

# BXjscls パッケージ (BXJS 文書クラス集) ソースコード説明書

八登崇之 (Takayuki YATO; aka. “ZR”)

v1.8 [2018/03/03]

---

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	9
3	和文フォントの変更	38
4	フォントサイズ	39
5	レイアウト	44
5.1	ページレイアウト	45
6	改ページ (日本語 TeX 開発コミュニティ版のみ)	59
7	ページスタイル	60
8	文書のマークアップ	64
8.1	表題	64
8.2	章・節	69
8.3	リスト環境	81
8.4	パラメータの設定	89
8.5	フロート	90
8.6	キャプション	91
9	フォントコマンド	93

10	相互参照	95
10.1	目次の類	95
10.2	参考文献	100
10.3	索引	102
10.4	脚注	103
11	段落の頭へのグルー挿入禁止	106
12	いろいろなロゴ	110
13	amsmath との衝突の回避	110
14	初期設定	111
付録 A	和文ドライバの仕様 	115
付録 B	和文ドライバ: minimal 	116
B.1	補助マクロ	116
B.2	(u)pTeX 用の設定	118
B.3	pdfTeX 用の処理	122
B.4	X <sub>Y</sub> TeX 用の処理	123
B.5	後処理 (エンジン共通)	124
付録 C	和文ドライバ: standard 	126
C.1	共通処理 (1)	127
C.2	pTeX 用設定	135
C.3	pdfTeX 用設定: CJK + bxcjkjatype	138
C.4	X <sub>Y</sub> TeX 用設定: xeCJK + zxjatype	140
C.5	LuaTeX 用設定: LuaTeX-ja	142
C.6	共通処理 (2)	145
付録 D	和文ドライバ: modern 	146
D.1	フォント設定	146
D.2	fixltx2e 読込	147
D.3	和文カテゴリコード	147
D.4	完了	147
付録 E	和文ドライバ: pandoc 	147
E.1	dupload システム	147
E.2	lang 変数	148
E.3	geometry 変数	149
E.4	CJKmainfont 変数	149
E.5	paragraph のマーク	149

E.6	全角空白文字 . . . . .	150
E.7	完了 . . . . .	150
付録 F	補助パッケージ一覧 	150
付録 G	補助パッケージ : bxjscompat 	151
G.1	準備 . . . . .	151
G.2	X <sub>q</sub> TeX 部分 . . . . .	151
G.3	LuaTeX 部分 . . . . .	152
G.4	完了 . . . . .	153
付録 H	補助パッケージ : bxjscjcat 	153
H.1	準備 . . . . .	153
H.2	和文カテゴリコードの設定 . . . . .	154
H.3	ギリシャ・キリル文字の扱い . . . . .	155
H.4	初期設定 . . . . .	161
H.5	完了 . . . . .	161
付録 I	補助パッケージ : bxjspandoc 	161
I.1	準備 . . . . .	162
I.2	パッケージ読込の阻止 . . . . .	162
I.3	fixltx2e パッケージ . . . . .	162
I.4	cmap パッケージ . . . . .	163
I.5	microtype パッケージ . . . . .	163
I.6	Unicode 文字変換対策 . . . . .	163
I.7	PandoLa モジュール . . . . .	164
I.8	完了 . . . . .	164

## 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「p<sub>A</sub>TeX 2<sub>ε</sub> 新ドキュメントクラス」に改変を加えたものである。本ドキュメントクラスに関する説明は全てこの形式の枠の中に記す。枠の外にあるものは原版著者による原版に対する解説である。

---

---

これは L<sup>A</sup>T<sub>E</sub>X3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいてもともと奥村晴彦により作成されたものです。現在は日本語 T<sub>E</sub>X 開発コミュニティにより GitHub で管理されています。

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

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

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

[2009-02-22] 田中琢爾氏による upL<sup>A</sup>T<sub>E</sub>X 対応パッチを取り込みました。

ここでは次のドキュメントクラス (スタイルファイル) を作ります。

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

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

以下では実際のコードに即して説明します。

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

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

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

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

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

#### ■ BXJS クラス特有の設定

---

---

長さ値の指定で式を利用可能にするため `calc` を読み込む。

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

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

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

クラスの本体ではこの他に geometry パッケージが読み込まれる。

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 (TeX の種類) の種別：`j = pTeX` 系、`x = XeTeX`、`p = pdfTeX` (含 DVI モード)、`l = LuaTeX`、`J = NTT jTeX`、`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) `upTeX` 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 `pTeX-ng` であるか。

```
39 \newif\ifjsWithpTeXng
```

```
40 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}
```

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

```
41 \newif\ifjsWitheTeX
42 \bxjs@test@engine\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}
```

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

```
53 \ifjsWitheTeX \let\bxjs@protected\protected
54 \else \let\bxjs@protected\empty
55 \fi
```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

```
56 \ifjsWitheTeX
57   \def\bxjs@robust@def{\protected\def}
58 \else
59   \def\bxjs@robust@def{\DeclareRobustCommand*}
60 \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。

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

`\bxjs@cond` `\bxjs@cond\ifXXX……\fi{〈真〉}{〈偽〉}`

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

```
69 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
70   #1\expandafter\@firstoftwo
71   \else\expandafter\@secondoftwo
72   \fi}
```

`\bxjs@cset` `\bxjs@cset{〈名前 1〉}\制御綴 :`

```

73 \def\bxjs@cslet#1{%
74   \expandafter\let\csname#1\endcsname}

\bxjs@csletcs \bxjs@csletcs{<名前 1>}{<名前 2>} :
75 \def\bxjs@csletcs#1#2{%
76   \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。
77 \def\bxjs@catopt#1#2{%
78   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

\bxjs@ifplus \@ifstar の + 版。
79 \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。
80 \let\c@bxjs@tempcnta\@tempcnta
81 \def\bxjs@gset@tempcnta{\setcounter{bxjs@tempcnta}}

\jsSetQHLengh \jsSetQHLengh\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）。
82 \def\jsSetQHLengh#1#2{%
83   \begingroup
84     \bxjs@parse@qh{#2}%
85     \ifx\bxjs@tmpb\relax
86       \setlength\@tempdima{#2}%
87       \xdef\bxjs@g@tmpa{\the\@tempdima}%
88     \else \global\let\bxjs@g@tmpa\bxjs@tmpb
89     \fi
90   \endgroup
91   #1=\bxjs@g@tmpa\relax}

\bxjs@parse@qh #1 が Q/H/trueQ/trueH/zw/zh で終わる場合、単位用の寸法値マクロ \bxjs@unit@XXX が
定義済なら、\bxjs@tmpb に #1 に等しい寸法の表現を返し、そうでないならエラーを出す。
それ以外では、\bxjs@tmpb は \relax になる。
※ (u)pLATEX の場合はこれらの和文単位はエンジンでサポートされる。しかし和文フォント
の設定が完了するまでは zw/zh の値は正しくない。
92 \if j\jsEngine \def\bxjs@parse@qh@all{zw,zh}
93 \else \def\bxjs@parse@qh@all{trueQ,trueH,Q,H,zw,zh}
94 \fi
95 \def\bxjs@parse@qh#1{%
96   \let\bxjs@tmpb\relax
97   \@for\bxjs@tmpa:=\bxjs@parse@qh@all\do{%
98     \ifx\bxjs@tmpb\relax
99       \edef\bxjs@next{{\bxjs@tmpa}{#1}}%
100      \expandafter\bxjs@parse@qh@a\csname bxjs@unit@\bxjs@tmpa\expandafter

```

```

101     \endcsname\bxjs@next
102   \fi}}
103 \def\bxjs@parse@qh@a#1#2#3{%
104   \def\bxjs@next##1#2\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}#1}%
105   \bxjs@next#3\@nil#2\@nil\@nnil}
106 \def\bxjs@parse@qh@b#1#2#3{%
107   \ifx\@nnil#2\@nnil\else
108     \ifx#3\relax
109       \ClassError\bxjs@clsname
110       {You cannot use '\bxjs@tmpa' here}{\@ehc}%
111       \def\bxjs@tmpb{Opt}%
112     \else
113       \@tempdimb#3\relax \@tempdimb#1\@tempdimb
114       \edef\bxjs@tmpb{\the\@tempdimb}%
115     \fi
116   \fi}

```

今の段階では Q/H だけが使用可能。

```

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

```

`\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。)

```

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

```

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。

```

120 \@onlypreamble\bxjs@restore@jltrcc
121 \let\bxjs@restore@jltrcc\@empty
122 \if 1\jsEngine
123 \def\bxjs@change@jltrcc#1{%
124   \xdef\bxjs@restore@jltrcc{%
125     \bxjs@restore@jltrcc
126     \catcode`#1=\the\catcode`#1\relax}%
127   \catcode`#1=11\relax}
128 \@tfor\bxjs@x:=西曆\do
129   {\expandafter\bxjs@change@jltrcc\bxjs@x}
130 \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。

```

131 \bxjs@robust@def\jsInhibitGlue{%
132   \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。

```

133 \if@compatibility
134   \ClassError\bxjs@clsname
135   {Something went chaotic!\MessageBreak
136     (How come '\string\documentstyle' is there?)\MessageBreak
137     I cannot go a single step further...}
138   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
139     then there'll still be hope....}
140   \expandafter\@firstofone
141 \else \expandafter\@gobble
142 \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。

```
143 \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。

```
144 \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。

```
145 %<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。

```
146 %<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。

---

```
147 %<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。

---

```
148 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

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

---

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

```
149 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
150 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
151 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
152 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
153 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
154 \DeclareOption{b4paper}{\bxjs@setpaper{{257truemm}{364truemm}}}
155 \DeclareOption{b5paper}{\bxjs@setpaper{{182truemm}{257truemm}}}
156 \DeclareOption{b6paper}{\bxjs@setpaper{{128truemm}{182truemm}}}
157 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
158 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
159 \DeclareOption{b4j}{\bxjs@setpaper{{257truemm}{364truemm}}}
160 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
161 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
162 \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。

```
163 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
164 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
165 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
166 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
167 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}
```

---

■横置き 用紙の縦と横の長さを入れ換えます。

```
168 \newif\if@landscape
169 \@landscapefalse
170 \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ため，簡単なた

め article のみで使えるオプションとしました。

```
171 \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。

```
172 %<!slide>\@slidefalse
```

```
173 %<slide>\@slidetrue
```

---

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

---

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

```
174 \newcommand{\@ptsize}{0}
```

```
175 \def\bxjs@param@basefontsize{10pt}
```

```
176 \def\jsUnusualPtSize{-20}
```

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

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

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

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

```
178 \jsSetQHLlength\@tempdima{#1}%
```

```
179 \edef\bxjs@param@basefontsize{\the\@tempdima}%
```

```
180 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
```

```
181 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
```

```
182 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
```

```
183 \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi}
```

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

```
184 \newif\ifjsc@mag
185 \newif\ifjsc@mag@xreal
186 %\let\jsc@magscale\@undefined
187 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
188 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
189 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
190 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
191 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
192 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
193 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
194 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
195 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
196 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
197 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
198 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
199 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
200 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
201 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
202 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
203 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
204 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
205 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}
```

JS クラス互換の `magstyle` 設定オプション。

```
206 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
207 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
208 \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  $\text{pLATEX} 2_{\epsilon}$  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  $\text{pLATEX} 2_{\epsilon}$  本体で宣言されています。

---

取りあえず、 $\text{pT}_{\text{E}}\text{X}$  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。

---

```
209 \if j\jsEngine
210 \hour\time \divide\hour by 60\relax
211 \@tempcnta\hour \multiply\@tempcnta 60\relax
212 \minute\time \advance\minute-\@tempcnta
213 \DeclareOption{tombow}{%
214   \tombowtrue \tombowdatetrue
215   \setlength{\@tombowwidth}{.1\p}%
216   \@bannertoken%
217   \jobname\space(\number\year-\two@digits\month-\two@digits\day
```

```

218 \space\two@digits\hour:\two@digits\minute}}%
219 \maketombowbox}
220 \DeclareOption{tombo}{%
221 \tombowtrue \tombowdatefalse
222 \setlength{\@tombowwidth}{.1\p@}%
223 \maketombowbox}
224 \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。

```

225 \if j\jsEngine
226 \DeclareOption{mentuke}{%
227 \tombowtrue \tombowdatefalse
228 \setlength{\@tombowwidth}{\z@}%
229 \maketombowbox}
230 \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。

```

231 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
232 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
233 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 `twocolumn` で二段組になります。

```

234 \DeclareOption{onecolumn}{\@twocolumnfalse}
235 \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。

```

236 \DeclareOption{titlepage}{\@titlepagetrue}
237 \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 I<sup>A</sup>T<sub>E</sub>X 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。

```

238 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
239 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
240 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}

```

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

eqnarray L<sup>A</sup>T<sub>E</sub>X 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。

```

241 \def\eqnarray{%
242   \stepcounter{equation}%
243   \def\@currentlabel{\p@equation\theequation}%
244   \global\@eqnswtrue
245   \m@th
246   \global\@eqcnt\z@
247   \tabskip\@centering
248   \let\@= \@eqncr
249   $$\everycr{}\halign to\displaywidth\bgroup
250     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
251     &\global\@eqcnt\@ne \hfil$\displaystyle{##}\hfil
252     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
253     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
254     \tabskip\z@skip
255   \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。

```

256 \DeclareOption{leqno}{\input{leqno.clo}}
257 \DeclareOption{fleqn}{\input{fleqn.clo}}
258 % fleqn 用の eqnarray 環境の再定義
259 \def\eqnarray{%
260   \stepcounter{equation}%
261   \def\@currentlabel{\p@equation\theequation}%
262   \global\@eqnswtrue\m@th
263   \global\@eqcnt\z@
264   \tabskip\mathindent
265   \let\@=\@eqncr
266   \setlength\abovedisplayskip{\topsep}%
267   \ifvmode
268     \addtolength\abovedisplayskip{\partopsep}%
269   \fi
270   \addtolength\abovedisplayskip{\parskip}%
271   \setlength\belowdisplayskip{\abovedisplayskip}%
272   \setlength\belowdisplayshortskip{\abovedisplayskip}%
273   \setlength\abovedisplayshortskip{\abovedisplayskip}%
274   $$\everycr{}\halign to\linewidth% $$
275   \bgroup
276     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
277     &\global\@eqcnt\@ne \hfil$\displaystyle{##}\hfil
278     &\global\@eqcnt\tw@
279     $\displaystyle{##}$\hfil \tabskip\@centering
280     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
281   \tabskip\z@skip\cr
282   }}

```

■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。

```
283 % \DeclareOption{openbib}{%
284 %   \AtEndOfPackage{%
285 %     \renewcommand\@openbib@code{%
286 %       \advance\leftmargin\bibindent
287 %       \itemindent -\bibindent
288 %       \listparindent \itemindent
289 %       \parsep \z@}%
290 %     \renewcommand\newblock{\par}}}
```

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

```
291 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。

292 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
293 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
294 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
295 \define@key{bxjs}{enablejfam}[true]{%
296   \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。

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

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

---

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

[2016-07-13] \ifdraft を定義するのをやめました。

---

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

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

```
298 \let\ifjsDraft\iffalse
299 \@onlypreamble\bxjs@draft
300 \def\bxjs@draft#1{%
301   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
302 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
303 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
304 \AtBeginDocument{%
305   \expandafter\ifx\csname ifdraft\endcsname\relax
306     \expandafter\let\csname ifdraft\endcsname\expandafter\endcsname
307     \csname ifjsDraft\endcsname
308   \fi}
```

---

■和文フォントメトリックの選択 ここでは新しい `jis` フォントメトリックを標準で使いますが、古い `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` メトリックが使えます。

---

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。

```
309 \newif\ifbxjs@papersize
310 \bxjs@papersizetrue
311 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
312 \DeclareOption{papersize}{\bxjs@papersizetrue}
```

---

■英語化 オプション `english` を新設しました。

```
313 \newif\if@english
314 \@englishfalse
315 \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。どちらでもお好きな方を使ってください。

---

BXJS では当初から `bxjsreport` クラスが用意されている。

---

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

---

BXJS クラスでは、`nojslogo` を既定とする。

---

```
316 \newif\if@jslogo \@jslogofalse
317 \DeclareOption{jslogo}{\@jslogotrue}
318 \DeclareOption{nojslogo}{\@jslogofalse}
```

#### ■BXJS 特有のオプションの一覧

---

- エンジンオプション：`xelatex` 等。
- ドライバオプション：`dvipdfmx` 等。
- 複合設定オプション：`pandoc` 等。
- `nopapersize`：`papersize`（既定で有効）の否定。
- `zw` / `noz`：`\jsZw` と等価な命令として `\zw` を定義する / しない。
- `js` / `nojs`：JS クラスを讀込済として扱う / 扱わない。
- `precisetext` / `noprecisetext`：`XYTeX` の “`generateactualtext`” を有効 / 無効にする。
- `simplejasetup` / `nosimplejasetup`：`XYTeX` の “`linebreaklocale`” を有効 / 無効にする。
- `bigcode` / `nobigcode`：`upTeX` で CMap として UTF8-UCS2 の代わりに UTF8-UTF16 を使う / 使わない。
- `oldfontcommands` / `nooldfontcommands`：古い “二文字フォント命令” に対する警告を抑止する / しない。
- `base=<dimen>`：基底フォントサイズを直接指定する。（`xxpt` オプションの代用なので、既定値は 10 pt である。）
- `jbase=<dimen>`：基底フォントサイズを “和文規準で” 直接指定する。
- `scale=<real>`：和文フォントのスケールを表すマクロ `\jsScale` の値を設定する。もちろんこの値を何らかの方法で和文処理モジュールに渡さないと意味を成さない。既定値は 0.924715 (= 13 Q/10 pt)。
- `noscale`：`scale=1` と等価。
- `mag=<int>`：`\mag` 値の直接設定。既定は `base` から算出する。
- `paper={<dimen:width>}{<dimen:height>}`：用紙サイズ設定。用紙サイズオプションの代用で、既定値は `a4paper` 相当。
- `ja=<name>`：使用する和文ドライバの指定。

- `jafont=<name>` : 和文フォントプリセットの指定。
- `japaram=<name>` : 和文フォントパラメタの指定。
- `magstyle=<name>` : “版面拡大” の実現方法の選択。
- `dvi=<name>` : DVI モードの時のみに参照されるドライバ指定。
- `geometry={class | user}` : `geometry` パッケージの読込を自動的に行うかユーザに任せるか。
- `fancyhdr=<bool>` : `fancyhdr` パッケージ用の調整を行うか。
- `layout=<name>` : レイアウト変種の指定。
- `textwidth-limit=<number>` : `bxjsbook` における、`\textwidth` の上限の全角単位での値。
- `paragraph-mark=<char>` : パラグラフのマーク。
- `whole-zw-lines=<bool>` : 行長を全角単位に丸めるか。
- `hyperref-enc=<bool>` : `hyperref` の文字コード指定補正を行うか。

`\bxjs@invscale` `\bxjs@invscale` は  $\TeX$  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。

```

319 \mathchardef\bxjs@isc@ll=128
320 \mathchardef\bxjs@isc@sl=259
321 \def\bxjs@isc@sl@h{65539 }
322 \def\bxjs@invscale#1#2{%
323   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
324     \ifdim\@tempdima<\bxjs@isc@ll\p@
325       \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
326       \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
327     \else
328       \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
329       \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
330     \fi
331     \@tempcntb\p@ \divide\@tempcntb\@tempdimb
332     \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
333     \@tempdimb\@tempcnta\@ne
334     \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
335     \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
336     \@whiledim\@tempdimb<\@tempdimc\do{%
337       \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
338       \advance\@tempcntb\@ne \divide\@tempcntb\tw@
339     \ifdim #2\@tempcntb>\@tempdima
340       \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
341     \else \@tempdimb=\@tempcntb\@ne \fi}%
342     \xdef\bxjs@gtmpa{\the\@tempdimb}%
343   \endgroup #1=\bxjs@gtmpa\relax}

```

---

---

## ■複合設定オプション

---

---

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。

`\bxjs@composite@proc` 複合設定オプションのための遅延処理マクロ。

```
344 \let\bxjs@composite@proc\relax
```

`pandoc` オプションは、Pandoc で L<sup>A</sup>T<sub>E</sub>X 用の既定テンプレートを用いて他形式から L<sup>A</sup>T<sub>E</sub>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。

```
345 \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。

```
346 \def\bxjs@composite@proc{%
347   \bxjs@oldfontcommandstrue
348   \setkeys{bxjs}{ja=pandoc}%
349   \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…。

```
350 \def\bxjs@driver@opt{dvipdfmx}%
351 \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。

```
352 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
353 %\let\bxjs@engine@opt\@undefined
```

エンジン明示指定のオプションの処理。

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

```
354 \DeclareOption{autodetect-engine}{%
355   \let\bxjs@engine@given=*}
356 \DeclareOption{latex}{%
357   \def\bxjs@engine@opt{latex}%
358   \let\bxjs@engine@given=n}
359 \DeclareOption{platex}{%
```

```

360 \def\bxjs@engine@opt{platex}%
361 \let\bxjs@engine@given=j}
362 \DeclareOption{uplatex}{%
363 \def\bxjs@engine@opt{uplatex}%
364 \let\bxjs@engine@given=u}
365 \DeclareOption{xelatex}{%
366 \def\bxjs@engine@opt{xelatex}%
367 \let\bxjs@engine@given=x}
368 \DeclareOption{pdflatex}{%
369 \def\bxjs@engine@opt{pdflatex}%
370 \let\bxjs@engine@given=p}
371 \DeclareOption{lualatex}{%
372 \def\bxjs@engine@opt{lualatex}%
373 \let\bxjs@engine@given=l}
374 \DeclareOption{platex-ng}{%
375 \def\bxjs@engine@opt{platex-ng}%
376 \let\bxjs@engine@given=g}
377 \DeclareOption{platex-ng*}{%
378 \def\bxjs@engine@opt{platex-ng*}%
379 \let\bxjs@platexng@nodrv=t%
380 \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類別。

```

381 %\let\bxjs@driver@given\@undefined
382 \let\bxjs@driver@@dvimode=0
383 \let\bxjs@driver@@dvi pdfmx=1
384 \let\bxjs@driver@@pdfmode=2
385 \let\bxjs@driver@@xetex=3
386 \let\bxjs@driver@@dvips=4
387 \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。

```

388 %\let\bxjs@driver@opt\@undefined
389 \DeclareOption{dvips}{%
390 \def\bxjs@driver@opt{dvips}%
391 \let\bxjs@driver@given\bxjs@driver@@dvips}
392 \DeclareOption{dviout}{%
393 \def\bxjs@driver@opt{dviout}%
394 \let\bxjs@driver@given\bxjs@driver@@dvimode}
395 \DeclareOption{xdvi}{%
396 \def\bxjs@driver@opt{xdvi}%
397 \let\bxjs@driver@given\bxjs@driver@@dvimode}
398 \DeclareOption{dvi pdfmx}{%
399 \def\bxjs@driver@opt{dvi pdfmx}%
400 \let\bxjs@driver@given\bxjs@driver@@dvi pdfmx}
401 \DeclareOption{nodvidriver}{%
402 \def\bxjs@driver@opt{nodvidriver}%
403 \let\bxjs@driver@given\bxjs@driver@@none}

```

```

404 \DeclareOption{pdftex}{%
405   \def\bxjs@driver@opt{pdftex}%
406   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
407 \DeclareOption{luatex}{%
408   \def\bxjs@driver@opt{luatex}%
409   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
410 \DeclareOption{xetex}{%
411   \def\bxjs@driver@opt{xetex}%
412   \let\bxjs@driver@given\bxjs@driver@@xetex}

```

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

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

```

413 \DeclareOption{dvipdfmx-if-dvi}{%
414   \setkeys{bxjs}{dvi=dvipdfmx}}

```

---

### ■その他の BXJS 独自オプション 密

---

`\ifbxjs@usezw` `\jsZw` の同義語として `\zw` を使えるようにするか。既定は真。

```
415 \newif\ifbxjs@usezw \bxjs@usezwtrue
```

`zw`、`noz` オプションの定義。

```

416 \DeclareOption{noz}{%
417   \bxjs@usezwfalse}
418 \DeclareOption{zw}{%
419   \bxjs@usezwtrue}

```

`\ifbxjs@disguise@js` JS クラスの派生クラスのふりをするか。既定は真。

```
420 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
```

`nojs`、`js` オプションの定義。

```

421 \DeclareOption{nojs}{%
422   \bxjs@disguise@jsfalse}
423 \DeclareOption{js}{%
424   \bxjs@disguise@jstrue}

```

`\ifbxjs@precisetext` X<sub>Y</sub>TEX の “generateactualtext” を有効にするか。既定は偽。

```
425 \newif\ifbxjs@precisetext
```

`noprecisetext` / `precisetext` オプションの定義。

```

426 \DeclareOption{noprecisetext}{%
427   \bxjs@precisetextfalse}
428 \DeclareOption{precisetext}{%
429   \bxjs@precisetexttrue}

```

`\ifbxjs@simplejasetup` X<sub>Y</sub>TEX の “linebreaklocale” を有効にするか。既定は真（であるが多くの場合は後に無効化される）。

```
430 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
```

nosimplejasetup / simplejasetup オプションの定義。

```
431 \DeclareOption{nosimplejasetup}{%
432   \bxjs@simplejasetupfalse}
433 \DeclareOption{simplejasetup}{%
434   \bxjs@simplejasetuptrue}
```

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

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

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

```
436 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
437 \def\bxjs@tmpb#1 #2#3\@nil{%
438   \ifx1#2\bxjs@bigcodetrue \fi}
439 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil
```

nobigcode / bigcode オプションの定義。

```
440 \DeclareOption{nobigcode}{%
441   \bxjs@bigcodefalse}
442 \DeclareOption{bigcode}{%
443   \bxjs@bigcodetrue}
```

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

```
444 \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。

```
445 \DeclareOption{nooldfontcommands}{%
446   \bxjs@oldfontcommandsfalse}
447 \DeclareOption{oldfontcommands}{%
448   \bxjs@oldfontcommandstrue}
```

---

## ■ keyval 型のオプション

---

```
449 \def\bxjs@setkey{%
450   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
451 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
452 \DeclareOption*{\bxjs@setkey}
```

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

```
453 \def\bxjs@safe@setkeys#1#2{%
454   \let\bxjs@KV@errx\KV@errx
```

```

455 \let\KV@errx\bxjs@safe@setkeys@a
456 \setkeys{#1}{#2}%
457 \let\KV@errx\bxjs@KV@errx}
458 \def\bxjs@safe@setkeys@a#1{

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

\jsScale [実数値マクロ] 和文スケール値。
470 \def\jsScale{0.924715}

\bxjs@base@opt base オプションの値 (そのもの)。
471 %\let\bxjs@base@opt\@undefined

    base オプションの処理。
472 \define@key{bxjs}{base}{%
473 \edef\bxjs@base@opt{#1}%
474 \bxjs@setbasefontsize{#1}}
475 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

\bxjs@jbase@opt jbase オプションの値。
476 %\let\bxjs@jbase@opt\@undefined

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

\bxjs@scale@opt scale オプションの値。
479 %\let\bxjs@jbase@opt\@undefined

    scale オプションの処理。
480 \define@key{bxjs}{scale}{%
481 \edef\bxjs@scale@opt{#1}%
482 \let\jsScale\bxjs@scale@opt}
483 \define@key{bxjs}{jsfontscale}{\setkeys{bxjs}{scale=#1}}

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

```

`\bxjs@param@mag` mag オプションの値。

```
485 \let\bxjs@param@mag\relax
```

mag オプションの処理。

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

paper オプションの処理。

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

`\bxjs@jadriver` 和文ドライバの名前。

```
488 \let\bxjs@jadriver\relax
```

```
489 %\let\bxjs@jadriver@given\@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)。

```
490 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
```

```
491 \define@key{bxjs}{ja}[\relax]{%
```

```
492 \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}
```

`\jsJaFont` 和文フォント設定の名前。

```
493 \let\jsJaFont\@empty
```

jafont オプションの処理。

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

`\jsJaParam` 和文ドライバパラメタの文字列。

```
495 \let\jsJaParam\@empty
```

japaram オプションの処理。

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

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

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

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

```
499 \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。

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

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

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

`\bxjs@magstyle@default` は既定の値を表す。

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

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

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

```
506 \fi\fi
```

```

507 \ifjswithptexng
508 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
509 \fi
510 \let\bxjs@magstyle\bxjs@magstyle@default

```

magstyle オプションの処理。

```

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

```

\bxjs@geometry geometry オプションの値。

```

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

```

geometry オプションの処理。

```

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

```

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

```

530 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

```

fancyhdr オプションの処理。

```

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

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

535 \newif\ifbxjs@dvi@opt

```

DVI モードのドライバとドライバ種別との対応。

```

536 \let\bxjs@dvidriver@@dvpdvmx=\bxjs@driver@@dvpdvmx
537 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
538 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
539 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
540 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

```

dvi オプションの処理。

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

\bxjs@driver@given を未定義にしていることに注意。

```
548   \def\bxjs@driver@opt{#1}%
549   \let\bxjs@driver@given\@undefined
550   \bxjs@dvi@opttrue
551 \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。

```
552 \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。

```
553 \newif\ifbxjs@force@chapterabstract
554 %<book>\bxjs@force@chapterabstracttrue
```

layout オプションの処理。

```
555 \@namedef{bxjs@kv@layout@v1}{%
556 %<book>\bxjs@layout@buggyhmargintrue
557 %<report>\bxjs@force@chapterabstracttrue
558 }
559 \@namedef{bxjs@kv@layout@v2}{%
560 %<book>\bxjs@layout@buggyhmarginfalse
561 %<report>\bxjs@force@chapterabstractfalse
562 }
563 \define@key{bxjs}{layout}{%
564   \bxjs@set@keyval{layout}{#1}{}}
```

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

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

\bxjs@textwidth@opt textwidth の指定値。

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

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

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

```
573 \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。

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

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

```
577 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
578 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
579 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
580 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}
```

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

```
581 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
582 \let\bxjs@kv@jaspaccmd@true\bxjs@jaspace@cmdtrue
583 \let\bxjs@kv@jaspaccmd@false\bxjs@jaspace@cmdfalse
584 \define@key{bxjs}{jaspac-cmd}[true]{\bxjs@set@keyval{jaspaccmd}{#1}{}}
585 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{jaspaccmd}{#1}{}}
```

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

```
586 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
587 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
588 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
589 \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。

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

`\bxjs@everyparhook` everyparhook の指定値。

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

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

```
606 \chardef\bxjs@label@section@none=0
607 \chardef\bxjs@label@section@compat=1
```

```

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

```

---

## ■ オプションの実行

---

LaTeX 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。

```

614 \let\bxjs@ltx@removeelement\@removeelement
615 \def\@removeelement#1#2#3{%
616   \def\reserved@a{#2}%
617   \ifx\reserved@a\@empty \let#3\@empty
618   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
619   \fi}

```

---

デフォルトのオプションを実行し、dvi ファイルの先頭に dvips の papersize special を書き込みます。この special は dvips や最近の dviout が対応しています。multicols や url を \RequirePackage するのはやめました。

```

620 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
621 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
622 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
623 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
624 \ProcessOptions\relax
625 \bxjs@composite@proc

```

後処理

```

626 \if@slide
627   \def\maybeblue{\@ifundefined{ver@color.sty}{}{\color{blue}}}
628 \fi
629 \if@landscape
630   \setlength\@tempdima {\paperheight}
631   \setlength\paperheight{\paperwidth}
632   \setlength\paperwidth {\@tempdima}
633 \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

込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。

```
634 \@onlypreamble\bxjs@purge@brace@elts
635 \def\bxjs@purge@brace@elts{%
636   \def\bxjs@tmpa{\@gobble}%
637   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
638   \let\@classoptionslist\bxjs@tmpa}
639 \@onlypreamble\bxjs@purge@be@a
640 \def\bxjs@purge@be@a#1,{%
641   \ifx\@nil#1\relax\else
642     \bxjs@purge@be@b#1{\@nil}
643     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
644     \expandafter\bxjs@purge@be@a
645     \fi}
646 \@onlypreamble\bxjs@purge@be@b
647 \def\bxjs@purge@be@b#1#\bxjs@purge@be@c}
648 \@onlypreamble\bxjs@purge@be@c
649 \def\bxjs@purge@be@c#1\@nil{%
650   \ifx\@nil#1\@nil \@tempwatrue \else \@tempwafalse \fi}
651 \bxjs@purge@brace@elts
```

`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。

```
652 \@expandtwoargs\@removeelement
653   {papersize}\@classoptionslist\@classoptionslist
654 \@expandtwoargs\@removeelement
655   {10pt}\@classoptionslist\@classoptionslist
656 \@expandtwoargs\@removeelement
657   {noscale}\@classoptionslist\@classoptionslist
```

---

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

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

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

---

現在の (正規化前の) 和文ドライバの値を `\bxjs@jadriver@given` に保存する。

```
658 \ifx\bxjs@jadriver\relax\else
659   \let\bxjs@jadriver@given\bxjs@jadriver
660 \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。

```
661 \let\bxjs@tmpb\jsEngine
662 \ifx j\bxjs@tmpb\ifjsWithpTeXng
663   \let\bxjs@tmpb=g
664 \fi\fi
665 \ifx j\bxjs@tmpb\ifjsWithupTeX
666   \let\bxjs@tmpb=u
667 \fi\fi
668 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
669   \let\bxjs@tmpb=n
670 \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。)

```
671 \ifx *\bxjs@engine@given
672   \let\bxjs@engine@given\bxjs@tmpb
```

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

```
673 \ifx j\bxjs@engine@given
674   \g@addto@macro\@classoptionslist{,latex}
675 \else\ifx u\bxjs@engine@given
676   \g@addto@macro\@classoptionslist{,uplatex}
677 \fi\fi
678 \fi
679 \ifx\bxjs@engine@given\@undefined\else
680   \ifx\bxjs@engine@given\bxjs@tmpb\else
681     \ClassError\bxjs@clsname
682       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
683   \fi
684 \fi
```

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

```
685 \ifjsWithpTeXng
686   \g@addto@macro\@classoptionslist{,uplatex}
687 \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。

```
688 \@tempwatrue
689 \ifx \bxjs@driver@given\@undefined\else
690   \ifjsInPdfMode
691     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
692       \@tempwafalse
693     \fi
694   \else\ifx x\jsEngine
695     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
```

```

696     \@tempswafalse
697     \fi
698   \else
699     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
700       \@tempswafalse
701     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
702       \@tempswafalse
703     \fi\fi
704     \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
705       \@tempswafalse
706     \fi\fi
707   \fi\fi
708 \fi
709 \if@tempswa\else
710   \ClassError\bxjs@clsname
711   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
712 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

713 \ifjsInPdfMode \@tempswafalse
714 \else\ifx x\jsEngine \@tempswafalse
715 \else\ifjsWithpTeXng \@tempswafalse
716 \else \@tempswatru
717 \fi\fi\fi
718 \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)。

```

719 \ifx\bxjs@driver@opt\undefined \ifx\bxjs@jadriver@given\undefined\else
720   \ClassWarningNoLine\bxjs@clsname
721   {No driver option is given}
722 \fi\fi

```

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

```

723 \ifbxjs@dvi@opt
724   \edef\bxjs@nxt{%
725     \let\noexpand\bxjs@driver@given
726     \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
727     \noexpand\g@addto@macro\noexpand\@classoptionslist
728     {,\bxjs@driver@opt}%
729   }\bxjs@nxt
730 \fi
731 \fi

```

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

```

732 \ifjsWithpTeXng

```

```

733 \ifx\bxjs@driver@given\bxjs@driver@dvipdfmx
734 \let\bxjs@platexng@nodrv\undefined
735 \else\ifx t\bxjs@platexng@nodrv\else
736 \g@addto@macro\@classoptionslist{,dvipdfmx}
737 \fi\fi
738 \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。

```

739 \ifx\bxjs@driver@given\bxjs@driver@none
740 \bxjs@papersizefalse
741 \fi

```

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

```

742 \def\bxjs@@minimal{minimal}
743 \ifx\bxjs@jadriver\relax
744 \ifx j\jsEngine
745 \def\bxjs@jadriver{standard}
746 \else
747 \let\bxjs@jadriver\bxjs@@minimal
748 \fi
749 \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。

```

750 \ifx\bxjs@jadriver@given\undefined\else
751 \ifx\bxjs@engine@given\undefined
752 \ClassError\bxjs@clsname
753 {An engine option must be explicitly given}%
754 {When you use a Japanese-driver you must specify a correct\MessageBreak
755 engine option.\MessageBreak\@ehc}
756 \fi\fi

```

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

```

757 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
758 \ifx\bxjs@magstyle\bxjs@magstyle@mag
759 \let\bxjs@magstyle\bxjs@magstyle@default
760 \ClassError\bxjs@clsname
761 {The engine does not support 'magstyle=usemag'}%
762 {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
763 The default value 'nomag*' is used instead.\MessageBreak \@ehc}
764 \fi
765 \fi

```

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

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

```
766 \ifx\bxjs@jbase@opt\@undefined\else
767 \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。

```
768 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
769 \bxjs@invscale\@tempdima\jsScale
770 \bxjs@setbasefontsize{\@tempdima}%
771 \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。

```
772 \ifx\bxjs@scale@opt\@undefined\else
773 \ClassWarningNoLine\bxjs@clsname
774 {Redundant 'scale' option is ignored}%
775 \fi
776 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
777 \@tempdimb=\bxjs@param@basefontsize\relax
778 \edef\jsScale{\strip@pt\@tempdimb}%
779 \bxjs@invscale\@tempdima\jsScale
780 \edef\jsScale{\strip@pt\@tempdima}%
781 \fi
782 \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。

```
783 \let\Cjascale\jsScale
```

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

```
784 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
785 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
786 \catcode\@tempcnta\active
787 \advance\@tempcnta\@ne
788 \repeat
789 \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。

```
790 \ifbxjs@disguise@js
791 %<book|report>\def\bxjs@js@clsname{jsbook}
792 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
793 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
794 \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。

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

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

```
796 \ifbxjs@oldfontcommands
797   \AtEndOfClass{\allowoldfontcommands}
798 \fi
```

---

■papersize スペシャルの出力 dvips の papersize スペシャルを出力します。これで dvips や新しい dviout で出力領域が自動設定できます。トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> はトンボ出力幅を両側に 1 インチとっていますので、dvips 使用時に

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

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

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

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

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

---

BXJS では出力用紙サイズ記録は geometry パッケージが行う。

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

---

```
799 \newdimen\stockwidth \newdimen\stockheight
800 \begingroup\expandafter\expandafter\expandafter\endgroup
801 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
802 % \newdimen\stockwidth \newdimen\stockheight
803   \setlength{\stockwidth}{\paperwidth}
804   \setlength{\stockheight}{\paperheight}
805   \advance \stockwidth 2in
806   \advance \stockheight 2in
807 \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。

```
808 %<slide>\def\n@baseline{13}%
809 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
810 %<!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。

```

811 \ifx\bxjs@magstyle\bxjs@magstyle@mag
812   \jsc@magtrue
813 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
814   \jsc@mag@xrealtrue
815 \fi\fi

```

---

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

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

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

---

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

```

816 \ifx\bxjs@param@mag\relax
817   \@tempdima=\bxjs@param@basefontsize
818   \advance\@tempdima.001pt \multiply\@tempdima25
819   \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
820   \edef\bxjs@param@mag{\the\@tempcmta}
821 \else
822 % mag 値が直接指定された場合
823   \bxjs@gset@tempcmta{\bxjs@param@mag}
824   \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
825 % 有効な mag 値の範囲は 1--32768
826   \edef\bxjs@param@mag{\the\@tempcmta}
827   \advance\@tempcmta100000
828   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
829   \expandafter\bxjs@tmpa\the\@tempcmta\@nil
830   \edef\bxjs@param@basefontsize{\the\@tempdima}
831 \fi
832 \@tempcmta\bxjs@param@mag \advance\@tempcmta100000
833 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
834 \expandafter\bxjs@tmpa\the\@tempcmta\@nil
835 \edef\jsc@magscale{\strip@pt\@tempdima}
836 \let\jsBaseFontSize\bxjs@param@basefontsize
837 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
838 %\typeout{\string\jsEngine: \meaning\jsEngine}
839 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
840 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
841 %\typeout{\string\jsc@magscale: \jsc@magscale}
842 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
843 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}

```

---

[2016-07-08] `\jsc@empt` 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@empt` 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@empt` 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。

```
844 \newdimen\jsc@empt
845 \newdimen\jsc@mmm
846 \ifjsc@mag
847   \jsc@empt=1\p@
848   \jsc@mmm=1mm
849   \catcode`\?=9 % \p@? read as \p@
850 \else
851   \jsc@empt=\jsc@magscale\p@
852   \jsc@mmm=\jsc@magscale mm
853   \catcode`\?=11 \let\p@?\jsc@empt
854 \fi
855 \chardef\bxjs@qmc=\catcode`\?\relax
```

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

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

```
856 \newdimen\jsZw
857 \jsZw=10\jsc@empt \jsZw=\jsScale\jsZw
858 \ifbxjs@usezw
859   \providecommand*\zw{\jsZw}
860 \fi
```

`\zwspace` 全角幅の水平空き。

```
861 \def\zwspace{\hskip\jsZw\relax}
```

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

```
862 \ifjsc@mag@xreal
863   \RequirePackage{type1cm}
864   \let\jsc@invscale\bxjs@invscale
```

ムニャムニャムニャ……。

---

```
865 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
866 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
867 \let\jsc@get@external@font\get@external@font
868 \def\get@external@font{%
869   \jsc@preadjust@extract@font
870   \jsc@get@external@font}
871 \def\jsc@fstrunc#1{%
```

```

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

```

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

```

\jsc@smallskip
\jsc@medskip 903 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 904 %\def\jsc@medskip{\vspace\jsc@medskipamount}
905 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 906 \newskip\jsc@smallskipamount
907 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
\jsc@bigskipamount 908 %\newskip\jsc@medskipamount
909 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
910 %\newskip\jsc@bigskipamount
911 %\jsc@bigskipamount =12\jsc@mpt plus 4\jsc@mpt minus 4\jsc@mpt

```

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

します。

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

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

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

```
912 % \ifpapersize
913 %   \setlength{\@tempdima}{\paperwidth}
914 %   \setlength{\@tempdimb}{\paperheight}
915 %   \iftombow
916 %     \advance \@tempdima 2truein
917 %     \advance \@tempdimb 2truein
918 %   \fi
919 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
920 % \fi
```

### 3 和文フォントの変更

---

---

和文フォントの設定は和文ドライバの管轄。

---

---

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

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

---

---

BXJS クラスでの変更点：

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

---

---

```
921 \chardef\bxjs@periodchar=`\ .
922 \bxjs@protected\def\bxjs@SE{%
923   \ifnum\spacefactor<\@m \spacefactor\@m
924   \else \spacefactor\sfcode\bxjs@periodchar
925   \fi}
926 \ifbxjs@fix@at@cmd
927   \def\@{\bxjs@SE{}}
928 \fi
```

## 4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

---

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

```
929 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
930 \expandafter\bxjs@tmpa\expandafter{%
931   \set@fontsize{#1}{#2}{#3}%
932 % 末尾にコードを追加
933   \expandafter\def\expandafter\size@update\expandafter{%
934     \size@update
935     \jsFontSizeChanged}%
936 }
```

`\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。

```
937 \newcommand*\jsFontSizeChanged{%
938   \jsZw=\f@size\p@
939   \jsZw=\jsScale \jsZw
940   \ifdim\parindent>\z@
941     \if@english \parindent=1em
942     \else       \parindent=1\jsZw
943     \fi
944   \fi\relax
945   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
946 \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。

```
947 \ifjscssmag
948   \let\jsc@setfontsize\@setfontsize
949 \else
950   \def\jsc@setfontsize#1#2#3{%
951     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
952 % microtype 対策
953 \ifjswitheTeX\if j\jsEngine\else
954   \def\jsc@setfontsize#1#2#3{%
955     \edef\bxjs@sfs@next{%
956       \unexpanded{\@setfontsize#1}%
957       {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
958     }\bxjs@sfs@next}
959   \fi\fi
960 \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。

---

```
961 \emergencystretch 3\jsZw
```

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

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

`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対して、しばしば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。

---

```
962 \newif\ifnarrowbaselines
963 \if@english
964   \narrowbaselinestrue
965 \fi
966 \def\narrowbaselines{%
967   \narrowbaselinestrue
968   \skip0=\abovedisplayskip
969   \skip2=\abovedisplayshortskip
970   \skip4=\belowdisplayskip
971   \skip6=\belowdisplayshortskip
972 % 一時的に警告を無効化する
973   \let\bxjs@ltx@nomath\@nomath
974   \let\@nomath\@gobble
975   \@currsize\selectfont
976   \let\@nomath\bxjs@ltx@nomath
977   \abovedisplayskip=\skip0
978   \abovedisplayshortskip=\skip2
979   \belowdisplayskip=\skip4
980   \belowdisplayshortskip=\skip6\relax}
981 \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。

```
982 \def\bxjs@if@narrowbaselines{%
983   \ifnarrowbaselines\expandafter\@firstoftwo
984   \else \expandafter\@secondoftwo
985   \fi
986 }
```

---

`\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 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。

---

```
987 \renewcommand{\normalsize}{%
988   \bxjs@if@narrowbaselines{%
989     \jsc@setfontsize\normalsize\@xpt\@xiipt
990   }{%else
991     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
992   }%
```

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

[2003-02-16] ちょっと変えました。

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

```
993 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
994 \abovedisplayskipshortskip \z@ \@plus3\p@?
995 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
996 \belowdisplayskipshortskip \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。

```
997 \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。

```
998 \normalsize
```

`\Cht` 基準となる長さの設定をします。pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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 pT<sub>E</sub>X 系以外では未定義なのでここで定義する。

```
999 \ifx\Cht\@undefined \newdimen\Cht \fi
1000 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1001 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1002 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1003 \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。

---

```
1004 \setlength\Cht{0.88\jsZw}
1005 \setlength\Cdp{0.12\jsZw}
```

```

1006 \setlength\Cwd{1\jsZw}
1007 \setlength\Cvs{\baselineskip}
1008 \setlength\Chs{1\jsZw}

```

`\small` `\small` も `\normalsize` と同様に設定します。行送りは、`\normalsize` が 16 ポイントなら、割合からすれば  $16 \times 0.9 = 14.4$  ポイントになりますが、`\small` の使われ方を考えて、ここでは和文 13 ポイント、欧文 11 ポイントとします。また、`\topsep` と `\parsep` は、元はそれぞれ  $4 \pm 2$ 、 $2 \pm 1$  ポイントでしたが、ここではゼロ (`\z@`) にしました。

---

microtype 対策のため if 文を避ける。後の `\footnotesize` も同様。

---

```

1009 \newcommand{\small}{%
1010   \bxjs@if@narrowbaselines{%
1011     <!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1012     <kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1013   }{%else
1014     <!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1015     <kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1016   }%
1017   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1018   \abovedisplayshortskip \z@ \@plus3\p@?
1019   \belowdisplayskip \abovedisplayskip
1020   \belowdisplayshortskip \belowdisplayskip
1021   \def\@listi{\leftmargin\leftmargini
1022             \topsep \z@
1023             \parsep \z@
1024             \itemsep \parsep}}

```

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

```

1025 \newcommand{\footnotesize}{%
1026   \bxjs@if@narrowbaselines{%
1027     <!kiyou>   \jsc@setfontsize\footnotesize\@viiipt{9.5}%
1028     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1029   }{%else
1030     <!kiyou>   \jsc@setfontsize\footnotesize\@viiipt{11}%
1031     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1032   }%
1033   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1034   \abovedisplayshortskip \z@ \@plus2\p@?
1035   \belowdisplayskip \abovedisplayskip
1036   \belowdisplayshortskip \belowdisplayskip
1037   \def\@listi{\leftmargin\leftmargini
1038             \topsep \z@
1039             \parsep \z@
1040             \itemsep \parsep}}

```

`\scriptsize` それ以外のサイズは、本文に使うことがないので、単にフォントサイズと行送りだけ変更し

```

\tiny
\large
\Large
\LARGE
\huge

```

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

[2004-11-03] `\HUGE` を追加。

```

1041 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1042 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
1043 \if@twocolumn
1044 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
1045 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1046 \else
1047 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
1048 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1049 \fi
1050 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1051 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1052 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1053 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1054 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1055 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

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

```

1056 \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。

```

1057 % \newcommand{\headfont}{\bfseries}
1058 \newcommand{\headfont}{\sffamily}
1059 % \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。

```

1060 %<kiyou>\setlength\columnsep{2\Cwd}
1061 %<kiyou>\setlength\columnsep{28truebp}
1062 \setlength\columnseprule{0\jsc@mp}

```

## ■段落

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

```

\lineskiplimit 1063 \setlength\lineskip{1\jsc@mp}
1064 \setlength\normallineskip{1\jsc@mp}
\normallineskiplimit 1065 \setlength\lineskiplimit{1\jsc@mp}
1066 \setlength\normallineskiplimit{1\jsc@mp}

```

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

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

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

```
1067 \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。

```

1068 \setlength\parskip{0\jsc@mp}
1069 \if@slide
1070 \setlength\parindent{0\p@}
1071 \else
1072 \setlength\parindent{1\Cwd}
1073 \fi

```

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

```

\@highpenalty 1074 \@lowpenalty 51
1075 \@medpenalty 151
1076 \@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。

```
1077 % \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。

```
1078 % \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

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
1079 \ifjsc@mag
1080 \mag=\bxjs@param@mag
1081 \fi
1082 \setlength{\topskip}{10\p0?}
```

`\jsSetQHLlength` のための和文単位の定義。

```
1083 \def\bxjs@unit@trueQ{0.25trueem}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1084 \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。

```
1085 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
1086 \def\bxjs@read@b{%
1087   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
1088   \else \expandafter\bxjs@read@d \fi}
1089 \def\bxjs@read@c#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1090 \def\bxjs@read@d#1\@nil{\bxjs@read@e#1,,\@nil}
1091 \def\bxjs@read@e#1,#2,#3\@nil{%
1092   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1093 \expandafter\bxjs@read@a\bxjs@param@paper\@empty\@empty\@nil
```

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

```
1094 \edef\bxjs@layout@paper{%
1095   \ifjsc@mag truedimen,\fi
1096   \if@landscape landscape,\fi
1097   \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。

```
1098 %<*article|report>
1099 \def\bxjs@layout@base{%
1100   headheight=\topskip,footskip=0.03367\paperheight,%
1101   headsep=\footskip-\topskip,includeheadfoot,%
1102 }
1103 \edef\bxjs@layout{\bxjs@layout@base
1104   hscale=0.76,hmarginratio=1:1,%
1105   vscale=0.83,vmarginratio=1:1,%
1106 }
1107 %</article|report>
1108 %<*book>
1109 \def\bxjs@layout@base{%
1110   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1111 }
1112 \ifbxjs@layout@buggyhmargin %---
1113 % アレ
```

```

1114 \edef\bxjs@layout{\bxjs@layout@base
1115   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1116   vscale=0.83,vmarginratio=1:1,%
1117 }
1118 \else %---
1119 % 非アレ
1120 \edef\bxjs@layout{\bxjs@layout@base
1121   hmargin=18\jsc@mmm,%
1122   vscale=0.83,vmarginratio=1:1,%
1123 }
1124 \fi %---
1125 %</book>
1126 %<*slide>
1127 \def\bxjs@layout@base{%
1128   noheadfoot,%
1129 }
1130 \edef\bxjs@layout{\bxjs@layout@base
1131   hscale=0.9,hmarginratio=1:1,%
1132   vscale=0.944,vmarginratio=1:1,%
1133 }
1134 %</slide>

```

`textwidth` オプションの設定を反映する。

```

1135 %<*!book>
1136 \ifx\bxjs@textwidth@opt\undefined\else
1137   \jsSetQHLLength\@tempdima{\bxjs@textwidth@opt}
1138   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1139 \fi
1140 %</!book>
1141 \ifx\bxjs@number@of@lines@opt\undefined\else
1142   \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1143   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1144 \fi

```

`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1145 \newdimen\fullwidth
```

`\jsTextWidthLimit` [実数値マクロ] `bxjsbook` における、`\textwidth` の上限の全角単位での値。

```

1146 %<*book>
1147 \newcommand\jsTextWidthLimit{40}
1148 \ifx\bxjs@textwidth@limit\undefined\else
1149   \bxjs@gset@tempcnta{\bxjs@textwidth@limit}
1150   \long\edef\jsTextWidthLimit{\the\@tempcnta}
1151 \fi
1152 %</book>

```

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

```

1153 \def\bxjs@postproc@layout{%
1154 % ドライバ再設定

```

```

1155 \ifx\bxjs@geometry@driver\relax\else
1156 \let\Gm@driver\bxjs@geometry@driver
1157 \fi
1158 % textwidth 調整
1159 \ifbxjs@whole@zw@lines
1160 \@tempdimb=\textwidth
1161 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1162 \advance\textwidth.005pt\relax
1163 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1164 \advance\@tempdimb-\textwidth
1165 \advance\oddsidemargin 0.5\@tempdimb
1166 \advance\evensidemargin 0.5\@tempdimb
1167 \fi
1168 \fullwidth=\textwidth
1169 %<*book>
1170 \@tempdima=\jsTextWidthLimit\Cwd
1171 \ifx\bxjs@textwidth@opt\undefined\else
1172 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1173 \fi
1174 \ifdim\textwidth>\@tempdima
1175 \textwidth=\@tempdima
1176 \addtolength\evensidemargin{\fullwidth-\textwidth}
1177 \fi
1178 %</book>
1179 % textheight 調整
1180 \@tempdimb=\textheight
1181 \advance\textheight-\topskip
1182 \advance\textheight.005pt\relax
1183 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1184 \advance\textheight\topskip
1185 \advance\@tempdimb-\textheight
1186 \advance\topmargin0.5\@tempdimb
1187 % headheight 調整
1188 \@tempdima=\topskip
1189 \advance\headheight\@tempdima
1190 \advance\topmargin-\@tempdima
1191 % marginpar 設定
1192 \setlength\marginparsep{\columnsep}
1193 \setlength\marginparpush{\baselineskip}
1194 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1195 -\textwidth-10\jsc@mmm-\marginparsep}
1196 \ifbxjs@whole@zw@lines
1197 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1198 \fi
1199 % 連動する変数
1200 \maxdepth=.5\topskip
1201 \stockwidth=\paperwidth
1202 \stockheight=\paperheight
1203 }

```

`\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。

```
1204 \edef\jsGeometryOptions{%
1205   \bxjs@layout@paper,\bxjs@layout}
```

---

## ■ `geometry` パッケージ読込

---

`geoemtry` オプションの値に応じて分岐する。

まずは `geometry=class` の場合。

```
1206 \ifx\bxjs@geometry\bxjs@geometry@class
```

`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。

```
1207 \def\bxjs@geometry@guard@on{%
1208   \edef\bxjs@gg@tmp{\the\ht\strutbox}\ht\strutbox=10\jsc@mpt}
1209 \def\bxjs@geometry@guard@off{%
1210   \ht\strutbox=\bxjs@gg@tmp\relax}
```

`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。

```
1211 \ifbxjs@papersize
1212   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1213     \PassOptionsToPackage{dvipdfm}{geometry}
1214   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1215     \PassOptionsToPackage{dvipdfm}{geometry}
1216   \fi\fi
1217   \let\bxPapersizeSpecialDone=t
1218 \else
1219   \PassOptionsToPackage{driver=none}{geometry}
1220 \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。

```
1221 \AtBeginDocument{\bxjs@pre@geometry@hook}
1222 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1223 \bxjs@geometry@guard@on
1224 \edef\bxjs@next{%
1225   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1226 }\bxjs@next
1227 \bxjs@geometry@guard@off
1228 \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。

```
1229 \let\bxjs@geometry@driver\Gm@driver
1230 \bxjs@postproc@layout
```

`\bxjs@pre@geometry@hook` 1.2 版より、`geometry` の 4.x 版の使用は非推奨とする。

※将来サポートを廃止する予定。

```
1231 \@onlypreamble\bxjs@pre@geometry@hook
1232 \def\bxjs@pre@geometry@hook{%
1233   \@ifpackageloaded{geometry}{%
1234     \@ifpackagelater{geometry}{2010/02/12}{}%\else
1235     \PackageWarningNoLine\bxjs@clsname
1236     {The 'geometry' package installed\MessageBreak
1237     is too old (< v5.0)}%
1238     \if x\jsEngine \ifnum\mag=\@m\else
1239     \def\bxjs@Gm@driver{pdftex}
1240     \ifx\pdfhorigin@\undefined \newdimen\pdfhorigin \fi
1241     \ifx\pdfvorigin@\undefined \newdimen\pdfvorigin \fi
1242     \fi\fi
1243   }%
1244   \ifjsWithpTeXng
1245     \ifx\Gm@driver@\empty
1246     \def\Gm@driver{pdftex}%
1247     \fi
1248   \fi
1249 }{}}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

- `\setpagelayout{<text>}` : 現在の `geometry` の設定の一部を修正する。
- `\setpagelayout*{<text>}` : 用紙以外の設定をリセットして、改めて設定を行う。

どちらも設定の後で後処理 `\bxjs@postproc@layout` を実行する。

```
1250 \def\setpagelayout{%
1251   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1252     \@ifstar{\bxjs@setpagelayout@a\@ne}{\bxjs@setpagelayout@a\z@}}
1253 \def\bxjs@setpagelayout@a#1#2{%
1254   \ifcase#1% modify
1255     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1256   \or% reset(*)
1257     \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1258   \or% semireset(+)
1259     \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1260   \fi
1261   \bxjs@geometry@guard@on
1262   \edef\bxjs@next{%
1263     \noexpand\geometry{\bxjs@next}%
1264   }\bxjs@next
```

```
1265 \bxjs@geometry@guard@off
1266 \bxjs@postproc@layout}
```

geometry=class の場合の処理はここで終わり。

次に、geometry=user の場合の処理。

```
1267 \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。

```
1268 \AtBeginDocument{\bxjs@check@page@layout}
1269 \@onlypreamble\bxjs@check@page@layout
1270 \def\bxjs@check@page@layout{%
1271   \ifdim\textwidth=.5\maxdimen
1272     \ClassError\bxjs@clsname
1273       {Page layout is not properly set}%
1274     {\@ehd}
1275   \fi}
1276 \def\jsUseMinimalPageLayout{%
1277   \setlength{\textwidth}{6.5in}%
1278   \setlength{\textheight}{8in}}
```

\setpagelayout はとりあえず無効にしておく。

```
1279 \let\bxjs@geometry@driver\relax
1280 \def\setpagelayout{%
1281   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1282     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1283 \def\bxjs@pagelayout@a#1{%
1284   \ClassError\bxjs@clsname
1285   {Command '\string\setpagelayout' is not supported,\MessageBreak
1286   because 'geometry' value is not 'class'}\@eha}
```

geometry=user の場合の処理はここで終わり。

```
1287 \fi\fi
```

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。

---

```
1288 %<*\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。

```
1289 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1290 \if@slide
1291   \setlength\headheight{0\jsc@mpt}
1292 \else
1293   \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1294 \fi
```

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

```
1295 %<*article|kiyou>
1296 \if@slide
1297   \setlength\footskip{0pt}
1298 \else
1299   \setlength\footskip{0.03367\paperheight}
1300   \ifdim\footskip<\baselineskip
1301     \setlength\footskip{\baselineskip}
1302   \fi
1303 \fi
1304 %</article|kiyou>
1305 %<jspf>\setlength\footskip{9\jsc@mmm}
1306 %<*book>
1307 \if@report
1308   \setlength\footskip{0.03367\paperheight}
1309   \ifdim\footskip<\baselineskip
1310     \setlength\footskip{\baselineskip}
1311   \fi
1312 \else
1313   \setlength\footskip{0pt}
1314 \fi
1315 %</book>
1316 %<*report>
1317 \setlength\footskip{0.03367\paperheight}
1318 \ifdim\footskip<\baselineskip
1319   \setlength\footskip{\baselineskip}
```

```
1320 \fi
1321 %</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)。

```
1322 %<*article>
1323 \if@slide
1324   \setlength\headsep{0\jsc@mpt}
1325   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1326   \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1327 \else
1328   \setlength\headsep{\footskip}
1329   \addtolength\headsep{-\topskip}
1330 \fi
1331 %</article>
1332 %<*book>
1333 \if@report
1334   \setlength\headsep{\footskip}
1335   \addtolength\headsep{-\topskip}
1336 \else
1337   \setlength\headsep{6\jsc@mmm}
1338   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1339   \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1340 \fi
1341 %</book>
1342 %<*report>
1343 \setlength\headsep{\footskip}
1344 \addtolength\headsep{-\topskip}
1345 %</report>
1346 %<*jspf>
1347 \setlength\headsep{9\jsc@mmm}
1348 \addtolength\headsep{-\topskip}
1349 %</jspf>
1350 %<*kiyou>
1351 \setlength\headheight{0\jsc@mpt}
1352 \setlength\headsep{0\jsc@mpt}
1353 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1354 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1355 %</kiyou>
```

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

```
1356 \setlength\maxdepth{.5\topskip}
```

### ■本文の幅と高さ

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

```
1357 \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。

```
1358 %<*article>
1359 \if@slide
1360   \setlength\fullwidth{0.9\paperwidth}
1361 \else
1362   \setlength\fullwidth{0.76\paperwidth}
1363 \fi
1364 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1365 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1366 \setlength\textwidth{\fullwidth}
1367 %</article>
1368 %<*book>
1369 \if@report
1370   \setlength\fullwidth{0.76\paperwidth}
1371 \else
1372   \setlength\fullwidth{\paperwidth}
1373   \addtolength\fullwidth{-36\jsc@mmm}
1374 \fi
1375 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1376 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1377 \setlength\textwidth{\fullwidth}
1378 \if@report \else
1379   \if@twocolumn \else
1380     \ifdim \fullwidth>40zw
1381       \setlength\textwidth{40zw}
1382     \fi
1383   \fi
1384 \fi
1385 %</book>
1386 %<*report>
```

```

1387 \setlength\fullwidth{0.76\paperwidth}
1388 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1389 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1390 \setlength\textwidth{\fullwidth}
1391 %</report>
1392 %<*jspf>
1393 \setlength\fullwidth{50zw}
1394 \addtolength\fullwidth{8\jsc@mmm}
1395 \setlength\textwidth{\fullwidth}
1396 %</jspf>
1397 %<*kiyou>
1398 \setlength\fullwidth{48zw}
1399 \addtolength\fullwidth{\columnsep}
1400 \setlength\textwidth{\fullwidth}
1401 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

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

[2016-10-08] article の slide のときに `\headheight` はゼロなので、さらに修正しました (2016-08-17 での修正漏れ)。

```

1402 %<*article|book|report>
1403 \if@slide
1404 \setlength{\textheight}{0.95\paperheight}
1405 \else
1406 \setlength{\textheight}{0.83\paperheight}
1407 \fi
1408 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1409 \addtolength{\textheight}{-\headsep}
1410 \addtolength{\textheight}{-\footskip}
1411 \addtolength{\textheight}{-\topskip}
1412 \divide\textheight\baselineskip
1413 \multiply\textheight\baselineskip
1414 %</article|book|report>
1415 %<jspf>\setlength{\textheight}{51\baselineskip}
1416 %<kiyou>\setlength{\textheight}{47\baselineskip}

```

```

1417 \addtolength{\textheight}{\topskip}
1418 \addtolength{\textheight}{0.1\jsc@mpt}
1419 %<jspf>\setlength{\mathindent}{10\jsc@mmm}

```

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

```

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

```

ですが、次のようにします。

```

1420 \def\flushbottom{%
1421 \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1422 \let\@texttop\relax}

```

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

```

1423 \setlength\marginparsep{\columnsep}
1424 \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` が使われます。T<sub>E</sub>X は上・左マージンに `1truein` を挿入しますが、トンボ関係のオプションが指定されると pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (`plcore.ltx`) はトンボの内側に `1in` のスペース (`1truein` ではなく) を挿入するので、場合分けしています。

```

1425 \setlength{\oddsidemargin}{\paperwidth}
1426 \addtolength{\oddsidemargin}{-\fullwidth}
1427 \setlength{\oddsidemargin}{.5\oddsidemargin}
1428 \iftombow
1429 \addtolength{\oddsidemargin}{-1in}
1430 \else
1431 \addtolength{\oddsidemargin}{-\inv@mag in}
1432 \fi
1433 \setlength{\evensidemargin}{\oddsidemargin}
1434 \if@mparswitch
1435 \addtolength{\evensidemargin}{\fullwidth}
1436 \addtolength{\evensidemargin}{-\textwidth}
1437 \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。

```

1438 \setlength\marginparwidth{\paperwidth}
1439 \addtolength\marginparwidth{-\oddsidemargin}
1440 \addtolength\marginparwidth{-\inv@mag in}
1441 \addtolength\marginparwidth{-\textwidth}
1442 \addtolength\marginparwidth{-10\jsc@mmm}
1443 \addtolength\marginparwidth{-\marginparsep}

```

```

1444 \@tempdima=1zw
1445 \divide\marginparwidth\@tempdima
1446 \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）。

```

1447 \setlength\topmargin{\paperheight}
1448 \addtolength\topmargin{-\textheight}
1449 \if@slide
1450 \addtolength\topmargin{-\headheight}
1451 \else
1452 \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1453 \fi
1454 \addtolength\topmargin{-\headsep}
1455 \addtolength\topmargin{-\footskip}
1456 \setlength\topmargin{0.5\topmargin}
1457 %<kiyou>\setlength\topmargin{81truebp}
1458 \iftombow
1459 \addtolength\topmargin{-1in}
1460 \else
1461 \addtolength\topmargin{-\inv@mag in}
1462 \fi
1463 %</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。

---

```
1464 \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。

1465 `\setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}`

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

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

[2003-08-23] ちょっと増やしました。

1466 `\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。

1467 `\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。

1468 `\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。

1469 `\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。

1470 `\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。

1471 `\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。

1472 `\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。

1473 `\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。

1474 `\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。

1475 `\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。`\intertextsep` は本文の途中に出力されるフロートと本  
`\intertextsep` 文との距離です。

```
1476 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1477 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1478 \setlength\intertextsep {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 1479 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1480 \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 1481 \setlength\@fptop{0\p@? \@plus 1fil}
1482 \setlength\@fpsep{8\p@? \@plus 2fil}
1483 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpptop` 段抜きフロートについての値です。

```
\@dblfpsep 1484 \setlength\@dblfpptop{0\p@? \@plus 1fil}
\@dblfpbot 1485 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1486 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

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

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

```
\pltx@cleartooddpage
\pltx@cleartoevenpage
```

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

となっています。

```
1487 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1488 % \ifodd\c@page
1489 % \iftdir
1490 % \hbox{}\thispagestyle{empty}\newpage
1491 % \if@twocolumn\hbox{}\newpage\fi
1492 % \fi
1493 % \else
1494 % \ifydir
1495 % \hbox{}\thispagestyle{empty}\newpage
1496 % \if@twocolumn\hbox{}\newpage\fi
1497 % \fi
1498 % \fi\fi}
1499 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
```

```

1500 % \ifodd\c@page
1501 %   \ifydir
1502 %     \hbox{}\thispagestyle{empty}\newpage
1503 %     \if@twocolumn\hbox{}\newpage\fi
1504 %   \fi
1505 % \else
1506 %   \iftdir
1507 %     \hbox{}\thispagestyle{empty}\newpage
1508 %     \if@twocolumn\hbox{}\newpage\fi
1509 %   \fi
1510 % \fi\fi}
1511 \def\pltx@cleartooddpage{\clearpage\if@twoside
1512 \ifodd\c@page\else
1513   \hbox{}\thispagestyle{empty}\newpage
1514   \if@twocolumn\hbox{}\newpage\fi
1515 \fi\fi}
1516 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1517 \ifodd\c@page
1518   \hbox{}\thispagestyle{empty}\newpage
1519   \if@twocolumn\hbox{}\newpage\fi
1520 \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。

```

1521 \let\pltx@cleartorightpage\pltx@cleartooddpage
1522 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

---

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

```

1523 %<*book|report>
1524 \if@openleft
1525   \let\cleardoublepage\pltx@cleartoleftpage
1526 \else\if@openright
1527   \let\cleardoublepage\pltx@cleartorightpage
1528 \fi\fi
1529 %</book|report>

```

## 7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

`\@evenhead` `\@oddhead`, `\@oddfoot`, `\@evenhead`, `\@evenfoot` は偶数・奇数ページの柱（ヘッダ,  
`\@oddhead` 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。

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

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

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

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

```
1530 % \def\ps@empty{%
1531 %   \let\mkboth\gobbletwo
1532 %   \let\@oddhead\@empty
1533 %   \let\@oddfoot\@empty
1534 %   \let\@evenhead\@empty
1535 %   \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。

```
1536 \def\ps@plainfoot{%
1537   \let\mkboth\gobbletwo
1538   \let\@oddhead\@empty
1539   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1540   \let\@evenhead\@empty
1541   \let\@evenfoot\@oddfoot}
1542 \def\ps@plainhead{%
1543   \let\mkboth\gobbletwo
1544   \let\@oddfoot\@empty
1545   \let\@evenfoot\@empty
1546   \def\@evenhead{%
1547     \ifmparswitch \hss \fi
1548     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1549     \ifmparswitch\else \hss \fi}%
1550   \def\@oddhead{%
1551     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1552 %<book>\let\ps@plain\ps@plainhead
```

```
1553 %<!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。

```
1554 %<*article|slide>
1555 \if@twoside
1556   \def\ps@headings{%
1557     \let\@oddfoot\@empty
1558     \let\@evenfoot\@empty
1559     \def\@evenhead{\if@mparswitch \hss \fi
1560       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1561     \if@mparswitch\else \hss \fi}%
1562     \def\@oddhead{%
1563       \underline{%
1564         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1565     \let\@mkboth\markboth
1566     \def\sectionmark##1{\markboth{%
1567       \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1568       ##1}{}}%
1569     \def\subsectionmark##1{\markright{%
1570       \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1571       ##1}}%
1572   }
1573 \else % if not twoside
1574   \def\ps@headings{%
1575     \let\@oddfoot\@empty
1576     \def\@oddhead{%
1577       \underline{%
1578         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1579     \let\@mkboth\markboth
1580     \def\sectionmark##1{\markright{%
1581       \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1582       ##1}}%
1583   \fi
1584 %</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。

---

```
1585 %<*book|report>
1586 \def\bxjs@maybe@autoxspacing{%
1587   \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1588 \newif\if@omit@number
1589 \def\ps@headings{%
```

```

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

```

最後は学会誌の場合です。

```

1613 %<*jspf>
1614 \def\ps@headings{%
1615   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1616   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1617   \def\@oddhead{\normalfont\hfil \@title \hfil}
1618   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1619 %</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。

```

1620 \def\ps@myheadings{%
1621   \let\@oddfoot\@empty\let\@evenfoot\@empty
1622   \def\@evenhead{%
1623     \if@mparswitch \hss \fi%
1624     \hbox to \fullwidth{\thepage\hfil\leftmark}}%
1625     \if@mparswitch\else \hss \fi}%
1626   \def\@oddhead{%
1627     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1628   \let\@mkboth\@gobbletwo
1629 %<book|report> \let\chaptermark\@gobble
1630 \let\sectionmark\@gobble
1631 %<!book&!report> \let\subsectionmark\@gobble
1632 }

```

## 8 文書のマークアップ

### 8.1 表題

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

```
\date 1633 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1634 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1635 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1636 % \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。

```
1637 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1638 %\let\bxjs@subtitle\undefined

      \title にフックを入れる。

1639 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1640 \AtBeginDocument{\bxjs@decl@subtitle}
1641 \def\bxjs@decl@subtitle{%
1642   \global\let\bxjs@decl@subtitle\relax
1643   \ifx\subtitle\undefined
1644     \global\let\subtitle\jsSubtitle
1645   \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。

```
1646 \def\bxjs@annihilate@subtitle{%
1647   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1648   \global\let\jsSubtitle\relax}
```

---

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

```
\eauthor 1649 %<*jspf>
\keywords 1650 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
          1651 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
          1652 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
          1653 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
          1654 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
          1655 %</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ります。

```

1656 \def\plainifnotempty{%
1657   \ifx \@oddhead \@empty
1658     \ifx \@oddfoot \@empty
1659       \else
1660         \thispagestyle{plainfoot}%
1661       \fi
1662     \else
1663       \thispagestyle{plainhead}%
1664     \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。

```

1665 %<*article|book|report|slide>
1666 \if@titlepage
1667   \newcommand{\maketitle}{%
1668     \begin{titlepage}%
1669       \let\footnotesize\small
1670       \let\footnoterule\relax
1671       \let\footnote\thanks
1672       \null\vfil
1673       \if@slide
1674         {\footnotesize \@date}%
1675       \begin{center}
1676         \mbox{} \\\[1\jsZw]
1677         \large
1678         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1679         \jsc@smallskip
1680         \@title
1681         \ifx\bxjs@subtitle\@undefined\else
1682           \par\vskip\z@
1683           {\small \bxjs@subtitle\par}
1684         \fi
1685         \jsc@smallskip
1686         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1687         \vfill
1688         {\small \@author}%
1689       \end{center}
1690     \else
1691       \vskip 60\p@?

```

```

1692     \begin{center}%
1693         {\LARGE \@title \par}%
1694     \ifx\bxjs@subtitle\@undefined\else
1695         \vskip5\p@?
1696         {\normalsize \bxjs@subtitle\par}
1697     \fi
1698     \vskip 3em%
1699     {\large
1700         \lineskip .75em
1701         \begin{tabular}[t]{c}%
1702             \@author
1703         \end{tabular}\par}%
1704     \vskip 1.5em
1705     {\large \@date \par}%
1706     \end{center}%
1707     \fi
1708     \par
1709     \@thanks\vfil\null
1710 \end{titlepage}%
1711 \setcounter{footnote}{0}%
1712 \global\let\thanks\relax
1713 \global\let\maketitle\relax
1714 \global\let\@thanks\@empty
1715 \global\let\@author\@empty
1716 \global\let\@date\@empty
1717 \global\let\@title\@empty
1718 \global\let\title\relax
1719 \global\let\author\relax
1720 \global\let\date\relax
1721 \global\let\and\relax
1722 \bxjs@annihilate@subtitle
1723 }%
1724 \else
1725     \newcommand{\maketitle}{\par
1726         \begingroup
1727             \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1728             \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1729             \long\def\@makefnmark##1{\advance\leftskip 3\jsZw
1730                 \parindent 1\jsZw\noindent
1731                 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1732             \if@twocolumn
1733                 \ifnum \col@number=\@ne
1734                     \@maketitle
1735                 \else
1736                     \twocolumn[\@maketitle]%
1737                 \fi
1738             \else
1739                 \newpage
1740                 \global\@topnum\z@ % Prevents figures from going at top of page.

```

```

1741     \@maketitle
1742     \fi
1743     \plainifnotempty
1744     \@thanks
1745     \endgroup
1746     \setcounter{footnote}{0}%
1747     \global\let\thanks\relax
1748     \global\let\maketitle\relax
1749     \global\let\@thanks\@empty
1750     \global\let\@author\@empty
1751     \global\let\@date\@empty
1752     \global\let\@title\@empty
1753     \global\let\title\relax
1754     \global\let\author\relax
1755     \global\let\date\relax
1756     \global\let\and\relax
1757     \bxjs@annihilate@subtitle
1758 }

```

`\@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。

```

1759 \def\@maketitle{%
1760     \newpage\null
1761     \vskip 2em
1762     \begin{center}%
1763         \let\footnote\thanks
1764         {\LARGE \@title \par}%
1765         \ifx\bxjs@subtitle\@undefined\else
1766             \vskip3\p@?
1767             {\normalsize \bxjs@subtitle\par}
1768         \fi
1769         \vskip 1.5em
1770         {\large
1771             \lineskip .5em
1772             \begin{tabular}[t]{c}%
1773                 \@author
1774             \end{tabular}\par}%
1775         \vskip 1em
1776         {\large \@date}%
1777     \end{center}%
1778     \par\vskip 1.5em
1779 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1780 }
1781 \fi
1782 %</article|book|report|slide>
1783 %<*jspf>
1784 \newcommand{\maketitle}{\par
1785     \begingroup
1786     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1787     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%

```

```

1788 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1789 \parindent 1\jsZw\noindent
1790 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1791 \twocolumn[\@maketitle]%
1792 \plainifnotempty
1793 \@thanks
1794 \endgroup
1795 \setcounter{footnote}{0}%
1796 \global\let\thanks\relax
1797 \global\let\maketitle\relax
1798 \global\let\@thanks\@empty
1799 \global\let\@author\@empty
1800 \global\let\@date\@empty
1801 % \global\let\@title\@empty % \@title は柱に使う
1802 \global\let\title\relax
1803 \global\let\author\relax
1804 \global\let\date\relax
1805 \global\let\and\relax
1806 \ifx\authors@mail\@undefined\else{%
1807 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1808 \footnotetext[0]{\itshape\authors@mail}%
1809 }\fi
1810 \global\let\authors@mail\@undefined}
1811 \def\@maketitle{%
1812 \newpage\null
1813 \vskip 6em % used to be 2em
1814 \begin{center}
1815 \let\footnote\thanks
1816 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1817 \lineskip .5em
1818 \ifx\@author\@undefined\else
1819 \vskip 1em
1820 \begin{tabular}[t]{c}%
1821 \@author
1822 \end{tabular}\par
1823 \fi
1824 \ifx\@etitle\@undefined\else
1825 \vskip 1em
1826 {\large \@etitle \par}%
1827 \fi
1828 \ifx\@eauthor\@undefined\else
1829 \vskip 1em
1830 \begin{tabular}[t]{c}%
1831 \@eauthor
1832 \end{tabular}\par
1833 \fi
1834 \vskip 1em
1835 \@date
1836 \end{center}

```

```

1837 \vskip 1.5em
1838 \centerline{\box\@abstractbox}
1839 \ifx\@keywords\@undefined\else
1840 \vskip 1.5em
1841 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1842 \fi
1843 \vskip 1.5em}
1844 %</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。

```

1845 \def\bxjs@label@sect#1{%
1846 \expandafter\ifx\csname label#1\endcsname\relax
1847 \csname the#1\endcsname
1848 \else \csname label#1\endcsname
1849 \fi}
1850 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1851 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1852 \def\@secapp{\presectionname}
1853 \def\@secpos{\postsectionname}
1854 \fi

```

`\labelsection` 節番号の出力書式。

```

1855 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1856 \def\labelsection{\@secapp\thesection\@secpos}
1857 \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。

```
1858 \def\@startsection#1#2#3#4#5#6{%
1859   \if@noskipsec \leavevmode \fi
1860   \par
1861 % 見出し上の空きを \@tempskipa にセットする
1862   \@tempskipa #4\relax
1863 % \@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
1864   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1865 % 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
1866   \ifdim \@tempskipa <\z@
1867     \@tempskipa -\@tempskipa \@afterindentfalse
1868   \fi
1869   \if@nobreak
1870 %   \everypar{\everyparhook}% これは間違い
1871     \everypar{}%
1872   \else
1873     \addpenalty\@secpenalty
1874 % 次の行は削除
1875 %   \addvspace\@tempskipa
1876 % 次の \noindent まで追加
1877     \ifdim \@tempskipa >\z@
1878       \if@slide\else
1879         \null
1880         \vspace*{-\baselineskip}%
1881       \fi
1882       \vskip\@tempskipa
1883     \fi
1884   \fi
1885   \noindent
1886 % 追加終わり
1887   \@ifstar
```

```

1888   {\@ssect{#3}{#4}{#5}{#6}}%
1889   {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}

```

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

---

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

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

```

1890 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1891   \let\bxjs@if@ceph\@firstofone
1892 \else \let\bxjs@if@ceph\@gobble
1893 \fi

```

---

```

1894 \def\@sect#1#2#3#4#5#6[#7]#8{%
1895   \ifnum #2>\c@secnumdepth
1896     \let\@svsec\@empty
1897   \else
1898     \refstepcounter{#1}%
1899     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1900   \fi
1901 % 見出し後の空きを \@tempkipa にセット
1902   \@tempkipa #5\relax
1903 % 条件判断の順序を入れ替えました
1904   \ifdim \@tempkipa<\z@
1905     \def\@svsechd{%
1906       #6{\hskip #3\relax
1907         \@svsec #8}%
1908       \csname #1mark\endcsname{#7}%
1909       \addcontentsline{toc}{#1}{%
1910         \ifnum #2>\c@secnumdepth \else
1911           \protect\numberline{\bxjs@label@sect{#1}}%
1912         \fi
1913         #7}}% 目次にフルネームを載せるなら #8
1914   \else
1915     \begingroup
1916       \interlinepenalty \@M % 下から移動
1917       #6{%
1918         \@hangfrom{\hskip #3\relax\@svsec}%
1919       \interlinepenalty \@M % 上に移動
1920       #8\@@par}%
1921     \endgroup
1922     \csname #1mark\endcsname{#7}%
1923     \addcontentsline{toc}{#1}{%
1924       \ifnum #2>\c@secnumdepth \else
1925         \protect\numberline{\bxjs@label@sect{#1}}%
1926       \fi
1927       #7}}% 目次にフルネームを載せるならここは #8

```

```
1928 \fi
1929 \@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。

```
1930 \def\@xsect#1{%
1931 % 見出しの後ろの空きを \@tempskipa にセット
1932 \@tempskipa #1\relax
1933 % 条件判断の順序を変えました
1934 \ifdim \@tempskipa<\z@
1935 \@nbreakfalse
1936 \global\@noskipsectrue
1937 \everypar{%
1938 \if@noskipsec
1939 \global\@noskipsecfalse
1940 {\setbox\z@\lastbox}%
1941 \clubpenalty\@M
1942 \begingroup \@svsechd \endgroup
1943 \unskip
1944 \@tempskipa #1\relax
1945 \hskip -\@tempskipa
1946 \else
1947 \clubpenalty \@clubpenalty
1948 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1949 \fi\bxjs@if@ceph\everyparhook}%
1950 \else
1951 \par \nbreak
1952 \vskip \@tempskipa
1953 \@afterheading
1954 \fi
1955 \if@slide
1956 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1957 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1958 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1959 \fi
1960 \par % 2000-12-18
1961 \ignorespaces}
1962 \def\@sssect#1#2#3#4#5{%
1963 \@tempskipa #3\relax
1964 \ifdim \@tempskipa<\z@
1965 \def\@svsechd{#4{\hskip #1\relax #5}}%
1966 \else
1967 \begingroup
1968 #4{%
1969 \@hangfrom{\hskip #1}%
1970 \interlinepenalty \@M #5\@@par}%
```

```

1971 \endgroup
1972 \fi
1973 \@xsect{#3}

```

### ■柱関係の命令

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

```

\subsectionmark 1974 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1975 % \newcommand*\sectionmark[1]{}
1976 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1977 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1978 % \newcommand*\paragraphmark[1]{}
1979 % \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。

```

1980 %<!book&!report>\setcounter{secnumdepth}{3}
1981 %<book|report>\setcounter{secnumdepth}{2}

```

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

`\c@section` 第 2 引数が増加するたびに 0 に戻されます。第 2 引数は定義済みのカウンタです。

```

\c@subsection 1982 \newcounter{part}
\c@subsubsection 1983 %<book|report>\newcounter{chapter}
1984 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1985 %<!book&!report>\newcounter{section}
\c@subparagraph 1986 \newcounter{subsection}[section]
1987 \newcounter{subsubsection}[subsection]
1988 \newcounter{paragraph}[subsubsection]
1989 \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。

```

1990 \renewcommand{\thepart}{\@Roman\c@part}
1991 %<!*book&!report>
1992 \ifnum\bxjs@label@section=\bxjs@label@section@compat
1993 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}

```

```

1994 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1995 \else
1996 \renewcommand{\thesection}{\@arabic\c@section}
1997 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1998 \fi
1999 %<!/book&!report>
2000 %<*book|report>
2001 \renewcommand{\thechapter}{\@arabic\c@chapter}
2002 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2003 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2004 %</book|report>
2005 \renewcommand{\thesubsubsection}{%
2006   \thesubsection.\@arabic\c@subsubsection}
2007 \renewcommand{\theparagraph}{%
2008   \thesubsubsection.\@arabic\c@paragraph}
2009 \renewcommand{\thesubparagraph}{%
2010   \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。

```

2011 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2012 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

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

```

2013 %<*book|report>
2014 \newcommand\frontmatter{%
2015   \pltx@cleartooddpage
2016   \@mainmatterfalse
2017   \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。

```

2018 \newcommand\mainmatter{%
2019   \pltx@cleartooddpage
2020   \@mainmattertrue
2021   \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。

```

2022 \newcommand\backmatter{%
2023   \if@openleft
2024     \cleardoublepage
2025   \else\if@openright
2026     \cleardoublepage
2027   \else
2028     \clearpage
2029   \fi\fi
2030   \@mainmatterfalse}
2031 %</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。

```

2032 %<!*book&!report>
2033 \newcommand\part{%
2034   \if@noskipsec \leavevmode \fi
2035   \par
2036   \addvspace{4ex}%
2037   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2038   \secdef\@part\@spart}
2039 %</!*book&!report>

```

`book` および `report` クラスの場合は、少し複雑です。

```

2040 %<*book|report>
2041 \newcommand\part{%
2042   \if@openleft
2043     \cleardoublepage
2044   \else\if@openright
2045     \cleardoublepage
2046   \else
2047     \clearpage
2048   \fi\fi
2049   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2050   \if@twocolumn
2051     \onecolumn

```

```

2052 \@restonecoltrue
2053 \else
2054 \@restonecolfalse
2055 \fi
2056 \null\vfil
2057 \secdef\@part\@spart}
2058 %</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。

```

2059 %<!*book&!report>
2060 \def\@part[#1]#2{%
2061 \ifnum \c@secnumdepth >\m@ne
2062 \refstepcounter{part}%
2063 \addcontentsline{toc}{part}{%
2064 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2065 \else
2066 \addcontentsline{toc}{part}{#1}%
2067 \fi
2068 \markboth{}{}%
2069 {\parindent\z@
2070 \raggedright
2071 \interlinepenalty \@M
2072 \normalfont
2073 \ifnum \c@secnumdepth >\m@ne
2074 \Large\headfont\prepartname\thepart\postpartname
2075 \par\nobreak
2076 \fi
2077 \huge \headfont #2%
2078 \markboth{}{\par}%
2079 \nobreak
2080 \vskip 3ex
2081 \@afterheading}
2082 %</!*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。

```

2083 %<*book|report>
2084 \def\@part[#1]#2{%
2085 \ifnum \c@secnumdepth >-2\relax
2086 \refstepcounter{part}%
2087 \addcontentsline{toc}{part}{%
2088 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2089 \else
2090 \addcontentsline{toc}{part}{#1}%
2091 \fi
2092 \markboth{}{}%
2093 {\centering
2094 \interlinepenalty \@M

```

```

2095 \normalfont
2096 \ifnum \c@secnumdepth >-2\relax
2097 \huge\headfont \prepartname\thepart\postpartname
2098 \par\vskip20\p@?
2099 \fi
2100 \Huge \headfont #2\par}%
2101 \@endpart}
2102 %</book|report>

```

`\@spart` 番号を付けない部です。

```

2103 %<!*book&!report>
2104 \def\@spart#1{%
2105 \parindent \z@ \raggedright
2106 \interlinepenalty \@M
2107 \normalfont
2108 \huge \headfont #1\par}%
2109 \nobreak
2110 \vskip 3ex
2111 \@afterheading}
2112 %</!book&!report>
2113 %<*book|report>
2114 \def\@spart#1{%
2115 \centering
2116 \interlinepenalty \@M
2117 \normalfont
2118 \Huge \headfont #1\par}%
2119 \@endpart}
2120 %</book|report>

```

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

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

```

2121 %<*book|report>
2122 \def\@endpart{\vfil\newpage
2123 \if@twoside
2124 \if@openleft %% added (2017/02/24)
2125 \null\thispagestyle{empty}\newpage
2126 \else\if@openright %% added (2016/12/13)
2127 \null\thispagestyle{empty}\newpage
2128 \fi\fi %% added (2016/12/13, 2017/02/24)
2129 \fi
2130 \if@restonecol
2131 \twocolumn
2132 \fi}
2133 %</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。

```
2134 %<*book|report>
2135 \newcommand{\chapter}{%
2136   \if@openleft\cleardoublepage\else
2137   \if@openright\cleardoublepage\else\clearpage\fi\fi
2138   \plainifnotempty % 元: \thispagestyle{plain}
2139   \global\@topnum\z@
2140   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2141   \secdef
2142     {\@omit@numberfalse\@chapter}%
2143     {\@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。

```
2144 \def\@chapter[#1]#2{%
2145   \ifnum \c@secnumdepth >\m@ne
2146     \if@mainmatter
2147       \refstepcounter{chapter}%
2148       \typeout{\@chapapp\thechapter\@chappos}%
2149       \addcontentsline{toc}{chapter}%
2150         {\protect\numberline
2151 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2152         {\@chapapp\thechapter\@chappos}%
2153         #1}%
2154     \else\addcontentsline{toc}{chapter}{#1}\fi
2155   \else
2156     \addcontentsline{toc}{chapter}{#1}%
2157   \fi
2158   \chaptermark{#1}%
2159   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}%
2160   \addtocontents{lot}{\protect\addvspace{10\jsc@empt}}%
2161   \if@twocolumn
2162     \@topnewpage[\@makechapterhead{#2}]%
2163   \else
2164     \@makechapterhead{#2}%
2165     \@afterheading
2166   \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。

```
2167 \def\@makechapterhead#1{%
2168   \vspace*{2\Cvs}% 欧文は 50pt
2169   {\parindent \z@ \raggedright \normalfont
2170     \ifnum \c@secnumdepth >\m@ne
2171       \if@mainmatter
2172         \huge\headfont \@chapapp\thechapter\@chappos
```

```

2173     \par\nobreak
2174     \vskip \Cvs % 欧文は 20pt
2175     \fi
2176     \fi
2177     \interlinepenalty\@M
2178     \Huge \headfont #1\par\nobreak
2179     \vskip 3\Cvs}} % 欧文は 40pt

```

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

```

2180 \def\@chapter#1{%
2181   \chaptermark{#1}%
2182   \if@twocolumn
2183     \topnewpage[\@makeschapterhead{#1}]%
2184   \else
2185     \@makeschapterhead{#1}\@afterheading
2186   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2187 \def\@makeschapterhead#1{%
2188   \vspace*{2\Cvs}% 欧文は 50pt
2189   {\parindent \z@ \raggedright
2190     \normalfont
2191     \interlinepenalty\@M
2192     \Huge \headfont #1\par\nobreak
2193     \vskip 3\Cvs}} % 欧文は 40pt
2194 %</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。

```

2195 \if@twocolumn
2196   \newcommand{\section}{%
2197 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2198   \@startsection{section}{1}{\z@}%
2199 %<kiyou>   {0.6\Cvs}{0.4\Cvs}%
2200 %<kiyou>   {\Cvs}{0.5\Cvs}%
2201 %   {\normalfont\large\headfont\@secapp}}
2202   {\normalfont\large\headfont\raggedright}}
2203 \else
2204   \newcommand{\section}{%
2205     \if@slide\clearpage\fi
2206     \@startsection{section}{1}{\z@}%
2207     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2208     {.5\Cvs \@plus.3\Cdp}% 後アキ
2209 %   {\normalfont\Large\headfont\@secapp}}
2210     {\normalfont\Large\headfont\raggedright}}

```

2211 \fi

\subsection 同上です。

```
2212 \if@twocolumn
2213   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2214     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2215     {\normalfont\normalsize\headfont}}
2216 \else
2217   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2218     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2219     {.5\Cvs \@plus.3\Cdp}% 後アキ
2220     {\normalfont\large\headfont}}
2221 \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)。

```
2222 \if@twocolumn
2223   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2224     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2225     {\normalfont\normalsize\headfont}}
2226 \else
2227   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2228     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2229     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2230     {\normalfont\normalsize\headfont}}
2231 \fi
```

\paragraph 見出しの後ろで改行されません。

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

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

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

---

※ BXJS クラスでは、1.1 版 [2016-02-14] から \jsParagraphMark をサポートしている。

段落のマーク (■) が必ず和文フォントで出力されるようにする。

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

---

```
2232 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2233 \let\bxjs@org@paragraph@mark\jsParagraphMark
2234 \ifx\bxjs@paragraph@mark\@empty
2235   \let\jsParagraphMark\@empty
2236 \else\ifx\bxjs@paragraph@mark\@undefined\else
```

```

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

```

`\subparagraph` 見出しの後ろで改行されません。

```

2252 \if@twocolumn
2253 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2254   {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2255   {\normalfont\normalsize\headfont}}
2256 \else
2257 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2258   {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2259   {\normalfont\normalsize\headfont}}
2260 \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 `\leftmargink` 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。

```

2261 \if@slide
2262 \setlength\leftmargini{1\jsZw}
2263 \else
2264 \if@twocolumn
2265 \setlength\leftmargini{2\jsZw}
2266 \else
2267 \setlength\leftmargini{3\jsZw}
2268 \fi
2269 \fi

```

`\leftmarginii` `ii`, `iii`, `iv` は `\labelsep` とそれぞれ ‘(m)’, ‘vii.’, ‘M.’ の幅との和より大きくすること

`\leftmarginiii`

`\leftmarginiv`

`\leftmarginv`

`\leftmarginvi`

になっています。ここでは全角幅の整数倍に丸めました。

```
2270 \if@slide
2271   \setlength\leftmarginii {1\jsZw}
2272   \setlength\leftmarginiii{1\jsZw}
2273   \setlength\leftmarginiv {1\jsZw}
2274   \setlength\leftmarginv  {1\jsZw}
2275   \setlength\leftmarginvi {1\jsZw}
2276 \else
2277   \setlength\leftmarginii {2\jsZw}
2278   \setlength\leftmarginiii{2\jsZw}
2279   \setlength\leftmarginiv {2\jsZw}
2280   \setlength\leftmarginv  {1\jsZw}
2281   \setlength\leftmarginvi {1\jsZw}
2282 \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。

```
2283 \setlength \labelsep {0.5\jsZw} % .5em
2284 \setlength \labelwidth{\leftmargini}
2285 \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。

```
2286 \setlength\partopsep{z@} % {2\p@ \@plus 1\p@ \@minus 1\p@}
```

`\@beginparpenalty` リストや段落環境の前、リスト項目間に挿入されるペナルティです。

```
\@endparpenalty 2287 \@beginparpenalty -\@lowpenalty
\@itempenalty 2288 \@endparpenalty -\@lowpenalty
2289 \@itempenalty -\@lowpenalty
```

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

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

```
2290 \def\@listi{\leftmargin\leftmargini
2291   \parsep \z@
2292   \topsep 0.5\baselineskip
2293   \itemsep \z@ \relax}
2294 \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)。

```
2295 \@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 2296 \def\@listii{\leftmargin\leftmarginii
\@listiv 2297 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2298 \topsep \z@
\@listv 2299 \parsep \z@
\@listvi 2300 \itemsep\parsep}
2301 \def\@listiii{\leftmargin\leftmarginiii
2302 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2303 \topsep \z@
2304 \parsep \z@
2305 \itemsep\parsep}
2306 \def\@listiv {\leftmargin\leftmarginiv
2307 \labelwidth\leftmarginiv
2308 \advance\labelwidth-\labelsep}
2309 \def\@listv {\leftmargin\leftmarginv
2310 \labelwidth\leftmarginv
2311 \advance\labelwidth-\labelsep}
2312 \def\@listvi {\leftmargin\leftmarginvi
2313 \labelwidth\leftmarginvi
2314 \advance\labelwidth-\labelsep}
```

■**enumerate 環境** `enumerate` 環境はカウンタ `enumi`, `enumii`, `enumiii`, `enumiv` を使います。`enumn` は第  $n$  レベルの番号です。

`\theenumi` 出力する番号の書式を設定します。これらは L<sup>A</sup>T<sub>E</sub>X 本体 (`ltlists.dtx` 参照) で定義済み  
`\theenumii` 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  
`\theenumiii` 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  
`\theenumiv` 力する命令です。

```
2315 \renewcommand{\theenumi}{\@arabic\c@enumi}
2316 \renewcommand{\theenumii}{\@alph\c@enumii}
2317 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2318 \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  
`\labelenumii` 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  
`\labelenumiii` 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。

---

```
2319 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2320 \newcommand*\jsInJaParen[1]{%
```

```

2321 \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2322 \newcommand{\labelenumi}{\theenumi.}
2323 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
2324 \newcommand{\labelenumiii}{\theenumiii.}
2325 \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 2326 \renewcommand{\p@enumii}{\theenumi}
2327 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
2328 \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 2329 \newcommand\labelitemi{\textbullet}
\labelitemiii 2330 \newcommand\labelitemii{\normalfont\bfseries \textendash}
2331 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2332 \newcommand\labelitemiv{\textperiodcentered}

```

### ■description 環境

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

```

2333 \newenvironment{description}{%
2334 \list{}{%
2335 \labelwidth=\leftmargin
2336 \labelsep=1\jsZw
2337 \advance \labelwidth by -\labelsep
2338 \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。

```

2339 \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。

```

2340 %<*book|report>
2341 \newenvironment{chapterabstract}{%
2342   \begin{list}{}{%
2343     \listparindent=1\jsZw
2344     \itemindent=\listparindent
2345     \rightmargin=\z@
2346     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2347 %</book|report>

```

“普通の” abstract 環境の定義。

```

2348 %<*article|report|slide>
2349 \newbox\@abstractbox
2350 \if@titlepage
2351   \newenvironment{abstract}{%
2352     \titlepage
2353     \null\vfil
2354     \@beginparpenalty\@lowpenalty
2355     \begin{center}%
2356       \headfont \abstractname
2357       \@endparpenalty\@M
2358     \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。

```

2359   \par}%
2360   {\par\vfil\null\endtitlepage}
2361 \else
2362   \newenvironment{abstract}{%
2363     \if@twocolumn
2364       \ifx\maketitle\relax
2365         \section*{\abstractname}%
2366       \else
2367         \global\setbox\@abstractbox\hbox\bgroup
2368         \begin{minipage}[b]{\textwidth}
2369           \small\parindent1\jsZw
2370           \begin{center}%
2371             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2372           \end{center}%
2373         \list{}{%
2374           \listparindent\parindent
2375           \itemindent \listparindent
2376           \rightmargin \leftmargin}%
2377         \item\relax
2378       \fi
2379 \else

```

```

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

```

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

```

2408 %<*book|report>
2409 \ifbxjs@force@chapterabstract
2410 \let\abstract\chapterabstract
2411 \let\endabstract\endchapterabstract
2412 \fi
2413 %</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。

```

2414 %<*jspf>
2415 %\newbox\@keywordsbox
2416 %\newenvironment{keywords}{%
2417 % \global\setbox\@keywordsbox\hbox\bgroup
2418 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2419 % \small\parindent0\jsZw}%

```

```
2420 % {\end{minipage}\egroup}
2421 %</jspf>
```

#### ■verse 環境

verse 詩のための verse 環境です。

```
2422 \newenvironment{verse}{%
2423   \let \=\@centercr
2424   \list{}{%
2425     \itemsep \z@
2426     \itemindent -2\jsZw % 元: -1.5em
2427     \listparindent\itemindent
2428     \rightmargin \z@
2429     \advance\leftmargin 2\jsZw}% 元: 1.5em
2430   \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。

```
2431 \newenvironment{quotation}{%
2432   \list{}{%
2433     \listparindent\parindent
2434     \itemindent\listparindent
2435     \rightmargin \z@}%
2436   \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。

```
2437 \newenvironment{quote}%
2438   {\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。

```
2439 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2440   \item[\hspace \labelsep{\headfont #1\ #2}]}
2441 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2442   \item[\hspace \labelsep{\headfont #1\ #2 (#3) }]}

```

`titlepage` タイトルを独立のページに出力するのに使われます。

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

```
2443 \newenvironment{titlepage}{%
2444 %<book> \pltx@cleartooddpage %% 2017-02-24
2445 \if@twocolumn
2446 \@restonecoltrue\onecolumn
2447 \else
2448 \@restonecolfalse\newpage
2449 \fi
2450 \thispagestyle{empty}%
2451 \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2452 24
2453 }%
2454 {\if@restonecol\twocolumn \else \newpage \fi
2455 \if@twoside\else
2456 \setcounter{page}\@ne
2457 \fi}
```

## ■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2457 %<!*book&!report>
2458 \newcommand{\appendix}{\par
2459 \setcounter{section}{0}%
2460 \setcounter{subsection}{0}%
2461 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2462 \gdef\presectionname{\appendixname}%
2463 \gdef\postsectionname{}}%
2464 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2465 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2466 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2467 \else
2468 \gdef\@secapp{\appendixname}%
2469 \gdef\@secpos{}}%
2470 \gdef\thesection{\@Alph\c@section}%
2471 \fi}
2472 %</!*book&!report>
2473 %<!*book|report>
2474 \newcommand{\appendix}{\par
2475 \setcounter{chapter}{0}%
2476 \setcounter{section}{0}%
2477 \gdef\@chapapp{\appendixname}%
```

```

2478 \gdef\chappos{}%
2479 \gdef\thechapter{\@Alph\c@chapter}}
2480 %</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。

```
2481 \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。

```
2482 \setlength\tabcolsep{6\p@?}
```

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

```
2483 \setlength\arrayrulewidth{.4\p@}
```

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

```
2484 \setlength\doublerulesep{2\p@}
```

### ■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2485 \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。

```
2486 \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。

```
2487 \setlength\fboxsep{3\p@?}
```

```
2488 \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。

```
2489 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```
2490 %<*book|report>
```

```
2491 \@addtoreset{equation}{chapter}
```

```
2492 \renewcommand\theequation
```

```
2493 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
```

```
2494 %</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。

```
2495 % \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。

```
2496 % \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。

```
2497 % \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。

```
2498 %<!*book&!report>
2499 \newcounter{figure}
2500 \renewcommand \thefigure {\@arabic\c@figure}
2501 %</!*book&!report>
2502 %<*book|report>
2503 \newcounter{figure}[chapter]
2504 \renewcommand \thefigure
2505     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2506 %</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 2507 \def\fps@figure{tbp}
2508 \def\ftype@figure{1}
\fnum@figure 2509 \def\ext@figure{lof}
2510 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` \* 形式は段抜きフロートです。

```

figure* 2511 \newenvironment{figure}%
2512         {\@float{figure}}%
2513         {\end@float}
2514 \newenvironment{figure*}%
2515         {\@dblfloat{figure}}%
2516         {\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。

```

2517 %<!*book&!report>
2518 \newcounter{table}
2519 \renewcommand\thetable{\@arabic\c@table}
2520 %<!/book&!report>
2521 %<*book|report>
2522 \newcounter{table}[chapter]
2523 \renewcommand \thetable
2524     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2525 %</book|report>

```

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

```

\ext@table 2526 \def\fps@table{tbp}
2527 \def\ftype@table{2}
\fnun@table 2528 \def\ext@table{lot}
2529 \def\fnun@table{\tablename\nobreak\thetable}

```

`table` \* は段抜きフロートです。

```

table* 2530 \newenvironment{table}%
2531         {\@float{table}}%
2532         {\end@float}
2533 \newenvironment{table*}%
2534         {\@dblfloat{table}}%
2535         {\end@dblfloat}

```

## 8.6 キャプション

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

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

```

2536 \newlength\abovecaptionskip

```

```

2537 \newlength\belowcaptionskip
2538 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2539 \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。

```

2540 %<*\jspf>
2541 % \long\def\@makecaption#1#2{\small
2542 % \advance\leftskip 10\jsc@mmm
2543 % \advance\rightskip 10\jsc@mmm
2544 % \vskip\abovecaptionskip
2545 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2546 % \ifdim \wd\@tempboxa >\hsize
2547 % #1\hskip1\jsZw\relax #2\par
2548 % \else
2549 % \global \@minipagefalse
2550 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2551 % \fi
2552 % \vskip\belowcaptionskip}}
2553 \long\def\@makecaption#1#2{\small
2554 \advance\leftskip .0628\linewidth
2555 \advance\rightskip .0628\linewidth
2556 \vskip\abovecaptionskip
2557 \sbox\@tempboxa{#1\zwspace#2}%
2558 \ifdim \wd\@tempboxa <\hsize \centering \fi
2559 #1\zwspace#2\par
2560 \vskip\belowcaptionskip}}
2561 %</\jspf>
2562 %<*\jspf>
2563 \long\def\@makecaption#1#2{%
2564 \vskip\abovecaptionskip
2565 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2566 \ifdim \wd\@tempboxa >\hsize
2567 {\small\sffamily
2568 \list{#1}{%
2569 \renewcommand{\makelabel}[1]{##1\hfil}
2570 \itemsep \z@
2571 \itemindent \z@
2572 \labelsep \z@
2573 \labelwidth 11\jsc@mmm
2574 \listparindent\z@
2575 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2576 \else
2577 \global \@minipagefalse
2578 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2579 \fi
2580 \vskip\belowcaptionskip}

```

```
2581 %</jspf>
```

## 9 フォントコマンド

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

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

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

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

```
2582 \newif\if@jsc@warnoldfontcmd
2583 \@jsc@warnoldfontcmdtrue
2584 \newif\if@jsc@warnoldfontcmdexception
2585 \@jsc@warnoldfontcmdexceptionfalse
```

```
\jsc@DeclareOldFontCommand
```

```
2586 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2587   \@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2588   \DeclareOldFontCommand{#1}{%
2589     \bxjs@oldfontcmd{#1}#2%
2590   }{%
2591     \bxjs@oldfontcmd{#1}#3%
2592   }%
2593 }
2594 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2595   \ClassInfo\bxjs@clsname
2596   {Old font command '\string#1' is used!!\MessageBreak
2597     The first occurrence is}%
2598 }
```

```
\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。
```

```
2599 \newcommand*{\allowoldfontcommands}{%
2600   \@jsc@warnoldfontcmdexceptiontrue}
2601 \newcommand*{\disallowoldfontcommands}{%
2602   \@jsc@warnoldfontcmdexceptionfalse}
```

```

2603 \let\bxjs@oldfontcmd@list\@empty
2604 \def\bxjs@oldfontcmd#1{%
2605   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2606 \def\bxjs@oldfontcmd@a#1#2{%
2607   \if@jsc@warnoldfontcmdexception\else
2608     \global\@jsc@warnoldfontcmdfalse
2609     \ifx#1\relax
2610       \global\let#1=t%
2611       \jsc@warnoldfontcmd{#2}%
2612     \fi
2613   \fi}
2614 \def\bxjs@warnoldfontcmd@final{%
2615 % \par
2616 \global\let\bxjs@warnoldfontcmd@final\@empty
2617 \let\@tempa\@empty
2618 \def\do##1{%
2619   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2620     \edef\@tempa{\@tempa \space\string##1}\fi}
2621 \bxjs@oldfontcmd@list
2622 \ifx\@tempa\@empty\else
2623   \ClassWarningNoLine\bxjs@clsname
2624   {Some old font commands were used in text:\MessageBreak
2625     \space\@tempa\MessageBreak
2626     You should note, that since 1994 LaTeX2e provides a\MessageBreak
2627     new font selection scheme called NFSS2 with several\MessageBreak
2628     new, combinable font commands. The
2629     class provides\MessageBreak
2630     the old font commands only for compatibility}
2631   \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。

```

2632 \def\bxjs@warnoldfontcmd@kick@final{%
2633   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2634 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

---

`\mc` フォントファミリーを変更します。

```

\gt 2635 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2636 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2637 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2638 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2639 \jsc@DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

```

`\bf` ボールドシリーズにします。通常のミーディアムシリーズに戻すコマンドは `\mdseries` です。

```

2640 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\it フォントシェイプを変えるコマンドです。斜体とスモールキャップスは数式中では何もしま
\sl せん（警告メッセージを出力します）。通常のアップライト体に戻すコマンドは \upshape
\sc です。
2641 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2642 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2643 \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 2644 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2645 \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。

```
2646 \newcommand\@pnumwidth{1.55em}
2647 \newcommand\@tocrmarg{2.55em}
2648 \newcommand\dotsep{4.5}
2649 %<!book&!report>\setcounter{tocdepth}{2}
2650 %<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)

```
2651 \newdimen\jsc@tocl@width
2652 \newcommand{\tableofcontents}{%
2653 %<*book|report>
2654 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2655 \settowidth\@tempdima{\headfont\appendixname}%
2656 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2657 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2658 \if@twocolumn
2659 \@restonecoltrue\onecolumn
2660 \else
2661 \@restonecolfalse
2662 \fi
2663 \chapter*{\contentsname}%
2664 \@mkboth{\contentsname}{}%
2665 %</book|report>
2666 %<*!book&!report>
2667 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2668 \settowidth\@tempdima{\headfont\appendixname}%
2669 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2670 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2671 \section*{\contentsname}%
2672 \@mkboth{\contentsname}{\contentsname}%
2673 %</!book&!report>
2674 \@starttoc{toc}%
2675 %<book|report> \if@restonecol\twocolumn\fi
2676 }
```

`\l@part` 部の目次です。

```
2677 \newcommand*{\l@part}[2]{%
2678 \ifnum \c@tocdepth >-2\relax
2679 %<!book&!report> \addpenalty\@secpenalty
2680 %<book|report> \addpenalty{-\@highpenalty}%
2681 \advspace{2.25em \@plus\p@?}%
```

```

2682 \begingroup
2683 \parindent \z@
2684 % \@pnumwidth should be \@tocrmarg
2685 % \rightskip \@pnumwidth
2686 \rightskip \@tocrmarg
2687 \parfillskip -\rightskip
2688 {\leavevmode
2689 \large \headfont
2690 \setlength\@lnumwidth{4\jsZw}%
2691 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2692 \nobreak
2693 %<book|report> \global\@nobreaktrue
2694 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2695 \endgroup
2696 \fi}

```

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

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

```

2697 %<*book|report>
2698 \newcommand*{\l@chapter}[2]{%
2699 \ifnum \c@tocdepth >\m@ne
2700 \addpenalty{-\@highpenalty}%
2701 \addvspace{1.0em \@plus\p@?}
2702 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2703 \begingroup
2704 \parindent\z@
2705 % \rightskip\@pnumwidth
2706 \rightskip\@tocrmarg
2707 \parfillskip-\rightskip
2708 \leavevmode\headfont
2709 % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2710 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2711 \advance\leftskip\@lnumwidth \hskip-\leftskip
2712 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2713 \penalty\@highpenalty
2714 \endgroup
2715 \fi}
2716 %</book|report>

```

`\l@section` 節の目次です。

```

2717 %<!*book&!report>
2718 \newcommand*{\l@section}[2]{%
2719 \ifnum \c@tocdepth >\z@
2720 \addpenalty{\@secpenalty}%
2721 \addvspace{1.0em \@plus\p@?}%
2722 \begingroup
2723 \parindent\z@
2724 % \rightskip\@pnumwidth

```

```

2725     \rightskip\@tocrmarg
2726     \parfillskip-\rightskip
2727     \leavevmode\headfont
2728 %     % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2729     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2730     \advance\leftskip\@lnumwidth \hskip-\leftskip
2731     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2732     \endgroup
2733     \fi}
2734 %</*!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。

```

2735 %<book|report> % \newcommand*\l@section{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

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

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

```

\l@subparagraph 2736 %</*!book&!report>
2737 % \newcommand*\l@subsection {\@dottedtocline{2}{1.5em}{2.3em}}
2738 % \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
2739 % \newcommand*\l@paragraph {\@dottedtocline{4}{7.0em}{4.1em}}
2740 % \newcommand*\l@subparagraph {\@dottedtocline{5}{10em}{5em}}
2741 %
2742 % \newcommand*\l@subsection {\@dottedtocline{2}{1zw}{3zw}}
2743 % \newcommand*\l@subsubsection{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2744 % \newcommand*\l@paragraph {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2745 % \newcommand*\l@subparagraph {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2746 %
2747 \newcommand*\l@subsection{%
2748     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2749     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2750 \newcommand*\l@subsubsection{%
2751     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2752     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2753 \newcommand*\l@paragraph{%
2754     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2755     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2756 \newcommand*\l@subparagraph{%
2757     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2758     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2759 %</*!book&!report>
2760 %<*book|report>
2761 % \newcommand*\l@subsection {\@dottedtocline{2}{3.8em}{3.2em}}
2762 % \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
2763 % \newcommand*\l@paragraph {\@dottedtocline{4}{10em}{5em}}
2764 % \newcommand*\l@subparagraph {\@dottedtocline{5}{12em}{6em}}
2765 \newcommand*\l@section{%

```

```

2766         \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2767         \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2768 \newcommand*\l@section}{%
2769         \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2770         \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2771 \newcommand*\l@subsubsection}{%
2772         \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2773         \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2774 \newcommand*\l@paragraph}{%
2775         \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2776         \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2777 \newcommand*\l@subparagraph}{%
2778         \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2779         \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2780 %</book|report>

```

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

```

2781 \newdimen\@lnumwidth
2782 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

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

```

2783 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2784   \vskip \z@ \@plus.2\p@?
2785   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2786     \parindent #2\relax\@afterindenttrue
2787     \interlinepenalty\@M
2788     \leavevmode
2789     \@lnumwidth #3\relax
2790     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2791     {#4}\nobreak
2792     \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2793       mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2794       \hfil\normalfont \normalcolor #5}\par}\fi}

```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2795 \newcommand{\listoffigures}{%
2796 %<*book|report>
2797   \if@twocolumn\@restonecoltrue\onecolumn
2798   \else\@restonecolfalse\fi
2799   \chapter*{\listfigurename}%
2800   \@mkboth{\listfigurename}{}%
2801 %</book|report>

```

```

2802 %<!*book&!report>
2803 \section*{\listfigurename}%
2804 \@mkboth{\listfigurename}{\listfigurename}%
2805 %</!book&!report>
2806 \@starttoc{lof}%
2807 %<book|report> \if@restonecol\twocolumn\fi
2808 }

```

`\l@figure` 図目次の項目を出力します。

```
2809 \newcommand*\l@figure{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```

2810 \newcommand{\listoftables}{%
2811 %<*book|report>
2812 \if@twocolumn\@restonecoltrue\onecolumn
2813 \else\@restonecolfalse\fi
2814 \chapter*{\listtablename}%
2815 \@mkboth{\listtablename}{}%
2816 %</book|report>
2817 %<!*book&!report>
2818 \section*{\listtablename}%
2819 \@mkboth{\listtablename}{\listtablename}%
2820 %</!book&!report>
2821 \@starttoc{lot}%
2822 %<book|report> \if@restonecol\twocolumn\fi
2823 }

```

`\l@table` 表目次は図目次と同じです。

```
2824 \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。

```

2825 \newdimen\bibindent
2826 \setlength\bibindent{2\jsZw}

```

`thebibliography` 参考文献リストを出力します。

[2016-07-16] L<sup>A</sup>T<sub>E</sub>X 2.09 で使われていたフォントコマンドの警告を、文献スタイル (.bst) ではよく `\bf` がいまだに用いられることが多いため、`thebibliography` 環境内では例外的に出さないようにしました。

```

2827 \newenvironment{thebibliography}[1]{%
2828 \@jsc@warnoldfontcmdexceptiontrue
2829 \global\let\presectionname\relax
2830 \global\let\postsectionname\relax
2831 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2832 %<*kiyou>
2833 \vspace{1.5\baselineskip}

```

```

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

```

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

```
2856 \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。

```
2857 \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。

```
2858 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

`\cite` 文献の番号を出力する部分は `ltbibl.dtx` で定義されていますが、コンマとカッコを和文

`\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必

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

```

2859 % \def\@citex[#1]#2{\leavevmode
2860 % \let\@citea\@empty
2861 % \@cite{\@for\@citeb:=#2\do
2862 % {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2863 % \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2864 % \if@files\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2865 % \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2866 % \G@refundefinedtrue
2867 % \@latex@warning

```

```

2868 %      {Citation '\@citeb' on page \thepage \space undefined}}%
2869 %      {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2870 % \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。
2871 % \DeclareRobustCommand\cite{\unskip
2872 %   \@ifnextchar [{\@tempwatrue\@citex}{\@tempwafalse\@citex[]}]
2873 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2874 %   , \jsInhibitGlue\ #2\fi}})}$}

```

### 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)。

```

2875 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2876   \if@twocolumn
2877     \onecolumn\@restonecolfalse
2878   \else
2879     \clearpage\@restonecoltrue
2880   \fi
2881   \columnseprule.4pt \columnsep 2\jsZw
2882   \ifx\multicols\@undefined
2883 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}]%
2884 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2885 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2886 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2887   \else
2888     \ifdim\textwidth<\fullwidth
2889       \setlength{\evensidemargin}{\oddsidemargin}
2890       \setlength{\textwidth}{\fullwidth}
2891       \setlength{\linewidth}{\fullwidth}
2892 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}]%
2893 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2894 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2895 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2896   \else
2897 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}]%
2898 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2899 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2900 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2901   \fi
2902   \fi
2903 %<book|report>      \@mkboth{\indexname}{}%
2904 %<!book&!report>    \@mkboth{\indexname}{\indexname}]%
2905   \plainifnotempty % \thispagestyle{plain}
2906   \parindent\z@
2907   \parskip\z@ \@plus .3\p@?\relax

```

```

2908 \let\item\@idxitem
2909 \raggedright
2910 \footnotesize\narrowbaselines
2911 }{
2912 \ifx\multicols\@undefined
2913 \if@restonecol\onecolumn\fi
2914 \else
2915 \end{multicols}
2916 \fi
2917 \clearpage
2918 }

```

`\@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 2919 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2920 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2921 \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。

```
2922 \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。⇒ ( $\rightarrow$ ) などもいいでしょう。

```

2923 \newcommand\seename{\if@english see\else →\fi}
2924 \newcommand\alsoname{\if@english see also\else →\fi}

```

## 10.4 脚注

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

---

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

---

```

2925 \ifx\pltx@foot@penalty\@undefined
2926 \let\footnotes@ve=\footnote
2927 \def\footnote{\jsInhibitGlue\footnotes@ve}
2928 \let\footnotemarks@ve=\footnotemark
2929 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2930 \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 pTeX 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 pLaTeX の変更に従いました (Thanks: 角藤さん)。pLaTeX 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。

---

pTeX 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 `newpertext` 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)。

```
2931 \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。

```
2932 % \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。

```
2933 \renewcommand{\footnoterule}{%
```

```
2934   \kern-2.6\p@? \kern-.4\p@
```

```
2935   \hrule width .4\columnwidth
```

```
2936   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2937 %<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, *TeX and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9)

[2016-08-25] コミュニティ版 pLaTeX 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 pLaTeX のバグ修正に従いました。

[2016-11-29] 古い pLaTeX で使用された場合を考慮してコードを改良。

```
2938 \long\def\@footnotetext{%
```

```
2939   \insert\footins\bgroup
```

```
2940     \normalfont\footnotesize
```

```
2941     \interlinepenalty\interfootnotelinepenalty
```

```
2942     \splittopskip\footnotesep
```

```
2943     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
```

```
2944     \hsize\columnwidth \@parboxrestore
```

```
2945     \protected@edef\@currentlabel{%
```

```

2946     \csname p@footnote\endcsname\@thefnmark
2947   }%
2948   \color@begingroup
2949     \@makefntext{%
2950       \rule\z@\footnotesep\ignorespaces}%
2951     \futurelet\next\fo@t}
2952 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
2953           \else \let\next\fo@t\fi \next}
2954 \def\fo@t{\bgroup\aftergroup\@foot\let\next}
2955 \def\fo@t#1{#1\@foot}
2956 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2957   \ifx\pltx@foot@penalty\undefined\else
2958     \ifhmode\null\fi
2959     \ifnum\pltx@foot@penalty=\z@\else
2960       \penalty\pltx@foot@penalty
2961       \pltx@foot@penalty\z@
2962     \fi
2963   \fi}

```

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

```

2964 \newcommand\@makefntext[1]{%
2965   \advance\leftskip 3\jsZw
2966   \parindent 1\jsZw
2967   \noindent
2968   \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。

```

2969 % \def\@xfootnotenext[#1]{%
2970 %   \begingroup
2971 %     \ifnum#1>\z@
2972 %       \csname c@\mpfn\endcsname #1\relax
2973 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2974 %     \else
2975 %       \unrestored@protected@xdef\@thefnmark{}%
2976 %     \fi
2977 %   \endgroup
2978 %   \@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

段落頭のかぎかっこなどを見かけ 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。

```
2979 \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。

```
2980 \def\everyparhook{\jsInhibitGlueAtParTop}
2981 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2982 \AtBeginDocument{\everypar{\everyparhook}}
2983 \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す使う危険性が高いので、専用の命令 `\jsc@ig@temp` に置き換えました (Issue #54)。

---

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

---

```
2984 \ifx j\jsEngine
2985 \def\@inhibitglue{%
2986   \futurelet\@let@token\@@inhibitglue}
2987 \begingroup
2988 \let\GDEF=\gdef
2989 \let\CATCODE=\catcode
2990 \let\ENDGROUP=\endgroup
2991 \CATCODE`k=12
2992 \CATCODE`a=12
2993 \CATCODE`n=12
2994 \CATCODE`j=12
2995 \CATCODE`i=12
2996 \CATCODE`c=12
2997 \CATCODE`h=12
2998 \CATCODE`r=12
2999 \CATCODE`t=12
```

```

3000 \CATCODE`e=12
3001 \GDEF\KANJI@CHARACTER{kanji character }
3002 \ENDGROUP
3003 \def\@@inhibitglue{%
3004   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@t
3005 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3006   \def\jsc@ig@temp{#1}%
3007   \ifx\jsc@ig@temp\@empty
3008     \ifnum\the\inhibitxspcode`#2=2\relax
3009       \inhibitglue
3010     \fi
3011   \fi}
3012 \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。

---

```

3013 \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。

```

3014 \def\@doendpe{%
3015   \@endpetrue
3016   \def\par{%
3017     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3018   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3019 \def\@setminipage{%
3020   \@minipagetrue
3021   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3022 }

```

`\item` 命令の直後です。

```

3023 \def\@item[#1]{%
3024   \if@noperitem
3025     \donoperitem
3026   \else
3027     \if@inlabel
3028       \indent \par
3029     \fi
3030     \ifhmode
3031       \unskip\unskip \par
3032     \fi
3033     \if@newlist

```

```

3034     \if@nobreak
3035         \@nbitem
3036     \else
3037         \addpenalty\@beginparpenalty
3038         \addvspace\@topsep
3039         \addvspace{-\parskip}%
3040     \fi
3041 \else
3042     \addpenalty\@itempenalty
3043     \addvspace\itemsep
3044 \fi
3045 \global\@inlabeltrue
3046 \fi
3047 \everypar{%
3048     \@minipagefalse
3049     \global\@newlistfalse
3050     \if@inlabel
3051         \global\@inlabelfalse
3052         {\setbox\z@\lastbox
3053         \ifvoid\z@
3054             \kern-\itemindent
3055         \fi}%
3056     \box\@labels
3057     \penalty\z@
3058 \fi
3059 \if@nobreak
3060     \@nobreakfalse
3061     \clubpenalty \@M
3062 \else
3063     \clubpenalty \@clubpenalty
3064     \everypar{\everyparhook}%
3065 \fi
3066 \everyparhook}%
3067 \if@noitemarg
3068     \@noitemargfalse
3069     \if@nmbrlist
3070         \refstepcounter\@listctr
3071     \fi
3072 \fi
3073 \sbox\@tempboxa{\makelabel{#1}}%
3074 \global\setbox\@labels\hbox{%
3075     \unhbox\@labels
3076     \hskip \itemindent
3077     \hskip -\labelwidth
3078     \hskip -\labelsep
3079     \ifdim \wd\@tempboxa >\labelwidth
3080         \box\@tempboxa
3081     \else
3082         \hbox to\labelwidth {\unhbox\@tempboxa}%

```

```

3083 \fi
3084 \hskip \labelsep}%
3085 \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。

```

3086 \def\@afterheading{%
3087 \@nbreaktrue
3088 \everypar{%
3089 \if@nbreak
3090 \@nbreakfalse
3091 \clubpenalty \@M
3092 \if@afterindent \else
3093 {\setbox\z@\lastbox}%
3094 \fi
3095 \else
3096 \clubpenalty \@clubpenalty
3097 \everypar{\everyparhook}%
3098 \fi\everyparhook}}

```

---

「`\everyparhook` 用のパッチ処理」はここまで。

---

```

3099 \fi

```

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

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

---

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

---

```

3100 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3101 \AtEndOfPackage{%
3102 \def\@gnewline #1{%
3103 \ifvmode
3104 \@nolnerr
3105 \else
3106 \unskip \reserved@e {\reserved@f#1}\nbreak \hfil \break \null
3107 \jsInhibitGlue \ignorespaces
3108 \fi}
3109 }
3110 \fi

```

## 12 いろいろなロゴ

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

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

---

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

※`\小`、`\上小` の制御綴は定義しない。

---

```
3111 \if@jslogo
3112   \IfFileExists{jslogo.sty}{%
3113     \RequirePackage{jslogo}%
3114   }{%
3115     \ClassWarningNoLine\bxjs@clsname
3116     {The package 'jslogo' is not installed.\MessageBreak
3117     It is included in the recent release of\MessageBreak
3118     the 'jsclasses' bundle}
3119   }
3120 \fi
```

## 13 amsmath との衝突の回避

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

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

[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。

```
3121 %\let\ltx@ifnextchar\@ifnextchar
3122 %\def\ProvidesFile#1{%
3123 %   \begingroup
3124 %     \catcode`\ 10 %
3125 %     \ifnum \endlinechar<256 %
3126 %       \ifnum \endlinechar>\m@ne
3127 %         \catcode\endlinechar 10 %
3128 %       \fi
3129 %     \fi
3130 %   \@makeother\%
```

```

3131 % \@makeother\&%
3132 % \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 3133 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3134 \newcommand{\postpartname}{\if@english\else 部\fi}
3135 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3136 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3137 \newcommand{\presectionname}{}% 第
\postsectionname 3138 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 3139 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3140 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3141 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3142 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3143 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3144 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3145 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3146 %<jspf>\newcommand{\figurename}{Fig.~}
3147 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3148 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3149 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3150 \newcommand{\appendixname}{\if@english \else 付録\fi}
3151 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

---

環境変数 `SOURCE_DATE_EPOCH` / `FORCE_SOURCE_DATE` が設定されている場合は“今日”が過去・未来の日付になる可能性がある。その場合、和暦表記は 1970 年から平成の終わりまでの期間でのみサポートする。

※新元号が公表されたら対応する。

---

```
3152 \@tempwafalse
```

```

3153 \if p\jsEngine \@tempwattrue \fi
3154 \if n\jsEngine \@tempwattrue \fi
3155 \if@tempswa \expandafter\@firstoftwo
3156 \else \expandafter\@secondoftwo
3157 \fi
3158 {%
3159 % 欧文 8bitTeX の場合
3160 \newif\ifjsSeireki \jsSeirekitrue
3161 \def\bxjs@decl@Seireki@cmds{%
3162 \def\西暦{\jsSeirekitrue}%
3163 \def\和暦{\jsSeirekifalse}}
3164 \def\Seireki{\jsSeirekitrue}
3165 \def\Wareki{\jsSeirekifalse}
3166 \def\bxjs@if@use@seireki{%
3167 \ifjsSeireki \expandafter\@firstoftwo
3168 \else \expandafter\@secondoftwo \fi}
3169 }{%
3170 \newif\if 西暦 \西暦 true
3171 \def\bxjs@decl@Seireki@cmds{%
3172 \def\西暦{\西暦 true}%
3173 \def\和暦{\西暦 false}}
3174 \def\Seireki{\西暦 true}
3175 \def\Wareki{\西暦 false}
3176 \def\bxjs@if@use@seireki{%
3177 \if 西暦 \expandafter\@firstoftwo
3178 \else \expandafter\@secondoftwo \fi}
3179 }
3180 \bxjs@decl@Seireki@cmds
3181 \let\bxjs@unxp\@firstofone \let\bxjs@onxp\@firstofone
3182 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded
3183 \def\bxjs@onxp{\unexpanded\expandafter}}
3184 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3185 \def\bxjs@iai{\noexpand~}
3186 \else \def\bxjs@iai{}
3187 \fi

```

`\jyear` 和暦における年の表記（元号 + 年数）。

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

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

```

3188 \newcount\bxjs@jyear
3189 \let\bxjs@gengo\@empty \bxjs@jyear\year
3190 \@tempcnta=\the\year\two@digits\month\two@digits\day\relax
3191 \ifnum\@tempcnta<19700101
3192 \else\ifnum\@tempcnta<19890108
3193 \def\bxjs@gengo{昭和}\advance\bxjs@jyear-1925\relax
3194 \let\heisei\bxjs@jyear
3195 \else\ifnum\@tempcnta<20190501
3196 \def\bxjs@gengo{平成}\advance\bxjs@jyear-1988\relax

```

```

3197 \let\heisei\bxjs@jyear
3198 \else
3199 % 新元号対応
3200 % \def\bxjs@gengo{新元号}\advance\bxjs@jyear-2018\relax
3201 \fi\fi\fi
3202 \edef\jyear{%
3203 \ifx\bxjs@gengo\@empty \the\bxjs@jyear\bxjs@iai
3204 \else \bxjs@onxp{\bxjs@gengo}%
3205 \ifnum\bxjs@jyear=\@ne \bxjs@unxp{元}%
3206 \else \bxjs@iai\the\bxjs@jyear\bxjs@iai
3207 \fi
3208 \fi}

```

\today 英語、西暦、和暦で場合分けをする。

```

3209 \edef\bxjs@today{%
3210 \if@english
3211 \ifcase\month\or
3212 January\or February\or March\or April\or May\or June\or
3213 July\or August\or September\or October\or November\or December\fi
3214 \space\number\day, \number\year
3215 \else
3216 \noexpand\bxjs@if@use@seireki{%
3217 \number\year\bxjs@iai\bxjs@unxp{年}%
3218 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3219 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3220 }{%
3221 \bxjs@onxp{\jyear}\bxjs@unxp{年}%
3222 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3223 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3224 }%
3225 \fi}
3226 \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。

```

3227 \AtBeginDocument{%
3228 \ifx\bb1@jpn@Seirekitrue\@undefined\else
3229 \bxjs@decl@Seireki@cmds
3230 \g@addto@macro\datejapanese{%
3231 \let\today\bxjs@today}%
3232 \fi}

```

---

■ハイフネーション例外 T<sub>E</sub>X のハイフネーションルールの補足です (ペンディング: eng-lish)

```

3233 \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。

```
3234 %<slide>\pagestyle{empty}%
3235 %<article|report>\pagestyle{plain}%
3236 %<book>\pagestyle{headings}%
3237 \pagenumbering{arabic}
3238 \if@twocolumn
3239   \twocolumn
3240   \sloppy
3241   \flushbottom
3242 \else
3243   \onecolumn
3244   \raggedbottom
3245 \fi
3246 %<*slide>
3247   \renewcommand\familydefault{\sfdefault}
3248   \raggedright
3249 %</slide>
```

■BXJS 独自の追加処理 

---

和文ドライバのファイルを読み込む。

```
3250 \catcode\?=12
3251 \ifx\bxjs@jadriver\relax\else
3252 \input{bxjsja-\bxjs@jadriver.def}
3253 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
3254 \bxjs@restore@jltrcc
3255 %</cls>
```

---

以上です。

## 付録 A 和文ドライバの仕様

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

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

用いられる「和文フォント」のメトリックに基づくのではなく、機械的に `\jsScale` × (フォントサイズ) であると定められている (フォントサイズ変更の度に再設定される)。従って、「和文コンポーネント」はこの設定と辻褃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。

```
3256 %<*drv>
```

## 付録 B 和文ドライバ : minimal

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

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

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

### B.1 補助マクロ

```
3257 %<*minimal>
```

```
3258 %% 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.
```

```
3259 \def\DeclareJaTextFontCommand#1#2{%
```

```
3260   \DeclareRobustCommand#1[1]{%
```

```
3261     \relax
```

```
3262     \ifmmode \expandafter\nfss@text \fi
```

```
3263     {#2##1}}%
```

```
3264 }
```

```
\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。
```

```
3265 \def\DeclareJaMathFontCommand#1#2{%
```

```
3266   \DeclareRobustCommand#1[1]{%
```

```
3267     \relax
```

```
3268     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3269     \nfss@text{\fontfamily\familydefault
```

```
3270       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3271       #2##1}}%
```

```

3272 }%
3273 }

```

`\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。

```

3274 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3275 \@onlypreamble\bxjs@if@sf@default
3276 \def\bxjs@if@sf@default#1{%
3277   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3278   \AtBeginDocument{%
3279     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3280 }

```

`\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。

```

3281 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3282 \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。

```

3283 \def\jsLetHeadChar#1#2{%
3284   \begingroup
3285     \escapechar=`\ %
3286     \let\bxjs@tmpa={% brace-match-hack
3287       \bxjs@let@hchar@exp#2}%
3288   \endgroup
3289   \let#1\bxjs@g@tmpa}
3290 \def\bxjs@let@hchar@exp{%
3291   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3292 \def\bxjs@let@hchar@exp@a{%
3293   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3294   \bxjs@let@hchar@out\let\relax
3295   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3296   \bxjs@let@hchar@out\let\space%
3297   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3298   \bxjs@let@hchar@out\let\@backslashchar
3299   }{\bxjs@let@hchar@exp@b}}
3300 \def\bxjs@let@hchar@exp@b#1{%
3301   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3302 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3303   %\message{<#1#2>}%
3304   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3305 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3306 \bxjs@let@hchar@out\let\relax
3307 }{%else
3308 \expandafter\bxjs@let@hchar@exp
3309 }%
3310 }{%else
3311 \bxjs@let@hchar@chr#1%
3312 }}
3313 \def\bxjs@let@hchar@chr#1{%
3314 \bxjs@let@hchar@out\def{#{1}}}}
3315 \def\bxjs@let@hchar@out#1#2{%
3316 \global#1\bxjs@g@tmpa#2\relax
3317 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3318 \chardef\bxjs@let@hchar@csta=128
3319 \chardef\bxjs@let@hchar@cstb=192
3320 \chardef\bxjs@let@hchar@cstc=224
3321 \chardef\bxjs@let@hchar@cstd=240
3322 \chardef\bxjs@let@hchar@cste=248
3323 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3324 \def\bxjs@let@hchar@chr@ue#1{%
3325 \@tempcnta=#1\relax
3326 %\message{\the\@tempcnta}%
3327 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3328 \bxjs@let@hchar@chr@ue@a#1%
3329 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3330 \bxjs@let@hchar@out\let\relax
3331 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3332 \bxjs@let@hchar@chr@ue@b
3333 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3334 \bxjs@let@hchar@chr@ue@c
3335 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3336 \bxjs@let@hchar@chr@ue@d
3337 }{%else
3338 \bxjs@let@hchar@out\let\relax
3339 }}}}}
3340 \def\bxjs@let@hchar@chr@ue@a#1{%
3341 \bxjs@let@hchar@out\def{#{1}}}}
3342 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3343 \bxjs@let@hchar@out\def{#{1#2}}}}
3344 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3345 \bxjs@let@hchar@out\def{#{1#2#3}}}}
3346 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3347 \bxjs@let@hchar@out\def{#{1#2#3#4}}}}

```

## B.2 (u)pTeX 用の設定

```
3348 \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。

```
3349 \def\bxjs@let@hchar@chr@pp#1#2{%
3350   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3351 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3352   \%message{(\meaning#3:\meaning#4)}%
3353   \bxjs@cond\if#1k\fi{%
3354     \bxjs@let@hchar@out\def{#4}}%
3355   }{%else
3356     \bxjs@let@hchar@chr@ue#3#4%
3357   }%
3358 \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。

```
3359 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3360 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3361 \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。

```
3362 \@onlypreamble\bxjs@declarefontshape
3363 \ifjsWithupTeX
3364 \def\bxjs@declarefontshape{%
3365   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3366   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3367   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3368   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3369 }
3370 \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。

```
3371 \else
3372 \def\bxjs@declarefontshape{%
3373   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3374   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3375   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{%
```

```

3376 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3377 }
3378 \def\bxjs@sizereference{jis}
3379 \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。

```

3380 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
3381   \def\bxjs@y{#5}}
3382 \ifjswithptexng \def\bxjs@y{10}%
3383 \else
3384 \expandafter\expandafter\expandafter\bxjs@tmpa
3385 \expandafter\string\the\jfont\relax
3386 \fi
3387 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3388               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3389   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
3390    \expandafter\let\csname\bxjs@x/\bxjs@y\endcsname=\@undefined}

```

■和文フォントスケールの補正 実には、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。

```

3391 \begingroup
3392 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3393 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3394 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3395 % 幅が丁度 10pt なら補正は不要
3396 \ifdim\wd\z@=10pt
3397   \global\let\bxjs@scale\jsScale
3398 \else
3399 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3400 \edef\bxjs@tmpa{\strip@pt\wd\z@}
3401 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3402 \bxjs@invscale\@tempdima\bxjs@tmpa
3403 \xdef\bxjs@scale{\strip@pt\@tempdima}
3404 \fi
3405 \endgroup
3406 %\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。

```

3407 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3408 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{}
3409 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{}
3410 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{}

```

```

3411 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{}
3412 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{}
3413 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{}
3414 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{}
3415 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{}
3416 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{}
3417 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{}
3418 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{}
3419 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{}
3420 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{}
3421 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{}

```

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

```

3422 \DeclareRobustCommand\rmfamily
3423   {\not@math@alphabet\rmfamily\mathrm}
3424   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3425 \DeclareRobustCommand\sffamily
3426   {\not@math@alphabet\sffamily\mathsf}
3427   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3428 \DeclareRobustCommand\ttfamily
3429   {\not@math@alphabet\ttfamily\mathtt}
3430   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3431 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3432 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3433 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3434 \fi
3435 \bxjs@if@sf@default{%
3436   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```
3437 \selectfont
```

`\bxjs@parse@qh` の処理は不要になるので無効化する。

```

3438 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3439 \let\bxjs@parse@qh@a\undefined
3440 \let\bxjs@parse@qh@b\undefined

```

#### ■パラメタの設定

```

3441 \prebreakpenalty\jis"2147=10000
3442 \postbreakpenalty\jis"2148=10000
3443 \prebreakpenalty\jis"2149=10000
3444 \inhibitxspcode`!=1
3445 \inhibitxspcode`〒=2
3446 \xspcode`+=3
3447 \xspcode`%=3

```

"80~"FF の範囲の `\xspcode` を 3 に変更。

```

3448 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3449   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```
3450 \let\jsInhibitGlueAtParTop\@inhibitglue
```

`\jsResetDimen` は空のままでよい。

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

```
3451 \begingroup
```

```
3452 \catcode\!=0
```

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

```
3453 \gdef\bxjs@ptex@dir{%
```

```
3454 !iftkdir t%
```

```
3455 !else!ifydir y%
```

```
3456 !else ?%
```

```
3457 !fi!fi}
```

新版の `pTeX` で脚注番号の周囲の空きが過大になる現象への対処。

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

```
3458 % 古い \@makefnmark の定義
```

```
3459 \long\def\bxjs@tmpa{\hbox{%
```

```
3460 !ifydir \@textsuperscript{\normalfont\@thefnmark}%
```

```
3461 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
```

```
3462 \ifx\@makefnmark\bxjs@tmpa
```

```
3463 \long\gdef\@makefnmark{%
```

```
3464 !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
```

```
3465 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
```

```
3466 \fi
```

```
3467 \endgroup
```

### B.3 pdfTeX 用の処理

```
3468 \else\ifx p\jsEngine
```

```
3469 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

```
3470 \@onlypreamble\bxjs@cjk@loaded
```

```
3471 \def\bxjs@cjk@loaded{%
```

```
3472 \def\@footnotemark{%
```

```
3473 \leavevmode
```

```
3474 \ifhmode
```

```
3475 \edef\@x@sf{\the\spacefactor}%
```

```
3476 \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
```

```
3477 \unkern\unkern
```

```
3478 \ifdim\lastskip>\z@ \unskip \fi
```

```
3479 \fi\fi
```

```
3480 \nobreak
```

```

3481 \fi
3482 \@makefnmark
3483 \ifhmode \spacefactor\@x@sf \fi
3484 \relax}%
3485 \let\bxjs@CJK@loaded\relax
3486 }
3487 \AtBeginDocument{%
3488 \ifpackageloaded{CJK}{%
3489 \bxjs@CJK@loaded
3490 }{}%
3491 }

```

## B.4 XeTeX 用の処理

```
3492 \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。

```

3493 \def\bxjs@let@hchar@chr#1{%
3494 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3495 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3496 \bxjs@let@hchar@chr@xe
3497 }{\bxjs@let@hchar@out\def{#1}}}}
3498 \def\bxjs@let@hchar@chr@xe#1{%
3499 \lccode`0=`#1\relax
3500 \lowercase{\bxjs@let@hchar@out\def{0}}}}

```

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

```

3501 \ifx\XeTeXgenerateactualtext\@undefined\else
3502 \def\bxjs@do@precisetext{%
3503 \XeTeXgenerateactualtext=\@ne}
3504 \fi

```

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

```

3505 \@onlypreamble\bxjs@do@simplejasetup
3506 \def\bxjs@do@simplejasetup{%
3507 \ifnum\XeTeXinterchartokenstate>\z@
3508 \else\ifnum\stricmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3509 \jsSimpleJaSetup
3510 \ClassInfo\bxjs@clsname
3511 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3512 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```

3513 \newcommand*\jsSimpleJaSetup{%
3514 \XeTeXlinebreaklocale "ja"\relax
3515 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3516 \XeTeXlinebreakpenalty=0\relax}

```

## B.5 後処理 (エンジン共通)

```
3517 \fi\fi\fi
      simplejasetup オプションの処理。
3518 \ifx\bxjs@do@simplejasetup\undefined\else
3519   \AtBeginDocument{%
3520     \ifbxjs@simplejasetup
3521       \bxjs@do@simplejasetup
3522     \fi}
3523 \fi

      precisetext オプションの処理。
3524 \ifbxjs@precisetext
3525   \ifx\bxjs@do@precisetext\undefined
3526     \ClassWarning\bxjs@clsname
3527       {The current engine does not supprt the\MessageBreak
3528       'precisetext' option\@gobble}
3529   \else
3530     \bxjs@do@precisetext
3531   \fi
3532 \fi
```

### ■段落頭でのグルー挿入禁止

`\bxjs@check@everyparhook` 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。

```
3533 \@onlypreamble\bxjs@check@everyparhook
3534 \def\bxjs@check@everyparhook{%
3535   \ifx\jsInhibitGlueAtParTop\@empty
3536     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3537     \ifx\everyparhook\bxjs@tmpa
3538       \let\everyparhook\@empty
3539     \fi
3540   \fi}
3541 \AtBeginDocument{\bxjs@check@everyparhook}
```

`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。

```
3542 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern
```

まず `\everypar` を“乗っ取る” 処理を行う。

```
3543 \let\bxjs@everypar\everypar
3544 \newtoks\everypar
3545 \everypar\bxjs@everypar
```

そして本物の `\everypar` では、最後に常に `\everyparhook` が実行されるようにする。

```
3546 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3547 \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。

3548 \ifbxjs@fancyhdr

\bxjs@adjust@fancyhdr fancyhdr の初期設定に関する変更の処理。fancyhdr 読込完了と \pagestyle{fancy} 実行の間で実行されるべき。

3549 \@onlypreamble\bxjs@adjust@fancyhdr

3550 \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。

3551 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%

3552 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%

3553 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3554 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3555 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3556 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3557 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%

3558 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%

3559 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3560 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3561 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3562 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3563 \def\bxjs@tmpa{\rm\thepage\strut}%

3564 \def\bxjs@tmpb{\thepage\strut}%

3565 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi

3566 \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。

3567 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth

3568 \setlength{\@tempdima}{\fullwidth-\textwidth}%

3569 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%

3570 }\bxjs@tmpa

3571 \fi\fi

3572 \PackageInfo\bxjs@clsname

3573 {Patch to fancyhdr is applied\@gobble}}

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

3574 \def\bxjs@pagestyle@hook{%

3575 \@ifpackageloaded{fancyhdr}{%

3576 \bxjs@adjust@fancyhdr

3577 \global\let\bxjs@adjust@fancyhdr\relax

3578 }{}}

`\pagestyle` にフックを入れ込む。

```
3579 \let\bxjs@org@pagestyle\pagestyle
3580 \def\pagestyle{%
3581   \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。

```
3582 \AtBeginDocument{%
3583   \bxjs@pagestyle@hook
3584   \global\let\bxjs@pagestyle@hook\relax}
```

### ■和文空白命令

```
3585 \ifbxjs@jaspace@cmd
```

`\jaenspace` 半角幅の水平空き。

```
3586 \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。

```
3587 \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。

```
3588 \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3589 \DeclareRobustCommand*{\jaspace}[1]{%
3590   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3591   \ClassError\bxjs@clsname
3592   {Unknown jaspac: #1}{\@eha}%
3593   \else
3594   \csname bxjs@jaspace@@#1\endcsname
3595   \fi}
3596 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3597 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3598 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
```

```
3599 \fi
```

終わり。

```
3600 \fi
```

以上で終わり。

```
3601 %</minimal>
```

## 付録 C 和文ドライバ : standard

標準のドライバ。

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

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

## C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
3602 %<*standard>
3603 %% このファイルは日本語文字を含みます
3604 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3605 \bxjs@simplejasetupfalse
```

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

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

```
3606 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。

3607 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3608 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3609 \define@key{bxjsStd}{jis2004}[true]{%
3610 \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。

```
3611 \newif\ifbxjs@jp@units

units オプションの処理。

3612 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3613 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3614 \define@key{bxjsStd}{units}[true]{%
3615 \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。

```
3616 \let\bxjs@jp@font\@empty
```

font オプションの処理。

```
3617 \define@key{bxjsStd}{font}{%
```

```
3618 \edef\bxjs@jp@font{#1}}
```

実際の `japaram` の値を適用する。

```
3619 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
```

```
3620 \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。

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

```
3622 \g@addto@macro\@classoptionslist{,jis2004}
```

```
3623 % \ifpackagewith 判定への対策
```

```
3624 \PassOptionsToPackage{jis2004}{otf}
```

```
3625 \fi
```

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

```
3626 \if j\jsEngine
```

```
3627 \bxjs@jp@unitsfalse
```

```
3628 \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。

```
3629 \ifbxjs@jp@units
```

```
3630 \IfFileExists{bxcalc.sty}{%
```

```
3631 \RequirePackage{bxcalc}[2018/01/28]%v1.0a
```

```
3632 \ifx\usepTeXunits\undefined
```

```
3633 \PackageWarningNoLine\bxjs@clsname
```

```
3634 {Cannot support pTeX units (zw etc.), since\MessageBreak
```

```
3635 the package 'bxcalc' is too old}%
```

```
3636 \bxjs@jp@unitsfalse
```

```
3637 \else \usepTeXunits
```

```
3638 \fi
```

```
3639 }{%else
```

```
3640 \PackageWarningNoLine\bxjs@clsname
```

```
3641 {Cannot support pTeX units (zw etc.), since\MessageBreak
```

```
3642 the package 'bxcalc' is unavailable}%
```

```
3643 \bxjs@jp@unitsfalse
```

```
3644 }
```

```
3645 \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。

```
3646 \ifbxjs@jp@units
```

```

3647 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3648 \let\bxjs@parse@qh@a\@undefined
3649 \let\bxjs@parse@qh@b\@undefined
3650 \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。

```

3651 \ifbxjs@jp@units
3652   \def\bxjs@let@lenexpr#1#2{%
3653     \edef#1{#2}%
3654     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3655 \else
3656   \def\bxjs@let@lenexpr{\edef}
3657 \fi

```

■共通命令の実装 `\jQ` 等の「単位」系の共通命令を実装する。まず  $\epsilon$ -TeX 拡張が使えるか検査する。

```
3658 \ifjsWithTeX
```

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

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

```

\jH 3659   \@tempdima=0.25mm
3660   \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3661   \let\jH\jQ

```

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

```

\trueH 3662   \ifjsc@mag
3663     \@tempdimb=\jsBaseFontSize\relax
3664     \edef\bxjs@tpa{\strip@pt\@tempdimb}%
3665     \@tempdima=2.5mm
3666     \bxjs@invscale\@tempdima\bxjs@tpa
3667     \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3668     \@tempdima=10pt
3669     \bxjs@invscale\@tempdima\bxjs@tpa
3670     \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3671   \else \let\trueQ\jQ \let\bxjs@truept\p@
3672   \fi
3673   \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。

```

3674   \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3675   \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3676   \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3677   \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}

```

3678 \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。

```
3679 \def\jafontsize#1#2{%
3680   \begingroup
3681     \bxjs@jafontsize@a{#1}%
3682     \@tempdimb\jsInverseScale\@tempdima
3683     \bxjs@jafontsize@a{#2}%
3684     \xdef\bxjs@g@tmpa{%
3685       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3686   \endgroup\bxjs@g@tmpa}
3687 \def\bxjs@jafontsize@a#1{%
3688   \bxjs@parse@qh{#1}%
3689   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3690   \@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。

```
3691 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3692 \newcommand*\setkanjiskip[1]{%
3693   \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3694   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3695 \newcommand*\getkanjiskip{%
3696   \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。

```
3697 \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 3698 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3699   \bxjs@kanjiskip@enabledtrue
3700   \bxjs@reset@kanjiskip}
3701 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3702   \bxjs@kanjiskip@enabledfalse
3703   \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。

```
3704 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3705   \ifbxjs@kanjiskip@enabled
3706     \setlength{\@tempkipa}{\bxjs@kanjiskip}%
3707   \else \@tempkipa\z@
```

```

3708 \fi
3709 \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 3710 \def\bxjs@xkanjiskip{Opt}
\getxkanjiskip 3711 \newcommand*\setxkanjiskip[1]{%
\ifbxjs@xkanjiskip@enabled 3712 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
3713 \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3714 \newcommand*\getxkanjiskip{%
\bxjs@disable@xkanjiskip 3715 \bxjs@xkanjiskip}
3716 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3717 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3718 \bxjs@xkanjiskip@enabledtrue
3719 \bxjs@reset@xkanjiskip}
3720 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3721 \bxjs@xkanjiskip@enabledfalse
3722 \bxjs@reset@xkanjiskip}
3723 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3724 \ifbxjs@xkanjiskip@enabled
3725 \setlength{\@tempkipa}{\bxjs@xkanjiskip}%
3726 \else \@tempkipa\z@
3727 \fi
3728 \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。

```

3729 \g@addto@macro\jsResetDimen{%
3730 \bxjs@reset@kanjiskip
3731 \bxjs@reset@xkanjiskip}
3732 \let\bxjs@apply@kanjiskip\relax
3733 \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。

```

3734 \@onlypreamble\bxjs@adjust@jafont
3735 \def\bxjs@adjust@jafont#1{%
3736 \ifx\jsJaFont\bxjs@auto
3737 \bxjs@get@kanjiEmbed
3738 \ifx\bxjs@kanjiEmbed\relax
3739 \let\bxjs@tmpa\@empty
3740 \else
3741 \let\bxjs@tmpa\bxjs@kanjiEmbed
3742 \fi
3743 \else
3744 \let\bxjs@tmpa\jsJaFont

```

```

3745 \fi
3746 \if f#1\ifx\bxjs@tmpa\bxjs@noEmbed
3747 \ClassWarningNoLine\bxjs@clsname
3748 {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3749 not available on the current situation}%
3750 \let\bxjs@tmpa\@empty
3751 \fi\fi
3752 }
3753 \def\bxjs@auto{auto}
3754 \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。

```
3755 \let\bxjs@kanjiEmbed\relax
```

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

```

3756 \@onlypreamble\bxjs@get@kanjiEmbed
3757 \def\bxjs@get@kanjiEmbed{%
3758 \begingroup\setbox\z@=\hbox{%
3759 \global\let\bxjs@g@tmpa\relax
3760 \endlinechar\m@ne
3761 \let\do\@makeother\dospecials
3762 \catcode32=10 \catcode12=10 %form-feed
3763 \let\bxjs@tmpa\@empty
3764 \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3765 \ifeof\@inputcheck\else
3766 \read\@inputcheck to\bxjs@tmpa
3767 \closein\@inputcheck
3768 \fi
3769 \ifx\bxjs@tmpa\@empty\else
3770 \openin\@inputcheck="\bxjs@tmpa"\relax
3771 \@tempswatrue
3772 \loop\if@tempswa
3773 \read\@inputcheck to\bxjs@tmpa
3774 \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3775 \ifx\bxjs@tmpb\relax\else
3776 \global\let\bxjs@g@tmpa\bxjs@tmpb
3777 \@tempswafalse
3778 \fi
3779 \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3780 \ifx\bxjs@tmpb\relax\else
3781 \global\let\bxjs@g@tmpa\bxjs@tmpb
3782 \@tempswafalse
3783 \fi
3784 \ifeof\@inputcheck \@tempswafalse \fi
3785 \repeat
3786 \fi
3787 }\endgroup
3788 \let\bxjs@kanjiEmbed\bxjs@g@tmpa

```

```

3789 }
3790 \@onlypreamble\bxjs@get@ke@a
3791 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3792   \ifx$#1$\def\bxjs@tmpb{#2}%
3793   \else \let\bxjs@tmpb\relax
3794   \fi}
3795 \@onlypreamble\bxjs@get@ke@b
3796 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3797   \ifx$#1$\def\bxjs@tmpb{#2}%
3798   \else \let\bxjs@tmpb\relax
3799   \fi}

```

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

```

3800 \newcommand*\jachar[1]{%
3801   \begingroup
3802     \jsLetHeadChar\bxjs@tmpa{#1}%
3803     \ifx\bxjs@tmpa\relax
3804       \ClassWarningNoLine\bxjs@clsname
3805         {Illegal argument given to \string\jachar}%
3806     \else
3807       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3808     \fi
3809   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```
3810 \let\jsJaChar\jachar
```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
3811 \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。

```
3812 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```

3813 \@onlypreamble\bxjs@fix@hyperref@unicode
3814 \def\bxjs@fix@hyperref@unicode#1{%
3815   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3816   \@namedef{KV@Hyp@bxjs/hook}##1{%
3817     \KV@Hyp@unicode{##1}%
3818     \def\KV@Hyp@unicode###1{%
3819       \expandafter\ifx\csname if###1\expandafter\endcsname
3820         \csname if###1\endcsname\else
3821         \ClassWarningNoLine\bxjs@clsname
3822         {Blcoked hyperref option 'unicode=###1'}%
3823       \fi
3824     }%

```

```
3825 }%
3826 }
```

`\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。

```
3827 \@onlypreamble\jsCheckHyperrefUnicode
3828 \let\jsCheckHyperrefUnicode\@empty
3829 \AtBeginDocument{\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。

```
3830 \@onlypreamble\bxjs@check@hyperref@unicode
3831 \def\bxjs@check@hyperref@unicode#1{%
3832   \g@addto@macro\jsCheckHyperrefUnicode{%
3833     \@tempswafalse
3834     \begingroup
3835       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3836         \aftergroup\@tempswatrue \fi
3837       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3838         \csname if#1\endcsname
3839         \aftergroup\@tempswatrue \fi
3840     \endgroup
3841     \if@tempswa\else
3842       \ClassError\bxjs@clsname
3843         {The value of hyperref 'unicode' key is not suitable\MessageBreak
3844          for the present engine (must be #1)}%
3845       {\@ehc}%
3846     \fi}}
```

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

```
3847 \@onlypreamble\bxjs@urgent@special
3848 \def\bxjs@urgent@special#1{%
3849   \AtBeginDvi{\special{#1}}%
3850   \AtBeginDocument{%
3851     \@ifpackageloaded{atbegshi}{%
3852       \begingroup
3853         \toks\z@{\special{#1}}%
3854         \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3855         \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3856       \endgroup
3857     }{}%
3858   }%
3859 }
```

`\bxjs@resolve@jafont@paren` jafont パラメタ値内の ( ) を解決する。

```
3860 \def\bxjs@resolve@jafont@paren#1{%
3861   \def\bxjs@tmpb{\let#1}%
3862   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3863 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3864   \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3865   \else
```

```

3866 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3867 \bxjs@tmpb\bxjs@tmpa
3868 \fi}

```

## C.2 pTeX 用設定

```
3869 \if j\jsEngine
```

### ■ 共通命令の実装

```

3870 \def\bxjs@apply@kanjiskip{%
3871 \kanjiskip\@tempkipa}
3872 \def\bxjs@apply@xkanjiskip{%
3873 \xkanjiskip\@tempkipa}

```

\jaJaChar のサブマクロ。

```

3874 \def\bxjs@jachar#1{%
3875 \bxjs@jachar@a#1...\@nil}
3876 \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。

```
3877 \ifx.#2#1%
```

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

```

3878 \else\ifx.#3%
3879 \@tempcnta`#1 \multiply\@tempcnta64
3880 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3881 \bxjs@jachar@b
3882 \else\ifx.#4%
3883 \@tempcnta`#1 \multiply\@tempcnta64
3884 \advance\@tempcnta`#2 \multiply\@tempcnta64
3885 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3886 \bxjs@jachar@b
3887 \else
3888 \@tempcnta`#1 \multiply\@tempcnta64
3889 \advance\@tempcnta`#2 \multiply\@tempcnta64
3890 \advance\@tempcnta`#3 \multiply\@tempcnta64
3891 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3892 \bxjs@jachar@b
3893 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3894 \ifjsWithupTeX
3895 \def\bxjs@jachar@b{\kchar\@tempcnta}
3896 \else
3897 \def\bxjs@jachar@b{%
3898 \ifx\bxUInt\@undefined\else
3899 \bxUInt{\@tempcnta}%
3900 \fi}
3901 \fi

```

和欧文間空白の命令 `\jathinspace` の実装。

```
3902 \ifbxjs@jaspace@cmd
3903   \def\jathinspace{\hskip\xkanjiskip}
3904 \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。

```
3905 \ifbxjs@jp@jismmiv
3906   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3907   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3908 \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。

```
3909 \let\bxjs@jafont@paren\@firstofone
3910 \let\bxjs@tmpa\jsJaFont
3911 \ifx\bxjs@tmpa\bxjs@@auto
3912   \let\bxjs@tmpa\@empty
3913 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3914   \def\bxjs@tmpa{noembed}
3915 \fi\fi
3916 \bxjs@resolve@jafont@paren\bxjs@tmpa
3917 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3918 \ifx\bxjs@tmpa\@empty\else
3919   \edef\bxjs@next{%
3920     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3921   }\bxjs@next
3922 \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……。)

```
3923 \begingroup
3924   \global\let\@gtempa\relax
3925   \catcode`\|=0 \catcode`\|=12
3926   |def|bxjs@check#1|@nil{%
3927     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3928   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3929     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3930   |catcode`\|=0 \catcode`\|=12
3931   \def\bxjs@check@b#1keyval#2\@nnil{%
3932     \ifx$#2$\else
```

```

3933     \xdef\@gtempa{%
3934         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3935     \fi}
3936 \@firstofone{%
3937     \catcode10=12 \endlinechar\m@ne
3938     \let\do\@makeother \dospecials \catcode32=10
3939     \openin\@inputcheck=otf.sty\relax
3940     \@tempwatrue
3941     \loop\if@tempswa
3942         \ifeof\@inputcheck \@tempswafalse \fi
3943         \if@tempswa
3944             \read\@inputcheck to\bxjs@line
3945             \expandafter\bxjs@check\bxjs@line\@nil
3946         \fi
3947     \repeat
3948     \closein\@inputcheck
3949 \endgroup}
3950 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3951 \ifbxjs@hyperref@enc
3952     \bxjs@check@hyperref@unicode{false}
3953 \fi

    tounicode special 命令を出力する。
3954 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3955     \else\ifjsWithpTeXng T\else F\fi\fi T%
3956     \ifnum\jis"2121="A1A1 %euc
3957         \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3958     \else\ifnum\jis"2121="8140 %sjis
3959         \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3960     \else\ifnum\jis"2121="3000 %uptex
3961         \ifbxjs@bigcode
3962             \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3963             \PassOptionsToPackage{bigcode}{pxjahyper}
3964         \else
3965             \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3966         \fi
3967     \fi\fi\fi
3968     \let\bxToUnicodeSpecialDone=t
3969 \fi

```

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

```

3970 \ifx f\bxjs@enablejfam\else
3971     \@enablejfamtrue
3972 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

3973 \if@enablejfam
3974 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3975 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3976 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3977 \jfam\symmincho
3978 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3979 \AtBeginDocument{%
3980 \ifx\reDeclareMathAlphabet\undefined\else
3981 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3982 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3983 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3984 \fi}
3985 \fi

```

### C.3 pdfTeX 用設定 : CJK + bxcjkatype

```
3986 \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。

```

3987 \bxjs@adjust@jafont{f}
3988 \let\bxjs@jafont@paren\@firstofone
3989 \bxjs@resolve@jafont@paren\bxjs@tmpa
3990 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3991 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
3992 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
3993 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
3994 \fi
3995 \edef\bxjs@next{%
3996 \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
3997 }\bxjs@next
3998 \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。

```

3999 \ifbxjs@hyperref@enc
4000 \PassOptionsToPackage{unicode}{hyperref}
4001 \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。

```

4002 \ifx\bxckjatypeHyperrefPatchDone\@undefined
4003 \begingroup
4004 \CJK@input{UTF8.bdg}
4005 \endgroup
4006 \g@addto@macro\pdfstringdefPreHook{%
4007 \@nameuse{CJK@UTF8Binding}%
4008 }
4009 \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。

```

4010 \ifx\bxckjatypeHyperrefPatchDone\@undefined
4011 \g@addto@macro\pdfstringdefPreHook{%
4012 \ifx-\bxjs@@CJKtilde
4013 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4014 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4015 \let-\@empty
4016 \fi
4017 }
4018 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4019 \def\bxjs@@tildecmd{~}
4020 \def\bxjs@LetUnexpandableSpace#1{%
4021 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4022 \bxjs@org@LetUnexpandableSpace#1%
4023 \fi}
4024 \fi

```

#### ■ 共通命令の実装

```

4025 \newskip\jsKanjiSkip
4026 \newskip\jsXKanjiSkip
4027 \ifx\CJKecglue\@undefined
4028 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4029 \fi
4030 \let\autospacing\bxjs@enable@kanjiskip
4031 \let\noautospacing\bxjs@disable@kanjiskip
4032 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4033 \def\bxjs@apply@kanjiskip{%
4034 \jsKanjiSkip\@tempkipa
4035 \let\CJKglue\bxjs@CJKglue}
4036 \let\autoxspacing\bxjs@enable@xkanjiskip
4037 \let\noautoxspacing\bxjs@disable@xkanjiskip
4038 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4039 \def\bxjs@apply@xkanjiskip{%
4040 \jsXKanjiSkip\@tempkipa
4041 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4042 \def\bxjs@jachar#1{%
4043 \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4044 \ifbxjs@jaspace@cmd
4045   \protected\def\jathinspace{\CJKecglue}
4046 \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。

```

4047 \ifx t\bxjs@enablejfam
4048   \ClassWarningNoLine\bxjs@clsname
4049   {You cannot use 'enablejfam=true', since the\MessageBreak
4050    CJK package does not support Japanese math}
4051 \fi

```

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

```

4052 \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。

```

4053 \RequirePackage{zxjatype}
4054 \PassOptionsToPackage{no-math}{fontspec}%!
4055 \PassOptionsToPackage{xetex}{graphicx}%!
4056 \PassOptionsToPackage{xetex}{graphics}%!
4057 \ifx\zxJaFamilyName\@undefined
4058   \ClassError\bxjs@clsname
4059   {xeCJK or zxjatype is too old}\@ehc
4060 \fi

```

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

```

4061 \bxjs@adjust@jafont{f}
4062 \let\bxjs@jafont@paren\@gobble
4063 \bxjs@resolve@jafont@paren\bxjs@tmpa
4064 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4065 \ifx\bxjs@tmpa\@empty
4066   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
4067   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
4068 \else
4069   \edef\bxjs@next{%
4070     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4071   }\bxjs@next
4072 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X<sub>Y</sub>TEX の場合は、`xdvipdfmx` が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は `special` 命令の文字列の文字コード変換は不要である。ところが、`hyperref` での方針としては、X<sub>Y</sub>TEX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、`unicode` を無効にしていると警告が出て強制的に有効化される。一方で、過

去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「Xe<sub>L</sub>TeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```
4073 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4074   \ifbxjs@hyperref@enc
4075     \PassOptionsToPackage{unicode}{hyperref}
4076   \fi
4077 \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。

```
4078 \AtEndOfPackage{%
4079 \def\@gnewline #1{%
4080   \ifvmode \@nolnerr
4081   \else
4082     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4083     \nobreak \hskip-1sp\hskip1sp\relax
4084     \ignorespaces
4085   \fi}
4086 }
```

#### ■共通命令の実装

```
4087 \newskip\jsKanjiSkip
4088 \newskip\jsXKanjiSkip
4089 \ifx\CJKecglue\@undefined
4090   \def\CJKtilde{\CJKglobal\def~{\CJKecglue\ignorespaces}}
4091 \fi
4092 \let\autospacing\bxjs@enable@kanjiskip
4093 \let\noautospacing\bxjs@disable@kanjiskip
4094 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4095 \def\bxjs@apply@kanjiskip{%
4096   \jsKanjiSkip\@tempskipa
4097   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4098 \let\autoxspacing\bxjs@enable@xkanjiskip
4099 \let\noautoxspacing\bxjs@disable@xkanjiskip
4100 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4101 \def\bxjs@apply@xkanjiskip{%
4102   \jsXKanjiSkip\@tempskipa
4103   \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。

```
4104 \ifx\mcfamily\@undefined
4105   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4106   \protected\def\gtfamily{\CJKfamily{\JKsfdefault}}
4107 \fi
```

`\jachar` のサブマクロの実装。

```
4108 \def\bxjs@jachar#1{%
4109   \xeCJKDeclareCharClass{CJK}{`#1}\relax
4110   #1}
```

`\jathinspace` の実装。

```
4111 \ifbxjs@jaspace@cmd
4112   \protected\def\jathinspace{\CJKecglue}
4113 \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。

```
4114 \ifx t\bxjs@enablejfam
4115   \@enablejfamtrue
4116 \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。

```
4117 \if@enablejfam
4118   \xeCJKsetup{CJKmath=true}
4119 \fi
```

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

```
4120 \else\if l\jsEngine
```

■Lua<sub>T</sub>E<sub>X</sub>-ja パッケージの読み込み `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。

```
4121 \let\zw\@undefined
4122 \RequirePackage{luatexja}
4123 \edef\bxjs@next{%
4124   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4125 }\bxjs@next
```

■和文フォント定義 `\jsJaFont` が指定された場合は、その値をオプションとして `luatexja-preset` を読み込む。非指定の場合は、`luatexja-preset` パッケージの `ipaex`

オプション (IPAex フォント使用) と等価な設定を用いる (luatexja-preset は読み込まない)。

```
4126 \bxjs@adjust@jafont{t}
4127 \ifx\bxjs@tmpa\bxjs@noEmbed
4128   \def\bxjs@tmpa{noembed}
4129 \fi
4130 \let\bxjs@jafont@paren@gobble
4131 \bxjs@resolve@jafont@paren\bxjs@tmpa
4132 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4133 \ifx\bxjs@tmpa@empty
4134   \defaultfontfeatures{ Kerning=Off }
4135   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
4136   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
4137 \else
4138   \edef\bxjs@next{%
4139     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4140   }\bxjs@next
4141 \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。

```
4142 \DeclareRobustCommand\rmfamily
4143   {\not@math@alphabet\rmfamily\mathrm}
4144   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4145 \DeclareRobustCommand\sffamily
4146   {\not@math@alphabet\sffamily\mathsf}
4147   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4148 \DeclareRobustCommand\ttfamily
4149   {\not@math@alphabet\ttfamily\mathtt}
4150   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4151 \AtBeginDocument{%
4152   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4153   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4154   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4155 \bxjs@if@sf@default{%
4156   \renewcommand\kanjifamilydefault{\gtdefault}}
```

#### ■和文パラメタの設定

```
4157 % 次の3つは既定値の通り
4158 %\ltjsetparameter{prebreakpenalty={`',10000}}
4159 %\ltjsetparameter{postbreakpenalty={`",10000}}
4160 %\ltjsetparameter{prebreakpenalty={`",10000}}
4161 \ltjsetparameter{jaxspmode={`!,1}}
4162 \ltjsetparameter{jaxspmode={`〒,2}}
4163 \ltjsetparameter{alxspmode={`+,3}}
4164 \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 Lua<sub>T</sub><sub>E</sub>X-ja で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合に備えて同等のものを用意する。

```
4165 \ifx\ltjfakeparbegin\@undefined
4166   \protected\def\ltjfakeparbegin{%
4167     \ifhmode
4168       \relax\directlua{%
4169         luatexja.jfmglue.create_beginpar_node()}}
4170   \fi}
4171 \fi
```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```
4172 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4173 \begingroup
4174   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4175   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}
4176   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4177     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}\fi
4178   \directlua{
4179     local function patchcmd(cs, code, from, to)
4180       tex.sprint(code:gsub(from:gsub("%W", "%%%0"), "%0"..to)
4181         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4182     end
4183     patchcmd(\@xsect, [[\meaning\xsect]],
4184       \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4185     patchcmd(\@item, [[\meaning\@item]],
4186       \bxjs@tmpa, \@{\ltjfakeparbegin})}
4187 \endgroup
4188 \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。

```
4189 \ifbxjs@hyperref@enc
4190   \PassOptionsToPackage{unicode}{hyperref}
4191   \bxjs@check@hyperref@unicode{true}
4192 \fi
```

■共通命令の実装

```
4193 \protected\def\autospacing{%
4194   \ltjsetparameter{autospacing=true}}
4195 \protected\def\noautospacing{%
4196   \ltjsetparameter{autospacing=false}}
4197 \protected\def\autoxspacing{%
4198   \ltjsetparameter{autoxspacing=true}}
4199 \protected\def\noautoxspacing{%
4200   \ltjsetparameter{autoxspacing=false}}
4201 \def\bxjs@apply@kanjiskip{%
```

```

4202 \ltjsetparameter{kanjiskip={\@tempskipa}}
4203 \def\bxjs@apply@xkanjiskip{%
4204 \ltjsetparameter{xkanjiskip={\@tempskipa}}
      \jachar のサブマクロの実装。
4205 \def\bxjs@jachar#1{%
4206 \ltjjachar`#1\relax}
      \jathinspace の実装。
4207 \ifbxjs@jaspace@cmd
4208 \protected\def\jathinspace{%
4209 \hskip\ltjgetparameter{xkanjiskip}\relax}
4210 \fi

```

■和文数式ファミリー LuaTeX-ja では和文数式ファミリーは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablejfam は常に真になる。

```

4211 \ifx f\bxjs@enablejfam
4212 \ClassWarningNoLine\bxjs@clsname
4213 {You cannot use 'enablejfam=false', since the\MessageBreak
4214 LuaTeX-ja always provides Japanese math families}
4215 \fi

```

## C.6 共通処理 (2)

```
4216 \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 4217 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4218 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4219 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4220 \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。

```

4221 \ifx\mathmc\undefined
4222 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4223 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4224 \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)。

```

4225 \ifbxjs@jaspace@cmd
4226 \bxjs@protected\def\bxjs@choice@jathinspace{%
4227 \relax\ifmmode \mskip\medmuskip

```

```

4228 \else \jathinspace\ignorespaces
4229 \fi}
4230 \jsAtEndOfClass{%
4231 \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4232 \else \def\>{\protect\bxjs@choice@jathinspace}%
4233 \fi}
4234 \fi

```

#### ■和文・和欧文間空白の初期値

```

4235 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4236 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4237 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4238 \fi

```

以上で終わり。

```
4239 %</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。

```

4240 %<*modern>
4241 \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。

```

4242 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4243 \def\encodingdefault{T1}%
4244 \input{t1enc.def}%
4245 \fontencoding\encodingdefault\selectfont
4246 \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。

```

4247 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4248 \renewcommand{\rmdefault}{lmr}
4249 \renewcommand{\sfdefault}{lmss}
4250 \renewcommand{\ttdefault}{lmtt}
4251 \fi

```

大型演算子用の数式フォントの設定。

※ `amsmath` パッケージと同等にする。

```
4252 \DeclareFontShape{OMX}{cmex}{m}{n}{%
```

```

4253 <-7.5>cmex7<7.5-8.5>cmex8%
4254 <8.5-9.5>cmex9<9.5->cmex10}{}%
4255 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
    amsmath 読込時に上書きされるのを防ぐ。
4256 \def\cmex@opt{10}

```

## D.2 fixltx2e 読込

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

```

4257 \ifx\@IncludeInRelease\@undefined
4258 \RequirePackage{fixltx2e}
4259 \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。

```

4260 \RequirePackage{bxjscjkcat}

```

## D.4 完了

おしまい。

```

4261 %</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。

```

4262 %<*pandoc>
4263 \input{bxjsja-standard.def}
4264 \RequirePackage{bxjspandoc}

```

### E.1 dupload システム

パッケージが重複して読み込まれたときに “option clash” の検査をスキップする。この時に何らかのコードを実行させることができる。

```

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

```

```

4265 \@onlypreamble\bxjs@set@dupload@proc
4266 \def\bxjs@set@dupload@proc#1{%
4267 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@d1p/#1\endcsname}

```

```

4268 \@onlypreamble\bxjs@set@dupload@proc@a
4269 \def\bxjs@set@dupload@proc@a#1{%
4270   \@onlypreamble#1\def#1##1}

```

\@if@options \@if@options の再定義。

```

4271 \@onlypreamble\bxjs@org@if@options
4272 \let\bxjs@org@if@options\@if@options
4273 \newif\ifbxjs@dlp
4274 \def\@if@options#1#2#3{%
4275   \bxjs@dlpfalse
4276   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4277   \ifx\bxjs@tmpa\bxjs@tmpb
4278     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4279       \bxjs@dlptrue \fi
4280   \fi
4281   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
4282   \else \expandafter\bxjs@org@if@options
4283   \fi {#1}{#2}{#3}}
4284 \AtBeginDocument{%
4285   \let\@if@options\bxjs@org@if@options}
4286 \@onlypreamble\bxjs@do@dupload@proc
4287 \def\bxjs@do@dupload@proc#1#2#3{%
4288   \csname bxjs@dlp/#2.#1\endcsname{#3}%
4289   \@firstoftwo}

```

## E.2 lang 変数

lang=ja という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐため、とりあえず両パッケージを無効化しておく。

```

4290 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0

```

Polyglossia について。

```

4291 \pandocSkipLoadPackage{polyglossia}
4292 \bxjs@set@dupload@proc{polyglossia.sty}{%
4293   \ClassWarning\bxjs@clsname
4294   {Loading of polyglossia is blocked}}
4295 \ifx\setmainlanguage\@undefined
4296 \newcommand*\setmainlanguage[2] [] {}
4297 \newcommand*\setotherlanguage[2] [] {%
4298   \ifcat_#2_\else
4299     \expandafter\let\csname #2\endcsname\@empty
4300     \expandafter\let\csname end#2\endcsname\@empty
4301     \expandafter\let\csname text#2\endcsname\@firstofone
4302   \fi}
4303 \newcommand*\setotherlanguages[2] [] {%
4304   \@for\bxjs@tmpa:={#2}\do{%
4305     \setotherlanguage{\bxjs@tmpa}}}

```

```
4306 \fi
4307 \else
```

Babel について。

```
4308 \pandocSkipLoadPackage{babel}
4309 \bxjs@set@dupload@proc{babel.sty}{%
4310   \ClassWarning\bxjs@clsname
4311     {Loading of babel is blocked}}
4312 \let\foreignlanguage\@secondoftwo
4313 \let\otherlanguage\@gobble
4314 \let\endotherlanguage\@empty
4315 \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。

```
4316 \bxjs@set@dupload@proc{geometry.sty}{%
4317   \setpagelayout*{#1}}
```

### E.4 CJKmainfont 変数

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

```
4318 \if 1\jsEngine
4319   \pandocSkipLoadPackage{xCJK}
4320   \providecommand*\setCJKmainfont{\setmainjfont}
4321 \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。

```
4322 \AtBeginDocument{%
4323   \@tempwafalse
```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```
4324   \ifx\oldparagraph\undefined\else
```

```
4325 \tempwatruue
4326 \fi
```

\paragraph が番号付きの場合は、マーク変更が必要である。

```
4327 \ifnum\c@secnumdepth>3
4328 \tempwatruue
4329 \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。

```
4330 \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4331 \let\jsParagraphMark\@empty
4332 \fi\fi}
```

## E.6 全角空白文字

```
4333 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4334 \catcode"3000=\active
4335 \begingroup \catcode`\!=7
4336 \protected\gdef!!!!3000{\zwspace}
4337 \endgroup
4338 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4339 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4340 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4341 \fi\fi
```

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

```
4342 \if j\jsEngine
4343 \bxjs@fix@hyperref@unicode{false}
4344 \else
4345 \bxjs@fix@hyperref@unicode{true}
4346 \fi
```

## E.7 完了

おしまい。

```
4347 %</pandoc>
```

和文ドライバ実装はここまで。

```
4348 %</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。

4349 %<\*anc>

## 付録 G 補助パッケージ : bxjscompat

ムニヤムニヤムニヤ……。

### G.1 準備

4350 %<\*compat>

4351 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4352 \let\bxac@engine=n

4353 \def\bxac@do#1#2{%

4354 \edef\bxac@tmpa{\string#1}%

4355 \edef\bxac@tmpb{\meaning#1}%

4356 \ifx\bxac@tmpa\bxac@tmpb #2\fi}

4357 \bxac@do\XeTeXversion{\let\bxac@engine=x}

4358 \bxac@do\luatexversion{\let\bxac@engine=1}

\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。

4359 \ifx\jsAtEndOfClass\@undefined

4360 \let\bxac@delayed@if@bxjs\@firstofone

4361 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass

4362 \fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

\RevokeOldLuaTeXBehavior 4363 \newif\ifbxac@in@old@behavior

4364 \let\ImposeOldLuaTeXBehavior\relax

4365 \let\RevokeOldLuaTeXBehavior\relax

### G.2 X<sub>Y</sub>TEX 部分

4366 \ifx x\bxac@engine

X<sub>Y</sub>TEX 文字クラスのムニヤムニヤ。

4367 \@onlypreamble\bxac@adjust@charclass

4368 \bxac@delayed@if@bxjs{%

4369 \@ifpackageloaded{xeCJK}{-}{%else

4370 \ifx\xe@alloc@intercharclass\@undefined\else

4371 \ifnum\xe@alloc@intercharclass=\z@

4372 \PackageInfo\bxac@pkgname

4373 {Setting up interchar class for CJK...\@gobble}%

4374 \InputIfFileExists{load-unicode-xetex-classes.tex}{%

4375 \xe@alloc@intercharclass=3

```

4376     }{%else
4377         \PackageWarning\bxac@pkgname
4378             {Cannot find file 'load-unicode-xetex-classes.tex'%
4379             \@gobble}%
4380     }%
4381 \fi\fi
4382 \ifnum\XeTeXcharclass"3041=\z@
4383     \PackageInfo\bxac@pkgname
4384         {Adjusting interchar class for CJK...\@gobble}%
4385     \@for\bxac@x:={%
4386         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4387         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4388         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4389         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4390         31FF%
4391     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4392 \fi
4393 }%
4394 }

```

以上。

```
4395 \fi
```

### G.3 LuaTeX 部分

```

4396 \ifx l\bxac@engine
      ムニヤムニヤ。
4397 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4398     \chardef\pdftexversion=200
4399     \def\pdftexrevision{0}
4400     \let\pdftexbanner\luatexbanner
4401 \fi\fi
\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior 4402 \begingroup\expandafter\expandafter\expandafter\endgroup
4403 \expandafter\ifx\csgname outputmode\endcsgname\relax\else
4404 \def\bxac@ob@list{%
4405     \do{\let}\pdfoutput{\outputmode}%
4406     \do{\let}\pdfpagewidth{\pagewidth}%
4407     \do{\let}\pdfpageheight{\pageheight}%
4408     \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4409     \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}%
4410 \def\bxac@ob@do#1#2{\begingroup
4411     \expandafter\bxac@ob@do@a\csgname bxac@\string#2\endcsgname{#1}#2}
4412 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4413     \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4414     \else \let#3#1\relax \let#1\@undefined
4415     \fi}
4416 \protected\def\ImposeOldLuaTeXBehavior{%
4417     \unless\ifbxac@in@old@behavior

```

```

4418 \bxac@in@old@behaviortrue
4419 \let\do\bxac@ob@do \bxac@ob@list
4420 \fi}
4421 \protected\def\RevokeOldLuaTeXBehavior{%
4422 \ifbxac@in@old@behavior
4423 \bxac@in@old@behaviorfalse
4424 \let\do\bxac@ob@do \bxac@ob@list
4425 \fi}
4426 \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。

```

4427 \ifnum\luatexversion>64 \directlua{
4428 local function range(cs, ce, cc, ff)
4429     if ff or not tex.getcatcode(cs) == cc then
4430         local setcc = tex.setcatcode
4431         for c = cs, ce do setcc(c, cc) end
4432     end
4433 end
4434 range(0x3400, 0x4DB5, 11, false)
4435 range(0x4DB5, 0x4DBF, 11, true)
4436 range(0x4E00, 0x9FCC, 11, false)
4437 range(0x9FCD, 0x9FFF, 11, true)
4438 range(0xAC00, 0xD7A3, 11, false)
4439 range(0x20000, 0x2A6D6, 11, false)
4440 range(0x2A6D7, 0x2A6FF, 11, true)
4441 range(0x2A700, 0x2B734, 11, false)
4442 range(0x2B735, 0x2B73F, 11, true)
4443 range(0x2B740, 0x2B81D, 11, false)
4444 range(0x2B81E, 0x2B81F, 11, true)
4445 range(0x2B820, 0x2CEA1, 11, false)
4446 range(0x2CEA2, 0x2FFFD, 11, true)
4447 }\fi

```

以上。

```
4448 \fi
```

## G.4 完了

おしまい。

```
4449 %</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

```

4450 %<*cjkcat>
4451 \def\bxjx@pkgname{bxjscjkcat}
4452 \newcount\bxjx@cnta

```

`\bxjx@engine` エンジンの種別。

```

4453 \let\bxjx@engine=n
4454 \def\bxjx@do#1#2{%
4455   \edef\bxjx@tmpa{\string#1}%
4456   \edef\bxjx@tmpb{\meaning#1}%
4457   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4458 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
4459 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
4460 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
4461 \bxjx@do\pdftexversion{\let\bxjx@engine=p}
4462 \bxjx@do\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。

```

4463 \def\bxjx@do#1#2{%
4464   \if#1\bxjx@engine
4465     \@ifpackageloaded{#2}{\%else
4466       \PackageError\bxjx@pkgname
4467         {Package '#2' must be loaded}%
4468         {Package loading is aborted.\MessageBreak\@ehc}%
4469       \endinput}
4470   \fi}
4471 \bxjx@do{p}{bxcjkatype}
4472 \bxjx@do{x}{xeCJK}
4473 \bxjx@do{l}{luatexja}

```

古い L<sup>A</sup>T<sub>E</sub>X の場合、`\TextOrMath` は `fixltx2e` パッケージで提供される。

```

4474 \ifx\TextOrMath\@undefined
4475   \RequirePackage{fixltx2e}
4476 \fi

```

## H.2 和文カテゴリコードの設定

upL<sup>A</sup>T<sub>E</sub>X の場合、和文カテゴリコードの設定を LuaT<sub>E</sub>X-ja と（ほぼ）等価なものに変更  
する。

※ LuaT<sub>E</sub>X-ja との相違点：A830、A960、1B000。

```

4477 \if u\bxjx@engine
4478   \@for\bxjx@x:={%
4479     0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4480     0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
4481     0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
4482     1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
4483     1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
4484     1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%

```

```

4485 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
4486 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
4487 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
4488 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
4489 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
4490 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4491 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4492 10480,10500,10530,10600,10800,10840,10860,10880,%
4493 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
4494 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
4495 11000,11080,110D0,11100,11150,11180,111E0,11200,%
4496 11280,112B0,11300,11480,11580,11600,11680,11700,%
4497 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
4498 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4499 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
4500 1F030,1FOA0,1F100,1F200,1F300,1F600,1F650,1F680,%
4501 1F700,1F780,1F800,1F900,E0000,F0000,100000%
4502 }\do{\kcatcode"\bxjx@x=15 }
4503 \fi

```

### H.3 ギリシャ・キリル文字の扱い

※ここで「ギリシャ・キリル文字」は Unicode と JIS X 0213 に共通して含まれるもののみを指すことにする。

`\bxjx@grkcyr@list` 対象のギリシャ・キリル文字に関するデータ。

```

4504 \def\bxjx@grkcyr@list{%
4505 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4506 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4507 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4508 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4509 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4510 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4511 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
4512 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4513 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4514 \do{039A}{LGR}{\textKappa}{K}%           % GR. C. L. KAPPA
4515 \do{039B}{LGR}{\textLambda}{\Lambda}%    % GR. C. L. LAMDA
4516 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4517 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
4518 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
4519 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
4520 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
4521 \do{03A1}{LGR}{\textRho}{P}%            % GR. C. L. RHO
4522 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4523 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
4524 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4525 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI

```

4526	$\backslash\text{do}\{03A7\}\{LGR\}\{\text{Chi}\}\{X\}\%$	% GR. C. L. CHI
4527	$\backslash\text{do}\{03A8\}\{LGR\}\{\text{Psi}\}\{\Psi\}\%$	% GR. C. L. PSI
4528	$\backslash\text{do}\{03A9\}\{LGR\}\{\text{Omega}\}\{\Omega\}\%$	% GR. C. L. OMEGA
4529	$\backslash\text{do}\{03B1\}\{LGR\}\{\text{alpha}\}\{\alpha\}\%$	% GR. S. L. ALPHA
4530	$\backslash\text{do}\{03B2\}\{LGR\}\{\text{beta}\}\{\beta\}\%$	% GR. S. L. BETA
4531	$\backslash\text{do}\{03B3\}\{LGR\}\{\text{gamma}\}\{\gamma\}\%$	% GR. S. L. GAMMA
4532	$\backslash\text{do}\{03B4\}\{LGR\}\{\text{delta}\}\{\delta\}\%$	% GR. S. L. DELTA
4533	$\backslash\text{do}\{03B5\}\{LGR\}\{\text{epsilon}\}\{\epsilon\}\%$	% GR. S. L. EPSILON
4534	$\backslash\text{do}\{03B6\}\{LGR\}\{\text{zeta}\}\{\zeta\}\%$	% GR. S. L. ZETA
4535	$\backslash\text{do}\{03B7\}\{LGR\}\{\text{eta}\}\{\eta\}\%$	% GR. S. L. ETA
4536	$\backslash\text{do}\{03B8\}\{LGR\}\{\text{theta}\}\{\theta\}\%$	% GR. S. L. THETA
4537	$\backslash\text{do}\{03B9\}\{LGR\}\{\text{iota}\}\{\iota\}\%$	% GR. S. L. IOTA
4538	$\backslash\text{do}\{03BA\}\{LGR\}\{\text{kappa}\}\{\kappa\}\%$	% GR. S. L. KAPPA
4539	$\backslash\text{do}\{03BB\}\{LGR\}\{\text{lambda}\}\{\lambda\}\%$	% GR. S. L. LAMDA
4540	$\backslash\text{do}\{03BC\}\{LGR\}\{\text{mu}\}\{\mu\}\%$	% GR. S. L. MU
4541	$\backslash\text{do}\{03BD\}\{LGR\}\{\text{nu}\}\{\nu\}\%$	% GR. S. L. NU
4542	$\backslash\text{do}\{03BE\}\{LGR\}\{\text{xi}\}\{\xi\}\%$	% GR. S. L. XI
4543	$\backslash\text{do}\{03BF\}\{LGR\}\{\text{omicron}\}\{\omicron\}\%$	% GR. S. L. OMICRON
4544	$\backslash\text{do}\{03C0\}\{LGR\}\{\text{pi}\}\{\pi\}\%$	% GR. S. L. PI
4545	$\backslash\text{do}\{03C1\}\{LGR\}\{\text{rho}\}\{\rho\}\%$	% GR. S. L. RHO
4546	$\backslash\text{do}\{03C2\}\{LGR\}\{\text{varsigma}\}\{\varsigma\}\%$	% GR. S. L. FINAL SIGMA
4547	$\backslash\text{do}\{03C3\}\{LGR\}\{\text{sigma}\}\{\sigma\}\%$	% GR. S. L. SIGMA
4548	$\backslash\text{do}\{03C4\}\{LGR\}\{\text{tau}\}\{\tau\}\%$	% GR. S. L. TAU
4549	$\backslash\text{do}\{03C5\}\{LGR\}\{\text{upsilon}\}\{\upsilon\}\%$	% GR. S. L. UPSILON
4550	$\backslash\text{do}\{03C6\}\{LGR\}\{\text{phi}\}\{\phi\}\%$	% GR. S. L. PHI
4551	$\backslash\text{do}\{03C7\}\{LGR\}\{\text{chi}\}\{\chi\}\%$	% GR. S. L. CHI
4552	$\backslash\text{do}\{03C8\}\{LGR\}\{\text{psi}\}\{\psi\}\%$	% GR. S. L. PSI
4553	$\backslash\text{do}\{03C9\}\{LGR\}\{\text{omega}\}\{\omega\}\%$	% GR. S. L. OMEGA
4554	$\backslash\text{do}\{0401\}\{T2A\}\{\text{CYRYO}\}\{\}$	% CY. C. L. IO
4555	$\backslash\text{do}\{0410\}\{T2A\}\{\text{CYRA}\}\{\}$	% CY. C. L. A
4556	$\backslash\text{do}\{0411\}\{T2A\}\{\text{CYRB}\}\{\}$	% CY. C. L. BE
4557	$\backslash\text{do}\{0412\}\{T2A\}\{\text{CYRV}\}\{\}$	% CY. C. L. VE
4558	$\backslash\text{do}\{0413\}\{T2A\}\{\text{CYRG}\}\{\}$	% CY. C. L. GHE
4559	$\backslash\text{do}\{0414\}\{T2A\}\{\text{CYRD}\}\{\}$	% CY. C. L. DE
4560	$\backslash\text{do}\{0415\}\{T2A\}\{\text{CYRE}\}\{\}$	% CY. C. L. IE
4561	$\backslash\text{do}\{0416\}\{T2A\}\{\text{CYRZH}\}\{\}$	% CY. C. L. ZHE
4562	$\backslash\text{do}\{0417\}\{T2A\}\{\text{CYRZ}\}\{\}$	% CY. C. L. ZE
4563	$\backslash\text{do}\{0418\}\{T2A\}\{\text{CYRI}\}\{\}$	% CY. C. L. I
4564	$\backslash\text{do}\{0419\}\{T2A\}\{\text{CYRISHRT}\}\{\}$	% CY. C. L. SHORT I
4565	$\backslash\text{do}\{041A\}\{T2A\}\{\text{CYRK}\}\{\}$	% CY. C. L. KA
4566	$\backslash\text{do}\{041B\}\{T2A\}\{\text{CYRL}\}\{\}$	% CY. C. L. EL
4567	$\backslash\text{do}\{041C\}\{T2A\}\{\text{CYRM}\}\{\}$	% CY. C. L. EM
4568	$\backslash\text{do}\{041D\}\{T2A\}\{\text{CYRN}\}\{\}$	% CY. C. L. EN
4569	$\backslash\text{do}\{041E\}\{T2A\}\{\text{CYRO}\}\{\}$	% CY. C. L. O
4570	$\backslash\text{do}\{041F\}\{T2A\}\{\text{CYRP}\}\{\}$	% CY. C. L. PE
4571	$\backslash\text{do}\{0420\}\{T2A\}\{\text{CYRR}\}\{\}$	% CY. C. L. ER
4572	$\backslash\text{do}\{0421\}\{T2A\}\{\text{CYRS}\}\{\}$	% CY. C. L. ES
4573	$\backslash\text{do}\{0422\}\{T2A\}\{\text{CYRT}\}\{\}$	% CY. C. L. TE
4574	$\backslash\text{do}\{0423\}\{T2A\}\{\text{CYRU}\}\{\}$	% CY. C. L. U

4575	\do{0424}{T2A}{\CYRF}{}	% CY. C. L. EF
4576	\do{0425}{T2A}{\CYRH}{}	% CY. C. L. HA
4577	\do{0426}{T2A}{\CYRC}{}	% CY. C. L. TSE
4578	\do{0427}{T2A}{\CYRCH}{}	% CY. C. L. CHE
4579	\do{0428}{T2A}{\CYRSH}{}	% CY. C. L. SHA
4580	\do{0429}{T2A}{\CYRSHCH}{}	% CY. C. L. SHCHA
4581	\do{042A}{T2A}{\CYRHRDSN}{}	% CY. C. L. HARD SIGN
4582	\do{042B}{T2A}{\CYRERY}{}	% CY. C. L. YERU
4583	\do{042C}{T2A}{\CYRSFTSN}{}	% CY. C. L. SOFT SIGN
4584	\do{042D}{T2A}{\CYREREV}{}	% CY. C. L. E
4585	\do{042E}{T2A}{\CYRYU}{}	% CY. C. L. YU
4586	\do{042F}{T2A}{\CYRYA}{}	% CY. C. L. YA
4587	\do{0430}{T2A}{\cyra}{}	% CY. S. L. A
4588	\do{0431}{T2A}{\cyrb}{}	% CY. S. L. BE
4589	\do{0432}{T2A}{\cyrv}{}	% CY. S. L. VE
4590	\do{0433}{T2A}{\cyrg}{}	% CY. S. L. GHE
4591	\do{0434}{T2A}{\cyrd}{}	% CY. S. L. DE
4592	\do{0435}{T2A}{\cyre}{}	% CY. S. L. IE
4593	\do{0436}{T2A}{\cyrzh}{}	% CY. S. L. ZHE
4594	\do{0437}{T2A}{\cyrz}{}	% CY. S. L. ZE
4595	\do{0438}{T2A}{\cyri}{}	% CY. S. L. I
4596	\do{0439}{T2A}{\cyrishrt}{}	% CY. S. L. SHORT I
4597	\do{043A}{T2A}{\cyrk}{}	% CY. S. L. KA
4598	\do{043B}{T2A}{\cyrl}{}	% CY. S. L. EL
4599	\do{043C}{T2A}{\cyrm}{}	% CY. S. L. EM
4600	\do{043D}{T2A}{\cyrn}{}	% CY. S. L. EN
4601	\do{043E}{T2A}{\cyro}{}	% CY. S. L. O
4602	\do{043F}{T2A}{\cyrp}{}	% CY. S. L. PE
4603	\do{0440}{T2A}{\cyrr}{}	% CY. S. L. ER
4604	\do{0441}{T2A}{\cyrs}{}	% CY. S. L. ES
4605	\do{0442}{T2A}{\cyrt}{}	% CY. S. L. TE
4606	\do{0443}{T2A}{\cyru}{}	% CY. S. L. U
4607	\do{0444}{T2A}{\cyrf}{}	% CY. S. L. EF
4608	\do{0445}{T2A}{\cyrh}{}	% CY. S. L. HA
4609	\do{0446}{T2A}{\cyrc}{}	% CY. S. L. TSE
4610	\do{0447}{T2A}{\cyrch}{}	% CY. S. L. CHE
4611	\do{0448}{T2A}{\cyrsh}{}	% CY. S. L. SHA
4612	\do{0449}{T2A}{\cyrshch}{}	% CY. S. L. SHCHA
4613	\do{044A}{T2A}{\cyrhrdsn}{}	% CY. S. L. HARD SIGN
4614	\do{044B}{T2A}{\cyrery}{}	% CY. S. L. YERU
4615	\do{044C}{T2A}{\cyrsftsn}{}	% CY. S. L. SOFT SIGN
4616	\do{044D}{T2A}{\cyrerev}{}	% CY. S. L. E
4617	\do{044E}{T2A}{\cyryu}{}	% CY. S. L. YU
4618	\do{044F}{T2A}{\cyrya}{}	% CY. S. L. YA
4619	\do{0451}{T2A}{\cyryo}{}	% CY. S. L. IO
4620	\do{00A7}{TS1}{\textsection}{\mathsection}%	SECTION SYMBOL
4621	\do{00A8}{TS1}{\textasciidieresis}{}	% DIAERESIS
4622	\do{00B0}{TS1}{\textdegree}{\mathdegree}%	% DEGREE SIGN
4623	\do{00B1}{TS1}{\textpm}{\pm}%	% PLUS-MINUS SIGN

```

4624 \do{00B4}{TS1}{\textasciicute}{}%          % ACUTE ACCENT
4625 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4626 \do{00D7}{TS1}{\texttimes}{\times}%        % MULTIPLICATION SIGN
4627 \do{00F7}{TS1}{\textdiv}{\div}%           % DIVISION SIGN
4628 }

4629 \providecommand*\mathdegree{{}^{\circ}}

```

`\ifbxjx@gcc@CJK` [スイッチ] ギリシャ・キリル文字を和文扱いにするか。

```
4630 \newif\ifbxjx@gcc@CJK
```

`\greekasCJK` ギリシャ・キリル文字を和文扱いにする。

`\nogreekasCJK` ギリシャ・キリル文字を欧文扱いにする。

```

4631 \newcommand*\greekasCJK{%
4632   \bxjx@gcc@CJKtrue}
4633 \newcommand*\nogreekasCJK{%
4634   \bxjx@gcc@CJKfalse}

```

`\bx@fake@grk` `\bx@fake@grk{(出力文字)}{(基準文字)}` :

```

4635 \def\bxjx@do#1\relax{%
4636   \def\bxjx@fake@grk##1##2{%
4637     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4638   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4639     \ifx\##1\%
4640       \bxjx@cnta##4\divide\bxjx@cnta\@cclvi
4641       \multiply\bxjx@cnta\@cclvi \advance\bxjx@cnta`##3\relax
4642       \mathchar\bxjx@cnta
4643     \else ##3\fi}
4644 }\expandafter\bxjx@do\string\mathchar\relax

```

#### ■pdfLaTeX・upLaTeX の場合

```
4645 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0
```

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。

```

4646 \@ifpackageloaded{inputenc}{}{%else
4647   \RequirePackage[utf8]{inputenc}}
4648 \def\bxjx@tmpa{utf8}
4649 \ifx\bxjx@tmpa\inputencdoingname
4650   \PackageWarningNoLine\bxjx@pkgname
4651     {Input encoding changed to utf8}%
4652   \inputencoding{utf8}%
4653 \fi

```

`upLaTeX` の場合は当該の文字を含むブロックをの和文カテゴリコードを変更する。

```

4654 \if u\bxjx@engine
4655 \kcatcode"0370=15
4656 \kcatcode"0400=15
4657 \kcatcode"0500=15
4658 \fi

```

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

```
4659 \def\do#1{%
4660   \@tempcnta="#1\relax
4661   \@tempcntb=\@tempcnta \divide\@tempcntb256
4662   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4663   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4664 \def\bxjx@do@a#1#2#3#4#5{%
4665   \ifx\#5\%
4666     \def\bxjx@tmpa{\@inmathwarn#4}%
4667     \else\ifcat A\@noexpand#5%
4668       \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4669         {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4670     \else \def\bxjx@tmpa{#5}%
4671     \fi\fi
4672     \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4673     \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
4674 \if u\bxjx@engine
4675 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
4676 \def\bxjx@do@b#1#2#3#4#5{%
4677   \kchardef#1=\@tempcnta
4678   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4679   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4680 \else\if p\bxjx@engine
4681 \def\bxjx@do@b#1#2#3#4#5{%
4682   \mathchardef#1=\@tempcnta
4683   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4684   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4685 \fi\fi
4686 \bxjx@grkcyr@list
4687 \let\bxjx@do@a\undefined
4688 \let\bxjx@do@b\undefined
```

`\bxjx@DeclareUnicodeCharacter` `\bxjx@DeclareUnicodeCharacter` を変更して、ギリシャ・キリル文字の場合に再定義を抑制したもの。

```
4689 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4690 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4691 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4692 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4693   \count@="#1\relax \bxjx@cnta\count@ \divide\bxjx@cnta256
4694   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cnta\endcsname\relax
4695     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4696   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4697     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4698   \else
4699     \wlog{ \space\space skipped defining Unicode char U+#1}%
4700   \fi\fi}
```

`\bxjx@ja@or@not` `\bxjx@ja@or@not`

```

4701 \def\bxjx@ja@or@not#1#2#3{%
    \greekasCJK の場合は、無条件に和文用コードを実行する。
4702 \ifbxjx@gcc@CJK #1%
    \nogreekasCJK の場合は、エンコーディングを固定して欧文用のコードを実行するが、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）和文用コードを使う。
4703 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4704 \else \UseTextSymbol{#2}{#3}%
4705 \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。

```

4706 \begingroup
4707 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}
4708 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4709 \noexpand\bxjx@swap@DUC@cmd
4710 \the\toks@
4711 \noexpand\bxjx@swap@DUC@cmd}}
4712 \endgroup\next
4713 \def\bxjx@swap@DUC@cmd{%
4714 \let\bxjx@tmpa\DeclareUnicodeCharacter
4715 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4716 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

#### ■ Xe<sub>La</sub>TeX・Lua<sub>La</sub>TeX の場合

```

4717 \else\ifnum0\if x\bxjx@engine1\fi\if 1\bxjx@engine1\fi>0
    各文字について、math active を設定する。
4718 \def\do#1{%
4719 \bxjx@canta="#1\relax
4720 \begingroup
4721 \lccode`~=\bxjx@canta
4722 \lowercase{\endgroup
4723 \bxjx@do@a{~}}{#1}}
4724 \def\bxjx@do@a#1#2#3#4#5{%
4725 \ifx\#5\\\let\bxjx@tmpa\relax
4726 \else\ifcat A\noexpand#5%
4727 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4728 {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4729 \else \def\bxjx@tmpa{#5}%
4730 \fi\fi
4731 \ifx\bxjx@tmpa\relax\else
4732 \mathcode\bxjx@canta"8000 \let#1\bxjx@tmpa
4733 \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、こちらの設定を有効にする。

```

4734 \mathchardef\bxjx@tmpa="119
4735 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4736 \let\bxjx@do@a\undefined

```

Lua $\TeX$  における  $\backslash(\text{no})\text{greekasCJK}$  の定義。jacharrange の設定を変更する。

```

4737 \if 1\bxjx@engine
4738   \protected\def\greekasCJK{%
4739     \bxjx@gcc@cjctrue
4740     \ltjsetparameter{jacharrange={+2, +8}}
4741   \protected\def\nogreekasCJK{%
4742     \bxjx@gcc@cjcfalse
4743     \ltjsetparameter{jacharrange={-2, -8}}
4744 \fi

```

X $\TeX$  における  $\backslash(\text{no})\text{greekasCJK}$  の定義。

```

4745 \if x\bxjx@engine
4746   \protected\def\greekasCJK{%
4747     \bxjx@gcc@cjctrue
4748     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4749     \bxjx@grkcyr@list}
4750   \protected\def\nogreekasCJK{%
4751     \bxjx@gcc@cjcfalse
4752     \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4753     \bxjx@grkcyr@list}
4754 \fi

```

以上。

```
4755 \fi\fi
```

## H.4 初期設定

ギリシャ・キリル文字を欧文扱いにする。

```
4756 \nogreekasCJK
```

## H.5 完了

おしまい。

```
4757 %</cjkat>
```

## 付録 I 補助パッケージ : bxjspandoc

Pandoc の L $\TeX$  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 $\TeX$  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。

## 1.1 準備

```
4758 %<*ancpandoc>
4759 %% このファイルは日本語文字を含みます.
4760 \def\bxjsp@pkgname{bxjscjkat}
```

`\bxjsp@engine` エンジンの種別。

```
4761 \let\bxjsp@engine=n
4762 \@onlypreamble\bxjsp@do
4763 \def\bxjsp@do#1#2{%
4764   \edef\bxjsp@tmpa{\string#1}%
4765   \edef\bxjsp@tmpb{\meaning#1}%
4766   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
4767 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
4768 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
4769 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
4770 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

## 1.2 パッケージ読込の阻止

`\pandocSkipLoadFile` `\pandocSkipLoadFile{〈ファイル名〉}` : 特定のファイルを (`\@filewithoptions` の処理に関して) 読込済であるとマークする。

```
4771 \newcommand*\pandocSkipLoadFile[1]{%
4772   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4773 \def\bxjsp@skip@load@file@a#1#2{%
4774   \ifx#1\relax
4775     \def#1{2001/01/01}%
4776     \PackageInfo\bxjsp@pkgname
4777       {File '#2' marked as loaded\@gobble}%
4778   \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。

```
4779 \newcommand*\pandocSkipLoadPackage[1]{%
4780   \pandocSkipLoadFile{#1.sty}}
```

## 1.3 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近 (2015 年版以降) の `LATEX` ではこれで警告が出る。これを抑止する。

`LATEX` カーネルが新しい場合は `fixltx2e` を読込済にする。

```
4781 \ifx\@IncludeInRelease\@undefined\else
4782   \pandocSkipLoadPackage{fixltx2e}
4783 \fi
```

## 1.4 cmap パッケージ

エンジンが (u)pL<sup>A</sup>T<sub>E</sub>X 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。)

```
4784 \if j\bxjisp@engine
4785   \pandocSkipLoadPackage{cmap}
4786 \fi
```

## 1.5 microtype パッケージ

警告が多すぎなので消す。

```
4787 \if j\bxjisp@engine \else
4788   \PassOptionsToPackage{verbose=silent}{microtype}
4789 \fi
```

エンジンが (u)pL<sup>A</sup>T<sub>E</sub>X 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。

```
4790 \if j\bxjisp@engine
4791   \pandocSkipLoadPackage{microtype}
4792   \newcommand*\UseMicrotypeSet[2] [] {}
4793 \fi
```

## 1.6 Unicode 文字変換対策

Pandoc で L<sup>A</sup>T<sub>E</sub>X 形式に書き出す場合は、元データ中の一部の Unicode 文字を「L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X では「L<sup>A</sup>T<sub>E</sub>X の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「\ldots{} を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の `\ldots` の実装。非数式でありかつ後続が `{}` の場合は代わりに `…` を実行する。

```
4794 \DeclareRobustCommand{\pandocLdots}{%
4795   \relax\ifmmode \expandafter\bxjisp@org@ldots
4796   \else \expandafter\bxjisp@ldots@a
4797   \fi}
4798 \def\bxjisp@ja@ellipsis{…}
4799 \let\bxjisp@org@ldots\ldots
4800 \def\bxjisp@ldots@a{%
```

```

4801 \futurelet\bxjsp@tok\bxjsp@ldots@b}
4802 \def\bxjsp@ldots@b{%
4803 \ifx\bxjsp@tok\bgroup \expandafter\bxjsp@ldots@c
4804 \else \expandafter\bxjsp@org@ldots
4805 \fi}
4806 \def\bxjsp@ldots@c{%
4807 \afterassignment\bxjsp@ldots@d \let\bxjsp@tok=}
4808 \def\bxjsp@ldots@d{%
4809 \futurelet\bxjsp@tok\bxjsp@ldots@e}
4810 \def\bxjsp@ldots@e{%
4811 \ifx\bxjsp@tok\egroup \expandafter\bxjsp@ldots@f
4812 \else \expandafter\bxjsp@ldots@g
4813 \fi}
4814 \def\bxjsp@ldots@f{%
4815 \bxjsp@ja@ellipsis \let\bxjsp@tok=}
4816 \def\bxjsp@ldots@g{%
4817 \expandafter\bxjsp@org@ldots\expandafter{\romannumeral-`} }

\ldots の実装を置き換える。
4818 \AtBeginDocument{%
4819 \let\bxjsp@org@ldots\ldots
4820 \let\ldots\pandocLdots}

```

## 1.7 PandoLa モジュール

インストール済であれば読み込む。

```

4821 \IfFileExists{bxpandola.sty}{%
4822 \RequirePackage{bxpandola}\relax
4823 \PackageInfo\bxjsp@pkgname
4824 {PandoLa module is loaded\@gobble}
4825 }{}

```

## 1.8 完了

おしまい。

```
4826 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
4827 %</anc>
```