

BXjscls パッケージ (BXJS 文書クラス集) ソースコード説明書

八登崇之 (Takayuki YATO; aka. “ZR”)





v2.1 [2020/05/28]

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

目次

1	はじめに	3
2	オプション	10
3	和文フォントの変更	39
4	フォントサイズ	40
5	レイアウト	46
5.1	ページレイアウト	47
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	61
7	ページスタイル	62
8	文書のマークアップ	65
8.1	表題	65
8.2	章・節	70
8.3	リスト環境	83
8.4	パラメータの設定	90
8.5	フロート	92
8.6	キャプション	93
9	フォントコマンド	94

10	相互参照	97
10.1	目次の類	97
10.2	参考文献	102
10.3	索引	104
10.4	脚注	105
11	段落の頭へのグルー挿入禁止	108
12	いろいろなロゴ	112
13	amsmath との衝突の回避	112
14	初期設定	113
付録 A	和文ドライバの仕様 罨	117
付録 B	和文ドライバ : minimal 罨	118
B.1	補助マクロ	118
B.2	(u)pTeX 用の設定	120
B.3	pdfTeX 用の処理	125
B.4	X _Y TeX 用の処理	126
B.5	後処理 (エンジン共通)	126
付録 C	和文ドライバ : standard 罨	129
C.1	共通処理 (1)	130
C.2	pTeX 用設定	137
C.3	pdfTeX 用設定 : CJK + bxcjkatype	141
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	143
C.5	LuaTeX 用設定 : LuaTeX-ja	145
C.6	共通処理 (2)	149
付録 D	和文ドライバ : modern 罨	150
D.1	フォント設定	150
D.2	fixltx2e 読込	150
D.3	和文カテゴリコード	151
D.4	完了	151
付録 E	和文ドライバ : pandoc 罨	151
E.1	dupload システム	151
E.2	lang 変数	152
E.3	geometry 変数	155
E.4	CJKmainfont 変数	155
E.5	paragraph のマーク	155

E.6	全角空白文字	156
E.7	完了	156
付録 F	補助パッケージ一覧 	157
付録 G	補助パッケージ : bxjscompat 	157
G.1	準備	157
G.2	X _Y TeX 部分	157
G.3	LuaTeX 部分	158
G.4	完了	159
付録 H	補助パッケージ : bxjscjcat 	160
H.1	準備	160
H.2	和文カテゴリコードの設定	161
H.3	ギリシャ・キリル文字の扱い	161
H.4	初期設定	169
H.5	完了	169
付録 I	補助パッケージ : bxjspandoc 	169
I.1	準備	169
I.2	パッケージオプション	169
I.3	パッケージ読込の阻止	170
I.4	fixltx2e パッケージ	170
I.5	cmap パッケージ	170
I.6	microtype パッケージ	171
I.7	Unicode 文字変換対策	171
I.8	PandoLa モジュール	172
I.9	完了	172

1 はじめに

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

```

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

```

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「pL^AT_EX 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てきました。アスキーのものが最近では modified BSD ライセンスになっていますので、私のものもそれに準じて modified BSD とすることにします。

[2016-07-13] 日本語 T_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 ($\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の種類) の種別: `j` = $\mathrm{p}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ 系、`x` = $\mathrm{X}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ 、`p` = $\mathrm{pdf}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ (含 DVI モード)、`l` = $\mathrm{Lua}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ 、`J` = $\mathrm{NTT}_{\mathrm{j}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}}$ 、`0` = Omega 系、`n` = 以上の何れでもない。

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

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

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

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

```

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

\ifjsWitheTeX [スイッチ] エンジンが  $\varepsilon$ -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  $\varepsilon$ -TeX 拡張が有効な場合にのみ \protected の効果をもつ。
56 \ifjsWitheTeX \let\bxjs@protected\protected
57 \else \let\bxjs@protected\@empty
58 \fi

\bxjs@robust@def 無引数の頑強な命令を定義する。
59 \ifjsWitheTeX
60   \def\bxjs@robust@def{\protected\def}
61 \else
62   \def\bxjs@robust@def{\DeclareRobustCommand*}
63 \fi

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

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

```

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

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

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

T_EX の if-文 (`\ifXXX……⟨真⟩\else⟨偽⟩\fi`) を末尾呼出形式に変換するためのマクロ。

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

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

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

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

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

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

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

`\bxjs@ifplus` `\@ifstar` の + 版。

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

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

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

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

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

```

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

```

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

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

```

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

```

今の段階では Q/H だけが使用可能。

```

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

```

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

```

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

```

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

```

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

```

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


```

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

```

`\jsAtEndOfClass` このクラスの読込終了時に対するフック。(補助パッケージ中で用いられる。)

```

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

```

一時的な手続き用の制御綴。

```

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

```

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

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

```

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

```

`\jsInhibitGlue` は `\inhibitglue` が定義されていればそれを実行し、未定義ならば何もしない。

```

149 \bxjs@robust@def\jsInhibitGlue{%
150   \ifx\inhibitglue\@undefined\else \inhibitglue \fi}

```

万が一「2.09 互換モード」になっていた場合は、これ以上進むと危険なので強制終了させる。

```

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

```

2 オプション

これらのクラスは `\documentclass{jsarticle}` あるいは `\documentclass[オプション]{jsarticle}` のように呼び出します。

まず、オプションに関連するいくつかのコマンドやスイッチ（論理変数）を定義します。

`\if@restonecol` 段組のときに真になる論理変数です。

```
161 \newif\if@restonecol
```

`\if@titlepage` これを真にすると表題、概要を独立したページに出力します。

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

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

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

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

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

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

BXJS では report 系でも定義されることに注意。

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

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

JS クラスと異なり、初期値は偽とする。

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

以下で各オプションを宣言します。

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

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

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

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

■横置き 用紙の縦と横の長さを入れ換えます。

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

■slide オプション `slide` を新設しました。

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

```
190 \newif\if@slide
```

BXJS ではスライド用のクラス `bxjsslide` を用意しているので、本来はこのスイッチは不要なはずである。しかし、JS クラスの一部のコードをそのまま使うために保持している。※この `\if@slide` という制御綴は、ユニークでないにも関わらず、衝突した場合に正常動作が保たれない、という問題を抱えている。

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

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

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

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

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

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

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

※クラスオプションのトークン列の中に展開可能なトークンがある場合、 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ はクラスファイルの読込の前にそれを展開しようとする。このため、この位置で `\jQ` をサポートすることは原理的に不可能である。

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

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

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

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

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

```

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

```

JS クラス互換の magstyle 設定オプション。

```

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

```

■ トンボオプション トンボ (crop marks) を出力します。実際の処理は p_{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。

```

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

```

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

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

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

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

■二段組 `twocolumn` で二段組になります。

```
255 \DeclareOption{onecolumn}{\@twocolumnfalse}
256 \DeclareOption{twocolumn}{\@twocolumntrue}
```

■表題ページ `titlepage` で表題・概要を独立したページに出力します。

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

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

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

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

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

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

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

```

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

```

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

```

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

```

■文献リスト 文献リストを open 形式（著者名や書名の後に改行が入る）で出力します。これは使われることはないのでコメントアウトしてあります。

```

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

```

```

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

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

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

```

312 %\let\bxjs@enablejfam\undefined

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

JS クラスとの互換のため disablejfam オプションを定義する。

```

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

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

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

[2016-07-13] \ifdraft を定義するのをやめました。

\ifjsDraft draft オプションが指定されているか。

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

```

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

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

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

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

BXJS クラスではここは和文ドライバの管轄。

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

BXJS クラスでは `geometry` パッケージがこの処理を行う。

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

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

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

■英語化 オプション `english` を新設しました。

※`\if@english` は非ユニークで衝突耐性がない。

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

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

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

BXJS では当初から `bxjsreport` クラスが用意されている。

■`jslogo` パッケージの読み込み IAT_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。

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

■複合設定オプション

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

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

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

※元の長さが 128 pt 以上の場合でも動作するように修正した。

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

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

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

```
357 \DeclareOption{pandoc}{%
```

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

※実際の和文ドライバ・エンジン設定より優先される。

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

ドライバオプションを `dvi=dvipdfmx` 相当に変更する。

※これは実際のドライバ設定で上書きできる（オプション宣言順に注意）。

TODO: できない気がする…。

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

■エンジン・ドライバオプション

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

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

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

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

エンジン明示指定のオプションの処理。

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

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

```

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

```

`\bxjs@driver@given` オプションで明示されたドライバの種別。

```

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

```

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

```

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

```

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

※ 2.0 版より `dvipdfmx-if-dvi` は非推奨となった。

```

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

```

```

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

```

■その他の BXJS 独自オプション

TODO: 互換用オプションを分離する (2.0 版で?)。

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

※ 2.0 版より、既定値を常に真とする。

```

432 \newif\ifbxjs@bigcode \bxjs@bigcodetrue

```

`nobigcode` / `bigcode` オプションの定義。

```

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

```

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

```

437 \newif\ifbxjs@oldfontcommands

```

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

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

```

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

```

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

```

442 \DeclareOption{winjis}{%
443   \ClassWarningNoLine{bxjs}{clsname
444     {This class does not support `winjis' option}}%
445 }
446 \DeclareOption{mingoth}{%
447   \ClassWarningNoLine{bxjs}{clsname
448     {This class does not support `mingoth' option}}%
449 }
450 \DeclareOption{jis}{%
451   \ClassWarningNoLine{bxjs}{clsname

```

```

452   {This class does not support `jis' option}%
453 }

```

■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。

```

454 \DeclareOption*{%
455   \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
456   \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。

```

457 \def\bxjs@safe@setkeys#1#2{%
458   \let\bxjs@saveKV@errx\KV@errx \let\KV@errx\@gobble
459   \setkeys{#1}{#2}%
460   \let\KV@errx\bxjs@saveKV@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。

```

461 \@onlypreamble\bxjs@declare@enum@option
462 \def\bxjs@declare@enum@option#1#2{%
463   \define@key{bxjs}{#1}{%
464     \expandafter\ifx\csname bxjs@#2@##1\endcsname\relax
465     \bxjs@error@keyval{#1}{##1}%
466     \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@##1}%
467     \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。

```

468 \@onlypreamble\bxjs@declare@bool@option
469 \def\bxjs@declare@bool@option#1#2{%
470   \define@key{bxjs}{#1}[true]{%
471     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
472     \bxjs@error@keyval{#1}{##1}%
473     \else \@nameuse{bxjs@#2##1}%
474     \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。

```

475 \def\bxjs@set@keyval#1#2#3{%
476   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
477   \ifx\bxjs@next\relax
478     \bxjs@error@keyval{#1}{#2}%
479     #3%
480   \else \bxjs@next

```

```

481 \fi}
482 \@onlypreamble\bxjs@error@keyval
483 \def\bxjs@error@keyval#1#2{%
484 \ClassError\bxjs@clsname
485 {Invalid value '#2' for option #1}\@ehc}

\jsScale 〔実数値マクロ〕 和文スケール値。
486 \def\jsScale{0.924715}

\bxjs@base@opt 明示された base オプションの値。
487 %\let\bxjs@base@opt\@undefined

base オプションの処理。
488 \define@key{bxjs}{base}{%
489 \edef\bxjs@base@opt{#1}%
490 \bxjs@setbasefontsize{#1}}
491 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

\bxjs@jbase@opt 明示された jbase オプションの値。
492 %\let\bxjs@jbase@opt\@undefined

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

\bxjs@scale@opt 明示された scale オプションの値。
495 %\let\bxjs@scale@opt\@undefined

scale オプションの処理。
496 \define@key{bxjs}{scale}{%
497 \edef\bxjs@scale@opt{#1}%
498 \let\jsScale\bxjs@scale@opt}
499 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

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

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

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

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

\bxjs@jadriver 和文ドライバの名前。
504 \let\bxjs@jadriver\relax
505 %\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)。

```
506 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver@opt{#1}}
507 \define@key{bxjs}{ja}[\relax]{%
508   \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}
```

\jsJaFont 和文フォント設定の名前。

```
509 \let\jsJaFont\@empty
```

jafont オプションの処理。

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

\jsJaParam 和文ドライバパラメタの文字列。

```
511 \let\jsJaParam\@empty
```

japaram オプションの処理。

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

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

```
513 \let\bxjs@magstyle@mag=m
514 \let\bxjs@magstyle@real=r
515 \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。

```
516 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
517 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
518 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

\bxjs@magstyle@default は既定の値を表す。

```
519 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
520 \ifx l\jsEngine \ifnum\luatexversion>86
521   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
522 \fi\fi
523 \ifjsWithpTeXng
524   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
525 \fi
526 \let\bxjs@magstyle\bxjs@magstyle@default
```

magstyle オプションの処理。

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



```

533     \let\bxjs@magstyle\bxjs@magstyle@default
534     \fi}

\bxjs@geometry geometry オプションの値。
535 \let\bxjs@geometry@class=c
536 \let\bxjs@geometry@user=u
537 \let\bxjs@geometry\bxjs@geometry@class

geometry オプションの処理。
538 \define@key{bxjs}{geometry}{%
539   \expandafter\let\expandafter\bxjs@geometry\csname
540     bxjs@geometry@#1\endcsname
541   \ifx\bxjs@geometry\relax
542     \ClassError\bxjs@clsname
543       {Invalid value '#1' for option geometry}\@ehc
544     \let\bxjs@geometry\bxjs@geometry@class
545   \fi}

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

fancyhdr オプションの処理。
547 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
548 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
549 \define@key{bxjs}{fancyhdr}[true]{%
550   \bxjs@set@keyval{fancyhdr}{#1}{}}

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

DVI モードのドライバとドライバ種別との対応。
552 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
553 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
554 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
555 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
556 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

dvi オプションの処理。
557 \define@key{bxjs}{dvi}{%
558   \expandafter\let\expandafter\bxjs@tmpa\csname
559     bxjs@dvidriver@@#1\endcsname
560   \ifx\bxjs@tmpa\relax
561     \ClassError\bxjs@clsname
562     {Invalid value '#1' for option dvi}\@ehc
563   \else

\bxjs@driver@given を未定義にしていることに注意。
564   \def\bxjs@driver@opt{#1}%
565   \let\bxjs@driver@given\undefined
566   \bxjs@dvi@opttrue

```

```

567 \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。
568 \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。
569 \newif\ifbxjs@force@chapterabstract
570 %<book>\bxjs@force@chapterabstracttrue

                             layout オプションの処理。
571 \@namedef{bxjs@kv@layout@v1}{%
572 %<book>\bxjs@layout@buggyhmargintrue
573 %<report>\bxjs@force@chapterabstracttrue
574 }
575 \@namedef{bxjs@kv@layout@v2}{%
576 %<book>\bxjs@layout@buggyhmarginfalse
577 %<report>\bxjs@force@chapterabstractfalse
578 }
579 \define@key{bxjs}{layout}{%
580 \bxjs@set@keyval{layout}{#1}{}}

\bxjs@textwidth@limit  textwidth-limit の指定値。
581 %\let\bxjs@textwidth@limit@opt\undefined
582 \define@key{bxjs}{textwidth-limit}{%
583 \edef\bxjs@textwidth@limit@opt{#1}}

\bxjs@textwidth@opt  textwidth の指定値。
584 %\let\bxjs@textwidth@opt\undefined
585 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
586 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

\bxjs@number@of@lines@opt  number-of-lines の指定値。
587 %\let\bxjs@number@of@lines@opt\undefined
588 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
589 \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。
590 %\let\bxjs@paragraph@mark\undefined
591 \define@key{bxjs}{paragraph-mark}{%
592 \edef\bxjs@paragraph@mark{#1}}

\ifbxjs@whole@zw@lines  〔スイッチ〕 whole-zw-lines の指定値。
593 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
594 \let\bxjs@kv@wholezwlines=true\bxjs@whole@zw@linestrue
595 \let\bxjs@kv@wholezwlines=false\bxjs@whole@zw@linesfalse
596 \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。

```
597 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
598 \let\bxjs@kv@jaspacecmd@true\bxjs@jaspace@cmdtrue
599 \let\bxjs@kv@jaspacecmd@false\bxjs@jaspace@cmdfalse
600 \define@key{bxjs}{jaspace-cmd}[true]{\bxjs@set@keyval{jaspacecmd}{#1}{}}
601 \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。

```
602 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
603 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
604 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
605 \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。

```
606 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
607 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
608 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
609 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}
```

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

```
610 \chardef\bxjs@everyparhook@none=0
611 \chardef\bxjs@everyparhook@compat=1
612 \chardef\bxjs@everyparhook@modern=2
613 \if j\jsEngine
614 \let\bxjs@everyparhook\bxjs@everyparhook@compat
615 \else
616 \let\bxjs@everyparhook\bxjs@everyparhook@modern
617 \fi
618 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
619 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
620 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
621 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}
```

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

```
622 \chardef\bxjs@label@section@none=0
623 \chardef\bxjs@label@section@compat=1
624 \chardef\bxjs@label@section@modern=2
625 \let\bxjs@label@section\bxjs@label@section@compat
626 \def\bxjs@kv@labelsection@none{\let\bxjs@label@section\bxjs@label@section@none}
627 \def\bxjs@kv@labelsection@compat{\let\bxjs@label@section\bxjs@label@section@compat}
628 \def\bxjs@kv@labelsection@modern{\let\bxjs@label@section\bxjs@label@section@modern}
629 \define@key{bxjs}{label-section}{\bxjs@set@keyval{labelsection}{#1}{}}
```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```
630 \newif\ifbxjs@usezw \bxjs@usezwtrue
631 \bxjs@declare@bool@option{use-zw}{usezw}
632 \DeclareOption{nozw}{\setkeys{bxjs}{use-zw=false}}
633 \DeclareOption{zw}{\setkeys{bxjs}{use-zw=true}}
```

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

```
634 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
635 \bxjs@declare@bool@option{disguise-js}{disguise@js}
636 \DeclareOption{nojs}{\setkeys{bxjs}{disguise-js=false}}
637 \DeclareOption{js}{\setkeys{bxjs}{disguise-js=true}}
```

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

```
638 \newif\ifbxjs@precisetext
639 \bxjs@declare@bool@option{precise-text}{precisetext}
640 \DeclareOption{noprecisetext}{\setkeys{bxjs}{precise-text=false}}
641 \DeclareOption{precisetext}{\setkeys{bxjs}{precise-text=true}}
```

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

```
642 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
643 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}
644 \DeclareOption{nosimplejasetup}{\setkeys{bxjs}{simple-ja-setup=false}}
645 \DeclareOption{simplejasetup}{\setkeys{bxjs}{simple-ja-setup=true}}
```

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

```
646 \newif\ifbxjs@strong@cmd \bxjs@strong@cmdtrue
647 \let\bxjs@kv@strongcmd@true\bxjs@strong@cmdtrue
648 \let\bxjs@kv@strongcmd@false\bxjs@strong@cmdfalse
649 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}
```

■ 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。

```
650 \let\bxjs@org@removeelement\@removeelement
651 \def\@removeelement#1#2#3{%
652   \def\reserved@a{#2}%
653   \ifx\reserved@a\@empty \let#3\@empty
654   \else \bxjs@org@removeelement{#1}{#2}{#3}%
655   \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。

```
656 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
657 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
658 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
659 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
```

```

660 \ProcessOptions\relax
661 \bxjs@post@option@hook

  後処理
662 \if@slide
663   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
664 \fi
665 \if@landscape
666   \setlength\@tempdima {\paperheight}
667   \setlength\paperheight{\paperwidth}
668   \setlength\paperwidth {\@tempdima}
669 \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。

```

670 \def\bxjs@tmpdo{%
671   \def\bxjs@tmpa{\@gobble}%
672   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
673   \let\@classoptionslist\bxjs@tmpa}
674 \def\bxjs@tmpdo@a#1,{%
675   \ifx\@nil#1\relax\else
676     \bxjs@tmpdo@b#1{\@nil
677     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
678     \expandafter\bxjs@tmpdo@a
679   \fi}
680 \def\bxjs@tmpdo@b#1#{\bxjs@tmpdo@c}
681 \def\bxjs@tmpdo@c#1\@nil{%
682   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
683 \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。

```

684 \@expandtwoargs\@removeelement
685 {papersize}\@classoptionslist\@classoptionslist
686 \@expandtwoargs\@removeelement
687 {10pt}\@classoptionslist\@classoptionslist
688 \@expandtwoargs\@removeelement
689 {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] p^LA^TE_X/ u^pL^AT_EX を自動判別するオプション `autodetect-engine` を新設

しました。upL^AT_EX の場合は、グローバルオプションに `uplatex` を追加することで、自動判定に応じて `otf` パッケージにも `uplatex` オプションが渡るようにします。

ここのコードを削除。

[2016-11-11] pL^AT_EX 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。

```
690 \ifx\bxjs@jadriver@opt\@undefined\else
691   \let\bxjs@jadriver\bxjs@jadriver@opt
692 \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。

```
693 \let\bxjs@tmpb\jsEngine
694 \ifx j\bxjs@tmpb\ifjsWithpTeXng
695   \let\bxjs@tmpb=g
696 \fi\fi
697 \ifx j\bxjs@tmpb\ifjsWithupTeX
698   \let\bxjs@tmpb=u
699 \fi\fi
700 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
701   \let\bxjs@tmpb=n
702 \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。)


```
703 \ifx *\bxjs@engine@given
704   \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。

```
705 \ifx j\bxjs@engine@given
706   \g@addto@macro\@classoptionslist{,latex}
707 \else\ifx u\bxjs@engine@given
708   \g@addto@macro\@classoptionslist{,uplatex}
709 \fi\fi
710 \fi
711 \ifx\bxjs@engine@given\@undefined\else
712   \ifx\bxjs@engine@given\bxjs@tmpb\else
713     \ClassError\bxjs@clsname
714       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
715   \fi
716 \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。

```
717 \ifjsWithpTeXng
718   \g@addto@macro\@classoptionslist{,uplatex}
719 \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。

```
720 \@tempswatrue
721 \ifx \bxjs@driver@given\@undefined\else
722   \ifjsInPdfMode
723     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
724       \@tempswafalse
725     \fi
726   \else\ifx x\jsEngine
727     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
728       \@tempswafalse
729     \fi
730   \else
731     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
732       \@tempswafalse
733     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
734       \@tempswafalse
735     \fi\fi
736   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
737     \@tempswafalse
738   \fi\fi
739 \fi\fi
740 \fi
741 \if@tempswa\else
742   \ClassError\bxjs@clsname
743   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
744 \fi
```

DVI 出力のエンジンである場合の追加処理。

```
745 \ifjsInPdfMode \@tempswafalse
746 \else\ifx x\jsEngine \@tempswafalse
747 \else\ifjsWithpTeXng \@tempswafalse
748 \else \@tempswatrue
749 \fi\fi\fi
750 \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)。

```
751 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@opt\@undefined\else
752   \ClassWarningNoLine\bxjs@clsname
753   {No driver option is given}
754 \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

ブションに XXX を追加する。)

```
755 \ifbxjs@dvi@opt
756   \edef\bxjs@next{%
757     \let\noexpand\bxjs@driver@given
758     \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
759     \noexpand\g@addto@macro\noexpand\@classoptionslist
760     {,\bxjs@driver@opt}%
761   }\bxjs@next
762 \fi
763 \fi
```

エンジンが pT_EX-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。

```
764 \ifjsWithpTeXng
765   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
766     \let\bxjs@platexng@nodrv\@undefined
767   \else\ifx t\bxjs@platexng@nodrv\else
768     \g@addto@macro\@classoptionslist{,dvipdfmx}
769   \fi\fi
770 \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。

```
771 \ifx\bxjs@driver@given\bxjs@driver@@none
772   \bxjs@papersizefalse
773 \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)pT_EX である場合は standard に変える。

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

```
774 \def\bxjs@@minimal{minimal}
775 \ifx\bxjs@jadriver\relax
776   \ifx j\jsEngine
777     \def\bxjs@jadriver{standard}
778   \else
779     \ClassWarningNoLine\bxjs@clsname
780     {The option 'ja' is MISSING!!\MessageBreak
781     So 'ja=minimal' is assumed as fallback, but\MessageBreak
782     such implicit setting is now DEPRECATED!\MessageBreak
783     You should write 'ja=minimal' explicitly,\MessageBreak
784     if it is intended}
785     \let\bxjs@jadriver\bxjs@@minimal
786   \fi
787 \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。

```
788 \ifx\bxjs@jadriver@opt\@undefined\else
```



```

789 \ifx\bxjs@engine@given\@undefined
790 \ClassError\bxjs@clsname
791 {An engine option must be explicitly given}%
792 {When you use a Japanese-driver you must specify a correct\MessageBreak
793 engine option.\MessageBreak\@ehc}
794 \fi\fi

```

新しい Lua_T_E_X (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。)

```

795 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
796 \ifx\bxjs@magstyle\bxjs@magstyle@mag
797 \let\bxjs@magstyle\bxjs@magstyle@default
798 \ClassError\bxjs@clsname
799 {The engine does not support 'magstyle=usemag'}%
800 {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
801 The default value 'nomag*' is used instead.\MessageBreak \@ehc}
802 \fi
803 \fi

```

base、jbase、scale の値を用いて和文スケール値を解決する。

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

```

804 \ifx\bxjs@jbase@opt\@undefined\else
805 \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。

```

806 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
807 \bxjs@invscale\@tempdima\jsScale
808 \bxjs@setbasefontsize{\@tempdima}%
809 \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。

```

810 \ifx\bxjs@scale@opt\@undefined\else
811 \ClassWarningNoLine\bxjs@clsname
812 {Redundant 'scale' option is ignored}%
813 \fi
814 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
815 \@tempdimb=\bxjs@param@basefontsize\relax
816 \edef\jsScale{\strip@pt\@tempdimb}%
817 \bxjs@invscale\@tempdima\jsScale
818 \edef\jsScale{\strip@pt\@tempdima}%
819 \fi
820 \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。

```

821 \let\Cjascale\jsScale

```

8bit 欧文 \TeX 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。)

```
822 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
823   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
824     \catcode\@tempcnta\active
825     \advance\@tempcnta\@ne
826   \repeat
827 \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。

```
828 \ifbxjs@disguise@js
829 %<book|report>\def\bxjs@js@clsname{jsbook}
830 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
831 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
832 \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。

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

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

```
834 \ifbxjs@oldfontcommands
835   \AtEndOfClass{\allowoldfontcommands}
836 \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。

```
837 \newdimen\stockwidth \newdimen\stockheight
838 \begingroup\expandafter\expandafter\expandafter\endgroup
839 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
840 % \newdimen\stockwidth \newdimen\stockheight
841 \setlength{\stockwidth}{\paperwidth}
842 \setlength{\stockheight}{\paperheight}
843 \advance \stockwidth 2in
844 \advance \stockheight 2in
845 \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。

```
846 %<slide>\def\n@baseline{13}%
847 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
848 %<!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。

```
849 \ifx\bxjs@magstyle\bxjs@magstyle@mag
850 \jsc@magtrue
851 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
852 \jsc@mag@xrealtrue
853 \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 `truedimen` を指定してください。
- `geometry` 側でオプション `mag` は使えません。

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

```
854 \ifx\bxjs@param@mag\relax
855 \@tempdima=\bxjs@param@basefontsize
856 \advance\@tempdima.001pt \multiply\@tempdima25
```

```

857 \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
858 \edef\bxjs@param@mag{\the\@tempcnta}
859 \else
860 % mag 値が直接指定された場合
861 \bxjs@gs@tempcnta{\bxjs@param@mag}
862 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
863 % 有効な mag 値の範囲は 1--32768
864 \edef\bxjs@param@mag{\the\@tempcnta}
865 \advance\@tempcnta100000
866 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
867 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
868 \edef\bxjs@param@basefontsize{\the\@tempdima}
869 \fi
870 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
871 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
872 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
873 \edef\jsc@magscale{\strip@pt\@tempdima}
874 \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。

```

875 \newdimen\jsc@mpt
876 \newdimen\jsc@mmm
877 \ifjsc@mag
878 \jsc@mpt=1\p@
879 \jsc@mmm=1mm
880 \catcode\?=9 % \p@? read as \p@
881 \else
882 \jsc@mpt=\jsc@magscale\p@
883 \jsc@mmm=\jsc@magscale mm
884 \catcode\?=11 \let\p@?\jsc@mpt
885 \fi
886 \chardef\bxjs@qmcc=\catcode\?\relax
887 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode\?=12\relax}

```

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

noz 非指定時は \zw を \jsZw と同義にする。

```

888 \newdimen\jsZw
889 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
890 \ifbxjs@usezw

```

```

891 \providecommand*\zw{\jsZw}
892 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

894 \ifjsc@mag@xreal
895 \RequirePackage{type1cm}
896 \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

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

```

```

933 \def\genb@sfcnt{%
934   \edef\mandatory@arg{%
935     \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@@}%
936   \empty@sfcnt}
937 \ifbxjs@TUenc\else
938   \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
939 \fi
940 \fi

```

[2016-11-16] latex.ltx (ltspace.dtx) で定義されている `\smallskip` の、単位 `pt` を `\jsc@empt` 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 941 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 942 %\def\jsc@medskip{\vspace\jsc@medskipamount}
943 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount 944 \newskip\jsc@smallskipamount
\jsc@bigskipamount 945 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
946 %\newskip\jsc@medskipamount
947 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
948 %\newskip\jsc@bigskipamount
949 %\jsc@bigskipamoun =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

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

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

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

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

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

```

950 % \ifpapersize
951 %   \setlength{\@tempdima}{\paperwidth}
952 %   \setlength{\@tempdimb}{\paperheight}
953 %   \iftombow
954 %     \advance \@tempdima 2truein
955 %     \advance \@tempdimb 2truein
956 %   \fi
957 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
958 % \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, L^AT_EX の `\def\@{\spacefactor\@m}` という定義 (`\@m` は 1000) では `I watch TV\@.` と書くと `V` とピリオドのペアカーニングが効かなくなります。そこで, 次のような定義に直し, `I watch TV.\@` と書くことにします。

[2016-07-14] 2015-01-01 の L^AT_EX 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。
-

```
959 \chardef\bxjs@periodchar=`\.  
960 \bxjs@protected\def\bxjs@SE{%  
961   \ifnum\spacefactor<\@m \spacefactor\@m  
962   \else \spacefactor\sfcode\bxjs@periodchar  
963   \fi}  
964 \ifbxjs@fix@at@cmd  
965   \def\@{\bxjs@SE{}}  
966 \fi
```

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

※

```
967 \ifbxjs@strong@cmd\jsAtEndOfClass{%  
968   \ifx\strong\@undefined\ifx\strongenv\@undefined  
969     \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%  
970     \DeclareTextFontCommand{\strong}{\strongenv}%
```

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

※`\strongfontdeclare` は試験的機能とする。

```
971   \@ifdefinable{\strongfontdeclare}{\bxjs@protected\def\strongfontdeclare#1{%  
972     \bxjs@strong@font@declare@a#1,\@nil}}%  
973   \def\bxjs@strong@font@declare@a#1,#2\@nil{\def\bxjs@strong@font{#1}}%
```

```

974 \def\bxjs@strong@font{\bfseries}%
975 \fi\fi
976 }\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 \LaTeX 2_ϵ で 0pt plus .4pt minus .5pt に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

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

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

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

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

```

977 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
978 \expandafter\bxjs@tmpa\expandafter{%
979 \set@fontsize{#1}{#2}{#3}%
980 % 末尾にコードを追加

```



```

981 \expandafter\def\expandafter\size@update\expandafter{%
982   \size@update
983   \jsFontSizeChanged}%
984 }

```

`\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。

```

985 \newcommand*\jsFontSizeChanged{%
986   \jsZw=\f@size\p@
987   \jsZw=\jsScale \jsZw
988   \ifdim\parindent>\z@
989     \if@english \parindent=1em
990     \else       \parindent=1\jsZw
991   \fi
992   \fi\relax
993   \jsResetDimen}

```

`\jsResetDimen` ユーザ定義用のフック。

```

994 \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。

```

995 \ifjsc@mag
996   \let\jsc@setfontsize\@setfontsize
997 \else
998   \def\jsc@setfontsize#1#2#3{%
999     \@setfontsize#1{#2\jsc@mpt}{#3\jsc@mpt}}
1000 % microtype 対策
1001 \ifjsWithTeX\if j\jsEngine\else
1002   \def\jsc@setfontsize#1#2#3{%
1003     \edef\bxjs@sfs@next{%
1004       \unexpanded{\@setfontsize#1}%
1005       {\the\dimexpr#2\jsc@mpt\relax}{\the\dimexpr#3\jsc@mpt\relax}%
1006     }\bxjs@sfs@next}
1007   \fi\fi
1008 \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。

```

1009 \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`

`\widebaselines`

[2003-06-30] 数式に入るところで `\narrowbaselines` を実行しているので `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対して、しっぽ愛好家さんが次の修正を教えてくださいました。

[2008-02-18] `english` オプションで最初の段落のインデントをしないようにしました。
TODO: Hasumi さん [qa:54539] のご指摘は考慮中です。

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

```
1010 \newif\ifnarrowbaselines
1011 \if@english
1012   \narrowbaselinestrue
1013 \fi
1014 \def\narrowbaselines{%
1015   \narrowbaselinestrue
1016   \skip0=\abovedisplayskip
1017   \skip2=\abovedisplayshortskip
1018   \skip4=\belowdisplayskip
1019   \skip6=\belowdisplayshortskip
1020 % 一時的に警告を無効化する
1021   \let\bxjs@save@nomath\@nomath
1022   \let\@nomath\@gobble
1023   \@currsize\selectfont
1024   \let\@nomath\bxjs@save@nomath
1025   \abovedisplayskip=\skip0
1026   \abovedisplayshortskip=\skip2
1027   \belowdisplayskip=\skip4
1028   \belowdisplayshortskip=\skip6\relax}
1029 \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。

```
1030 \def\bxjs@if@narrowbaselines{%
1031   \ifnarrowbaselines\expandafter\@firstoftwo
1032   \else \expandafter\@secondoftwo
1033   \fi
1034 }
```

`\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。

```
1035 \renewcommand{\normalsize}{%
1036   \bxjs@if@narrowbaselines{%
1037     \jsc@setfontsize\normalsize\@xpt\@xipt
1038   }{%else
1039     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1040   }%
```

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

[2003-02-16] ちょっと変えました。

[2009-08-26] `TEX` Q & A 52569 から始まる議論について逡巡していましたが、結局、微調節してみることにしました。

```
1041 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1042 \abovedisplayshortskip \z@ \@plus3\p@?
1043 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1044 \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。

```
1045 \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。

```
1046 \normalsize
```

`\Cht` 基準となる長さの設定をします。p_{La}T_EX 2_ε カーネル (`plfonts.dtx`) で宣言されている `\Cdp` パラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (1zw) です。

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

`\Chs`

`\Cwd` 等の変数は p_T_EX 系以外では未定義なのでここで定義する。

```
1047 \ifx\Cht\@undefined \newdimen\Cht \fi
1048 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1049 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1050 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1051 \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。

```
1052 \setlength\Cht{0.88\jsZw}
1053 \setlength\Cdp{0.12\jsZw}
1054 \setlength\Cwd{1\jsZw}
1055 \setlength\Cvs{\baselineskip}
1056 \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。

```
1057 \newcommand{\small}{%
1058   \bxjs@if@narrowbaselines{%
1059     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1060     %<kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1061   }{%else
1062     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1063     %<kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1064   }%
1065   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1066   \abovedisplayshortskip \z@ \@plus3\p@?
1067   \belowdisplayskip \abovedisplayskip
1068   \belowdisplayshortskip \belowdisplayskip
1069   \def\@listi{\leftmargin\leftmargini
1070             \topsep \z@
1071             \parsep \z@
1072             \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。

```
1073 \newcommand{\footnotesize}{%
1074   \bxjs@if@narrowbaselines{%
1075     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1076     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1077   }{%else
1078     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1079     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1080   }%
1081   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1082   \abovedisplayshortskip \z@ \@plus2\p@?
1083   \belowdisplayskip \abovedisplayskip
1084   \belowdisplayshortskip \belowdisplayskip
1085   \def\@listi{\leftmargin\leftmargini
```

```

1086          \topsep \z@
1087          \parsep \z@
1088          \itemsep \parsep}}

```

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

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1089 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1090 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 1091 \if@twocolumn
1092 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1093 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1094 \else
1095 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
1096 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1097 \fi
1098 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1099 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1100 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1101 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1102 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1103 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

本文中の数式の中では `\narrowbaselines` 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。

```

1104 \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。『`pLATEX 2ε` 美文書作成入門』（1997 年）では `\sffamily \fontseries{sbc}` として新ゴ M と合わせましたが、`\fontseries{sbc}` はちょっと幅が狭いように感じました。

```

1105 % \newcommand{\headfont}{\bfseries}
1106 \newcommand{\headfont}{\sffamily}
1107 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

5 レイアウト

■二段組

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

```
1108 %<!kiyou>\setlength\columnsep{2\Cwd}  
1109 %<kiyou>\setlength\columnsep{28truebp}  
1110 \setlength\columnseprule{\z@}
```

■段落

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

```
\lineskiplimit 1111 \setlength\lineskip{1\jsc@empt}  
1112 \setlength\normallineskip{1\jsc@empt}  
\normallineskiplimit 1113 \setlength\lineskiplimit{1\jsc@empt}  
1114 \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。

```
1115 \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。

```
1116 \setlength\parskip{\z@}  
1117 \if@slide  
1118 \setlength\parindent{0\p@}  
1119 \else  
1120 \setlength\parindent{1\Cwd}  
1121 \fi
```

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

```
\@highpenalty 1122 \@lowpenalty 51  
1123 \@medpenalty 151  
1124 \@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。

```
1125 % \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。

```
1126 % \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。

```
1127 \@onlypreamble\bxjs@bd@pre@geometry@hook
1128 \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。

```
1129 \ifjsc@mag
1130 \mag=\bxjs@param@mag
1131 \fi
1132 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLength` のための和文単位の定義。

```
1133 \def\bxjs@unit@trueQ{0.25trueem}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1134 \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。

```
1135 \def\bxjs@tmpdo{\futurelet\bxjs@tmpa\bxjs@tmpdo@a}
1136 \def\bxjs@tmpdo@a{%
1137   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@tmpdo@b
1138   \else \expandafter\bxjs@tmpdo@c \fi}
1139 \def\bxjs@tmpdo@b#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1140 \def\bxjs@tmpdo@c#1\@nil{\bxjs@tmpdo@d#1,,\@nil}
1141 \def\bxjs@tmpdo@d#1,#2,#3\@nil{%
1142   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1143 \expandafter\bxjs@tmpdo\bxjs@param@paper\@empty\@empty\@nil
```

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

```
1144 \edef\bxjs@layout@paper{%
1145   \ifjsc@mag truedimen,\fi
1146   \if@landscape landscape,\fi
1147   \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。

```

1148 %<*article|report>
1149 \def\bxjs@layout@base{%
1150   headheight=\topskip,footskip=0.03367\paperheight,%
1151   headsep=\footskip-\topskip,includeheadfoot,%
1152 }
1153 \edef\bxjs@layout{\bxjs@layout@base
1154   hscale=0.76,hmarginratio=1:1,%
1155   vscale=0.83,vmarginratio=1:1,%
1156 }
1157 %</article|report>
1158 %<*book>
1159 \def\bxjs@layout@base{%
1160   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1161 }
1162 \ifbxjs@layout@buggyhmargin      %---
1163 % アレ
1164 \edef\bxjs@layout{\bxjs@layout@base
1165   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1166   vscale=0.83,vmarginratio=1:1,%
1167 }
1168 \else                            %---
1169 % 非アレ
1170 \edef\bxjs@layout{\bxjs@layout@base
1171   hmargin=18\jsc@mmm,%
1172   vscale=0.83,vmarginratio=1:1,%
1173 }
1174 \fi                               %---
1175 %</book>
1176 %<*slide>
1177 \def\bxjs@layout@base{%
1178   noheadfoot,%
1179 }
1180 \edef\bxjs@layout{\bxjs@layout@base
1181   hscale=0.9,hmarginratio=1:1,%
1182   vscale=0.944,vmarginratio=1:1,%
1183 }
1184 %</slide>

```

textwidth オプションの設定を反映する。

```

1185 %<!*book>
1186 \ifx\bxjs@textwidth@opt\@undefined\else
1187   \jsSetQHLengh\@tempdima{\bxjs@textwidth@opt}
1188   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1189 \fi
1190 %</!*book>
1191 \ifx\bxjs@number@of@lines@opt\@undefined\else
1192   \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1193   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1194 \fi

```


`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1195 \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。

```
1196 %<*book>
1197 \newcommand\jsTextWidthLimit{40}
1198 \@tempdima=\jsTextWidthLimit\Cwd
1199 \ifx\bxjs@textwidth@limit@opt\@undefined\else
1200   \bxjs@set@tempcnta{\bxjs@textwidth@limit@opt}
1201   \@tempdima=\@tempcnta\Cwd
1202 \fi
1203 \ifx\bxjs@textwidth@opt\@undefined\else
1204   \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1205 \fi
1206 \edef\bxjs@textwidth@limit{\the\@tempdima}
1207 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1208   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1209   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1210 \fi
1211 %</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。

```
1212 \def\bxjs@preproc@layout{%
1213   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mpt}
```

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

```
1214 \def\bxjs@postproc@layout{%
  geometry のドライバを再設定する。
1215   \ifx\bxjs@geometry@driver\relax\else
1216     \let\Gm@driver\bxjs@geometry@driver
1217   \fi
  \ht\strutbox の値を元に戻す。
1218   \ht\strutbox=\bxjs@save@ht@strutbox\relax
  \textwidth の値を補正する。
1219   \ifbxjs@whole@zw@lines
1220     \@tempdimb=\textwidth
1221     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1222     \advance\textwidth.005pt\relax
1223     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1224     \advance\@tempdimb-\textwidth
1225     \advance\oddsidemargin 0.5\@tempdimb
```

```

1226 \advance\evensidemargin 0.5\@tempdimb
1227 \fi
1228 \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。

```

1229 %<*book>
1230 \@tempdima=\bxjs@textwidth@limit\relax
1231 \ifbxjs@whole@zw@lines
1232 \advance\@tempdima.005pt\relax
1233 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1234 \fi
1235 \ifdim\textwidth>\@tempdima
1236 \textwidth=\@tempdima
1237 \addtolength\evensidemargin{\fullwidth-\textwidth}
1238 \fi
1239 %</book>

```

\textheight 関連の調整。

```

1240 \@tempdimb=\textheight
1241 \advance\textheight-\topskip
1242 \advance\textheight.005pt\relax
1243 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1244 \advance\textheight\topskip
1245 \advance\@tempdimb-\textheight
1246 \advance\topmargin0.5\@tempdimb

```

\headheight 関連の調整。

```

1247 \@tempdima=\topskip
1248 \advance\headheight\@tempdima
1249 \advance\topmargin-\@tempdima

```

marginpar 関連の調整。

```

1250 \setlength\marginparsep{\columnsep}
1251 \setlength\marginparpush{\baselineskip}
1252 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1253 -\textwidth-10\jsc@mmm-\marginparsep}
1254 \ifbxjs@whole@zw@lines
1255 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1256 \fi

```

連動する変数。

```

1257 \maxdepth=.5\topskip
1258 \stockwidth=\paperwidth
1259 \stockheight=\paperheight
1260 }

```

\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。

```

1261 \edef\jsGeometryOptions{%

```

```
1262 \bxjs@layout@paper,\bxjs@layout}
```

■geometry パッケージを読み込む

geomentry=class の場合に、実際に geometry パッケージを読み込む。

```
1263 \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。

```
1264 \ifbxjs@papersize
1265   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1266     \PassOptionsToPackage{dvipdfm}{geometry}
1267   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1268     \PassOptionsToPackage{dvipdfm}{geometry}
1269   \fi\fi
1270   \let\bxPapersizeSpecialDone=t
1271 \else
1272   \PassOptionsToPackage{driver=none}{geometry}
1273 \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。

```
1274 \AtBeginDocument{\bxjs@bd@pre@geometry@hook}
1275 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1276 \bxjs@preproc@layout
1277 \edef\bxjs@next{%
1278   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1279 }\bxjs@next
1280 \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。

```
1281 \let\bxjs@geometry@driver\Gm@driver
1282 \bxjs@postproc@layout
```

geometry のドライバ自動判別に対する前処理。

```
1283 \g@addto@macro\bxjs@bd@pre@geometry@hook{%
```

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

```
1284   \@ifpackagelater{geometry}{2010/02/12}{\relax}%
1285   \PackageError\bxjs@clsname
1286     {Your 'geometry' package is too old (< v5.0)}%
1287     {\@ehc}%
1288   \let\Gm@driver\relax}%
```

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

```
1289 \ifjsWithpTeXng
1290 \ifx\Gm@driver\@empty
1291 \def\Gm@driver{pdftex}%
1292 \fi
1293 \fi}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```
1294 \def\setpagelayout{%
1295 \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1296 \@ifstar{\bxjs@setpagelayout@a\@ne}{\bxjs@setpagelayout@a\z@}}%
1297 \def\bxjs@setpagelayout@a#1#2{%
1298 \ifcase#1% modify
1299 \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1300 \or% reset(*)
1301 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1302 \or% semireset(+)
1303 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1304 \fi
1305 \bxjs@preproc@layout
1306 \edef\bxjs@next{%
1307 \noexpand\geometry{\bxjs@next}%
1308 }\bxjs@next
1309 \bxjs@postproc@layout}
```

■ geometry パッケージを読み込まない

`geometry=user` の場合の処理。

```
1310 \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。

```
1311 \g@addto@macro\bxjs@begin@document@hook{%
1312 \ifdim\textwidth=.5\maxdimen
1313 \ClassError\bxjs@clsname
1314 {Page layout is not properly set}%
1315 {\@ehd}%
1316 \fi}
1317 \def\jsUseMinimalPageLayout{%
1318 \setlength{\textwidth}{6.5in}%
1319 \setlength{\textheight}{8in}}
```

`\setpagelayout` はとりあえず無効にしておく。

```
1320 \let\bxjs@geometry@driver\relax
```

```

1321 \def\setpagelayout{%
1322   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1323     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1324 \def\bxjs@pagelayout@a#1{%
1325   \ClassError\bxjs@clsname
1326   {Command '\string\setpagelayout' is not supported,\MessageBreak
1327     because 'geometry' value is not 'class'}\@eha}
1328 %
1329 \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。

```

1330 %<*jsclasses>

```

■ 縦方向のスペース

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

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

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

```

1331 \setlength\topskip{1.38zw}% from 10\jsc@mpt (2016-08-17)
1332 \if@slide
1333   \setlength\headheight{0\jsc@mpt}
1334 \else
1335   \setlength\headheight{20\jsc@mpt}% from 2\topskip (2016-08-17); from \topskip (2003-
1336   06-26)

```

\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。

```

1337 %<*article|kiyou>
1338 \if@slide
1339   \setlength\footskip{0pt}
1340 \else
1341   \setlength\footskip{0.03367\paperheight}
1342   \ifdim\footskip<\baselineskip
1343     \setlength\footskip{\baselineskip}
1344   \fi
1345 \fi
1346 %</article|kiyou>
1347 %<jspf>\setlength\footskip{9\jsc@mmm}
1348 %<*book>
1349 \if@report
1350   \setlength\footskip{0.03367\paperheight}
1351   \ifdim\footskip<\baselineskip
1352     \setlength\footskip{\baselineskip}
1353   \fi
1354 \else
1355   \setlength\footskip{0pt}
1356 \fi
1357 %</book>
1358 %<*report>
1359 \setlength\footskip{0.03367\paperheight}
1360 \ifdim\footskip<\baselineskip
1361   \setlength\footskip{\baselineskip}
1362 \fi
1363 %</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)。

```

1364 %<*article>
1365 \if@slide
1366   \setlength\headsep{0\jsc@empt}
1367   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1368   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1369 \else
1370   \setlength\headsep{\footskip}
1371   \addtolength\headsep{-\topskip}
1372 \fi
1373 %</article>
1374 %<*book>
1375 \if@report
1376   \setlength\headsep{\footskip}
1377   \addtolength\headsep{-\topskip}
1378 \else

```

```

1379 \setlength\headsep{6\jsc@mmm}
1380 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1381 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1382 \fi
1383 %</book>
1384 %<*report>
1385 \setlength\headsep{\footskip}
1386 \addtolength\headsep{-\topskip}
1387 %</report>
1388 %<*jspf>
1389 \setlength\headsep{9\jsc@mmm}
1390 \addtolength\headsep{-\topskip}
1391 %</jspf>
1392 %<*kiyou>
1393 \setlength\headheight{0\jsc@mpt}
1394 \setlength\headsep{0\jsc@mpt}
1395 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1396 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1397 %</kiyou>

```

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

```

1398 \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。

```

1399 \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。

```

1400 %<*article>
1401 \if@slide
1402 \setlength\fullwidth{0.9\paperwidth}
1403 \else
1404 \setlength\fullwidth{0.76\paperwidth}

```

```

1405 \fi
1406 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1407 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1408 \setlength\textwidth{\fullwidth}
1409 %</article>
1410 %<*book>
1411 \if@report
1412   \setlength\fullwidth{0.76\paperwidth}
1413 \else
1414   \setlength\fullwidth{\paperwidth}
1415   \addtolength\fullwidth{-36\jsc@mmm}
1416 \fi
1417 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1418 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1419 \setlength\textwidth{\fullwidth}
1420 \if@report \else
1421   \if@twocolumn \else
1422     \ifdim \fullwidth>40zw
1423       \setlength\textwidth{40zw}
1424     \fi
1425   \fi
1426 \fi
1427 %</book>
1428 %<*report>
1429 \setlength\fullwidth{0.76\paperwidth}
1430 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1431 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1432 \setlength\textwidth{\fullwidth}
1433 %</report>
1434 %<*jspf>
1435 \setlength\fullwidth{50zw}
1436 \addtolength\fullwidth{8\jsc@mmm}
1437 \setlength\textwidth{\fullwidth}
1438 %</jspf>
1439 %<*kiyou>
1440 \setlength\fullwidth{48zw}
1441 \addtolength\fullwidth{\columnsep}
1442 \setlength\textwidth{\fullwidth}
1443 %</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)。

```
1444 %<*article|book|report>
1445 \if@slide
1446   \setlength{\textheight}{0.95\paperheight}
1447 \else
1448   \setlength{\textheight}{0.83\paperheight}
1449 \fi
1450 \addtolength{\textheight}{-10\jsc@mppt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1451 \addtolength{\textheight}{-\headsep}
1452 \addtolength{\textheight}{-\footskip}
1453 \addtolength{\textheight}{-\topskip}
1454 \divide\textheight\baselineskip
1455 \multiply\textheight\baselineskip
1456 %</article|book|report>
1457 <jspf>\setlength{\textheight}{51\baselineskip}
1458 <kiyou>\setlength{\textheight}{47\baselineskip}
1459 \addtolength{\textheight}{\topskip}
1460 \addtolength{\textheight}{0.1\jsc@mppt}
1461 <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。

```
1462 \def\flushbottom{%
1463   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mppt}%
1464   \let\@texttop\relax}
```

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

```
1465 \setlength\marginparsep{\columnsep}
1466 \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。

```
1467 \setlength{\oddsidemargin}{\paperwidth}
1468 \addtolength{\oddsidemargin}{-\fullwidth}
1469 \setlength{\oddsidemargin}{.5\oddsidemargin}
1470 \iftombow
1471   \addtolength{\oddsidemargin}{-1in}
1472 \else
1473   \addtolength{\oddsidemargin}{-\inv@mag in}
1474 \fi
1475 \setlength{\evensidemargin}{\oddsidemargin}
1476 \if@mparswitch
1477   \addtolength{\evensidemargin}{\fullwidth}
1478   \addtolength{\evensidemargin}{-\textwidth}
1479 \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。

```
1480 \setlength\marginparwidth{\paperwidth}
1481 \addtolength\marginparwidth{-\oddsidemargin}
1482 \addtolength\marginparwidth{-\inv@mag in}
1483 \addtolength\marginparwidth{-\textwidth}
1484 \addtolength\marginparwidth{-10\jsc@mmm}
1485 \addtolength\marginparwidth{-\marginparsep}
1486 \@tempdima=1zw
1487 \divide\marginparwidth\@tempdima
1488 \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）。

```
1489 \setlength\topmargin{\paperheight}
1490 \addtolength\topmargin{-\textheight}
1491 \if@slide
1492   \addtolength\topmargin{-\headheight}
1493 \else
1494   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
     \headheight (2003-06-26)
1495 \fi
1496 \addtolength\topmargin{-\headsep}
1497 \addtolength\topmargin{-\footskip}
1498 \setlength\topmargin{0.5\topmargin}
1499 %<kiyou>\setlength\topmargin{81truebp}
1500 \iftombow
```

```

1501 \addtolength\topmargin{-1in}
1502 \else
1503 \addtolength\topmargin{-\inv@mag in}
1504 \fi
1505 %</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。

```
1506 \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。

```
1507 \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。

```
1508 \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。

```
1509 \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。

```
1510 \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。

```
1511 \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。

1512 `\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。

1513 `\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。

1514 `\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。

1515 `\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。

1516 `\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。

1517 `\renewcommand{\dblfloatpagefraction}{.8}`

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

1518 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 1519 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`
 1520 `\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` 1521 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 1522 `\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` 1523 `\setlength\@fptop{0\p@? \@plus 1fil}`
 1524 `\setlength\@fpsep{8\p@? \@plus 2fil}`
 1525 `\setlength\@fpbot{0\p@? \@plus 1fil}`

`\@dblftop` 段抜きフロートについての値です。

`\@dblfpsep` 1526 `\setlength\@dblftop{0\p@? \@plus 1fil}`
`\@dblfpbot` 1527 `\setlength\@dblfpsep{8\p@? \@plus 2fil}`
 1528 `\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 pL^AT_EX の標準クラス 2017/02/15 に合わせて、同じ命令を追
`\pltx@cleartoleftpage` 加しました。

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

となっています。

```
1529 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1530 % \ifodd\c@page
1531 % \iftdir
1532 % \hbox{}\thispagestyle{empty}\newpage
1533 % \if@twocolumn\hbox{}\newpage\fi
1534 % \fi
1535 % \else
1536 % \ifydir
1537 % \hbox{}\thispagestyle{empty}\newpage
1538 % \if@twocolumn\hbox{}\newpage\fi
1539 % \fi
1540 % \fi\fi}
1541 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1542 % \ifodd\c@page
1543 % \ifydir
1544 % \hbox{}\thispagestyle{empty}\newpage
1545 % \if@twocolumn\hbox{}\newpage\fi
1546 % \fi
1547 % \else
1548 % \iftdir
1549 % \hbox{}\thispagestyle{empty}\newpage
1550 % \if@twocolumn\hbox{}\newpage\fi
1551 % \fi
1552 % \fi\fi}
1553 \def\pltx@cleartooddpage{\clearpage\if@twoside
1554 \ifodd\c@page\else
1555 \hbox{}\thispagestyle{empty}\newpage
1556 \if@twocolumn\hbox{}\newpage\fi
1557 \fi\fi}
1558 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1559 \ifodd\c@page
1560 \hbox{}\thispagestyle{empty}\newpage
1561 \if@twocolumn\hbox{}\newpage\fi
1562 \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。

```
1563 \let\pltx@cleartorightpage\pltx@cleartooddpage
1564 \let\pltx@cleartoleftpage\pltx@cleartoevenpage
```

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

```
1565 %<*book|report>
1566 \if@openleft
1567   \let\cleardoublepage\pltx@cleartoleftpage
1568 \else\if@openright
1569   \let\cleardoublepage\pltx@cleartorightpage
1570 \fi\fi
1571 %</book|report>
```

7 ページスタイル

ページスタイルとして、 \LaTeX 2 ϵ (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 \LaTeX 2 ϵ 本体で定義されています。

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

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

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

`\ps@empty` `empty` ページスタイルの定義です。 \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。

```

1572 % \def\ps@empty{%
1573 %   \let\@mkboth\@gobbletwo
1574 %   \let\@oddhead\@empty
1575 %   \let\@oddfoot\@empty
1576 %   \let\@evenhead\@empty
1577 %   \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。

```

1578 \def\ps@plainfoot{%
1579   \let\@mkboth\@gobbletwo
1580   \let\@oddhead\@empty
1581   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1582   \let\@evenhead\@empty
1583   \let\@evenfoot\@oddfoot}
1584 \def\ps@plainhead{%
1585   \let\@mkboth\@gobbletwo
1586   \let\@oddfoot\@empty
1587   \let\@evenfoot\@empty
1588   \def\@evenhead{%
1589     \if@mparswitch \hss \fi
1590     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1591     \if@mparswitch\else \hss \fi}%
1592   \def\@oddhead{%
1593     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1594 %<book>\let\ps@plain\ps@plainhead
1595 %<!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。

```

1596 %<*article|slide>
1597 \if@twoside
1598   \def\ps@headings{%
1599     \let\@oddfoot\@empty
1600     \let\@evenfoot\@empty
1601     \def\@evenhead{\if@mparswitch \hss \fi
1602       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1603     \if@mparswitch\else \hss \fi}%
1604     \def\@oddhead{%
1605       \underline{%
1606         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1607     \let\@mkboth\markboth
1608     \def\sectionmark##1{\markboth{%
1609       \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1610       ##1}{}}%
1611     \def\subsectionmark##1{\markright{%

```

```

1612     \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1613     ##1}}}%
1614 }
1615 \else % if not twoside
1616     \def\ps@headings{%
1617         \let\@oddfoot\@empty
1618         \def\@oddhead{%
1619             \underline{%
1620                 \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1621         \let\@mkboth\markboth
1622         \def\sectionmark##1{\markright{%
1623             \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1624             ##1}}}%
1625 \fi
1626 %</article|slide>

```

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

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

```

1627 %<*book|report>
1628 \def\bxjs@maybe@autoxspacing{%
1629     \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1630 \newif\if@omit@number
1631 \def\ps@headings{%
1632     \let\@oddfoot\@empty
1633     \let\@evenfoot\@empty
1634     \def\@evenhead{%
1635         \if@mparswitch \hss \fi
1636         \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1637             \textbf{\thepage}\hfil\leftmark}}}%
1638         \if@mparswitch\else \hss \fi}%
1639     \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1640         {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1641     \let\@mkboth\markboth
1642     \def\chaptermark##1{\markboth{%
1643         \ifnum \c@secnumdepth >\m@ne
1644             \if@mainmatter
1645                 \if@omit@number\else
1646                     \@chapapp\thechapter\@chappos\hskip1\jsZw
1647                 \fi
1648             \fi
1649         ##1}{}}}%
1650     \def\sectionmark##1{\markright{%
1651         \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi

```



```

1653     ##1}}}%
1654 %</book|report>

    最後は学会誌の場合です。

1655 %<*jspf>
1656 \def\ps@headings{%
1657     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1658     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1659     \def\@oddhead{\normalfont\hfil \@title \hfil}
1660     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1661 %</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。

```

1662 \def\ps@myheadings{%
1663     \let\@oddfoot\@empty\let\@evenfoot\@empty
1664     \def\@evenhead{%
1665         \if@mparswitch \hss \fi%
1666         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1667         \if@mparswitch\else \hss \fi}%
1668     \def\@oddhead{%
1669         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1670     \let\@mkboth\@gobbletwo
1671 %<book|report> \let\chaptermark\@gobble
1672 \let\sectionmark\@gobble
1673 %<!book&!report> \let\subsectionmark\@gobble
1674 }

```

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。

```

\author
\date
1675 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1676 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1677 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1678 % \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。

```

1679 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1680 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1681 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1682 \AtBeginDocument{\bxjs@decl@subtitle}
1683 \def\bxjs@decl@subtitle{%
1684   \global\let\bxjs@decl@subtitle\relax
1685   \ifx\subtitle\@undefined
1686     \global\let\subtitle\jsSubtitle
1687   \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。

```

1688 \def\bxjs@annihilate@subtitle{%
1689   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1690   \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 1691 %<*\jspf>
\keywords 1692 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1693 \newcommand*{\author}[1]{\gdef\@author{#1}}
1694 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1695 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1696 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1697 %</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。

```

1698 \def\plainifnotempty{%
1699   \ifx \@oddhead \@empty
1700     \ifx \@oddfoot \@empty
1701       \else
1702         \thispagestyle{plainfoot}%
1703       \fi
1704     \else
1705       \thispagestyle{plainhead}%
1706     \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。

```
1707 %<*article|book|report|slide>
1708 \if@titlepage
1709   \newcommand{\maketitle}{%
1710     \begin{titlepage}%
1711       \let\footnotesize\small
1712       \let\footnoterule\relax
1713       \let\footnote\thanks
1714       \null\vfil
1715       \if@slide
1716         {\footnotesize \@date}%
1717         \begin{center}
1718           \mbox{} \\\[1\jsZw]
1719           \large
1720           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1721           \jsc@smallskip
1722           \@title
1723           \ifx\bxjs@subtitle\@undefined\else
1724             \par\vskip\z@
1725             {\small \bxjs@subtitle\par}
1726           \fi
1727           \jsc@smallskip
1728           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1729           \vfill
1730           {\small \@author}%
1731         \end{center}
1732       \else
1733       \vskip 60\p@?
1734       \begin{center}%
1735         {\LARGE \@title \par}%
1736         \ifx\bxjs@subtitle\@undefined\else
1737           \vskip5\p@?
1738           {\normalsize \bxjs@subtitle\par}
1739         \fi
1740         \vskip 3em%
1741         {\large
1742           \lineskip .75em
1743           \begin{tabular}[t]{c}%
1744             \@author
1745           \end{tabular}\par}%
1746         \vskip 1.5em
1747         {\large \@date \par}%
1748       \end{center}%
1749       \fi
1750       \par
1751       \@thanks\vfil\null
1752     \end{titlepage}%
1753     \setcounter{footnote}{0}%
```

```

1754 \global\let\thanks\relax
1755 \global\let\maketitle\relax
1756 \global\let\@thanks\@empty
1757 \global\let\@author\@empty
1758 \global\let\@date\@empty
1759 \global\let\@title\@empty
1760 \global\let\title\relax
1761 \global\let\author\relax
1762 \global\let\date\relax
1763 \global\let\and\relax
1764 \bxjs@annihilate@subtitle
1765 }%
1766 \else
1767 \newcommand{\maketitle}{\par
1768 \begin{group}
1769 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1770 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1771 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1772 \parindent 1\jsZw\noindent
1773 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1774 \if@twocolumn
1775 \ifnum \col@number=\@ne
1776 \maketitle
1777 \else
1778 \twocolumn[\maketitle]%
1779 \fi
1780 \else
1781 \newpage
1782 \global\@topnum\z@ % Prevents figures from going at top of page.
1783 \maketitle
1784 \fi
1785 \plainifnotempty
1786 \@thanks
1787 \endgroup
1788 \setcounter{footnote}{0}%
1789 \global\let\thanks\relax
1790 \global\let\maketitle\relax
1791 \global\let\@thanks\@empty
1792 \global\let\@author\@empty
1793 \global\let\@date\@empty
1794 \global\let\@title\@empty
1795 \global\let\title\relax
1796 \global\let\author\relax
1797 \global\let\date\relax
1798 \global\let\and\relax
1799 \bxjs@annihilate@subtitle
1800 }

```

`\@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。

```
1801 \def\@maketitle{%
1802   \newpage\null
1803   \vskip 2em
1804   \begin{center}%
1805     \let\footnote\thanks
1806     {\LARGE \@title \par}%
1807     \ifx\bxjs@subtitle\@undefined\else
1808       \vskip3\p@?
1809       {\normalsize \bxjs@subtitle\par}
1810     \fi
1811     \vskip 1.5em
1812     {\large
1813       \lineskip .5em
1814       \begin{tabular}[t]{c}%
1815         \@author
1816       \end{tabular}\par}%
1817     \vskip 1em
1818     {\large \@date}%
1819   \end{center}%
1820   \par\vskip 1.5em
1821 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1822 }
1823 \fi
1824 %</article|book|report|slide>
1825 %<*jspf>
1826 \newcommand{\maketitle}{\par
1827   \begingroup
1828     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1829     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1830     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1831       \parindent 1\jsZw\noindent
1832       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1833     \twocolumn[\@maketitle]%
1834     \plainifnotempty
1835     \@thanks
1836   \endgroup
1837   \setcounter{footnote}{0}%
1838   \global\let\thanks\relax
1839   \global\let\maketitle\relax
1840   \global\let\@thanks\@empty
1841   \global\let\@author\@empty
1842   \global\let\@date\@empty
1843 % \global\let\@title\@empty % \@title は柱に使う
1844   \global\let\title\relax
1845   \global\let\author\relax
1846   \global\let\date\relax
1847   \global\let\and\relax
```

```

1848 \ifx\authors@mail\@undefined\else{%
1849 \def\@makefnctext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1850 \footnotetext[0]{\itshape\authors@mail}%
1851 }\fi
1852 \global\let\authors@mail\@undefined}
1853 \def\@maketitle{%
1854 \newpage\null
1855 \vskip 6em % used to be 2em
1856 \begin{center}
1857 \let\footnote\thanks
1858 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1859 \lineskip .5em
1860 \ifx\@author\@undefined\else
1861 \vskip 1em
1862 \begin{tabular}[t]{c}%
1863 \@author
1864 \end{tabular}\par
1865 \fi
1866 \ifx\@etitle\@undefined\else
1867 \vskip 1em
1868 {\large \@etitle \par}%
1869 \fi
1870 \ifx\@eauthor\@undefined\else
1871 \vskip 1em
1872 \begin{tabular}[t]{c}%
1873 \@eauthor
1874 \end{tabular}\par
1875 \fi
1876 \vskip 1em
1877 \@date
1878 \end{center}
1879 \vskip 1.5em
1880 \centerline{\box\@abstractbox}
1881 \ifx\@keywords\@undefined\else
1882 \vskip 1.5em
1883 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1884 \fi
1885 \vskip 1.5em}
1886 %</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。

```
1887 \def\bxjs@label@sect#1{%
```

```

1888 \expandafter\ifx\csname label#1\endcsname\relax
1889 \csname the#1\endcsname
1890 \else \csname label#1\endcsname
1891 \fi}
1892 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1893 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1894 \def\@secapp{\presectionname}
1895 \def\@secpos{\postsectionname}
1896 \fi

```

`\labelsection` 節番号の出力書式。

```

1897 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1898 \def\labelsection{\@secapp\thesection\@secpos}
1899 \fi

```

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

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

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```

1900 \def\@startsection#1#2#3#4#5#6{%
1901   \if@noskipsec \leavevmode \fi
1902   \par
1903 % 見出し上の空きを \@tempskipa にセットする
1904   \@tempskipa #4\relax
1905 % \@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
1906   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1907 % 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
1908   \ifdim \@tempskipa <\z@
1909     \@tempskipa -\@tempskipa \@afterindentfalse
1910   \fi
1911   \if@nobreak
1912 %   \everypar{\everyparhook}% これは間違い
1913     \everypar{}%
1914   \else
1915     \addpenalty\@secpenalty
1916 % 次の行は削除
1917 %   \addvspace\@tempskipa
1918 % 次の \noindent まで追加
1919     \ifdim \@tempskipa >\z@
1920       \if@slide\else
1921         \null
1922         \vspace*{-\baselineskip}%
1923       \fi
1924       \vskip\@tempskipa
1925     \fi
1926   \fi
1927   \noindent
1928 % 追加終わり
1929   \@ifstar
1930     {\@ssect{#3}{#4}{#5}{#6}}%
1931     {\@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のように、多少変えてあります。 \everyparhook も挿入しています。

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

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

```

1932 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1933   \let\bxjs@if@ceph\@firstofone
1934 \else \let\bxjs@if@ceph\@gobble
1935 \fi

```

```

1936 \def\@sect#1#2#3#4#5#6[#7]#8{%
1937   \ifnum #2>\c@secnumdepth
1938     \let\@svsec\@empty
1939   \else

```



```

1940 \refstepcounter{#1}%
1941 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1942 \fi
1943 % 見出し後の空きを \@tempskipa にセット
1944 \@tempskipa #5\relax
1945 % 条件判断の順序を入れ替えました
1946 \ifdim \@tempskipa<\z@
1947 \def\@svsechd{%
1948 #6{\hskip #3\relax
1949 \@svsec #8}%
1950 \csname #1mark\endcsname{#7}%
1951 \addcontentsline{toc}{#1}{%
1952 \ifnum #2>\c@secnumdepth \else
1953 \protect\numberline{\bxjs@label@sect{#1}}%
1954 \fi
1955 #7}}% 目次にフルネームを載せるなら #8
1956 \else
1957 \begingroup
1958 \interlinepenalty \@M % 下から移動
1959 #6{%
1960 \@hangfrom{\hskip #3\relax\@svsec}%
1961 % \interlinepenalty \@M % 上に移動
1962 #8\@@par}%
1963 \endgroup
1964 \csname #1mark\endcsname{#7}%
1965 \addcontentsline{toc}{#1}{%
1966 \ifnum #2>\c@secnumdepth \else
1967 \protect\numberline{\bxjs@label@sect{#1}}%
1968 \fi
1969 #7}% 目次にフルネームを載せるならここは #8
1970 \fi
1971 \@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。

```

1972 \def\@xsect#1{%
1973 % 見出しの後ろの空きを \@tempskipa にセット
1974 \@tempskipa #1\relax
1975 % 条件判断の順序を変えました
1976 \ifdim \@tempskipa<\z@
1977 \nobreakfalse
1978 \global\@noskipsecttrue
1979 \everypar{%
1980 \if@noskipsec
1981 \global\@noskipsecfalse
1982 {\setbox\z@\lastbox}%

```

```

1983      \clubpenalty\@M
1984      \begingroup \@svsechd \endgroup
1985      \unskip
1986      \@tempskipa #1\relax
1987      \hskip -\@tempskipa
1988      \else
1989      \clubpenalty \@clubpenalty
1990      \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1991      \fi\bxjs@if@ceph\everyparhook}%
1992 \else
1993      \par \nobreak
1994      \vskip \@tempskipa
1995      \@afterheading
1996 \fi
1997 \if@slide
1998      {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1999      \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
2000      \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
2001 \fi
2002 \par % 2000-12-18
2003 \ignorespaces}
2004 \def\@ssect#1#2#3#4#5{%
2005     \@tempskipa #3\relax
2006     \ifdim \@tempskipa<\z@
2007         \def\@svsechd{#4{\hskip #1\relax #5}}%
2008     \else
2009         \begingroup
2010             #4{%
2011                 \@hangfrom{\hskip #1}%
2012                 \interlinepenalty \@M #5\@@par}%
2013         \endgroup
2014     \fi
2015     \@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 \LaTeX 本体で定義済みです。

```

\subsectionmark 2016 \newcommand*\chaptermark[1]{}
\subsubsectionmark 2017 % \newcommand*\sectionmark[1]{}
2018 % \newcommand*\subsectionmark[1]{}
\paragraphmark 2019 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 2020 % \newcommand*\paragraphmark[1]{}
2021 % \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。

```
2022 %<!book&!report>\setcounter{secnumdepth}{3}
2023 %<book|report>\setcounter{secnumdepth}{2}
```

`\c@chapter` 見出し番号のカウントです。`\newcounter` の第 1 引数が新たに作るカウントです。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トです。

```
\c@subsection 2024 \newcounter{part}
2025 %<book|report>\newcounter{chapter}
\c@subsubsection 2026 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2027 %<!book&!report>\newcounter{section}
\c@subparagraph 2028 \newcounter{subsection}[section]
2029 \newcounter{subsubsection}[subsection]
2030 \newcounter{paragraph}[subsubsection]
2031 \newcounter{subparagraph}[paragraph]
```

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

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

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```
2032 \renewcommand{\thepart}{\@Roman\c@part}
2033 %<*!book&!report>
2034 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2035 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2036 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2037 \else
2038 \renewcommand{\thesection}{\@arabic\c@section}
2039 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2040 \fi
2041 %</!book&!report>
2042 %<*book|report>
2043 \renewcommand{\thechapter}{\@arabic\c@chapter}
2044 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2045 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2046 %</book|report>
2047 \renewcommand{\thesubsubsection}{\%
2048   \thesubsection.\@arabic\c@subsubsection}
2049 \renewcommand{\theparagraph}{\%
2050   \thesubsubsection.\@arabic\c@paragraph}
2051 \renewcommand{\thesubparagraph}{\%
2052   \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。

```
2053 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2054 %<book|report>\newcommand{\@chappos}{\postchaptername}
```

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

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

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

```
2055 %<*book|report>
2056 \newcommand\frontmatter{%
2057   \pltx@cleartooddpage
2058   \@mainmatterfalse
2059   \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。

```
2060 \newcommand\mainmatter{%
2061   \pltx@cleartooddpage
2062   \@mainmattertrue
2063   \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。

```
2064 \newcommand\backmatter{%
2065   \if@openleft
2066     \cleardoublepage
2067   \else\if@openright
2068     \cleardoublepage
2069   \else
2070     \clearpage
2071   \fi\fi
2072   \@mainmatterfalse}
2073 %</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。

```
2074 %<!*book&!report>
2075 \newcommand\part{%
2076   \if@noskipsec \leavevmode \fi
2077   \par
2078   \addvspace{4ex}%
2079   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2080   \secdef\@part\@spart}
2081 %</!*book&!report>
```

`book` および `report` クラスの場合は、少し複雑です。

```
2082 %<*book|report>
2083 \newcommand\part{%
2084   \if@openleft
2085     \cleardoublepage
2086   \else\if@openright
2087     \cleardoublepage
2088   \else
2089     \clearpage
2090   \fi\fi
2091   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2092   \if@twocolumn
2093     \onecolumn
2094     \@restonecoltrue
2095   \else
2096     \@restonecolfalse
2097   \fi
2098   \null\vfil
2099   \secdef\@part\@spart}
2100 %</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。

```
2101 %<!*book&!report>
2102 \def\@part[#1]#2{%
2103   \ifnum \c@secnumdepth >\m@ne
2104     \refstepcounter{part}%
2105     \addcontentsline{toc}{part}{%
2106       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2107   \else
```

```

2108 \addcontentsline{toc}{part}{#1}%
2109 \fi
2110 \markboth{}{}%
2111 {\parindent\z@
2112 \raggedright
2113 \interlinepenalty \@M
2114 \normalfont
2115 \ifnum \c@secnumdepth >\m@ne
2116 \Large\headfont\prepartname\thepart\postpartname
2117 \par\nobreak
2118 \fi
2119 \huge \headfont #2%
2120 \markboth{}{}\par}%
2121 \nobreak
2122 \vskip 3ex
2123 \@afterheading}
2124 %</*!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。

```

2125 %<*book|report>
2126 \def\@part[#1]#2{%
2127 \ifnum \c@secnumdepth >-2\relax
2128 \refstepcounter{part}%
2129 \addcontentsline{toc}{part}{%
2130 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2131 \else
2132 \addcontentsline{toc}{part}{#1}%
2133 \fi
2134 \markboth{}{}%
2135 {\centering
2136 \interlinepenalty \@M
2137 \normalfont
2138 \ifnum \c@secnumdepth >-2\relax
2139 \huge\headfont \prepartname\thepart\postpartname
2140 \par\vskip20\p@?
2141 \fi
2142 \Huge \headfont #2\par}%
2143 \@endpart}
2144 %</book|report>

```

\@spart 番号を付けない部です。

```

2145 %<*!book&!report>
2146 \def\@spart#1{%
2147 \parindent \z@ \raggedright
2148 \interlinepenalty \@M
2149 \normalfont
2150 \huge \headfont #1\par}%
2151 \nobreak
2152 \vskip 3ex

```

```

2153 \afterheading}
2154 %</!book&!report>
2155 %<*book|report>
2156 \def\@spart#1{%
2157     \centering
2158     \interlinepenalty \@M
2159     \normalfont
2160     \Huge \headfont #1\par}%
2161 \@endpart}
2162 %</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。

```

2163 %<*book|report>
2164 \def\@endpart{\vfil\newpage
2165     \if@twoside
2166     \if@openleft %% added (2017/02/24)
2167         \null\thispagestyle{empty}\newpage
2168     \else\if@openright %% added (2016/12/13)
2169         \null\thispagestyle{empty}\newpage
2170     \fi\fi %% added (2016/12/13, 2017/02/24)
2171     \fi
2172     \if@restonecol
2173         \twocolumn
2174     \fi}
2175 %</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。

```

2176 %<*book|report>
2177 \newcommand{\chapter}{%
2178     \if@openleft\cleardoublepage\else
2179     \if@openright\cleardoublepage\else\clearpage\fi\fi
2180     \plainifnotempty % 元: \thispagestyle{plain}
2181     \global\@topnum\z@
2182     \if@english \@afterindentfalse \else \@afterindenttrue \fi
2183     \secdef
2184         {\@omit@numberfalse\@chapter}%
2185         {\@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。

```

2186 \def\@chapter[#1]#2{%
2187   \ifnum \c@secnumdepth >\m@ne
2188     \if@mainmatter
2189       \refstepcounter{chapter}%
2190       \typeout{\@chapapp\thechapter\@chappos}%
2191       \addcontentsline{toc}{chapter}%
2192         {\protect\numberline
2193 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2194         {\@chapapp\thechapter\@chappos}%
2195         #1}%
2196     \else\addcontentsline{toc}{chapter}{#1}\fi
2197   \else
2198     \addcontentsline{toc}{chapter}{#1}%
2199   \fi
2200   \chaptermark{#1}%
2201   \addtocontents{lof}{\protect\addvspace{10\jsc@mp}}%
2202   \addtocontents{lot}{\protect\addvspace{10\jsc@mp}}%
2203   \if@twocolumn
2204     \@topnewpage[\@makechapterhead{#2}]%
2205   \else
2206     \@makechapterhead{#2}%
2207     \@afterheading
2208   \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。

```

2209 \def\@makechapterhead#1{%
2210   \vspace*{2\Cvs}% 欧文は 50pt
2211   {\parindent \z@ \raggedright \normalfont
2212     \ifnum \c@secnumdepth >\m@ne
2213       \if@mainmatter
2214         \huge\headfont \@chapapp\thechapter\@chappos
2215         \par\nobreak
2216         \vskip \Cvs % 欧文は 20pt
2217       \fi
2218     \fi
2219     \interlinepenalty\@M
2220     \Huge \headfont #1\par\nobreak
2221     \vskip 3\Cvs}} % 欧文は 40pt

```

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

```

2222 \def\@schapter#1{%
2223   \chaptermark{#1}%
2224   \if@twocolumn
2225     \@topnewpage[\@makeschapterhead{#1}]%
2226   \else
2227     \@makeschapterhead{#1}\@afterheading
2228   \fi}

```

\@makeschapterhead 番号なしの章見出しです。


```

2229 \def\@makeschapterhead#1{%
2230   \vspace*{2\Cvs}% 欧文は 50pt
2231   {\parindent \z@ \raggedright
2232     \normalfont
2233     \interlinepenalty\@M
2234     \Huge \headfont #1\par\nobreak
2235     \vskip 3\Cvs}} % 欧文は 40pt
2236 %</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。

```

2237 \if@twocolumn
2238   \newcommand{\section}{%
2239     <jspf>\ifx\maketitle\relax\else\maketitle\fi
2240     \@startsection{section}{1}{\z@}%
2241     <kiyou> {0.6\Cvs}{0.4\Cvs}%
2242     <kiyou> {\Cvs}{0.5\Cvs}%
2243     % {\normalfont\large\headfont\@secapp}%
2244     {\normalfont\large\headfont\raggedright}}
2245 \else
2246   \newcommand{\section}{%
2247     \if@slide\clearpage\fi
2248     \@startsection{section}{1}{\z@}%
2249     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2250     {.5\Cvs \@plus.3\Cdp}% 後アキ
2251     % {\normalfont\Large\headfont\@secapp}%
2252     {\normalfont\Large\headfont\raggedright}}
2253 \fi

```

`\subsection` 同上です。

```

2254 \if@twocolumn
2255   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2256     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2257     {\normalfont\normalsize\headfont}}
2258 \else
2259   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2260     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2261     {.5\Cvs \@plus.3\Cdp}% 後アキ
2262     {\normalfont\large\headfont}}
2263 \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)。

```

2264 \if@twocolumn
2265   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%

```

```

2266     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2267     {\normalfont\normalsize\headfont}}
2268 \else
2269   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2270     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2271     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2272     {\normalfont\normalsize\headfont}}
2273 \fi

```

`\paragraph` 見出しの後ろで改行されません。

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

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

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

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

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

```

2274 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2275 \let\bxjs@org@paragraph@mark\jsParagraphMark
2276 \ifx\bxjs@paragraph@mark\@empty
2277   \let\jsParagraphMark\@empty
2278 \else\ifx\bxjs@paragraph@mark\@undefined\else
2279   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2280 \fi\fi
2281 \let\jsJaChar\@empty
2282 \if@twocolumn
2283   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2284     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2285     %<jspf> {\normalfont\normalsize\headfont}}
2286     %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2287 \else
2288   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2289     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2290     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2291     %<jspf> {\normalfont\normalsize\headfont}}
2292     %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2293 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```
2294 \if@twocolumn
```

```

2295 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2296     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2297     {\normalfont\normalsize\headfont}}
2298 \else
2299 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2300     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2301     {\normalfont\normalsize\headfont}}
2302 \fi

```

8.3 リスト環境

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

`\leftmargini` 二段組であるかないかに応じてそれぞれ 2em, 2.5em でしたが、ここでは全角幅の 2 倍にしました。

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2303 \if@slide
2304   \setlength\leftmargini{1\jsZw}
2305 \else
2306   \if@twocolumn
2307     \setlength\leftmargini{2\jsZw}
2308   \else
2309     \setlength\leftmargini{3\jsZw}
2310   \fi
2311 \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 2312 \if@slide
\leftmarginiv 2313   \setlength\leftmarginii {1\jsZw}
\leftmarginiv 2314   \setlength\leftmarginiii{1\jsZw}
\leftmarginiv 2315   \setlength\leftmarginiv {1\jsZw}
\leftmarginiv 2316   \setlength\leftmarginv  {1\jsZw}
\leftmarginiv 2317   \setlength\leftmarginvi {1\jsZw}
\leftmarginiv 2318 \else
\leftmarginiv 2319   \setlength\leftmarginii {2\jsZw}
\leftmarginiv 2320   \setlength\leftmarginiii{2\jsZw}
\leftmarginiv 2321   \setlength\leftmarginiv {2\jsZw}
\leftmarginiv 2322   \setlength\leftmarginv  {1\jsZw}
\leftmarginiv 2323   \setlength\leftmarginvi {1\jsZw}
\leftmarginiv 2324 \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。

```

2325 \setlength \labelsep {0.5\jsZw} % .5em

```

```
2326 \setlength \labelwidth{\leftmargini}
2327 \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。

```
2328 \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 2329 \@beginparpenalty -\@lowpenalty
\@itempenalty 2330 \@endparpenalty -\@lowpenalty
2331 \@itempenalty -\@lowpenalty
```

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

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

```
2332 \def\@listi{\leftmargin\leftmargini
2333 \parsep \z@
2334 \topsep 0.5\baselineskip
2335 \itemsep \z@ \relax}
2336 \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）。

```
2337 \@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 2338 \def\@listii{\leftmargin\leftmarginii
\@listiv 2339 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2340 \topsep \z@
\@listv 2341 \parsep \z@
\@listvi 2342 \itemsep\parsep}
2343 \def\@listiii{\leftmargin\leftmarginiii
2344 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2345 \topsep \z@
2346 \parsep \z@
2347 \itemsep\parsep}
2348 \def\@listiv {\leftmargin\leftmarginiv
2349 \labelwidth\leftmarginiv
2350 \advance\labelwidth-\labelsep}
2351 \def\@listv {\leftmargin\leftmarginv
2352 \labelwidth\leftmarginv
2353 \advance\labelwidth-\labelsep}
2354 \def\@listvi {\leftmargin\leftmarginvi
```

```

2355 \labelwidth\leftmarginvi
2356 \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。

```

2357 \renewcommand{\theenumi}{\@arabic\c@enumi}
2358 \renewcommand{\theenumii}{\@alph\c@enumii}
2359 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2360 \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。

```

2361 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2362 \newcommand*{\jsInJaParen}[1]{%
2363   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2364 \newcommand{\labelenumi}{\theenumi.}
2365 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2366 \newcommand{\labelenumiii}{\theenumiii.}
2367 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2368 \renewcommand{\p@enumii}{\theenumi}
2369 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2370 \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 2371 \newcommand\labelitemi{\textbullet}
2372 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 2373 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2374 \newcommand\labelitemiv{\textperiodcentered}

```

■description 環境

description 本来の **description** 環境では、項目名が短いと、説明部分の頭がそれに引きずられて左に出てしまいます。これを解決した新しい **description** の実装です。

```
2375 \newenvironment{description}{%
2376   \list{}{%
2377     \labelwidth=\leftmargin
2378     \labelsep=1\jsZw
2379     \advance \labelwidth by -\labelsep
2380     \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。

```
2381 \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。

```
2382 %<*book|report>
2383 \newenvironment{chapterabstract}{%
2384   \begin{list}{}{%
2385     \listparindent=1\jsZw
2386     \itemindent=\listparindent
2387     \rightmargin=0pt
2388     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2389 %</book|report>
```

“普通の” **abstract** 環境の定義。

```
2390 %<*article|report|slide>
2391 \newbox\@abstractbox
2392 \if@titlepage
2393   \newenvironment{abstract}{%
```

```

2394 \titlepage
2395 \null\vfil
2396 \@beginparpenalty\@lowpenalty
2397 \begin{center}%
2398 \headfont \abstractname
2399 \@endparpenalty\@M
2400 \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。

```

2401 \par}%
2402 {\par\vfil\null\endtitlepage}
2403 \else
2404 \newenvironment{abstract}{%
2405 \if@twocolumn
2406 \ifx\maketitle\relax
2407 \section*{\abstractname}%
2408 \else
2409 \global\setbox\@abstractbox\hbox\bgroup
2410 \begin{minipage}[b]{\textwidth}
2411 \small\parindent1\jsZw
2412 \begin{center}%
2413 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2414 \end{center}%
2415 \list{}{%
2416 \listparindent\parindent
2417 \itemindent \listparindent
2418 \rightmargin \leftmargin}%
2419 \item\relax
2420 \fi
2421 \else
2422 \small
2423 \begin{center}%
2424 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2425 \end{center}%
2426 \list{}{%
2427 \listparindent\parindent
2428 \itemindent \listparindent
2429 \rightmargin \leftmargin}%
2430 \item\relax
2431 \fi}{\if@twocolumn
2432 \ifx\maketitle\relax
2433 \else
2434 \endlist\end{minipage}\egroup
2435 \fi
2436 \else
2437 \endlist
2438 \fi}
2439 \fi
2440 %</article|report|slide>

```

```

2441 %<*jspf>
2442 \newbox\@abstractbox
2443 \newenvironment{abstract}{%
2444   \global\setbox\@abstractbox\hbox\bgroup
2445   \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2446     \small
2447     \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2448   {\end{minipage}\egroup}
2449 %</jspf>

```

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

```

2450 %<*book|report>
2451 \ifbxjs@force@chapterabstract
2452   \let\abstract\chapterabstract
2453   \let\endabstract\endchapterabstract
2454 \fi
2455 %</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。

```

2456 %<*jspf>
2457 %\newbox\@keywordsbox
2458 %\newenvironment{keywords}{%
2459 %  \global\setbox\@keywordsbox\hbox\bgroup
2460 %  \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2461 %    \small\parindent0\jsZw}%
2462 %  {\end{minipage}\egroup}
2463 %</jspf>

```

■verse 環境

verse 詩のための `verse` 環境です。

```

2464 \newenvironment{verse}{%
2465   \let \\\=\@centercr
2466   \list{}{%
2467     \itemsep \z@
2468     \itemindent -2\jsZw % 元: -1.5em
2469     \listparindent\itemindent
2470     \rightmargin \z@
2471     \advance\leftmargin 2\jsZw}% 元: 1.5em
2472   \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。

```
2473 \newenvironment{quotation}{%
2474   \list{}{%
2475     \listparindent\parindent
2476     \itemindent\listparindent
2477     \rightmargin \z@}%
2478   \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。

```
2479 \newenvironment{quote}%
2480   {\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。

```
2481 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2482   \item[\hskip \labelsep{\headfont #1\ #2}]}
2483 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2484   \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。

```
2485 \newenvironment{titlepage}{%
2486   %<book> \pltx@cleartooddpage %% 2017-02-24
2487   \if@twocolumn
2488     \@restonecoltrue\onecolumn
2489   \else
2490     \@restonecolfalse\newpage
2491   \fi
2492   \thispagestyle{empty}}%
```

```

2493 \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2494 24
2495 }%
2496 {\if@restonecol\twocolumn \else \newpage \fi
2497 \if@twoside\else
2498 \setcounter{page}\@ne
2499 \fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```

2499 %<*!book&!report>
2500 \newcommand{\appendix}{\par
2501 \setcounter{section}{0}%
2502 \setcounter{subsection}{0}%
2503 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2504 \gdef\presectionname{\appendixname}%
2505 \gdef\postsectionname{}}%
2506 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2507 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2508 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2509 \else
2510 \gdef\@secapp{\appendixname}%
2511 \gdef\@secpos{}}%
2512 \gdef\thesection{\@Alph\c@section}%
2513 \fi}
2514 %</!book&!report>
2515 %<*book|report>
2516 \newcommand{\appendix}{\par
2517 \setcounter{chapter}{0}%
2518 \setcounter{section}{0}%
2519 \gdef\@chapapp{\appendixname}%
2520 \gdef\@chappos{}}%
2521 \gdef\thechapter{\@Alph\c@chapter}}
2522 %</book|report>

```

8.4 パラメータの設定

■array と tabular 環境

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

```
2523 \setlength\arraycolsep{5\p@?}
```

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

```
2524 \setlength\tabcolsep{6\p@?}
```

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

```
2525 \setlength\arrayrulewidth{.4\p@}
```

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

```
2526 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` `\'` コマンドで入るアキです。

```
2527 \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。

```
2528 \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。

```
2529 \setlength\fboxsep{3\p@?}
```

```
2530 \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。

```
2531 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```
2532 %<*book|report>
```

```
2533 \@addtoreset{equation}{chapter}
```

```
2534 \renewcommand\theequation
```

```
2535 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

```
2536 %</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。

```
2537 % \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。

```
2538 % \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。

```
2539 % \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。

```
2540 %<!*book&!report>
2541 \newcounter{figure}
2542 \renewcommand \thefigure {\@arabic\c@figure}
2543 %<!/book&!report>
2544 %<*book|report>
2545 \newcounter{figure}[chapter]
2546 \renewcommand \thefigure
2547     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2548 %</book|report>
```

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

```
\ext@figure 2549 \def\fps@figure{tbp}
2550 \def\ftype@figure{1}
\fnum@figure 2551 \def\ext@figure{lof}
2552 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2553 \newenvironment{figure}%
2554     {\@float{figure}}%
2555     {\end@float}
2556 \newenvironment{figure*}%
2557     {\@dblfloat{figure}}%
2558     {\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。

```

2559 %<!*book&!report>
2560 \newcounter{table}
2561 \renewcommand\thetable{\@arabic\c@table}
2562 %</!*book&!report>
2563 %<*book|report>
2564 \newcounter{table}[chapter]
2565 \renewcommand \thetable
2566     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2567 %</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 2568 \def\fps@table{tbp}
2569 \def\ftype@table{2}
\fnun@table 2570 \def\ext@table{lot}
2571 \def\fnun@table{\tablename\nobreak\thetable}

```

table * は段抜きのフロートです。

```

table* 2572 \newenvironment{table}%
2573     {\@float{table}}%
2574     {\end@float}
2575 \newenvironment{table*}%
2576     {\@dblfloat{table}}%
2577     {\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
\belowcaptionskip 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。

```

2578 \newlength\abovecaptionskip
2579 \newlength\belowcaptionskip
2580 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2581 \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。

```

2582 %<!*jspf>
2583 % \long\def\@makecaption#1#2{\small
2584 %     \advance\leftskip10\jsc@mmm
2585 %     \advance\rightskip10\jsc@mmm
2586 %     \vskip\abovecaptionskip
2587 %     \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%

```

```

2588 % \ifdim \wd\@tempboxa >\hsize
2589 % #1\hskip1\jsZw\relax #2\par
2590 % \else
2591 % \global \@minipagefalse
2592 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2593 % \fi
2594 % \vskip\belowcaptionskip}}
2595 \long\def\@makecaption#1#2{\small
2596 \advance\leftskip .0628\linewidth
2597 \advance\rightskip .0628\linewidth
2598 \vskip\abovecaptionskip
2599 \sbox\@tempboxa{#1\zwspace#2}%
2600 \ifdim \wd\@tempboxa <\hsize \centering \fi
2601 #1\zwspace#2\par
2602 \vskip\belowcaptionskip}}
2603 %</!jspf>
2604 %<*jspf>
2605 \long\def\@makecaption#1#2{%
2606 \vskip\abovecaptionskip
2607 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2608 \ifdim \wd\@tempboxa >\hsize
2609 {\small\sffamily
2610 \list{#1}{%
2611 \renewcommand{\makelabel}[1]{##1\hfil}
2612 \itemsep \z@
2613 \itemindent \z@
2614 \labelsep \z@
2615 \labelwidth 11\jsc@mmm
2616 \listparindent\z@
2617 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2618 \else
2619 \global \@minipagefalse
2620 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2621 \fi
2622 \vskip\belowcaptionskip}
2623 %</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@warnoldfontcmd` は BXJS クラスでは不使用。

`\if@jsc@warnoldfontcmdexception` `\if@jsc@warnoldfontcmdexception` は `\allow/disallowoldfontcommands` の状態を表す。

```

2624 \newif\if@jsc@warnoldfontcmd
2625 \@jsc@warnoldfontcmdtrue
2626 \newif\if@jsc@warnoldfontcmdexception
2627 \@jsc@warnoldfontcmdexceptionfalse

```

`\jsc@DeclareOldFontCommand`

```

2628 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2629   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2630   \DeclareOldFontCommand{#1}{%
2631     \bxjs@oldfontcmd{#1}#2%
2632   }{%
2633     \bxjs@oldfontcmd{#1}#3%
2634   }%
2635 }
2636 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2637   \ClassInfo\bxjs@clsname
2638   {Old font command '\string#1' is used!!\MessageBreak
2639     The first occurrence is}%
2640 }

```

`\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。

```

2641 \newcommand*{\allowoldfontcommands}{%
2642   \@jsc@warnoldfontcmdexceptiontrue}
2643 \newcommand*{\disallowoldfontcommands}{%
2644   \@jsc@warnoldfontcmdexceptionfalse}

2645 \let\bxjs@oldfontcmd@list\@empty
2646 \def\bxjs@oldfontcmd#1{%
2647   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2648 \def\bxjs@oldfontcmd@a#1#2{%
2649   \if@jsc@warnoldfontcmdexception\else
2650     \global\@jsc@warnoldfontcmdfalse
2651     \ifx#1\relax
2652       \global\let#1=t%
2653       \jsc@warnoldfontcmd{#2}%
2654     \fi
2655   \fi}
2656 \def\bxjs@warnoldfontcmd@final{%
2657 % \par
2658 \global\let\bxjs@warnoldfontcmd@final\@empty

```

```

2659 \let\@tempa\@empty
2660 \def\do##1{%
2661   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2662     \edef\@tempa{\@tempa \space\string##1}\fi}
2663 \bxjs@oldfontcmd@list
2664 \ifx\@tempa\@empty\else
2665   \ClassWarningNoLine\bxjs@clsname
2666     {Some old font commands were used in text:\MessageBreak
2667       \space\@tempa\MessageBreak
2668       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2669       new font selection scheme called NFSS2 with several\MessageBreak
2670       new, combinable font commands. The
2671       class provides\MessageBreak
2672       the old font commands only for compatibility}
2673 \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。

```

2674 \def\bxjs@warnoldfontcmd@kick@final{%
2675   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2676 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2677 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2678 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 2679 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf 2680 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2681 \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。

```

2682 \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。

```

2683 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2684 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2685 \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 2686 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2687 \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` \geq `\@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。

```
2688 \newcommand\@pnumwidth{1.55em}
2689 \newcommand\@tocrmarg{2.55em}
2690 \newcommand\@dotsep{4.5}
2691 %<!book&!report>\setcounter{tocdepth}{2}
2692 %<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)

```
2693 \newdimen\jsc@tocl@width
2694 \newcommand{\tableofcontents}{%
2695 %<*book|report>
2696   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2697   \settowidth\@tempdima{\headfont\appendixname}%
2698   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2699   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2700   \if@twocolumn
2701     \@restonecoltrue\onecolumn
2702   \else
2703     \@restonecolfalse
2704   \fi
2705   \chapter*{\contentsname}%
2706   \@mkboth{\contentsname}{}%
2707 %</book|report>
2708 %<*!book&!report>
2709   \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2710   \settowidth\@tempdima{\headfont\appendixname}%
2711   \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2712   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2713   \section*{\contentsname}%
2714   \@mkboth{\contentsname}{\contentsname}%
2715 %</!book&!report>
2716   \@starttoc{toc}%
2717 %<book|report> \if@restonecol\twocolumn\fi
2718 }
```

`\l@part` 部の目次です。

```
2719 \newcommand*{\l@part}[2]{%
2720   \ifnum \c@tocdepth >-2\relax
2721   %<!book&!report> \addpenalty\@secpenalty
2722   %<book|report> \addpenalty{-\@highpenalty}%
2723   \addvspace{2.25em \@plus\p@?}%
2724   \begingroup
2725     \parindent \z@
2726   % \@pnumwidth should be \@tocrmarg
2727   % \rightskip \@pnumwidth
2728   \rightskip \@tocrmarg
2729   \parfillskip -\rightskip
2730   {\leavevmode
2731     \large \headfont
2732     \setlength\@lnumwidth{4\jsZw}%
2733     #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2734   \nobreak
2735   %<book|report> \global\@nobreaktrue
2736   %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
```

```

2737 \endgroup
2738 \fi}

```

\l@chapter 章の目次です。 \@l@numwidth を 4.683zw に増やしました。

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

```

2739 %<*book|report>
2740 \newcommand*{\l@chapter}[2]{%
2741 \ifnum \c@tocdepth >\m@ne
2742 \addpenalty{-\@highpenalty}%
2743 \addvspace{1.0em \@plus\p@}%
2744 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2745 \begingroup
2746 \parindent\z@
2747 % \rightskip\@pnumwidth
2748 \rightskip\@tocrmarg
2749 \parfillskip-\rightskip
2750 \leavevmode\headfont
2751 % % \if@english\setlength\@l@numwidth{5.5em}\else\setlength\@l@numwidth{4.683\jsZw}\fi
2752 \setlength\@l@numwidth{\jsc@tocl@width}\advance\@l@numwidth 2.683\jsZw
2753 \advance\leftskip\@l@numwidth \hskip-\leftskip
2754 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2755 \penalty\@highpenalty
2756 \endgroup
2757 \fi}
2758 %</book|report>

```

\l@section 節の目次です。

```

2759 %<!*book&!report>
2760 \newcommand*{\l@section}[2]{%
2761 \ifnum \c@tocdepth >\z@
2762 \addpenalty{\@secpenalty}%
2763 \addvspace{1.0em \@plus\p@}%
2764 \begingroup
2765 \parindent\z@
2766 % \rightskip\@pnumwidth
2767 \rightskip\@tocrmarg
2768 \parfillskip-\rightskip
2769 \leavevmode\headfont
2770 % % \setlength\@l@numwidth{4\jsZw}% 元 1.5em [2003-03-02]
2771 \setlength\@l@numwidth{\jsc@tocl@width}\advance\@l@numwidth 2\jsZw
2772 \advance\leftskip\@l@numwidth \hskip-\leftskip
2773 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2774 \endgroup
2775 \fi}
2776 %</!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。

```

2777 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

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

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

```
\l@subparagraph 2778 %<!*book&!report>
2779 % \newcommand*{\l@section} {\@dottedtocline{2}{1.5em}{2.3em}}
2780 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2781 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2782 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2783 %
2784 % \newcommand*{\l@section} {\@dottedtocline{2}{1zw}{3zw}}
2785 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2786 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2787 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2788 %
2789 \newcommand*{\l@section}{%
2790     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2791     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2792 \newcommand*{\l@subsubsection}{%
2793     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2794     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2795 \newcommand*{\l@paragraph}{%
2796     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2797     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2798 \newcommand*{\l@subparagraph}{%
2799     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2800     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2801 %</!*book&!report>
2802 %<!*book|report>
2803 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2804 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2805 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2806 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2807 \newcommand*{\l@section}{%
2808     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2809     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2810 \newcommand*{\l@section}{%
2811     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2812     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2813 \newcommand*{\l@subsubsection}{%
2814     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2815     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2816 \newcommand*{\l@paragraph}{%
2817     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2818     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2819 \newcommand*{\l@subparagraph}{%
2820     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
```

```

2821 \dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2822 %</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。

```

2823 \newdimen\@lnumwidth
2824 \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 `\jsTocLine` 変えています。

[2018-06-23] デフォルトでは のようにベースラインになります。これを変更可能にするため、`\jsTocLine` というマクロに切り出しました。例えば、仮想ボディの中央 に変更したい場合は

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

とします。

```

2825 \def\jsTocLine{\leaders\hbox{%
2826   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2827 \def\dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2828   \vskip \z@ \@plus.2\p@?
2829   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2830    \parindent #2\relax\@afterindenttrue
2831    \interlinepenalty\@M
2832    \leavevmode
2833    \@lnumwidth #3\relax
2834    \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2835    {#4}\nobreak
2836    \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2837      \hfil\normalfont \normalcolor #5\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2838 \newcommand{\listoffigures}{%
2839 %<*book|report>
2840 \if@twocolumn\@restonecoltrue\onecolumn
2841 \else\@restonecolfalse\fi
2842 \chapter*{\listfigurename}%
2843 \mkboth{\listfigurename}{}%
2844 %</book|report>
2845 %<!*book&!report>
2846 \section*{\listfigurename}%
2847 \mkboth{\listfigurename}{\listfigurename}%
2848 %</*!book&!report>
2849 \@starttoc{lof}%

```

```

2850 %<book|report> \if@restonecol\twocolumn\fi
2851 }

```

`\l@figure` 図目次の項目を出力します。

```

2852 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2853 \newcommand{\listoftables}{%
2854 %<*book|report>
2855 \if@twocolumn\@restonecoltrue\onecolumn
2856 \else\@restonecolfalse\fi
2857 \chapter*{\listtablename}%
2858 \@mkboth{\listtablename}{}%
2859 %</book|report>
2860 %<!*book&!report>
2861 \section*{\listtablename}%
2862 \@mkboth{\listtablename}{\listtablename}%
2863 %</!*book&!report>
2864 \@starttoc{lot}%
2865 %<book|report> \if@restonecol\twocolumn\fi
2866 }

```

`\l@table` 表目次は図目次と同じです。

```

2867 \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。

```

2868 \newdimen\bibindent
2869 \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。

```

2870 \newenvironment{thebibliography}[1]{%
2871 \@jsc@warnoldfontcmdexceptiontrue
2872 \global\let\presectionname\relax
2873 \global\let\postsectionname\relax
2874 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2875 %<*kiyou>
2876 \vspace{1.5\baselineskip}
2877 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2878 \vspace{0.5\baselineskip}
2879 %</kiyou>
2880 %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2881 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%

```

```

2882 \list{\@biblabel{\@arabic\c@enumiv}}%
2883 {\settowidth\labelwidth{\@biblabel{#1}}%
2884 \leftmargin\labelwidth
2885 \advance\leftmargin\labelsep
2886 \@openbib@code
2887 \usecounter{enumiv}%
2888 \let\p@enumiv\@empty
2889 \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2890 %<kiyou> \small
2891 \sloppy
2892 \clubpenalty4000
2893 \@clubpenalty\clubpenalty
2894 \widowpenalty4000%
2895 \sfcode`.\@m}
2896 {\def\@noitemerr
2897 {\@latex@warning{Empty `thebibliography' environment}}}%
2898 \endlist}

```

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

```

2899 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

```

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

```

2900 \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。

```

2901 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

`\cite` 文献の番号を出力する部分は ltbibl.dtx で定義されていますが、コンマとカッコを和文 `\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必要に応じて生かしてください。かつこの前後に入るグルーを `\jsInhibitGlue` で取っていますので、オリジナル同様、Knuth-`\cite{knu}` のように半角空白で囲んでください。

```

2902 % \def\@citex[#1]#2{\leavevmode
2903 % \let\@citea\@empty
2904 % \@cite{\@for\@citeb:=#2\do
2905 % {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2906 % \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2907 % \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2908 % \ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2909 % \G@refundefinedtrue
2910 % \latex@warning
2911 % {Citation `'\@citeb' on page \thepage \space undefined}}%
2912 % {\@cite@ofmt{\csname b@\@citeb\endcsname}}{\#1}}
2913 % \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。

```
2914 % \DeclareRobustCommand\cite{\unskip
2915 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2916 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempwa
2917 %   , \jsInhibitGlue\ #2\fi}) }}$}
```

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)。

```
2918 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2919   \if@twocolumn
2920     \onecolumn\@restonecolfalse
2921   \else
2922     \clearpage\@restonecoltrue
2923   \fi
2924   \columnseprule.4pt \columnsep 2\jsZw
2925   \ifx\multicols\@undefined
2926 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2927 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2928 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2929 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2930   \else
2931     \ifdim\textwidth<\fullwidth
2932       \setlength{\evensidemargin}{\oddsidemargin}
2933       \setlength{\textwidth}{\fullwidth}
2934       \setlength{\linewidth}{\fullwidth}
2935 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
2936 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2937 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2938 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2939   \else
2940 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}%
2941 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2942 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2943 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2944   \fi
2945   \fi
2946 %<book|report>      \@mkboth{\indexname}{}%
2947 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2948   \plainifnotempty % \thispagestyle{plain}
2949   \parindent\z@
2950   \parskip\z@ \@plus .3\p@?\relax
2951   \let\item\@idxitem
2952   \raggedright
2953   \footnotesize\narrowbaselines
2954 }{
```



```

2955 \ifx\multicols\@undefined
2956 \if@restonecol\onecolumn\fi
2957 \else
2958 \end{multicols}
2959 \fi
2960 \clearpage
2961 }

```

`\@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 2962 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2963 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2964 \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。

```

2965 \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。

```

2966 \newcommand\seename{\if@english see\else →\fi}
2967 \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。p_{La}T_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。

```

2968 \ifx\pltx@foot@penalty\@undefined
2969 \let\footnotes@ve=\footnote
2970 \def\footnote{\jsInhibitGlue\footnotes@ve}
2971 \let\footnotemarks@ve=\footnotemark
2972 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2973 \fi

```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

[2016-07-11] コミュニティ版 p_AT_EX の変更に従いました (Thanks: 角藤さん)。p_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。

p_TE_X 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
2974 \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。

```
2975 % \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。

```
2976 \renewcommand{\footnoterule}{%
2977   \kern-2.6\p@? \kern-.4\p@
2978   \hrule width .4\columnwidth
2979   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2980 %<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_AT_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_AT_EX のバグ修正に従いました。

[2016-11-29] 古い p_AT_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。

```
2981 \long\def\@footnotetext{%
2982   \insert\footins\bgroup
2983     \normalfont\footnotesize
2984     \interlinepenalty\interfootnotelinepenalty
2985     \splittopskip\footnotesep
2986     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2987     \hsize\columnwidth \@parboxrestore
2988     \protected@edef\@currentlabel{%
2989       \csname p@footnote\endcsname\@thefnmark
```

```

2990 }%
2991 \color@begingroup
2992 \makefnmark{%
2993 \rule{z@footnotesep}{\ignorespaces}%
2994 \futurelet\jsc@next\jsc@fo@t}
2995 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@f@t
2996 \else \let\jsc@next\jsc@f@t\fi \jsc@next}
2997 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
2998 \def\jsc@f@t#1{#1\jsc@@foot}
2999 \def\jsc@@foot{\finalstrut\strutbox\color@endgroup\egroup
3000 \ifx\pltx@foot@penalty\undefined\else
3001 \ifhmode\null\fi
3002 \ifnum\pltx@foot@penalty=z@\else
3003 \penalty\pltx@foot@penalty
3004 \pltx@foot@penaltyz@
3005 \fi
3006 \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。

```

3007 \newcommand\makefnmark[1]{%
3008 \advance\leftskip 3\jsZw
3009 \parindent 1\jsZw
3010 \noindent
3011 \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。

```

3012 % \def\xfootnotenext[#1]{%
3013 % \begingroup
3014 % \ifnum#1>z@
3015 % \csname c@\mpfn\endcsname #1\relax
3016 % \unrestored@protected@xdef\thefnmark{\thempfn}%
3017 % \else
3018 % \unrestored@protected@xdef\thefnmark{}%
3019 % \fi
3020 % \endgroup
3021 % \@footnotetext}

```

ここまでのコードは JS クラスを踏襲する。

11 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

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

```
3022 \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。

```
3023 \def\everyparhook{\jsInhibitGlueAtParTop}
3024 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
3025 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3026 \fi
```

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

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

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

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

```
3027 \ifx j\jsEngine
3028 \def\@inhibitglue{%
3029   \futurelet\@let@token\@@inhibitglue}
3030 \begingroup
3031 \let\GDEF=\gdef
3032 \let\CATCODE=\catcode
3033 \let\ENDGROUP=\endgroup
3034 \CATCODE`k=12
3035 \CATCODE`a=12
3036 \CATCODE`n=12
3037 \CATCODE`j=12
3038 \CATCODE`i=12
3039 \CATCODE`c=12
3040 \CATCODE`h=12
3041 \CATCODE`r=12
3042 \CATCODE`t=12
```

```

3043 \CATCODE`e=12
3044 \GDEF\KANJI@CHARACTER{kanji character }
3045 \ENDGROUP
3046 \def\@@inhibitglue{%
3047   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
3048 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3049   \def\jsc@ig@temp{#1}%
3050   \ifx\jsc@ig@temp\@empty
3051     \ifnum\the\inhibitxspcode`#2=2\relax
3052       \inhibitglue
3053     \fi
3054   \fi}
3055 \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。

```

3056 \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。

```

3057 \def\@doendpe{%
3058   \@endpetrue
3059   \def\par{%
3060     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3061   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3062 \def\@setminipage{%
3063   \@minipagetrue
3064   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3065 }

```

`\item` 命令の直後です。

```

3066 \def\@item[#1]{%
3067   \if@noparitem
3068     \@donoparitem
3069   \else
3070     \if@inlabel
3071       \indent \par
3072     \fi
3073     \ifhmode
3074       \unskip\unskip \par
3075     \fi
3076     \if@newlist

```

```

3077     \if@nobreak
3078         \@nbitem
3079     \else
3080         \addpenalty\@beginparpenalty
3081         \addvspace\@topsep
3082         \addvspace{-\parskip}%
3083     \fi
3084 \else
3085     \addpenalty\@itempenalty
3086     \addvspace\itemsep
3087 \fi
3088 \global\@inlabeltrue
3089 \fi
3090 \everypar{%
3091     \@minipagefalse
3092     \global\@newlistfalse
3093     \if@inlabel
3094         \global\@inlabelfalse
3095         {\setbox\z@\lastbox
3096         \ifvoid\z@
3097             \kern-\itemindent
3098         \fi}%
3099     \box\@labels
3100     \penalty\z@
3101 \fi
3102 \if@nobreak
3103     \@nobreakfalse
3104     \clubpenalty \@M
3105 \else
3106     \clubpenalty \@clubpenalty
3107     \everypar{\everyparhook}%
3108 \fi
3109 \everyparhook}%
3110 \if@noitemarg
3111     \@noitemargfalse
3112     \if@nmbrlist
3113         \refstepcounter\@listctr
3114     \fi
3115 \fi
3116 \sbox\@tempboxa{\makelabel{#1}}%
3117 \global\setbox\@labels\hbox{%
3118     \unhbox\@labels
3119     \hskip \itemindent
3120     \hskip -\labelwidth
3121     \hskip -\labelsep
3122     \ifdim \wd\@tempboxa >\labelwidth
3123         \box\@tempboxa
3124     \else
3125         \hbox to\labelwidth {\unhbox\@tempboxa}%

```

```

3126 \fi
3127 \hskip \labelsep}%
3128 \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。

```

3129 \def\@afterheading{%
3130 \@nobreakefalse
3131 \everypar{%
3132 \if@nobreake
3133 \@nobreakefalse
3134 \clubpenalty \@M
3135 \if@afterindent \else
3136 {\setbox\z@\lastbox}%
3137 \fi
3138 \else
3139 \clubpenalty \@clubpenalty
3140 \everypar{\everyparhook}%
3141 \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

```

3142 \fi

```

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

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

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

```

3143 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3144 \AtEndOfPackage{%
3145 \def\@gnewline #1{%
3146 \ifvmode
3147 \@nolnerr
3148 \else
3149 \unskip \reserved@a {\reserved@f#1}\nobreake \hfil \break \null
3150 \jsInhibitGlue \ignorespaces
3151 \fi}
3152 }
3153 \fi

```

12 いろいろなロゴ

L^AT_EX 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。

```
3154 \if@jslogo
3155   \IfFileExists{jslogo.sty}{%
3156     \RequirePackage{jslogo}%
3157   }{%
3158     \ClassWarningNoLine{bxjs@clsname}
3159       {The package 'jslogo' is not installed.\MessageBreak
3160        It is included in the recent release of\MessageBreak
3161        the 'jsclasses' bundle}
3162   }
3163 \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` 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 `ltxclass.dtx` 2004/01/28 v1.1g で修正されているのでコメントアウトしました。

```
3164 %\let\ltx@ifnextchar\@ifnextchar
3165 %\def\ProvidesFile#1{%
3166 %   \begingroup
3167 %     \catcode`\ 10 %
3168 %     \ifnum \endlinechar<256 %
3169 %       \ifnum \endlinechar>\m@ne
3170 %         \catcode\endlinechar 10 %
3171 %       \fi
3172 %     \fi
3173 %     \@makeother\/%
```



```

3174 % \makeother\&%
3175 % \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 3176 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3177 \newcommand{\postpartname}{\if@english\else 部\fi}
3178 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3179 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3180 \newcommand{\presectionname}{}% 第
\postsectionname 3181 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3182 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3183 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3184 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname

\bibname 3185 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3186 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3187 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 3188 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3189 %<jspf>\newcommand{\figurename}{Fig.~}
3190 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3191 %<jspf>\newcommand{\tablename}{Table~}

\appendixname

\abstractname 3192 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3193 \newcommand{\appendixname}{\if@english \else 付録\fi}
3194 %<book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

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

※ 2.0 版より、完全に `bxwareki` に任せる。

```

3195 \@onlypreamble\bxjs@decl@Seireki@cmds

```

```

3196 \@tempswafalse
3197 \if p\jsEngine \@tempswatruetrue \fi
3198 \if n\jsEngine \@tempswatruetrue \fi
3199 \bxjs@cond\if@tempswa\fi{%
3200 % 欧文 8bitTeX の場合
3201 \newif\ifjsSeireki \jsSeirekitruetrue
3202 \def\bxjs@decl@Seireki@cmds{%
3203   \def\西暦{\jsSeirekitruetrue}%
3204   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3205 \def\Seireki{\jsSeirekitruetrue}
3206 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3207 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3208 \def\bxjs@iai{\noexpand~}
3209 }{%
3210 \newif\if 西暦 \西暦 true
3211 \def\bxjs@decl@Seireki@cmds{%
3212   \def\西暦{\西暦 true}%
3213   \def\和暦{\西暦 false\bxjs@wareki@used}}
3214 \def\Seireki{\西暦 true}
3215 \def\Wareki{\西暦 false\bxjs@wareki@used}
3216 \def\bxjs@if@use@seireki{\bxjs@cond\if 西暦\fi}
3217 \let\bxjs@iai@empty
3218 }
3219 \bxjs@decl@Seireki@cmds
3220 \let\bxjs@unxp@firstofone \let\bxjs@onxp@firstofone
3221 \bxjs@test@engine\unexpanded{%
3222   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}

```

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

```

3223 \newif\ifbxjs@bxwareki@avail
3224 \IfFileExists{bxwareki.sty}{%
3225   \RequirePackage{bxwareki}[]%
3226   \bxjs@bxwareki@availtrue}{%

```

\bxjs@wareki@used 和暦が非対応の場合に警告を出す。

```

3227 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\empty
3228 \else
3229   \bxjs@robust@def\bxjs@wareki@used{%
3230     \global\let\bxjs@wareki@used\empty
3231     \ClassWarning\bxjs@clsname
3232       {Wareki mode is not supported, since\MessageBreak
3233         'bxwareki' is unavailable, found}}
3234   \g@addto@macro\bxjs@begin@document@hook{%
3235     \let\bxjs@wareki@used\empty}
3236 \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。

```

3237 \ifbxjs@bxwareki@avail
3238   \let\jyear\warekiyear
3239   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3240     \newcount\heisei \heisei=\value{warekiyear}
3241   \fi

   bxwareki が使えない場合は西暦表示にする。

3242 \else
3243   \edef\jyear{\the\year \bxjs@iai}
3244 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3245 \let\bxjs@next\relax
3246 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3247   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3248 \fi\fi
3249 \edef\bxjs@today{%
3250   \if@english
3251     \ifcase\month\or
3252       January\or February\or March\or April\or May\or June\or
3253       July\or August\or September\or October\or November\or December\fi
3254     \space\number\day, \number\year
3255   \else
3256     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3257     \else \noexpand\bxjs@if@use@seireki
3258     \fi {%
3259       \number\year\bxjs@iai\bxjs@unxp{年}%
3260       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3261       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3262     }\bxjs@next}%
3263 \fi}
3264 \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。

```

3265 \g@addto@macro\bxjs@begin@document@hook{%
3266   \ifx\bbl@jpn@maybekansuji\@undefined\else
3267     \bxjs@decl@Seireki@cmds
3268     \g@addto@macro\datejapanese{%
3269       \let\today\bxjs@today}%
3270   \fi}

```

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


```

3271 \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。

```
3272 %<slide>\pagestyle{empty}%
3273 %<article|report>\pagestyle{plain}%
3274 %<book>\pagestyle{headings}%
3275 \pagenumbering{arabic}
3276 \if@twocolumn
3277   \twocolumn
3278   \sloppy
3279   \flushbottom
3280 \else
3281   \onecolumn
3282   \raggedbottom
3283 \fi
3284 %<*slide>
3285 \renewcommand\familydefault{\sfdefault}
3286 \raggedright
3287 %</slide>
```

■ BXJS 独自の追加処理 

フックを実行する。

```
3288 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3289 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

```
3290 %</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。

```
3291 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

ただし、(u)p_T_EX 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

```
3292 %<*minimal>
```

```
3293 %% 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.
```

```
3294 \def\DeclareJaTextFontCommand#1#2{%
```

```
3295   \DeclareRobustCommand#1[1]{%
```

```
3296     \relax
```

```
3297     \ifmmode \expandafter\nfss@text \fi
```

```
3298     {#2##1}}%
```

```
3299 }
```

```
\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。
```

```
3300 \def\DeclareJaMathFontCommand#1#2{%
```

```
3301   \DeclareRobustCommand#1[1]{%
```

```
3302     \relax
```

```
3303     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3304     \nfss@text{\fontfamily\familydefault
```

```
3305       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3306       #2##1}}%
```

```

3307 }%
3308 }

```

`\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。

```

3309 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3310 \@onlypreamble\bxjs@if@sf@default
3311 \def\bxjs@if@sf@default#1{%
3312   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3313   \g@addto@macro\bxjs@begin@document@hook{%
3314     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3315 }

```

`\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。

```

3316 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3317 \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。

```

3318 \def\jsLetHeadChar#1#2{%
3319   \begingroup
3320   \escapechar=`\ %
3321   \let\bxjs@tmpa={% brace-match-hack
3322     \bxjs@let@hchar@exp#2}%
3323   \endgroup
3324   \let#1\bxjs@g@tmpa}
3325 \def\bxjs@let@hchar@exp{%
3326   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3327 \def\bxjs@let@hchar@exp@a{%
3328   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3329   \bxjs@let@hchar@out\let\relax
3330   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3331   \bxjs@let@hchar@out\let\space%
3332   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3333   \bxjs@let@hchar@out\let\@backslashchar
3334   }{\bxjs@let@hchar@exp@b}}}%
3335 \def\bxjs@let@hchar@exp@b#1{%
3336   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3337 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3338   %\message{<#1#2>}%
3339   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3340 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3341 \bxjs@let@hchar@out\let\relax
3342 }{%else
3343 \expandafter\bxjs@let@hchar@exp
3344 }%
3345 }{%else
3346 \bxjs@let@hchar@chr#1%
3347 }}
3348 \def\bxjs@let@hchar@chr#1{%
3349 \bxjs@let@hchar@out\def{{#1}}}}
3350 \def\bxjs@let@hchar@out#1#2{%
3351 \global#1\bxjs@g@tmpa#2\relax
3352 \toks@{bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3353 \chardef\bxjs@let@hchar@csta=128
3354 \chardef\bxjs@let@hchar@cstb=192
3355 \chardef\bxjs@let@hchar@cstc=224
3356 \chardef\bxjs@let@hchar@cstd=240
3357 \chardef\bxjs@let@hchar@cste=248
3358 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3359 \def\bxjs@let@hchar@chr@ue#1{%
3360 \@tempcnta=#1\relax
3361 %\message{\the\@tempcnta}%
3362 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3363 \bxjs@let@hchar@chr@ue@a#1%
3364 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3365 \bxjs@let@hchar@out\let\relax
3366 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3367 \bxjs@let@hchar@chr@ue@b
3368 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3369 \bxjs@let@hchar@chr@ue@c
3370 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3371 \bxjs@let@hchar@chr@ue@d
3372 }{%else
3373 \bxjs@let@hchar@out\let\relax
3374 }}}}
3375 \def\bxjs@let@hchar@chr@ue@a#1{%
3376 \bxjs@let@hchar@out\def{{#1}}}}
3377 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3378 \bxjs@let@hchar@out\def{{#1#2}}}}
3379 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3380 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3381 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3382 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定


```
3383 \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。

```
3384 \def\bxjs@let@hchar@chr@pp#1#2{%
3385   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3386 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3387   \%message{(\meaning#3:\meaning#4)}%
3388   \bxjs@cond\if#1k\fi{%
3389     \bxjs@let@hchar@out\def{#4}}%
3390   }{%else
3391     \bxjs@let@hchar@chr@ue#3#4%
3392   }}
3393 \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。

```
3394 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3395 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3396 \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。

```
3397 \@onlypreamble\bxjs@declarefontshape
3398 \ifjsWithupTeX
3399 \def\bxjs@declarefontshape{%
3400   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3401   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3402   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3403   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3404 }
3405 \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。

```
3406 \else
3407 \def\bxjs@declarefontshape{%
3408   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3409   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3410   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3411 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3412 }
3413 \def\bxjs@sizereference{jis}
3414 \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。

```

3415 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3416   \def\bxjs@tmpb{#5}}
3417 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3418 \else
3419 \expandafter\expandafter\expandafter\bxjs@next
3420 \expandafter\string\the\jfont\relax
3421 \fi
3422 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3423                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3424   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3425    \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。

```

3426 \begingroup
3427 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3428 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3429 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3430 % 幅が丁度 10pt なら補正は不要
3431 \ifdim\wd\z@=10pt
3432   \global\let\bxjs@scale\jsScale
3433 \else
3434 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3435   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3436   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3437   \bxjs@invscale\@tempdima\bxjs@tmpa
3438   \xdef\bxjs@scale{\strip@pt\@tempdima}
3439 \fi
3440 \endgroup
3441 %\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。

```

3442 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3443 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3444 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3445 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3446 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3447 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3448 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3449 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3450 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{f}
3451 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{f}
3452 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{f}
3453 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{f}
3454 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{f}
3455 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{f}
3456 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{f}
3457 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{f}
3458 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3459 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3460 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3461 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3462 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3463 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3464 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3465 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{f}
3466 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{f}
3467 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{f}
3468 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{f}
3469 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{f}
3470 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{f}
3471 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{f}
3472 \DeclareFontShape{\jsc@JTn}{gt}{b}{sl}{<->ssub*gt/bx/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。

※ 2020-02-02 の NFSS の改修に対する jsclasses の対策を取り入れた。

```

3473 \ifx\@rmfamilyhook\@undefined % old
3474 \DeclareRobustCommand\rmfamily
3475   {\not@math@alphabet\rmfamily\mathrm
3476     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3477 \DeclareRobustCommand\sffamily
3478   {\not@math@alphabet\sffamily\mathsf
3479     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3480 \DeclareRobustCommand\ttfamily
3481   {\not@math@alphabet\ttfamily\mathtt
3482     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3483 \g@addto@macro\bxjs@begin@document@hook{%
3484   \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3485     % my definitions above should have been overwritten, recover it!
3486     % \selectfont is executed twice but I don't care about speed...
3487     \expandafter\g@addto@macro\csname rmfamily \endcsname
3488       {\kanjifamily\mcdefault\selectfont}%
3489     \expandafter\g@addto@macro\csname sffamily \endcsname
3490       {\kanjifamily\gtdefault\selectfont}%

```

```

3491 \expandafter\g@addto@macro\csname ttfamily \endcsname
3492 {\kanjifamily\gtdefault\selectfont}%
3493 \fi}
3494 \else % 2020-02-02
3495 \g@addto@macro\@rmfamilyhook
3496 {\prepare@family@series@update@kanji{mc}\mcdefault}
3497 \g@addto@macro\@sffamilyhook
3498 {\prepare@family@series@update@kanji{gt}\gtdefault}
3499 \g@addto@macro\@ttfamilyhook
3500 {\prepare@family@series@update@kanji{gt}\gtdefault}
3501 \fi
3502 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3503 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3504 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3505 \fi
3506 \bxjs@if@sf@default{%
3507 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3508 \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）。

```

3509 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3510 \let\bxjs@parse@qh@a\@undefined
3511 \let\bxjs@parse@qh@b\@undefined

```

■パラメタの設定

```

3512 \prebreakpenalty\jis"2147=10000
3513 \postbreakpenalty\jis"2148=10000
3514 \prebreakpenalty\jis"2149=10000
3515 \inhibitxspcode`!=1
3516 \inhibitxspcode`〒=2
3517 \xspcode`+=3
3518 \xspcode`%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3519 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3520 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3521 \let\jsInhibitGlueAtParTop\@inhibitglue

```

`\jsResetDimen` は空のままでよい。

■組方向依存の処理 組方向判定の `if-トークン` (`\if?dir`) は p_TE_X 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。

```

3522 \begingroup
3523 \catcode`\!=0

```

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

```
3524 \gdef\bxjs@ptex@dir{%
3525   !iftdir t%
3526   !else!ifydir y%
3527   !else ?%
3528   !fi!fi}
```

新版の p_TE_X で脚注番号の周囲の空きが過大になる現象への対処。

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

```
3529 % 古い \@makefnmark の定義
3530 \long\def\bxjs@tmpa{\hbox{%
3531   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3532   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3533 \ifx\@makefnmark\bxjs@tmpa
3534 \long\gdef\@makefnmark{%
3535   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3536   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3537 \fi
3538 \endgroup
```

B.3 pdf_TE_X 用の処理

```
3539 \else\ifx p\jsEngine
3540 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3541 \@onlypreamble\bxjs@cjk@loaded
3542 \def\bxjs@cjk@loaded{%
3543   \def\@footnotemark{%
3544     \leavevmode
3545     \ifhmode
3546       \edef\x@sf{\the\spacefactor}%
3547       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3548         \unkern\unkern
3549         \ifdim\lastskip>\z@ \unskip \fi
3550       \fi\fi
3551       \nobreak
3552     \fi
3553     \@makefnmark
3554     \ifhmode \spacefactor\x@sf \fi
3555     \relax}%
3556 \let\bxjs@cjk@loaded\relax
3557 }
3558 \g@addto@macro\bxjs@begin@document@hook{%
3559   \@ifpackageloaded{CJK}{%
3560     \bxjs@cjk@loaded
3561   }{}%
3562 }
```

B.4 X_YTeX 用の処理

3563 \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。

```
3564 \def\bxjs@let@hchar@chr#1{%
3565   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3566   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3567     \bxjs@let@hchar@chr@xe
3568   }\bxjs@let@hchar@out\def{{#1}}}}
3569 \def\bxjs@let@hchar@chr@xe#1{%
3570   \lccode`0=`#1\relax
3571   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3572 \@onlypreamble\bxjs@do@precisetext
3573 \ifx\XeTeXgenerateactualtext\@undefined\else
3574   \def\bxjs@do@precisetext{%
3575     \XeTeXgenerateactualtext=\@ne}
3576 \fi
```

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

```
3577 \@onlypreamble\bxjs@do@simplejasetup
3578 \def\bxjs@do@simplejasetup{%
3579   \ifnum\XeTeXinterchartokenstate>\z@
3580   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3581     \jsSimpleJaSetup
3582     \ClassInfo\bxjs@clsname
3583     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3584   \fi\fi}
```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```
3585 \newcommand*{\jsSimpleJaSetup}{%
3586   \XeTeXlinebreaklocale "ja"\relax
3587   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3588   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

3589 \fi\fi\fi

simplejasetup オプションの処理。

```
3590 \ifx\bxjs@do@simplejasetup\@undefined\else
3591   \g@addto@macro\bxjs@begin@document@hook{%
3592     \ifbxjs@simplejasetup
3593       \bxjs@do@simplejasetup
3594     \fi}
3595 \fi
```

precisetext オプションの処理。

```

3596 \ifbxjs@precisetext
3597   \ifx\bxjs@do@precisetext\@undefined
3598     \ClassWarning\bxjs@clsname
3599       {The current engine does not support the\MessageBreak
3600         'precise-text' option\@gobble}
3601   \else
3602     \bxjs@do@precisetext
3603   \fi
3604 \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。

```

3605 \g@addto@macro\bxjs@begin@document@hook{%
3606   \ifx\jsInhibitGlueAtParTop\@empty
3607     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3608     \ifx\everyparhook\bxjs@tmpa
3609       \let\everyparhook\@empty
3610     \fi
3611   \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。

```

3612 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3613   \let\bxjs@everypar\everypar
3614   \newtoks\everypar
3615   \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。

```

3616   \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3617 \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。

```

3618 \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。

```

3619 \@onlypreamble\bxjs@adjust@fancyhdr
3620 \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。

```

3621 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3622 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3623 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3624 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3625 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3626 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3627 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3628 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3629 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3630 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3631 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3632 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3633 \def\bxjs@tmpa{\rm\thepage\strut}%
3634 \def\bxjs@tmpb{\thepage\strut}%
3635 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3636 \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。

```

3637 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3638   \setlength{\@tempdima}{\fullwidth-\textwidth}%
3639   \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3640   }\bxjs@tmpa
3641 \fi\fi
3642 \PackageInfo\bxjs@clsname
3643 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3644 \def\bxjs@pagestyle@hook{%
3645   \@ifpackageloaded{fancyhdr}{%
3646     \bxjs@adjust@fancyhdr
3647     \global\let\bxjs@adjust@fancyhdr\relax
3648   }{}}

```

\pagestyle にフックを入れ込む。

```

3649 \let\bxjs@org@pagestyle\pagestyle
3650 \def\pagestyle{%
3651   \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。

```

3652 \g@addto@macro\bxjs@begin@document@hook{%
3653   \bxjs@pagestyle@hook
3654   \global\let\bxjs@pagestyle@hook\relax}

```

■和文空白命令

```

3655 \ifbxjs@jaspace@cmd

```


`\jaenspace` 半角幅の水平空き。

```
3656 \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。

```
3657 \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。

```
3658 \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3659 \DeclareRobustCommand*{\jaspace}[1]{%
3660   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3661     \ClassError\bxjs@clsname
3662     {Unknown jaspaces: #1}{\@eha}%
3663   \else
3664     \csname bxjs@jaspace@@#1\endcsname
3665   \fi}
3666 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3667 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3668 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3669 \fi
```

終わり。

```
3670 \fi
```

以上で終わり。

```
3671 %</minimal>
```

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

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

C.1 共通処理 (1)

まず minimal ドライバを読み込む。

```
3672 %<*standard>
3673 %% このファイルは日本語文字を含みます
3674 \input{bxjsja-minimal.def}
```

simplejasetup は standard では無効になる。

```
3675 \bxjs@simplejasetupfalse
```

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

\ifbxjs@jp@jismmiv 2004JIS 字形を優先させるか。

```
3676 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。

3677 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3678 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3679 \define@key{bxjsStd}{jis2004}[true]{%
3680 \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。

```
3681 \newif\ifbxjs@jp@units

units オプションの処理。

3682 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3683 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3684 \define@key{bxjsStd}{units}[true]{%
3685 \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。

```
3686 \let\bxjs@jp@font\@empty

font オプションの処理。

3687 \define@key{bxjsStd}{font}{%
3688 \edef\bxjs@jp@font{#1}}

実際の japaram の値を適用する。

3689 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3690 \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。

```

3691 \ifbxjs@jp@jismmiv
3692   \g@addto@macro\@classoptionslist{,jis2004}
3693 % \@ifpackagewith 判定への対策
3694   \PassOptionsToPackage{jis2004}{otf}
3695 \fi

```

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

```

3696 \if j\jsEngine
3697   \bxjs@jp@unitsfalse
3698 \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。

```

3699 \ifbxjs@jp@units
3700   \IfFileExists{bxcalc.sty}{%
3701     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3702     \ifx\usepTeXunits\undefined
3703       \PackageWarningNoLine\bxjs@clsname
3704         {Cannot support pTeX units (zw etc.), since\MessageBreak
3705           the package 'bxcalc' is too old}%
3706       \bxjs@jp@unitsfalse
3707     \else \usepTeXunits
3708     \fi
3709   }{%else
3710     \PackageWarningNoLine\bxjs@clsname
3711       {Cannot support pTeX units (zw etc.), since\MessageBreak
3712         the package 'bxcalc' is unavailable}%
3713     \bxjs@jp@unitsfalse
3714   }
3715 \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。

```

3716 \ifbxjs@jp@units
3717 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3718 \let\bxjs@parse@qh@a\undefined
3719 \let\bxjs@parse@qh@b\undefined
3720 \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。

```

3721 \ifbxjs@jp@units
3722   \def\bxjs@let@lenexpr#1#2{%
3723     \edef#1{#2}%
3724     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3725 \else
3726   \def\bxjs@let@lenexpr{\edef}
3727 \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。

```
3728 \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 3729 \@tempdima=0.25mm
      3730 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
      3731 \let\jH\jQ
```

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

```
\trueH 3732 \ifjsc@mag
      3733 \@tempdimb=\jsBaseFontSize\relax
      3734 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
      3735 \@tempdima=2.5mm
      3736 \bxjs@invscale\@tempdima\bxjs@tmpa
      3737 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
      3738 \@tempdima=10pt
      3739 \bxjs@invscale\@tempdima\bxjs@tmpa
      3740 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
      3741 \else \let\trueQ\jQ \let\bxjs@truept\p@
      3742 \fi
      3743 \let\trueH\trueQ
```

`\ascQ` `\ascQ` は `\trueQ` を和文スケール値で割った値。例えば、`\fontsize{12\ascQ}{16\trueH}`
`\ascpt` とすると、和文が 12Q になる。

同様に、`\ascpt` は `truept` を和文スケールで割った値。

```
3744 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3745 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3746 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3747 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3748 \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。

```
3749 \def\jafontsize#1#2{%
3750 \begingroup
3751 \bxjs@jafontsize@a{#1}%
3752 \@tempdimb\jsInverseScale\@tempdima
3753 \bxjs@jafontsize@a{#2}%
3754 \xdef\bxjs@g@tmpa{%
3755 \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3756 \endgroup\bxjs@g@tmpa}
3757 \def\bxjs@jafontsize@a#1{%
```

```

3758 \bxjs@parse@qh{#1}%
3759 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3760 \@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。

```
3761 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```

3762 \newcommand*\setkanjiskip[1]{%
3763 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3764 \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3765 \newcommand*\getkanjiskip{%
3766 \bxjs@kanjiskip}

```

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

```
3767 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

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

```

\bxjs@disable@kanjiskip 3768 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3769 \bxjs@kanjiskip@enabledtrue
3770 \bxjs@reset@kanjiskip}
3771 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3772 \bxjs@kanjiskip@enabledfalse
3773 \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。

```

3774 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3775 \ifbxjs@kanjiskip@enabled
3776 \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3777 \else \@tempskipa\z@
3778 \fi
3779 \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 3780 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3781 \newcommand*\setxkanjiskip[1]{%
```

```
3782 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3783 \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3784 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3785 \bxjs@xkanjiskip}
```

```
3786 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3787 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3788 \bxjs@xkanjiskip@enabledtrue
```

```
3789 \bxjs@reset@xkanjiskip}
```

```

3790 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3791   \bxjs@xkanjiskip@enabledfalse
3792   \bxjs@reset@xkanjiskip}
3793 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3794   \ifbxjs@xkanjiskip@enabled
3795     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3796   \else \@tempskipa\z@
3797   \fi
3798   \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。

```

3799 \g@addto@macro\jsResetDimen{%
3800   \bxjs@reset@kanjiskip
3801   \bxjs@reset@xkanjiskip}
3802 \let\bxjs@apply@kanjiskip\relax
3803 \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。

```

3804 \@onlypreamble\bxjs@adjust@jafont
3805 \def\bxjs@adjust@jafont#1{%
3806   \ifx\jsJaFont\bxjs@@auto
3807     \bxjs@get@kanjiEmbed
3808     \ifx\bxjs@kanjiEmbed\relax
3809       \let\bxjs@tmpa\@empty
3810     \else
3811       \let\bxjs@tmpa\bxjs@kanjiEmbed
3812     \fi
3813   \else
3814     \let\bxjs@tmpa\jsJaFont
3815   \fi
3816   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3817     \ClassWarningNoLine\bxjs@clsname
3818     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3819     not available on the current situation}%
3820     \let\bxjs@tmpa\@empty
3821   \fi\fi
3822 }
3823 \def\bxjs@@auto{auto}
3824 \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。

```

3825 \let\bxjs@kanjiEmbed\relax

```

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

```
3826 \@onlypreamble\bxjs@get@kanjiEmbed
3827 \def\bxjs@get@kanjiEmbed{%
3828   \begingroup\setbox\z@=\hbox{%
3829     \global\let\bxjs@g@tmpa\relax
3830     \endlinechar\m@ne
3831     \let\do\@makeother\dospecials
3832     \catcode32=10 \catcode12=10 %form-feed
3833     \let\bxjs@tmpa\@empty
3834     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3835     \ifeof\@inputcheck\else
3836       \read\@inputcheck to\bxjs@tmpa
3837       \closein\@inputcheck
3838     \fi
3839     \ifx\bxjs@tmpa\@empty\else
3840       \openin\@inputcheck="\bxjs@tmpa"\relax
3841       \@tempswatrue
3842       \loop\if@tempswa
3843         \read\@inputcheck to\bxjs@tmpa
3844         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3845         \ifx\bxjs@tmpb\relax\else
3846           \global\let\bxjs@g@tmpa\bxjs@tmpb
3847           \@tempswafalse
3848         \fi
3849         \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3850         \ifx\bxjs@tmpb\relax\else
3851           \global\let\bxjs@g@tmpa\bxjs@tmpb
3852           \@tempswafalse
3853         \fi
3854         \ifeof\@inputcheck \@tempswafalse \fi
3855       \repeat
3856     \fi
3857   }\endgroup
3858   \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3859 }
3860 \@onlypreamble\bxjs@get@ke@a
3861 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3862   \ifx$#1$\def\bxjs@tmpb{#2}%
3863   \else \let\bxjs@tmpb\relax
3864   \fi}
3865 \@onlypreamble\bxjs@get@ke@b
3866 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3867   \ifx$#1$\def\bxjs@tmpb{#2}%
3868   \else \let\bxjs@tmpb\relax
3869   \fi}
```

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

```
3870 \newcommand*\jachar[1]{%
3871   \begingroup
```

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

```
3872 \jsLetHeadChar\bxjs@tmpa{#1}%
3873 \ifx\bxjs@tmpa\relax
3874 \ClassWarningNoLine\bxjs@clsname
3875 {Illegal argument given to \string\jachar}%
3876 \else
3877 \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3878 \fi
3879 \endgroup}
```

\jsJaChar を \jachar と等価にする。

```
3880 \let\jsJaChar\jachar
```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```
3881 \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。

```
3882 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3883 \@onlypreamble\bxjs@fix@hyperref@unicode
3884 \def\bxjs@fix@hyperref@unicode#1{%
3885 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3886 \@namedef{KV@Hyp@bxjs/hook}##1{%
3887 \KV@Hyp@unicode{##1}%
3888 \def\KV@Hyp@unicode####1{%
3889 \expandafter\ifx\csname if##1\expandafter\endcsname
3890 \csname if####1\endcsname\else
3891 \ClassWarningNoLine\bxjs@clsname
3892 {Blcoked hyperref option 'unicode=####1'}%
3893 \fi
3894 }%
3895 }%
3896 }
```

\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。

```
3897 \@onlypreamble\jsCheckHyperrefUnicode
3898 \let\jsCheckHyperrefUnicode\@empty
3899 \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。

```
3900 \@onlypreamble\bxjs@check@hyperref@unicode
3901 \def\bxjs@check@hyperref@unicode#1{%
3902 \g@addto@macro\jsCheckHyperrefUnicode{%
3903 \@tempwafalse
3904 \begingroup
3905 \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3906 \aftergroup\@tempwatruue \fi
```



```

3907 \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3908 \csname if#1\endcsname
3909 \aftergroup\@tempwattrue \fi
3910 \endgroup
3911 \if@tempswa\else
3912 \ClassError\bxjs@clsname
3913 {The value of hyperref 'unicode' key is not suitable\MessageBreak
3914 for the present engine (must be #1)}%
3915 {\@ehc}%
3916 \fi}}

```

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

```

3917 \@onlypreamble\bxjs@urgent@special
3918 \def\bxjs@urgent@special#1{%
3919 \AtBeginDvi{\special{#1}}%
3920 \g@addto@macro\bxjs@begin@document@hook{%
3921 \@ifpackageloaded{atbegshi}{%
3922 \begingroup
3923 \toks\z@{\special{#1}}%
3924 \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3925 \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3926 \endgroup
3927 }{}%
3928 }%
3929 }

```

`\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。

```

3930 \@onlypreamble\bxjs@resolve@jafont@paren
3931 \def\bxjs@resolve@jafont@paren#1{%
3932 \def\bxjs@tmpb{\let#1}%
3933 \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3934 \@onlypreamble\bxjs@resolve@jafont@paren@a
3935 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3936 \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3937 \else
3938 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3939 \bxjs@tmpb\bxjs@tmpa
3940 \fi}

```

C.2 pTeX 用設定

```

3941 \if j\jsEngine

```

■ 共通命令の実装

```

3942 \def\bxjs@apply@kanjiskip{%
3943 \kanjiskip\@tempskipa}
3944 \def\bxjs@apply@xkanjiskip{%
3945 \xkanjiskip\@tempskipa}

```

`\jaJaChar` のサブマクロ。

```
3946 \def\bxjs@jachar#1{%
3947   \bxjs@jachar@a#1...\@nil}
3948 \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。

```
3949   \ifx.#2#1%
```

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

```
3950   \else\ifx.#3%
3951     \@tempcnta`#1 \multiply\@tempcnta64
3952     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3953     \bxjs@jachar@b
3954   \else\ifx.#4%
3955     \@tempcnta`#1 \multiply\@tempcnta64
3956     \advance\@tempcnta`#2 \multiply\@tempcnta64
3957     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3958     \bxjs@jachar@b
3959   \else
3960     \@tempcnta`#1 \multiply\@tempcnta64
3961     \advance\@tempcnta`#2 \multiply\@tempcnta64
3962     \advance\@tempcnta`#3 \multiply\@tempcnta64
3963     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3964     \bxjs@jachar@b
3965   \fi\fi\fi}
```

符号値が `\@tempcnta` の和文文字を出力する処理。

```
3966 \ifjsWithupTeX
3967   \def\bxjs@jachar@b{\kchar\@tempcnta}
3968 \else
3969   \def\bxjs@jachar@b{%
3970     \ifx\bxUInt\@undefined\else
3971       \bxUInt{\@tempcnta}%
3972     \fi}
3973 \fi
```

和欧文間空白の命令 `\jathinspace` の実装。

```
3974 \ifbxjs@jaspace@cmd
3975   \def\jathinspace{\hskip\xkanjiskip}
3976 \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。

```
3977 \ifbxjs@jp@jismmiv
3978   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3979   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3980 \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。

```

3981 \let\bxjs@jafont@paren\@firstofone
3982 \let\bxjs@tmpa\jsJaFont
3983 \ifx\bxjs@tmpa\bxjs@@auto
3984   \let\bxjs@tmpa\@empty
3985 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3986   \def\bxjs@tmpa{noembed}
3987 \fi\fi
3988 \bxjs@resolve@jafont@paren\bxjs@tmpa
3989 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3990 \ifx\bxjs@tmpa\@empty\else
3991   \edef\bxjs@next{%
3992     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3993   }\bxjs@next
3994 \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……。)

```

3995 \begingroup
3996   \global\let\bxjs@g@tmpa\relax
3997   \catcode`\|=0 \catcode`\|=12
3998   |def|bxjs@tmpdo#1|@nil{%
3999     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4000   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4001     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4002   |catcode`\|=0 \catcode`\|=12
4003   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4004     \ifx$#2$\else
4005       \xdef\bxjs@g@tmpa{%
4006         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4007     \fi}
4008 \@firstofone{%
4009   \catcode10=12 \endlinechar\m@ne
4010   \let\do\@makeother \dospecials \catcode32=10
4011   \openin\@inputcheck=otf.sty\relax
4012   \@tempswatrue
4013   \loop\if@tempswa
4014     \ifeof\@inputcheck \@tempswafalse \fi
4015     \if@tempswa
4016       \read\@inputcheck to\bxjs@next

```

```

4017 \expandafter\bxjs@tmpdo\bxjs@next\@nil
4018 \fi
4019 \repeat
4020 \closein\@inputcheck
4021 \endgroup}
4022 \bxjs@g@tmpa

```

■hyperref 対策 unicode にはいけない。

```

4023 \ifbxjs@hyperref@enc
4024 \bxjs@check@hyperref@unicode{false}
4025 \fi

```

tounicode special 命令を出力する。

```

4026 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4027 \else\ifjsWithpTeXng T\else F\fi\fi T%
4028 \ifnum\jis"2121="A1A1 %euc
4029 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4030 \else\ifnum\jis"2121="8140 %sjis
4031 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4032 \else\ifnum\jis"2121="3000 %uptex
4033 \ifbxjs@bigcode
4034 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4035 \PassOptionsToPackage{bigcode}{pxjahyper}
4036 \else
4037 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4038 \fi
4039 \fi\fi\fi
4040 \let\bxToUnicodeSpecialDone=t
4041 \fi

```

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

```

4042 \ifx f\bxjs@enablejfam\else
4043 \@enablejfamtrue
4044 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

4045 \if@enablejfam
4046 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4047 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4048 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4049 \jfam\symmincho
4050 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4051 \g@addto@macro\bxjs@begin@document@hook{%
4052 \ifx\reDeclareMathAlphabet\undefined\else
4053 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4054 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4055 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4056 \fi}

```

4057 \fi

C.3 pdfTeX 用設定 : CJK + bxcjkatype

4058 \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。

```
4059 \bxjs@adjust@jafont{f}
4060 \let\bxjs@jafont@paren\@firstofone
4061 \bxjs@resolve@jafont@paren\bxjs@tmpa
4062 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4063 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4064 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4065   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4066 \fi
4067 \edef\bxjs@next{%
4068   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4069 }\bxjs@next
4070 \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。

```
4071 \ifbxjs@hyperref@enc
4072   \PassOptionsToPackage{unicode}{hyperref}
4073 \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。

```
4074 \ifx\bxchkatypeHyperrefPatchDone\@undefined
4075 \begingroup
4076   \CJK@input{UTF8.bdg}
4077 \endgroup
4078 \g@addto@macro\pdfstringdefPreHook{%
4079   \@nameuse{CJK@UTF8Binding}%
4080 }
4081 \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。

```
4082 \ifx\bxchkatypeHyperrefPatchDone\@undefined
4083 \g@addto@macro\pdfstringdefPreHook{%
```

```

4084 \ifx~\bxjs@@CJKtilde
4085 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4086 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4087 \let~\@empty
4088 \fi
4089 }
4090 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4091 \def\bxjs@@tildecmd{~}
4092 \def\bxjs@LetUnexpandableSpace#1{%
4093 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4094 \bxjs@org@LetUnexpandableSpace#1%
4095 \fi}
4096 \fi

```

■ 共通命令の実装

```

4097 \newskip\jsKanjiSkip
4098 \newskip\jsXKanjiSkip
4099 \ifx\CJKecglue\@undefined
4100 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4101 \fi
4102 \let\autospacing\bxjs@enable@kanjiskip
4103 \let\noautospacing\bxjs@disable@kanjiskip
4104 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4105 \def\bxjs@apply@kanjiskip{%
4106 \jsKanjiSkip\@tempskipa
4107 \let\CJKglue\bxjs@CJKglue}
4108 \let\autoxspacing\bxjs@enable@xkanjiskip
4109 \let\noautoxspacing\bxjs@disable@xkanjiskip
4110 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4111 \def\bxjs@apply@xkanjiskip{%
4112 \jsXKanjiSkip\@tempskipa
4113 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4114 \def\bxjs@jachar#1{%
4115 \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4116 \ifbxjs@jaspace@cmd
4117 \protected\def\jathinspace{\CJKecglue}
4118 \fi

```

■ 和文数式ファミリ CJK パッケージは（恐らく）数式文字として CJK 文字をサポートしていない。従って @enablejfam は常に偽になる。

```

4119 \ifx t\bxjs@enablejfam
4120 \ClassWarningNoLine\bxjs@clsname
4121 {You cannot use 'enablejfam=true', since the\MessageBreak
4122 CJK package does not support Japanese math}
4123 \fi

```

C.4 Xe_{La}TeX 用設定 : xeCJK + zxjatype

```
4124 \else\if x\jsEngine
```

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

```
4125 \RequirePackage{zxjatype}
4126 \PassOptionsToPackage{no-math}{fontspec}%!
4127 \PassOptionsToPackage{xetex}{graphicx}%!
4128 \PassOptionsToPackage{xetex}{graphics}%!
4129 \ifx\zxJaFamilyName\@undefined
4130   \ClassError{bxjs}{clsname}
4131   {xeCJK or zxjatype is too old}\@ehc
4132 \fi
```

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

※ 2.0 版より既定を IPAex から原ノ味に変更。

```
4133 \bxjs@adjust@jafont{f}
4134 \let\bxjs@jafont@paren\@gobble
4135 \bxjs@resolve@jafont@paren\bxjs@tmpa
4136 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4137 \ifx\bxjs@tmpa\@empty
4138   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4139   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4140 \else
4141   \edef\bxjs@next{%
4142     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4143   }\bxjs@next
4144 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

Xe_{La}TeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、Xe_{La}TeX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「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。

```
4145 \ifnum\stricmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4146   \ifbxjs@hyperref@enc
4147     \PassOptionsToPackage{unicode}{hyperref}
```

```

4148 \fi
4149 \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。

```

4150 \AtEndOfPackage{%
4151 \def\@gnewline #1{%
4152   \ifvmode \@nolnerr
4153   \else
4154     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4155     \nobreak \hskip-1sp\hskip1sp\relax
4156     \ignorespaces
4157   \fi}
4158 }

```

■共通命令の実装

```

4159 \newskip\jsKanjiSkip
4160 \newskip\jsXKanjiSkip
4161 \ifx\CJKecglue\@undefined
4162   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4163 \fi
4164 \let\autospacing\bxjs@enable@kanjiskip
4165 \let\noautospacing\bxjs@disable@kanjiskip
4166 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4167 \def\bxjs@apply@kanjiskip{%
4168   \jsKanjiSkip\@tempskipa
4169   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4170 \let\autoxspacing\bxjs@enable@xkanjiskip
4171 \let\noautoxspacing\bxjs@disable@xkanjiskip
4172 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4173 \def\bxjs@apply@xkanjiskip{%
4174   \jsXKanjiSkip\@tempskipa
4175   \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。

```

4176 \ifx\mcfamily\@undefined
4177   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4178   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4179 \fi

```

\jachar のサブマクロの実装。

```

4180 \def\bxjs@jachar#1{%
4181   \xeCJKDeclareCharClass{CJK}{`#1}\relax

```



```

4182   #1}

      \jathinspace の実装。
4183 \ifbxjs@jaspace@cmd
4184   \protected\def\jathinspace{\CJKecglue}
4185 \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。

```

4186 \ifx t\bxjs@enablejfam
4187   \@enablejfamtrue
4188 \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。

```

4189 \if@enablejfam
4190   \xeCJKsetup{CJKmath=true}
4191 \fi

```

C.5 LuaTeX 用設定 : LuaTeX-ja

```

4192 \else\if l\jsEngine

```

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

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

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

```

4193 \let\zw\@undefined
4194 \RequirePackage{luatexja}
4195 \edef\bxjs@next{%
4196   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4197 }\bxjs@next

```

フォント代替の明示的定義。

```

4198 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4199 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4200 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4201 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4202 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4203 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4204 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4205 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4206 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4207 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4208 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }

```

```

4209 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4210 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4211 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4212 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4213 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4214 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4215 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4216 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4217 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4218 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4219 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4220 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4221 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4222 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4223 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4224 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4225 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4226 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4227 \DeclareFontShape{JT3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }

```

■和文フォント定義 \jsJaFont が指定された場合は、その値をオプションとして luatexja-preset を読み込む。非指定の場合は原ノ味フォントを指定する (luatexja-preset は読み込まない)。

※ 2.0 版より既定を IPAex から原ノ味に変更。

```

4228 \bxjs@adjust@jafont{t}
4229 \ifx\bxjs@tmpa\bxjs@@noEmbed
4230   \def\bxjs@tmpa{noembed}
4231 \fi
4232 \let\bxjs@jafont@paren\@gobble
4233 \bxjs@resolve@jafont@paren\bxjs@tmpa
4234 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4235 \ifx\bxjs@tmpa\@empty
4236   \defaultjfontfeatures{ Kerning=Off }
4237   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4238   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4239 \else
4240   \edef\bxjs@next{%
4241     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4242   }\bxjs@next
4243 \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。

```

4244 \@ifpackagelater{luatexja}{2016/03/31}{ }{\%else
4245 \DeclareRobustCommand\rmfamily
4246   {\not@math@alphabet\rmfamily\mathrm

```

```

4247 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4248 \DeclareRobustCommand\sffamily
4249 {\not@math@alphabet\sffamily\mathsf
4250 \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
4251 \DeclareRobustCommand\ttfamily
4252 {\not@math@alphabet\ttfamily\mathtt
4253 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4254 }
4255 \long\def\jttdefault{\gtdefault}
4256 \unless\ifx\@ltj@match@familytrue\@undefined
4257 \@ltj@match@familytrue
4258 \fi
4259 \g@addto@macro\bxjs@begin@document@hook{%
4260 \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4261 \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4262 \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4263 \bxjs@if@sf@default{%
4264 \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

4265 % 次の3つは既定値の通り
4266 \ltjsetparameter{prebreakpenalty}={`,10000}}
4267 \ltjsetparameter{postbreakpenalty}={` ",10000}}
4268 \ltjsetparameter{prebreakpenalty}={` ",10000}}
4269 \ltjsetparameter{jaxspmode}={` !,1}}
4270 \ltjsetparameter{jaxspmode}={` 〒,2}}
4271 \ltjsetparameter{alxspmode}={` +,3}}
4272 \ltjsetparameter{alxspmode}={` %,3}}

```

■段落頭でのグルー挿入禁止 基本的に現状の `ltjs*` クラスの処理に合わせる。

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の `LuaTeX-j` で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合に備えて同等のものを用意する。

```

4273 \ifx\ltjfakeparbegin\@undefined
4274 \protected\def\ltjfakeparbegin{%
4275 \ifhmode
4276 \relax\directlua{%
4277 luatexja.jfmglue.create_beginpar_node()}}
4278 \fi}
4279 \fi

```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```

4280 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4281 \begingroup
4282 \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4283 \@gobble\if\def\bxjs@tmpa{\@{\everypar}\fi}}
4284 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat

```

```

4285 \gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}\fi}}\fi
4286 \directlua{
4287   local function patchcmd(cs, code, from, to)
4288     tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4289       :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4290   end
4291   patchcmd(\@xsect, [[\meaning\xsect]],
4292     \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4293   patchcmd(\@item, [[\meaning\item]],
4294     \bxjs@tmpa, \@ltjfakeparbegin)}
4295 \endgroup
4296 \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。

```

4297 \ifbxjs@hyperref@enc
4298   \PassOptionsToPackage{unicode}{hyperref}
4299   \bxjs@check@hyperref@unicode{true}
4300 \fi

```

■共通命令の実装

```

4301 \protected\def\autospacing{%
4302   \ltjsetparameter{autospacing=true}}
4303 \protected\def\noautospacing{%
4304   \ltjsetparameter{autospacing=false}}
4305 \protected\def\autoxspacing{%
4306   \ltjsetparameter{autoxspacing=true}}
4307 \protected\def\noautoxspacing{%
4308   \ltjsetparameter{autoxspacing=false}}
4309 \def\bxjs@apply@kanjiskip{%
4310   \ltjsetparameter{kanjiskip={\@tempskipa}}}
4311 \def\bxjs@apply@xkanjiskip{%
4312   \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4313 \def\bxjs@jachar#1{%
4314   \ltjjachar`#1\relax}

```

\jathinspace の実装。

```

4315 \ifbxjs@jaspace@cmd
4316   \protected\def\jathinspace{%
4317     \hskip\ltjgetparameter{xkanjiskip}\relax}
4318 \fi

```

■和文数式ファミリ Lua_T_EX-j_a では和文数式ファミリは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablej_{fam} は常に真になる。

```

4319 \ifx f\bxjs@enablejfam
4320   \ClassWarningNoLine\bxjs@clsname

```

```

4321 {You cannot use 'enablejfam=false', since the\MessageBreak
4322 LuaTeX-japanese always provides Japanese math families}
4323 \fi

```

C.6 共通処理 (2)

```
4324 \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 4325 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4326 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4327 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4328 \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。

```

4329 \ifx\mathmc\@undefined
4330 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4331 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4332 \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)。

```

4333 \ifbxjs@jaspace@cmd
4334 \bxjs@protected\def\bxjs@choice@jathinspace{%
4335 \relax\ifmmode \mskip\medmuskip
4336 \else \jathinspace\ignorespaces
4337 \fi}
4338 \jsAtEndOfClass{%
4339 \ifjsWiththeTeX \let\>\bxjs@choice@jathinspace
4340 \else \def\>{\protect\bxjs@choice@jathinspace}%
4341 \fi}
4342 \fi

```

■和文・和欧文間空白の初期値

```

4343 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4344 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4345 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4346 \fi

```

以上で終わり。

```
4347 %</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。

```
4348 %<*modern>
4349 \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。

```
4350 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4351 \def\encodingdefault{T1}%
4352 \input{t1enc.def}%
4353 \fontencoding\encodingdefault\selectfont
4354 \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。

```
4355 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4356 \renewcommand{\rmdefault}{lmr}
4357 \renewcommand{\sfdefault}{lms}
4358 \renewcommand{\ttdefault}{lmtt}
4359 \fi
```

大型演算子用の数式フォントの設定。

※`amsmaths` パッケージと同等にする。

```
4360 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4361   <-7.5>cmex7<7.5-8.5>cmex8%
4362   <8.5-9.5>cmex9<9.5->cmex10}{}%
4363 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
```

`amsmaths` 読み込時に上書きされるのを防ぐ。

```
4364 \def\cmex@opt{10}
```

D.2 fixltx2e 読み込

※`fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```
4365 \ifx\@IncludeInRelease\@undefined
4366 \RequirePackage{fixltx2e}
4367 \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。

```
4368 \RequirePackage{bxjscjkat}
```

D.4 完了

おしまい。

```
4369 %</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。

```
4370 %<*pandoc>
```

```
4371 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4372 \RequirePackage{bxjspandoc}
```

`\bxjs@endpreamble@hook` etoolbox の `\AtEndPreamble` で実行される BXJS クラス用のフック。

※ ϵ - $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```
4373 \@onlypreamble\bxjs@endpreamble@hook
```

```
4374 \let\bxjs@endpreamble@hook\@empty
```

```
4375 \ifjsWitheTeX
```

```
4376 \RequirePackage{etoolbox}[2010/08/21]% v2.0
```

```
4377 \AtEndPreamble{\bxjs@endpreamble@hook}
```

```
4378 \fi
```

E.1 dupload システム

パッケージが重複して読み込まれたときに “option clash” の検査をスキップする。この時に何らかのコードを実行させることができる。

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

```
4379 \@onlypreamble\bxjs@set@dupload@proc
```

```
4380 \def\bxjs@set@dupload@proc#1{%
```

```
4381 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
4382 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
4383 \def\bxjs@set@dupload@proc@a#1{%
```

```
4384 \@onlypreamble#1\def#1##1}
```

```

4385 \def\bxjs@unset@dupload@proc#1{%
4386   \expandafter\let\csname bxjs@dlp/#1\endcsname\@undefined}

```

\@if@ptions \@if@ptions の再定義。

```

4387 \@onlypreamble\bxjs@org@if@ptions
4388 \let\bxjs@org@if@ptions\@if@ptions
4389 \@onlypreamble\bxjs@org@reset@ptions
4390 \let\bxjs@org@reset@ptions\relax
4391 \def\@if@ptions#1#2#3{%
4392   \let\bxjs@next\@secondoftwo
4393   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@current}%
4394   \ifx\bxjs@tmpa\bxjs@tmpb
4395     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4396       \let\bxjs@next\@firstoftwo \fi
4397   \fi
4398   \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@ptions{#1}{#2}{#3}}
4399 \g@addto@macro\bxjs@begin@document@hook{%
4400   \let\@if@ptions\bxjs@org@if@ptions}
4401 \@onlypreamble\bxjs@do@dupload@proc
4402 \def\bxjs@do@dupload@proc#1#2#3{%
4403   \ifx\bxjs@org@reset@ptions\relax
4404     \let\bxjs@org@reset@ptions\@reset@ptions
4405   \fi
4406   \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4407   \def\@reset@ptions{%
4408     \let\@reset@ptions\bxjs@org@reset@ptions
4409     \@reset@ptions
4410     \bxjs@next{#3}}%
4411   \@firstoftwo}

```

E.2 lang 変数

lang=ja という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐための対策を行う。

\bxjs@polyglossia@options Polyglossia のオプション列のテキスト。“実際には読み込まれていない” 場合は \relax になる。

```

4412 \let\bxjs@polyglossia@options\relax

```

\bxjs@babel@options Babel のオプション列のテキスト。“実際には読み込まれていない” 場合は \relax になる。

```

4413 \let\bxjs@babel@options\relax

```

■Polyglossia について つまり Xe_{La}TeX・Lua_{La}TeX の場合。

※この場合 etoolbox が使用可能になっている。

```

4414 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0

```

パッケージの読込を検知するため読込済のマークを付けて dupload の処理を仕込む。


```

4415 \pandocSkipLoadPackage{polyglossia}
4416 \bxjs@set@dupload@proc{polyglossia.sty}{%
4417   \bxjs@unset@dupload@proc{polyglossia.sty}%
4418   \ClassWarning\bxjs@clsname
4419   {Package polyglossia is required}%
4420   \def\bxjs@polyglossia@options{#1}%

```

polyglossia の読込が指示された場合、直後に \setmainlanguage が実行されることを想定して、フック用の \setmainlanguage を定義する。

※先に \setmainlanguage 以外が実行された場合はエラーになる。

```

4421 \newcommand*\setmainlanguage[2][]{%

```

もし、言語名が空の \setmainlanguage{} が実行された場合は、lang=ja が指定されたと見なす。

```

4422   \ifblank{##2}{%
4423     \ClassWarning\bxjs@clsname
4424     {Main language is invalid, thus the fallback\MessageBreak
4425       definitions will be employed}%
4426     \bxjs@pandoc@polyglossia@ja

```

言語名が非空ならば、本来の polyglossia の処理を実行する。

```

4427   }{%else
4428     \ClassWarning\bxjs@clsname
4429     {Main language is valid (##2),\MessageBreak
4430       thus polyglossia will be loaded}%
4431     \csundef{ver@polyglossia.sty}%
4432     \edef\bxjs@next{%
4433       \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4434     }\bxjs@next
4435     \setmainlanguage[##1]{##2}%
4436   }}}

```

プレアンブルで polyglossia の読込が指示されなかった場合、Polyglossia と連携するパッケージの誤動作を防ぐため、読込済マークを外す。

```

4437 \g@addto@macro\bxjs@endpreamble@hook{%
4438   \ifx\bxjs@polyglossia@options\relax
4439     \csundef{ver@polyglossia.sty}%
4440   \fi}

```

\bxjs@pandoc@polyglossia@ja Pandoc 側で lang=ja が指定されていた場合の処理。この場合は Polyglossia の処理を無効化するためにダミーの定義を行う。その時点でダミーの \setotherlanguage(s) を定義する。

```

4441 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4442 \def\bxjs@pandoc@polyglossia@ja{%
4443   \renewcommand*\setmainlanguage[2][]{}%
4444   \newcommand*\setotherlanguage[2][]{%
4445     \ifblank{##2}{-}{%else
4446       \cslet{##2}\@empty \cslet{end##2}\@empty
4447       \cslet{text##2}\@firstofone}}}%

```

```

4448 \newcommand*\setotherlanguages[2][]{%
4449   \@for\bxjs@tmpa:={##2}\do{%
4450     \setotherlangauge{\bxjs@tmpa}}}%

```

Polyglossia の読込済マークは外れるようにしておく。

```

4451 \let\bxjs@polyglossia@options\relax}%
4452 \fi

```

■Babel について Xe_{La}TeX 以外の場合。

※ Pandoc 2.6 版において、Lua_{La}TeX で用いられる多言語パッケージが Polyglossia から Babel に変更された。

```

4453 \if x\jsEngine\else

```

パッケージの読込を検知するため読込済のマークを付けて dupload の処理を仕込む。

```

4454 \pandocSkipLoadPackage{babel}
4455 \bxjs@set@dupload@proc{babel.sty}{%
4456   \bxjs@unset@dupload@proc{babel.sty}%
4457   \ClassWarning\bxjs@clsname
4458   {Package babel is required}}%

```

パッケージオプションに言語名が空の main= があるかを調べる。ある場合は lang=ja 対策を実行する。

※\bxjs@babel@options には main= を除いたオプション列を格納する。

```

4459 \@tempswafalse \let\bxjs@babel@options\@empty
4460 \def\bxjs@tmpb{main=}%
4461 \@for\bxjs@tmpa:={#1}\do{%
4462   \ifx\bxjs@tmpa\bxjs@tmpb \@tempswattrue
4463   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4464   \fi}%
4465 \if@tempswa
4466   \ClassWarning\bxjs@clsname
4467   {Main language is invalid, so fallback\MessageBreak
4468     definitions will be employed}%
4469   \bxjs@pandoc@polyglossia@ja

```

ない場合は、本来の babel の処理を実行する。

```

4470 \else
4471   \ClassWarning\bxjs@clsname
4472   {Main language is valid,\MessageBreak
4473     thus babel will be loaded}%
4474   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4475   \RequirePackage[#1]{babel}[]%
4476 \fi}

```

プレアンブルで babel の読込が指示されなかった場合、読込済マークを外す。

```

4477 \g@addto@macro\bxjs@endpreamble@hook{%
4478   \ifx\bxjs@babel@options\relax
4479     \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4480   \fi}

```

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

```
4481 \onlypreamble\bxjs@pandoc@babel@ja
4482 \def\bxjs@pandoc@babel@ja{%
4483   \expandafter\let\csname ver@babel.sty\endcsname\undefined
4484   \edef\bxjs@next{%
4485     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4486   }\bxjs@next
4487   \if j\jsEngine
4488     \RequirePackage[main=japanese]{pxbabel}[]%
4489   \else
4490     \RequirePackage{bxorigcapt}[]%
4491   \fi}

    lang 対策はこれで終わり。
4492 \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。

```
4493 \bxjs@set@dupload@proc{geometry.sty}{%
4494   \setpagelayout*{#1}}
```

E.4 CJKmainfont 変数

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

```
4495 \if l\jsEngine
4496   \pandocSkipLoadPackage{xeCJK}
4497   \providecommand*\setCJKmainfont{\setmainjfont}
4498 \fi
```

ムニャムニャムニャ。

```
4499 \if x\jsEngine
4500   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4501     ,space}
4502 \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。

```
4503 \g@addto@macro\bxjs@begin@document@hook{%
4504   \@tempswafalse
```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```
4505   \ifx\oldparagraph\@undefined\else
4506     \@tempwattrue
4507   \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4508   \ifnum\c@secnumdepth>3
4509     \@tempwattrue
4510   \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。

```
4511   \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4512     \let\jsParagraphMark\@empty
4513   \fi\fi}
```

E.6 全角空白文字

```
4514 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4515   \catcode"3000=\active
4516   \begingroup \catcode`\!=7
4517   \protected\gdef!!!3000{\zwspace}
4518   \endgroup
4519 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4520   \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4521   \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4522 \fi\fi
```

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

```
4523 \if j\jsEngine
4524   \bxjs@fix@hyperref@unicode{false}
4525 \else
4526   \bxjs@fix@hyperref@unicode{true}
4527 \fi
```

E.7 完了

おしまい。

```
4528 %</pandoc>
```

和文ドライバ実装はここまで。

```
4529 %</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。

```
4530 %<*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

```
4531 %<*compat>
```

```
4532 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
4533 \let\bxac@engine=n
```

```
4534 \def\bxac@do#1#2{%
```

```
4535   \edef\bxac@tmpa{\string#1}%
```

```
4536   \edef\bxac@tmpb{\meaning#1}%
```

```
4537   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
4538 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
4539 \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。

```
4540 \ifx\jsAtEndOfClass\undefined
```

```
4541   \let\bxac@delayed@if@bxjs\@firstofone
```

```
4542 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
4543 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4544 \newif\ifbxac@in@old@behavior
```

```
4545 \let\ImposeOldLuaTeXBehavior\relax
```

```
4546 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 X_YTEX 部分

```
4547 \ifx x\bxac@engine
```

X_YTEX 文字クラスのムニャムニャ。

```

4548 \@onlypreamble\bxac@adjust@charclass
4549 \bxac@delayed@if@bxjs{%
4550   \@ifpackageloaded{xeCJK}{-}{%else
4551     \ifx\XeTeXalloc@intercharclass\undefined\else
4552       \ifnum\XeTeXalloc@intercharclass=\z@
4553         \PackageInfo\bxac@pkgname
4554           {Setting up interchar class for CJK...\@gobble}%
4555         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4556           \XeTeXalloc@intercharclass=3
4557         }{%else
4558           \PackageWarning\bxac@pkgname
4559             {Cannot find file 'load-unicode-xetex-classes.tex'%
4560             \@gobble}%
4561         }%
4562       \fi\fi
4563     \ifnum\XeTeXcharclass"3041=\z@
4564       \PackageInfo\bxac@pkgname
4565         {Adjusting interchar class for CJK...\@gobble}%
4566       \@for\bxac@x={%
4567         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4568         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4569         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4570         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4571         31FF%
4572       }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4573     \fi
4574   }%
4575 }

```

以上。

```
4576 \fi
```

G.3 LuaTeX 部分

```

4577 \ifx l\bxac@engine
      ムニヤムニヤ。
4578 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4579   \chardef\pdfTeXversion=200
4580   \def\pdfTeXrevision{0}
4581   \let\pdfTeXbanner\luatexbanner
4582 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior 4583 \begingroup\expandafter\expandafter\expandafter\endgroup
4584 \expandafter\ifx\csname outputmode\endcsname\relax\else
4585 \def\bxac@ob@list{%
4586   \do{\let}\pdfoutput{\outputmode}%
4587   \do{\let}\pdfpagewidth{\pagewidth}%
4588   \do{\let}\pdfpageheight{\pageheight}%
4589   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%

```

```

4590 \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}}
4591 \def\bxac@ob@do#1#2{\begingroup
4592 \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4593 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4594 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4595 \else \let#3#1\relax \let#1\@undefined
4596 \fi}
4597 \protected\def\ImposeOldLuaTeXBehavior{%
4598 \unless\ifbxac@in@old@behavior
4599 \bxac@in@old@behaviortrue
4600 \let\do\bxac@ob@do \bxac@ob@list
4601 \fi}
4602 \protected\def\RevokeOldLuaTeXBehavior{%
4603 \ifbxac@in@old@behavior
4604 \bxac@in@old@behaviorfalse
4605 \let\do\bxac@ob@do \bxac@ob@list
4606 \fi}
4607 \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。

```

4608 \ifnum\luatexversion>64 \directlua{
4609   local function range(cs, ce, cc, ff)
4610     if ff or not tex.getcatcode(cs) == cc then
4611       local setcc = tex.setcatcode
4612       for c = cs, ce do setcc(c, cc) end
4613     end
4614   end
4615   range(0x3400, 0x4DB5, 11, false)
4616   range(0x4DB5, 0x4DBF, 11, true)
4617   range(0x4E00, 0x9FCC, 11, false)
4618   range(0x9FCD, 0x9FFF, 11, true)
4619   range(0xAC00, 0xD7A3, 11, false)
4620   range(0x20000, 0x2A6D6, 11, false)
4621   range(0x2A6D7, 0x2A6FF, 11, true)
4622   range(0x2A700, 0x2B734, 11, false)
4623   range(0x2B735, 0x2B73F, 11, true)
4624   range(0x2B740, 0x2B81D, 11, false)
4625   range(0x2B81E, 0x2B81F, 11, true)
4626   range(0x2B820, 0x2CEA1, 11, false)
4627   range(0x2CEA2, 0x2FFFD, 11, true)
4628 } \fi

```

以上。

```

4629 \fi

```

G.4 完了

おしまい。

```

4630 %</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

```
4631 %<*cjkcat>
4632 \def\bxjx@pkgname{bxjscjkat}
4633 \newcount\bxjx@cna
4634 \@onlypreamble\bxjx@tmpdo
4635 \@onlypreamble\bxjx@tmpdo@a
4636 \@onlypreamble\bxjx@tmpdo@b
```

\bxjx@engine エンジンの種別。

```
4637 \let\bxjx@engine=n
4638 \def\bxjx@tmpdo#1#2{%
4639   \edef\bxjx@tmpa{\string#1}%
4640   \edef\bxjx@tmpb{\meaning#1}%
4641   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4642 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4643 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4644 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4645 \bxjx@tmpdo\pdftexversion{\let\bxjx@engine=p}
4646 \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。

```
4647 \def\bxjx@tmpdo#1#2{%
4648   \if#1\bxjx@engine
4649     \@ifpackageloaded{#2}{\}%else
4650     \PackageError\bxjx@pkgname
4651       {Package '#2' must be loaded}%
4652       {Package loading is aborted.\MessageBreak\@ehc}%
4653     \endinput}
4654 \fi}
4655 \bxjx@tmpdo{p}{bxcjkatype}
4656 \bxjx@tmpdo{x}{xeCJK}
4657 \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。

```
4658 \ifx\TextOrMath\@undefined
4659   \RequirePackage{fixltx2e}
4660 \fi
```


H.2 和文カテゴリコードの設定

upLaTeX の場合、和文カテゴリコードの設定を LuaTeX-jā と（ほぼ）等価なものに変更する。

※ LuaTeX-jā との相違点：A830、A960、1B000。

```
4661 \if u\bxjx@engine
4662 \@for\bxjx@tmpa:={%
4663 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4664 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4665 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4666 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4667 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4668 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4669 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4670 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4671 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4672 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4673 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4674 FE70,FFF0,%
4675 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4676 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4677 10480,104B0,10500,10530,10600,10800,10840,10860,%
4678 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4679 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4680 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4681 11200,11280,112B0,11300,11400,11480,11580,11600,%
4682 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4683 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4684 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4685 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4686 1EE00,1F000,1F030,1FOA0,1F300,1F600,1F650,1F680,%
4687 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4688 00C0%
4689 }\do{%
4690 \@tempcnta="\bxjx@tmpa\relax
4691 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4692 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4693 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4694 \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。

```

4695 \@onlypreamble\bxjx@grkcyr@list
4696 \def\bxjx@grkcyr@list{%
4697 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4698 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4699 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4700 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4701 \do{0395}{LGR}{\textEpsilon}{E}%       % GR. C. L. EPSILON
4702 \do{0396}{LGR}{\textZeta}{Z}%          % GR. C. L. ZETA
4703 \do{0397}{LGR}{\textEta}{H}%           % GR. C. L. ETA
4704 \do{0398}{LGR}{\textTheta}{\Theta}%    % GR. C. L. THETA
4705 \do{0399}{LGR}{\textIota}{I}%          % GR. C. L. IOTA
4706 \do{039A}{LGR}{\textKappa}{K}%         % GR. C. L. KAPPA
4707 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4708 \do{039C}{LGR}{\textMu}{M}%            % GR. C. L. MU
4709 \do{039D}{LGR}{\textNu}{N}%           % GR. C. L. NU
4710 \do{039E}{LGR}{\textXi}{\Xi}%         % GR. C. L. XI
4711 \do{039F}{LGR}{\textOmicron}{O}%      % GR. C. L. OMICRON
4712 \do{03A0}{LGR}{\textPi}{\Pi}%         % GR. C. L. PI
4713 \do{03A1}{LGR}{\textRho}{P}%          % GR. C. L. RHO
4714 \do{03A3}{LGR}{\textSigma}{\Sigma}%    % GR. C. L. SIGMA
4715 \do{03A4}{LGR}{\textTau}{T}%          % GR. C. L. TAU
4716 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4717 \do{03A6}{LGR}{\textPhi}{\Phi}%       % GR. C. L. PHI
4718 \do{03A7}{LGR}{\textChi}{X}%          % GR. C. L. CHI
4719 \do{03A8}{LGR}{\textPsi}{\Psi}%       % GR. C. L. PSI
4720 \do{03A9}{LGR}{\textOmega}{\Omega}%   % GR. C. L. OMEGA
4721 \do{03B1}{LGR}{\textalpha}{\alpha}%   % GR. S. L. ALPHA
4722 \do{03B2}{LGR}{\textbeta}{\beta}%     % GR. S. L. BETA
4723 \do{03B3}{LGR}{\textgamma}{\gamma}%   % GR. S. L. GAMMA
4724 \do{03B4}{LGR}{\textdelta}{\delta}%   % GR. S. L. DELTA
4725 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
4726 \do{03B6}{LGR}{\textzeta}{\zeta}%     % GR. S. L. ZETA
4727 \do{03B7}{LGR}{\texteta}{\eta}%       % GR. S. L. ETA
4728 \do{03B8}{LGR}{\texttheta}{\theta}%   % GR. S. L. THETA
4729 \do{03B9}{LGR}{\textiota}{\iota}%     % GR. S. L. IOTA
4730 \do{03BA}{LGR}{\textkappa}{\kappa}%   % GR. S. L. KAPPA
4731 \do{03BB}{LGR}{\textlambda}{\lambda}% % GR. S. L. LAMDA
4732 \do{03BC}{LGR}{\textmu}{\mu}%         % GR. S. L. MU
4733 \do{03BD}{LGR}{\textnu}{\nu}%        % GR. S. L. NU
4734 \do{03BE}{LGR}{\textxi}{\xi}%        % GR. S. L. XI
4735 \do{03BF}{LGR}{\textomicron}{o}%       % GR. S. L. OMICRON
4736 \do{03C0}{LGR}{\textpi}{\pi}%        % GR. S. L. PI

```

4737 \do{03C1}{LGR}{\textrho}{\rho}% % GR. S. L. RHO
 4738 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
 4739 \do{03C3}{LGR}{\textsigma}{\sigma}% % GR. S. L. SIGMA
 4740 \do{03C4}{LGR}{\texttau}{\tau}% % GR. S. L. TAU
 4741 \do{03C5}{LGR}{\textupsilon}{\upsilon}% % GR. S. L. UPSILON
 4742 \do{03C6}{LGR}{\textphi}{\phi}% % GR. S. L. PHI
 4743 \do{03C7}{LGR}{\textchi}{\chi}% % GR. S. L. CHI
 4744 \do{03C8}{LGR}{\textpsi}{\psi}% % GR. S. L. PSI
 4745 \do{03C9}{LGR}{\textomega}{\omega}% % GR. S. L. OMEGA
 4746 \do{0401}{T2A}{\CYRYO}{}% % CY. C. L. IO
 4747 \do{0410}{T2A}{\CYRA}{}% % CY. C. L. A
 4748 \do{0411}{T2A}{\CYRB}{}% % CY. C. L. BE
 4749 \do{0412}{T2A}{\CYRV}{}% % CY. C. L. VE
 4750 \do{0413}{T2A}{\CYRG}{}% % CY. C. L. GHE
 4751 \do{0414}{T2A}{\CYRD}{}% % CY. C. L. DE
 4752 \do{0415}{T2A}{\CYRE}{}% % CY. C. L. IE
 4753 \do{0416}{T2A}{\CYRZH}{}% % CY. C. L. ZHE
 4754 \do{0417}{T2A}{\CYRZ}{}% % CY. C. L. ZE
 4755 \do{0418}{T2A}{\CYRI}{}% % CY. C. L. I
 4756 \do{0419}{T2A}{\CYRISHRT}{}% % CY. C. L. SHORT I
 4757 \do{041A}{T2A}{\CYRK}{}% % CY. C. L. KA
 4758 \do{041B}{T2A}{\CYRL}{}% % CY. C. L. EL
 4759 \do{041C}{T2A}{\CYRM}{}% % CY. C. L. EM
 4760 \do{041D}{T2A}{\CYRN}{}% % CY. C. L. EN
 4761 \do{041E}{T2A}{\CYRO}{}% % CY. C. L. O
 4762 \do{041F}{T2A}{\CYRP}{}% % CY. C. L. PE
 4763 \do{0420}{T2A}{\CYRR}{}% % CY. C. L. ER
 4764 \do{0421}{T2A}{\CYRS}{}% % CY. C. L. ES
 4765 \do{0422}{T2A}{\CYRT}{}% % CY. C. L. TE
 4766 \do{0423}{T2A}{\CYRU}{}% % CY. C. L. U
 4767 \do{0424}{T2A}{\CYRF}{}% % CY. C. L. EF
 4768 \do{0425}{T2A}{\CYRH}{}% % CY. C. L. HA
 4769 \do{0426}{T2A}{\CYRC}{}% % CY. C. L. TSE
 4770 \do{0427}{T2A}{\CYRCH}{}% % CY. C. L. CHE
 4771 \do{0428}{T2A}{\CYRSH}{}% % CY. C. L. SHA
 4772 \do{0429}{T2A}{\CYRSHCH}{}% % CY. C. L. SHCHA
 4773 \do{042A}{T2A}{\CYRHRDSN}{}% % CY. C. L. HARD SIGN
 4774 \do{042B}{T2A}{\CYRERY}{}% % CY. C. L. YERU
 4775 \do{042C}{T2A}{\CYRSFTSN}{}% % CY. C. L. SOFT SIGN
 4776 \do{042D}{T2A}{\CYREREV}{}% % CY. C. L. E
 4777 \do{042E}{T2A}{\CYRYU}{}% % CY. C. L. YU
 4778 \do{042F}{T2A}{\CYRYA}{}% % CY. C. L. YA
 4779 \do{0430}{T2A}{\cyra}{}% % CY. S. L. A
 4780 \do{0431}{T2A}{\cyrb}{}% % CY. S. L. BE
 4781 \do{0432}{T2A}{\cyrv}{}% % CY. S. L. VE
 4782 \do{0433}{T2A}{\cyrg}{}% % CY. S. L. GHE
 4783 \do{0434}{T2A}{\cyrd}{}% % CY. S. L. DE
 4784 \do{0435}{T2A}{\cyre}{}% % CY. S. L. IE
 4785 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE

```

4786 \do{0437}{T2A}{\cyrz}{}% % CY. S. L. ZE
4787 \do{0438}{T2A}{\cyri}{}% % CY. S. L. I
4788 \do{0439}{T2A}{\cyrishrt}{}% % CY. S. L. SHORT I
4789 \do{043A}{T2A}{\cyrk}{}% % CY. S. L. KA
4790 \do{043B}{T2A}{\cyr1}{}% % CY. S. L. EL
4791 \do{043C}{T2A}{\cyrn}{}% % CY. S. L. EM
4792 \do{043D}{T2A}{\cyrn}{}% % CY. S. L. EN
4793 \do{043E}{T2A}{\cyro}{}% % CY. S. L. O
4794 \do{043F}{T2A}{\cyrp}{}% % CY. S. L. PE
4795 \do{0440}{T2A}{\cyrr}{}% % CY. S. L. ER
4796 \do{0441}{T2A}{\cyrs}{}% % CY. S. L. ES
4797 \do{0442}{T2A}{\cyr1}{}% % CY. S. L. TE
4798 \do{0443}{T2A}{\cyr1}{}% % CY. S. L. U
4799 \do{0444}{T2A}{\cyr1}{}% % CY. S. L. EF
4800 \do{0445}{T2A}{\cyrh}{}% % CY. S. L. HA
4801 \do{0446}{T2A}{\cyr1}{}% % CY. S. L. TSE
4802 \do{0447}{T2A}{\cyrch}{}% % CY. S. L. CHE
4803 \do{0448}{T2A}{\cyrsh}{}% % CY. S. L. SHA
4804 \do{0449}{T2A}{\cyrshch}{}% % CY. S. L. SHCHA
4805 \do{044A}{T2A}{\cyrhrdsn}{}% % CY. S. L. HARD SIGN
4806 \do{044B}{T2A}{\cyrery}{}% % CY. S. L. YERU
4807 \do{044C}{T2A}{\cyr1}{}% % CY. S. L. SOFT SIGN
4808 \do{044D}{T2A}{\cyrerev}{}% % CY. S. L. E
4809 \do{044E}{T2A}{\cyr1}{}% % CY. S. L. YU
4810 \do{044F}{T2A}{\cyr1}{}% % CY. S. L. YA
4811 \do{0451}{T2A}{\cyr1}{}% % CY. S. L. IO
4812 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4813 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4814 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4815 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN
4816 \do{00B4}{TS1}{\textasciicircum}{}% % ACUTE ACCENT
4817 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4818 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4819 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN
4820 }

```

`\mathdegree` 面倒なので補っておく。

```

4821 \providecommand*\mathdegree{{}^{\circ}}

```

`\ifbxjx@gcc@cjkl` [スイッチ]「特定 CJK 曖昧文字」を和文扱いにするか。

```

4822 \newif\ifbxjx@gcc@cjkl

```

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

```

4823 \newcommand*\greekasCJK{%

```

```

4824   \bxjx@gcc@cjkltrue}

```

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

```

4825 \newcommand*\nogreekasCJK{%

```

```

4826   \bxjx@gcc@cjklfalse}

```

`\bxjx@fake@grk` `\bxjx@fake@grk{〈出力文字〉}{〈基準文字〉}` : ラテン文字で代用される数式ギリシャ文字の出力を行う。〈基準文字〉 (`mathchardef` の制御綴) の数式クラスと数式ファミリを引き継いで、〈出力文字〉 (ASCII 文字トークン) の文字コードの数式文字を出力する。例えば、`\Pi` の意味が `\mathchar"7005` である場合、`\bxjx@fake@grk{B}{\Pi}` は `\mathchar"7042` を実行する。

※フォントパッケージ使用時の再定義を考慮して、〈基準文字〉が `mathchardef` であるかを検査し、そうでない場合はフォールバックとして単に 〈出力文字〉 を実行する。

```
4827 \def\bxjx@tmpdo#1\relax{%
4828   \def\bxjx@fake@grk##1##2{%
4829     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4830   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4831     \ifx\\##1\\%
4832       \bxjx@ccta##4\divide\bxjx@ccta\@cclvi
4833       \multiply\bxjx@ccta\@cclvi \advance\bxjx@ccta`##3\relax
4834       \mathchar\bxjx@ccta
4835     \else ##3\fi}
4836 }\expandafter\bxjx@tmpdo\string\mathchar\relax
```

■pdfTeX・upTeX の場合

```
4837 \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。

```
4838 \@ifpackageloaded{inputenc}{\fi}{%else
4839   \RequirePackage[utf8]{inputenc}}
4840 \def\bxjx@tmpa{utf8}
4841 \ifx\bxjx@tmpa\inputencdoingname
4842   \PackageWarningNoLine\bxjx@pkgname
4843     {Input encoding changed to utf8}%
4844   \inputencoding{utf8}%
4845 \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。

```
4846 \if u\bxjx@engine
4847   \kcatcode"0370=15
4848   \kcatcode"0400=15
4849   \kcatcode"0500=15
4850 \fi
```

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

```
4851 \def\bxjx@tmpdo#1{%
4852   \@tempccta="#1\relax
```

```

4853 \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4854 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

引数 = \backslash [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。

```

4855 \ifx\#5\%
4856 \def\bxjx@tmpa{\@inmathwarn#4}%

```

〈数式 LICR〉が英字である場合は \backslash bxjx@fake@grk で出力する。大文字なら \backslash Pi、小文字なら \backslash pi を基準文字にする。

```

4857 \else\ifcat A\noexpand#5%
4858 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4859 { \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。

```

4860 \else \def\bxjx@tmpa{#5}%
4861 \fi\fi
4862 \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4863 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

4864 \if u\bxjx@engine
4865 \def\bxjx@tmpdo@b#1#2#3#4#5{%

```

引数 = \backslash [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 \backslash bxjx@ja@or@not に帰着させる。この際に、和文用の定義として当該の kchardef を使用し、その制御綴として \backslash [bxjx@KC/...] を流用している。

```

4866 \kchardef#1=\@tempcnta
4867 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4868 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdfTeX の場合も処理はほとんど同じ。ただし、和文用の定義として \backslash UTF{〈符号値〉} を使う（ \backslash UTF は bxcjkatype の命令）。 \backslash [bxjx@KC/...] は使わないが定義済にする必要がある。

```

4869 \else\if p\bxjx@engine
4870 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4871 \mathchardef#1=\@tempcnta
4872 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4873 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4874 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

4875 \let\do\bxjx@tmpdo \bxjx@grkcyr@list

```

\backslash bxjx@DeclareUnicodeCharacter \backslash 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。

```

4876 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter

```

```

4877 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4878 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4879 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4880   \count@=#1\relax
4881   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4882     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4883   \else
4884     \wlog{ \space\space skipped defining Unicode char U+#1}%
4885   \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。

```

4886 \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。

```

4887   \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。

```

4888   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4889   \else \UseTextSymbol{#2}{#3}%
4890   \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。

```

4891 \begingroup
4892 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4893 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4894   \noexpand\bxjx@swap@DUC@cmd
4895   \the\toks@
4896   \noexpand\bxjx@swap@DUC@cmd}}
4897 \endgroup\next
4898 \def\bxjx@swap@DUC@cmd{%
4899   \let\bxjx@tmpa\DeclareUnicodeCharacter
4900   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4901   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
4902   \let\bxjx@tmpa\relax}

```

以上。

■ $\mathrm{X}\mathrm{\TeX}$ ・ $\mathrm{Lua}\mathrm{\TeX}$ の場合

```

4903 \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。

```

4904 \def\bxjx@tmpdo#1{%
4905   \bxjx@cmta=#1\relax
4906   \begingroup
4907     \lccode`~=\bxjx@cmta
4908   \lowercase{\endgroup

```

```

4909 \bxjx@tmpdo@a{~}{#1}}
4910 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

〈数式 LICR〉が空なら何もしない。空でない場合、up $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の場合と同じ方法で“数式中の動作”を決定し、当該の文字を math active にしてその動作を設定する。

```

4911 \ifx\#5\\\let\bxjx@tmpa\relax
4912 \else\ifcat A\noexpand#5%
4913 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4914 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4915 \else \def\bxjx@tmpa{#5}%
4916 \fi\fi
4917 \ifx\bxjx@tmpa\relax\else
4918 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4919 \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。

```

4920 \mathchardef\bxjx@tmpa="119
4921 \ifx\bxjx@tmpa\pi \let\do\bxjx@tmpdo \bxjx@grkcyr@list \fi

```

次に、テキストにおいて「特定 CJK 曖昧文字」の扱いが `\[no]greekasCJK` で切り替わるようにする。

Lua $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の場合は、Lua $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -ja の `jacharrange` の設定を変更する。

※ “範囲 2” がギリシャ・キリル文字、“範囲 8” が Latin-1 の記号。

```

4922 \if 1\bxjx@engine
4923 \protected\def\greekasCJK{%
4924 \bxjx@gcc@cjctrue
4925 \ltjsetparameter{jacharrange={+2, +8}}}
4926 \protected\def\nogreekasCJK{%
4927 \bxjx@gcc@cjkfalse
4928 \ltjsetparameter{jacharrange={-2, -8}}}
4929 \fi

```

X $\mathrm{E}_{\mathrm{L}}\mathrm{TeX}$ の場合、xeCJK は X $\mathrm{E}_{\mathrm{L}}\mathrm{TeX}$ 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。

```

4930 \if x\bxjx@engine
4931 \let\bxjx@gcc@cjkl@list\@empty
4932 \def\do#1#2#3#4{%
4933 \edef\bxjx@gcc@cjkl@list{\bxjx@gcc@cjkl@list
4934 \noexpand\XeTeXcharclass"#1\bxjx@cmta}}
4935 \bxjx@grkcyr@list
4936 \protected\def\greekasCJK{%
4937 \bxjx@gcc@cjctrue
4938 \bxjx@cmta=\@ne \bxjx@gcc@cjkl@list}
4939 \protected\def\nogreekasCJK{%
4940 \bxjx@gcc@cjkfalse
4941 \bxjx@cmta=\z@ \bxjx@gcc@cjkl@list}
4942 \fi

```


以上。

```
4943 \fi\fi
```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
4944 \nogreekasCJK
```

H.5 完了

おしまい。

```
4945 %</cjkat>
```

付録 I 補助パッケージ : bxjspanc

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

```
4946 %<*ancpandoc>
```

```
4947 %% このファイルは日本語文字を含みます.
```

```
4948 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```
4949 \let\bxjsp@engine=n
```

```
4950 \@onlypreamble\bxjsp@do
```

```
4951 \def\bxjsp@do#1#2{%
```

```
4952   \edef\bxjsp@tmpa{\string#1}%
```

```
4953   \edef\bxjsp@tmpb{\meaning#1}%
```

```
4954   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
```

```
4955 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
```

```
4956 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
```

```
4957 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
```

```
4958 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

I.2 パッケージオプション

english オプションが指定されている場合、\ldots の調整を抑止する。

※つまり、「グローバルの english オプション」が指定されている場合も抑止の対象になる。BXJS クラスの英語モードを想定しているが、それ以外の場合でも、一般的な L^AT_EX の

習慣として、グローバルの `english` は「その文書の基底言語が英語である」ことを示す。

```
4959 \newif\ifbxjisp@english
4960 \DeclareOption{english}{\bxjisp@englishttrue}

      オプション定義はおしまい。
4961 \ProcessOptions*
```

1.3 パッケージ読込の阻止

`\pandocSkipLoadFile` `\pandocSkipLoadFile{〈ファイル名〉}` : 特定のファイルを (`\@filewithoptions` の処理に関して) 読込済であるとマークする。

```
4962 \@onlypreamble\pandocSkipLoadFile
4963 \newcommand*\pandocSkipLoadFile[1]{%
4964   \expandafter\bxjisp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4965 \def\bxjisp@skip@load@file@a#1#2{%
4966   \ifx#1\relax
4967     \def#1{2001/01/01}%
4968     \PackageInfo\bxjisp@pkgname
4969       {File '#2' marked as loaded\@gobble}%
4970   \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。

```
4971 \@onlypreamble\pandocSkipLoadPackage
4972 \newcommand*\pandocSkipLoadPackage[1]{%
4973   \pandocSkipLoadFile{#1.sty}}
```

1.4 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近 (2015 年版以降) の \LaTeX ではこれで警告が出る。これを抑止する。

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

```
4974 \ifx\@IncludeInRelease\@undefined\else
4975   \pandocSkipLoadPackage{fixltx2e}
4976 \fi
```

1.5 cmap パッケージ

エンジンが (u)p \LaTeX のときに `cmap` パッケージが読み込まれるのを阻止する。(実際は警告が出るだけで無害であるが。)

```
4977 \if j\bxjisp@engine
4978   \pandocSkipLoadPackage{cmap}
4979 \fi
```

I.6 microtype パッケージ

警告が多すぎなので消す。

```
4980 \if j\bxjsp@engine \else
4981   \PassOptionsToPackage{verbose=silent}{microtype}
4982 \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。

```
4983 \if j\bxjsp@engine
4984   \pandocSkipLoadPackage{microtype}
4985   \newcommand*\UseMicrotypeSet[2][]{\fi}
4986 \fi
```

I.7 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。非数式である場合は代わりに … を実行する。

※以前は「Pandoc が必ず `\ldots{}` の形で書き出す」ことを利用して後続に `{}` があるかで「元が … であるか」を判断していた。ところが、Pandoc 2.7 版で `{}` を必ずしも付けなくなったため、1.9f 版で非数式の `\ldots` を全て … に戻す動作に変更した。

```
4987 \DeclareRobustCommand{\pandocLdots}{%
4988   \relax\ifmmode \expandafter\bxjsp@org@ldots
4989   \else \expandafter\bxjsp@ja@ellipsis
4990   \fi}
4991 \def\bxjsp@ja@ellipsis{…}
4992 \let\bxjsp@org@ldots\ldots
```

`\ldots` の実装を置き換える。

※ Benglish オプション指定時は置き換えない。

```
4993 \AtBeginDocument{%
4994   \let\bxjsp@org@ldots\ldots
4995   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}
```

`\ldots` の直後の文字が非英字の場合、Pandoc は「`\ldots。`」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが $\text{X}\text{q}\text{T}\text{E}\text{X}$ ・ $\text{L}\text{u}\text{a}\text{T}\text{E}\text{X}$ は英字と見なす（または将来その可能性がある）」文字で、特に日本語文書に現れるものについて、非英字扱いにしておく。

※ Pandoc は「Unicode 7.0 で GC が Letter」な文字を英字と判定している。

```

4996 \chardef\bxjsp@cc@other=12
4997 \@onlypreamble\bxjsp@makeother@range
4998 \def\bxjsp@makeother@range#1#2{%
4999   \@tempcnta"#1\relax \@tempcntb"#2\relax
5000   \loop\ifnum\@tempcnta<\@tempcntb
5001     \catcode\@tempcnta\bxjsp@cc@other
5002     \advance\@tempcnta\@ne
5003   \repeat}
5004 \ifnum0\if x\bxjsp@engine1\fi\if l\bxjsp@engine1\fi>0
5005   \catcode"1F23B=\bxjsp@cc@other
5006   \bxjsp@makeother@range{9FCD}{A000}
5007   \bxjsp@makeother@range{1B002}{1B170}
5008   \bxjsp@makeother@range{2B820}{2EBF0}
5009 \fi

```

1.8 PandoLa モジュール

インストール済であれば読み込む。

```

5010 \IfFileExists{bxpandola.sty}{%
5011   \RequirePackage{bxpandola}\relax
5012   \PackageInfo\bxjsp@pkgname
5013   {PandoLa module is loaded\@gobble}
5014 }{}

```

1.9 完了

おしまい。

```
5015 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
5016 %</anc>
```