM
 4619 \do{041D}{T2A}{\CYRN}{\}% % CY. C. L. EN
 4620 \do{041E}{T2A}{\CYRO}{\}% % CY. C. L. O
 4621 \do{041F}{T2A}{\CYRP}{\}% % CY. C. L. PE
 4622 \do{0420}{T2A}{\CYRR}{\}% % CY. C. L. ER
 4623 \do{0421}{T2A}{\CYRS}{\}% % CY. C. L. ES
 4624 \do{0422}{T2A}{\CYRT}{\}% % CY. C. L. TE
 4625 \do{0423}{T2A}{\CYRU}{\}% % CY. C. L. U
 4626 \do{0424}{T2A}{\CYRF}{\}% % CY. C. L. EF
 4627 \do{0425}{T2A}{\CYRH}{\}% % CY. C. L. HA
 4628 \do{0426}{T2A}{\CYRC}{\}% % CY. C. L. TSE
 4629 \do{0427}{T2A}{\CYRCH}{\}% % CY. C. L. CHE
 4630 \do{0428}{T2A}{\CYRSH}{\}% % CY. C. L. SHA
 4631 \do{0429}{T2A}{\CYRSHCH}{\}% % CY. C. L. SHCHA

4632	\do{042A}{T2A}{\CYRHRDSN}{}	% CY. C. L. HARD SIGN
4633	\do{042B}{T2A}{\CYRERY}{}	% CY. C. L. YERU
4634	\do{042C}{T2A}{\CYRSFTSN}{}	% CY. C. L. SOFT SIGN
4635	\do{042D}{T2A}{\CYREREV}{}	% CY. C. L. E
4636	\do{042E}{T2A}{\CYRYU}{}	% CY. C. L. YU
4637	\do{042F}{T2A}{\CYRYA}{}	% CY. C. L. YA
4638	\do{0430}{T2A}{\cyra}{}	% CY. S. L. A
4639	\do{0431}{T2A}{\cyrb}{}	% CY. S. L. BE
4640	\do{0432}{T2A}{\cyrv}{}	% CY. S. L. VE
4641	\do{0433}{T2A}{\cyrg}{}	% CY. S. L. GHE
4642	\do{0434}{T2A}{\cyrd}{}	% CY. S. L. DE
4643	\do{0435}{T2A}{\cyre}{}	% CY. S. L. IE
4644	\do{0436}{T2A}{\cyrrzh}{}	% CY. S. L. ZHE
4645	\do{0437}{T2A}{\cyrrz}{}	% CY. S. L. ZE
4646	\do{0438}{T2A}{\cyri}{}	% CY. S. L. I
4647	\do{0439}{T2A}{\cyrishrt}{}	% CY. S. L. SHORT I
4648	\do{043A}{T2A}{\cyrk}{}	% CY. S. L. KA
4649	\do{043B}{T2A}{\cyrl}{}	% CY. S. L. EL
4650	\do{043C}{T2A}{\cyrm}{}	% CY. S. L. EM
4651	\do{043D}{T2A}{\cyrn}{}	% CY. S. L. EN
4652	\do{043E}{T2A}{\cyro}{}	% CY. S. L. O
4653	\do{043F}{T2A}{\cyrrp}{}	% CY. S. L. PE
4654	\do{0440}{T2A}{\cyrr}{}	% CY. S. L. ER
4655	\do{0441}{T2A}{\cyrs}{}	% CY. S. L. ES
4656	\do{0442}{T2A}{\cyrt}{}	% CY. S. L. TE
4657	\do{0443}{T2A}{\cyru}{}	% CY. S. L. U
4658	\do{0444}{T2A}{\cyrf}{}	% CY. S. L. EF
4659	\do{0445}{T2A}{\cyrrh}{}	% CY. S. L. HA
4660	\do{0446}{T2A}{\cyrc}{}	% CY. S. L. TSE
4661	\do{0447}{T2A}{\cyrrch}{}	% CY. S. L. CHE
4662	\do{0448}{T2A}{\cyrrsh}{}	% CY. S. L. SHA
4663	\do{0449}{T2A}{\cyrrshch}{}	% CY. S. L. SHCHA
4664	\do{044A}{T2A}{\cyrhrdsn}{}	% CY. S. L. HARD SIGN
4665	\do{044B}{T2A}{\cyrrery}{}	% CY. S. L. YERU
4666	\do{044C}{T2A}{\cyrsftsn}{}	% CY. S. L. SOFT SIGN
4667	\do{044D}{T2A}{\cyrrerev}{}	% CY. S. L. E
4668	\do{044E}{T2A}{\cyrryu}{}	% CY. S. L. YU
4669	\do{044F}{T2A}{\cyrrya}{}	% CY. S. L. YA
4670	\do{0451}{T2A}{\cyrryo}{}	% CY. S. L. IO
4671	\do{00A7}{TS1}{\textsection}{\mathsection}%	SECTION SYMBOL
4672	\do{00A8}{TS1}{\textasciidieresis}{}	% DIAERESIS
4673	\do{00B0}{TS1}{\textdegree}{\mathdegree}%	% DEGREE SIGN
4674	\do{00B1}{TS1}{\textpm}{\pm}%	% PLUS-MINUS SIGN
4675	\do{00B4}{TS1}{\textasciicircum}{}	% ACUTE ACCENT
4676	\do{00B6}{TS1}{\textparagraph}{\mathparagraph}%	PILCROW SIGN
4677	\do{00D7}{TS1}{\texttimes}{\times}%	% MULTIPLICATION SIGN
4678	\do{00F7}{TS1}{\textdiv}{\div}%	% DIVISION SIGN
4679	}	

`\mathdegree` 面倒なので補っておく。

```
4680 \providecommand*\mathdegree{{}^{\circ}}
```

`\ifbxjx@gcc@CJK` [スイッチ]「特定 CJK 曖昧文字」を和文扱いにするか。

```
4681 \newif\ifbxjx@gcc@CJK
```

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

```
4682 \newcommand*\greekasCJK{%
```

```
4683   \bxjx@gcc@CJKtrue}
```

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

```
4684 \newcommand*\nogreekasCJK{%
```

```
4685   \bxjx@gcc@CJKfalse}
```

`\bxjx@fake@grk` `\bxjx@fake@grk{〈出力文字〉}{〈基準文字〉}` : ラテン文字で代用される数式ギリシャ文字の出力を行う。〈基準文字〉(`\mathchardef` の制御綴) の数式クラスと数式ファミリを引き継いで、〈出力文字〉(ASCII 文字トークン) の文字コードの数式文字を出力する。例えば、`\Pi` の意味が `\mathchar"7005` である場合、`\bxjx@fake@grk{B}{\Pi}` は `\mathchar"7042` を実行する。

※フォントパッケージ使用時の再定義を考慮して、〈基準文字〉が `\mathchardef` であるかを検査し、そうでない場合はフォールバックとして単に 〈出力文字〉 を実行する。

```
4686 \def\bxjx@tmpdo#1\relax{%
4687   \def\bxjx@fake@grk##1##2{%
4688     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4689   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4690     \ifx\##1\%
4691       \bxjx@canta##4\divide\bxjx@canta\@cclvi
4692       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
4693       \mathchar\bxjx@canta
4694     \else ##3\fi}
4695 }\expandafter\bxjx@tmpdo\string\mathchar\relax
```

■pdfTeX・upTeX の場合

```
4696 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0
```

- `\[bxjx@KC/〈符号値〉]` : その文字が「特定曖昧 CJK 文字」に該当する場合に定義済になる。

まず `inputenc` を読み込んで入力エンコーディングを `utf8` に変更する。

※「既定 UTF-8 化」後の L^AT_EX においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```
4697 \@ifpackageloaded{inputenc}{}{%else
4698   \RequirePackage[utf8]{inputenc}}
4699 \def\bxjx@tmpa{utf8}
4700 \ifx\bxjx@tmpa\inputencdoingname
4701   \PackageWarningNoLine\bxjx@pkgname
```



```

4702   {Input encoding changed to utf8}%
4703   \inputencoding{utf8}%
4704 \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。

```

4705 \if u\bxjx@engine
4706 \kcatcode"0370=15
4707 \kcatcode"0400=15
4708 \kcatcode"0500=15
4709 \fi

```

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

```

4710 \def\bxjx@tmpdo#1{%
4711   \@tempcnta="#1\relax
4712   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4713 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

引数 = \[bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式 LICR>}

“数式中の動作”を決定する。<数式 LICR> が空（数式非対応）なら警告を出す。

```

4714   \ifx\\#5\\%
4715     \def\bxjx@tmpa{\@inmathwarn#4}%

```

<数式 LICR> が英字である場合は \bxjx@fake@grk で出力する。大文字なら \Pi、小文字なら \pi を基準文字にする。

```

4716   \else\ifcat A\noexpand#5%
4717     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4718       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%

```

それ以外は <数式 LICR> をそのまま実行する。

```

4719   \else \def\bxjx@tmpa{#5}%
4720   \fi\fi
4721   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4722   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

4723 \if u\bxjx@engine
4724 \def\bxjx@tmpdo@b#1#2#3#4#5{%

```

引数 = \[bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式中の動作>}

当該の Unicode 文字の動作は「テキストでは <LICR>、数式では <数式中の動作>」となる。LICR は現在エンコーディングで有効な定義がある場合はそれが実行されるはずである。（つまり、現在が LGR である場合はギリシャ文字は常に欧文扱いになる。）それ以外の場合は LICR を \bxjx@ja@or@not に帰着させる。この際に、和文用の定義として当該の kchardef を使用し、その制御綴として \[bxjx@KC/...] を流用している。

```

4725   \kchardef#1=\@tempcnta
4726   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4727   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdf_T_EX 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 `\UTF{〈符号値〉}` を使う (`\UTF` は `bxCJKatype` の命令)。`\[bxjx@KC/...]` は使わないが定義済にする必要がある。

```
4728 \else\if p\bxjx@engine
4729 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4730   \mathchardef#1=\@tempcnta
4731   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4732   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4733 \fi\fi
```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```
4734 \let\do\bxjx@tmpdo \bxjx@grkcyr@list
```

`\bxjx@DeclareUnicodeCharacter` `\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。

```
4735 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4736 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4737 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4738 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4739   \count@="#1\relax
4740   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4741     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4742   \else
4743     \wlog{ \space\space skipped defining Unicode char U+#1}%
4744   \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。

```
4745 \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。

```
4746   \ifbxjx@gcc@cjk #1%
```

`\nogreekasCJK` の場合は、対象のエンコーディングに変更して LICR を実行するが、そのエンコーディングが未定義の場合は（フォールバックとして）和文用定義を使う。

```
4747   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4748   \else \UseTextSymbol{#2}{#3}%
4749   \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。

```
4750 \begingroup
4751 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4752 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4753   \noexpand\bxjx@swap@DUC@cmd
4754   \the\toks@
4755   \noexpand\bxjx@swap@DUC@cmd}}
4756 \endgroup\next
4757 \def\bxjx@swap@DUC@cmd{%
4758   \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```

4759 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4760 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
4761 \let\bxjx@tmpa\relax}

```

以上。

■Xe_{La}TeX・Lua_{TeX} の場合

```

4762 \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。

```

4763 \def\bxjx@tmpdo#1{%
4764   \bxjx@cmta="#1\relax
4765   \begingroup
4766     \lccode`~=\bxjx@cmta
4767     \lowercase{\endgroup
4768       \bxjx@tmpdo@a{~}}{#1}}
4769 \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_{La}TeX 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。

```

4770   \ifx\\#5\\\let\bxjx@tmpa\relax
4771   \else\ifcat A\noexpand#5%
4772     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4773       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4774   \else \def\bxjx@tmpa{#5}%
4775   \fi\fi
4776   \ifx\bxjx@tmpa\relax\else
4777     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4778   \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。

```

4779 \mathchardef\bxjx@tmpa="119
4780 \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_{TeX} の場合は、Lua_{TeX}-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。

```

4781 \if 1\bxjx@engine
4782   \protected\def\greekasCJK{%
4783     \bxjx@gcc@cjktrue
4784     \ltjsetparameter{jacharrange={+2, +8}}
4785   \protected\def\nogreekasCJK{%
4786     \bxjx@gcc@cjkfalse
4787     \ltjsetparameter{jacharrange={-2, -8}}
4788 \fi

```

X_eL_ATeX の場合、xeCJK は Xe_{La}TeX 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。

```

4789 \if x\bxjx@engine
4790   \let\bxjx@gcc@cjkl@list\@empty
4791   \def\do#1#2#3#4{%
4792     \edef\bxjx@gcc@cjkl@list{\bxjx@gcc@cjkl@list
4793       \noexpand\XeTeXcharclass"#1\bxjx@canta}}
4794   \bxjx@grkcyrl@list
4795   \protected\def\greekasCJK{%
4796     \bxjx@gcc@cjkltrue
4797     \bxjx@canta=\@ne \bxjx@gcc@cjkl@list}
4798   \protected\def\nogreekasCJK{%
4799     \bxjx@gcc@cjklfalse
4800     \bxjx@canta=\z@ \bxjx@gcc@cjkl@list}
4801 \fi

    以上。
4802 \fi\fi

```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
4803 \nogreekasCJK
```

H.5 完了

おしまい。

```
4804 %</cjkat>
```

付録 I 補助パッケージ : bxjspancod

Pandoc の L^AT_EX 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_EX 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

```

4805 %<*ancpandoc>
4806 %% このファイルは日本語文字を含みます。
4807 \def\bxjsp@pkgname{bxjscjkat}

\bxjsp@engine エンジンの種別。

4808 \let\bxjsp@engine=n
4809 \@onlypreamble\bxjsp@do
4810 \def\bxjsp@do#1#2{%
4811   \edef\bxjsp@tmpa{\string#1}%

```

```

4812 \edef\bxjsp@tmpb{\meaning#1}%
4813 \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
4814 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
4815 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
4816 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
4817 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}

```

1.2 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。

```

4818 \@onlypreamble\pandocSkipLoadFile
4819 \newcommand*\pandocSkipLoadFile[1]{%
4820   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4821 \def\bxjsp@skip@load@file@a#1#2{%
4822   \ifx#1\relax
4823     \def#1{2001/01/01}%
4824     \PackageInfo\bxjsp@pkgname
4825       {File '#2' marked as loaded\@gobble}%
4826   \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。

```

4827 \@onlypreamble\pandocSkipLoadPackage
4828 \newcommand*\pandocSkipLoadPackage[1]{%
4829   \pandocSkipLoadFile{#1.sty}}

```

1.3 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近 (2015 年版以降) の `LATEX` ではこれで警告が出る。これを抑止する。

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

```

4830 \ifx\@IncludeInRelease\@undefined\else
4831   \pandocSkipLoadPackage{fixltx2e}
4832 \fi

```

1.4 cmap パッケージ

エンジンが (u)p`LATEX` のときに `cmap` パッケージが読み込まれるのを阻止する。(実際は警告が出るだけで無害であるが。)

```

4833 \if j\bxjsp@engine
4834   \pandocSkipLoadPackage{cmap}
4835 \fi

```

I.5 microtype パッケージ

警告が多すぎなので消す。

```
4836 \if j\bxjsp@engine \else
4837   \PassOptionsToPackage{verbose=silent}{microtype}
4838 \fi
```

エンジンが (u)pL^AT_EX 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。

```
4839 \if j\bxjsp@engine
4840   \pandocSkipLoadPackage{microtype}
4841   \newcommand*\UseMicrotypeSet[2][]{\fi}
4842 \fi
```

I.6 Unicode 文字変換対策

Pandoc で L^AT_EX 形式に書き出す場合は、元データ中の一部の Unicode 文字を「L^AT_EX 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。

…→\ldots{} ‘→` ’→' “→`”→''

日本語 L^AT_EX では「L^AT_EX の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「\ldots{} を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の `\ldots` の実装。非数式でありかつ後続が `{}` の場合は代わりに … を実行する。

```
4843 \DeclareRobustCommand{\pandocLdots}{%
4844   \relax\ifmmode \expandafter\bxjsp@org@ldots
4845   \else \expandafter\bxjsp@ldots@a
4846   \fi}
4847 \def\bxjsp@ja@ellipsis{…}
4848 \let\bxjsp@org@ldots\ldots
4849 \def\bxjsp@ldots@a{%
4850   \futurelet\bxjsp@tok\bxjsp@ldots@b}
4851 \def\bxjsp@ldots@b{%
4852   \ifx\bxjsp@tok\bgroup \expandafter\bxjsp@ldots@c
4853   \else \expandafter\bxjsp@org@ldots
4854   \fi}
4855 \def\bxjsp@ldots@c{%
4856   \afterassignment\bxjsp@ldots@d \let\bxjsp@tok=}
4857 \def\bxjsp@ldots@d{%
4858   \futurelet\bxjsp@tok\bxjsp@ldots@e}
```

```

4859 \def\bxjsp@ldots@e{%
4860   \ifx\bxjsp@tok\egroup \expandafter\bxjsp@ldots@f
4861   \else \expandafter\bxjsp@ldots@g
4862   \fi}
4863 \def\bxjsp@ldots@f{%
4864   \bxjsp@ja@ellipsis \let\bxjsp@tok=}
4865 \def\bxjsp@ldots@g{%
4866   \expandafter\bxjsp@org@ldots\expandafter{\romannumeral-`} }

```

\ldots の実装を置き換える。

```

4867 \g@addto@macro\bxjs@begin@document@hook{%
4868   \let\bxjsp@org@ldots\ldots
4869   \let\ldots\pandocLdots}

```

I.7 PandoLa モジュール

インストール済であれば読み込む。

```

4870 \IfFileExists{bxpandola.sty}{%
4871   \RequirePackage{bxpandola}\relax
4872   \PackageInfo\bxjsp@pkgname
4873   {PandoLa module is loaded\@gobble}
4874 }{}

```

I.8 完了

おしまい。

```

4875 %</ancpandoc>

```

補助パッケージ実装はここまで。

```

4876 %</anc>

```