

# 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”)

v1.7 [2017/10/21]

---





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

---

## 目次

1	はじめに	3
2	オプション	7
3	和文フォントの変更	36
4	フォントサイズ	36
5	レイアウト	42
5.1	ページレイアウト . . . . .	43
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	56
7	ページスタイル	57
8	文書のマークアップ	60
8.1	表題 . . . . .	60
8.2	章・節 . . . . .	65
8.3	リスト環境 . . . . .	77
8.4	パラメータの設定 . . . . .	85
8.5	フロート . . . . .	86
8.6	キャプション . . . . .	87
9	フォントコマンド	89

10	相互参照	91
10.1	目次の類 . . . . .	91
10.2	参考文献 . . . . .	96
10.3	索引 . . . . .	98
10.4	脚注 . . . . .	99
11	段落の頭へのグルー挿入禁止	102
12	いろいろなロゴ	106
13	amsmath との衝突の回避	106
14	初期設定	107
付録 A	和文ドライバの仕様 罨	111
付録 B	和文ドライバ : minimal 罨	112
B.1	補助マクロ . . . . .	112
B.2	(u)pTeX 用の設定 . . . . .	114
B.3	pdfTeX 用の処理 . . . . .	118
B.4	X <sub>Y</sub> TeX 用の処理 . . . . .	119
B.5	後処理 (エンジン共通) . . . . .	119
付録 C	和文ドライバ : standard 罨	122
C.1	共通処理 (1) . . . . .	123
C.2	pTeX 用設定 . . . . .	130
C.3	pdfTeX 用設定 : CJK + bxcjkatype . . . . .	133
C.4	X <sub>Y</sub> TeX 用設定 : xeCJK + zxcjatype . . . . .	135
C.5	LuaTeX 用設定 : LuaTeX-ja . . . . .	137
C.6	共通処理 (2) . . . . .	140
付録 D	和文ドライバ : modern 罨	141
D.1	フォント設定 . . . . .	141
D.2	fixltx2e 読込 . . . . .	142
D.3	和文カテゴリコード . . . . .	142
D.4	完了 . . . . .	142
付録 E	和文ドライバ : pandoc 罨	142
E.1	dupload システム . . . . .	143
E.2	lang 変数 . . . . .	143
E.3	geometry 変数 . . . . .	144
E.4	CJKmainfont 変数 . . . . .	144
E.5	paragraph のマーク . . . . .	145

E.6	全角空白文字 . . . . .	145
E.7	完了 . . . . .	146
付録 F	補助パッケージ一覧 	146
付録 G	補助パッケージ : bxjscompat 	146
G.1	準備 . . . . .	146
G.2	X <sub>g</sub> TeX 部分 . . . . .	147
G.3	LuaTeX 部分 . . . . .	147
G.4	完了 . . . . .	149
付録 H	補助パッケージ : bxjscjkat 	149
H.1	準備 . . . . .	149
H.2	和文カテゴリコードの設定 . . . . .	150
H.3	ギリシャ・キリル文字の扱い . . . . .	150
H.4	初期設定 . . . . .	157
H.5	完了 . . . . .	157
付録 I	補助パッケージ : bxjspandoc 	157
I.1	準備 . . . . .	157
I.2	パッケージ読込の阻止 . . . . .	157
I.3	fixltx2e パッケージ . . . . .	158
I.4	cmap パッケージ . . . . .	158
I.5	microtype パッケージ . . . . .	158
I.6	Unicode 文字変換対策 . . . . .	159
I.7	PandoLa モジュール . . . . .	160
I.8	完了 . . . . .	160

## 1 はじめに

---

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

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

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「p<sub>l</sub>A<sub>T</sub>E<sub>X</sub> 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` に基づいて奥村が改変したものです。権利については両者のものに従います。奥村は何の権利も主張しません。

[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。

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 = XYTEX、p =
pdfTEX (含 DVI モード)、l = LuaTEX、J = NTT jTEX、O = Omega 系、n = 以上の何
れでもない。

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

\ifjsWithupTeX 〔スイッチ〕 エンジンが (内部漢字コードが Unicode の) 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\epsilonTeXversion{\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] pdf $\TeX$  / Lua $\TeX$  が PDF モードで動作しているか。
※ Lua $\TeX$  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@cslet \bxjs@cslet{⟨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}

\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。)
79 \def\jsAtEndOfClass{%
80 \expandafter\g@addto@macro\csname\bxjs@clsname.cls-h@k\endcsname}

```

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

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

```

81 \@onlypreamble\bxjs@restore@jltrcc
82 \let\bxjs@restore@jltrcc\@empty
83 \if 1\jsEngine
84 \def\bxjs@change@jltrcc#1{%
85   \xdef\bxjs@restore@jltrcc{%
86     \bxjs@restore@jltrcc
87     \catcode`#1=\the\catcode`#1\relax}%
88   \catcode`#1=11\relax}
89 \@tfor\bxjs@x:=西暦\do
90   {\expandafter\bxjs@change@jltrcc\bxjs@x}
91 \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。

```

92 \bxjs@robust@def\jsInhibitGlue{%
93   \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。

```

94 \if@compatibility
95   \ClassError\bxjs@clsname
96   {Something went chaotic!\MessageBreak
97     (How come '\string\documentstyle' is there?)\MessageBreak
98     I cannot go a single step further...}
99   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
100    then there'll still be hope....}
101   \expandafter\@firstofone
102 \else \expandafter\@gobble
103 \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。

```
104 \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。

```
105 \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。

```
106 %<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。
```

```
107 %<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。

---

```
108 %<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。

---

```
109 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

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

---

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

```
110 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
111 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
112 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
113 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
114 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
115 \DeclareOption{b4paper}{\bxjs@setpaper{{257true mm}{364true mm}}}
116 \DeclareOption{b5paper}{\bxjs@setpaper{{182true mm}{257true mm}}}
117 \DeclareOption{b6paper}{\bxjs@setpaper{{128true mm}{182true mm}}}
118 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
119 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
120 \DeclareOption{b4j}{\bxjs@setpaper{{257true mm}{364true mm}}}
```



```

121 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
122 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
123 \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。

```

124 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
125 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
126 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
127 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
128 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

129 \newif\if@landscape
130 \@landscapefalse
131 \DeclareOption{landscape}{\@landscapetrue}

```

■slide オプション slide を新設しました。

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

```

132 \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。

```

133 %<!slide>\@slidefalse
134 %<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。

```
135 \newcommand{\@ptsize}{0}
136 \def\bxjs@param@basefontsize{10pt}
137 \def\jsUnusualPtSize{-20}
```

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

```
138 \def\bxjs@setbasefontsize#1{%
139   \bxjs@setbasefontlength\@tempdima{#1}%
140   \edef\bxjs@param@basefontsize{\the\@tempdima}%
141   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
142   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
143   \else\ifdim\@tempdima=12pt  \long\def\@ptsize{2}%
144   \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi}
```

`\bxjs@setbasefontlength` `base`、`jbase` で指定される長さ（式）のための特別な `\setlength`。与えられた式が“(実数)Q”の形の場合、Q 単位の長さを代入する（この場合“式”は使えない）。

※クラスオプションのトークン列の中に展開可能なトークンがある場合、 $\text{\LaTeX}$  は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。

```
145 \def\bxjs@setbasefontlength#1#2{%
```

ここで `true` の長さが使われるのは不合理なので、式が“true”を含む場合には警告を出す。

```
146   \bxjs@setbasefontlength@a#2true\@nnil
```

式の末尾が“Q”である時は特別に扱い、それ以外は `\setlength` に移譲する。

```
147   \ifx j\jsEngine \setlength#1{#2}%
148   \else
149     \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
150     \ifx\bxjs@tmpa\relax \setlength#1{#2}%
151     \else \@tempdimc0.25mm #1=\bxjs@tmpa\@tempdimc
152     \fi
153   \fi}
154 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
155   \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
156   \else \def\bxjs@tmpa{#1}%
157   \fi}
158 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
159   \ifx\@nnil#2\@nnil\else
160     \ClassWarningNoLine\bxjs@clsname
161     {You should not use 'true' lengths here}%
162   \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。

```
163 \newif\ifjsc@mag
164 \newif\ifjsc@mag@xreal
165 %\let\jsc@magscale\@undefined
166 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
```

```

167 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
168 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
169 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
170 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
171 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
172 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
173 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
174 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
175 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
176 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
177 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
178 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
179 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
180 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
181 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
182 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
183 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
184 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

185 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
186 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
187 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}

```

---

■ トンボオプション    トンボ (crop marks) を出力します。実際の処理は p<sub>La</sub>T<sub>E</sub>X 2<sub>ε</sub> 本体で行います (plcore.dtx 参照)。オプション `tombow` で日付付きのトンボ、オプション `tombo` で日付なしのトンボを出力します。これらはアスキー版のままです。カウンタ `\hour`, `\minute` は p<sub>La</sub>T<sub>E</sub>X 2<sub>ε</sub> 本体で宣言されています。

---

取りあえず、p<sub>T</sub>E<sub>X</sub> 系の場合に限り、JS クラスのトンボ関連のコードをそのまま活かしておく。正常に動作する保証はない。

---

```

188 \if j\jsEngine
189 \hour\time \divide\hour by 60\relax
190 \@tempcnta\hour \multiply\@tempcnta 60\relax
191 \minute\time \advance\minute-\@tempcnta
192 \DeclareOption{tombow}{%
193   \tombowtrue \tombowdatetrue
194   \setlength{\@tombowwidth}{.1\p}%
195   \@bannertoken{%
196     \jobname\space(\number\year-\two@digits\month-\two@digits\day
197     \space\two@digits\hour:\two@digits\minute)}}%
198   \maketombowbox}
199 \DeclareOption{tombo}{%
200   \tombowtrue \tombowdatefalse
201   \setlength{\@tombowwidth}{.1\p}%

```

```
202 \maketombowbox}
203 \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。

```
204 \if j\jsEngine
205 \DeclareOption{mentuke}{%
206 \tombowtrue \tombowdatefalse
207 \setlength{\@tombowwidth}{\z@}%
208 \maketombowbox}
209 \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。

```
210 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
211 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
212 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
213 \DeclareOption{onecolumn}{\@twocolumnfalse}
214 \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。

```
215 \DeclareOption{titlepage}{\@titlepagetrue}
216 \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` と表すことにしています。 `openany` で偶数ページからでも始まるようになります。

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

```
217 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
218 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
219 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}
```

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

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

```
220 \def\eqnarray{%
221 \stepcounter{equation}%
222 \def\@currentlabel{\p@equation\theequation}%
```

```

223 \global\@eqnswtrue
224 \m@th
225 \global\@eqcnt\z@
226 \tabskip\@centering
227 \let\\\@eqnocr
228 $$\everycr{}\halign to\displaywidth\bgroup
229     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
230     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
231     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
232     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
233     \tabskip\z@skip
234 \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。森本さんにしただって訂正しました。

```

235 \DeclareOption{leqno}{\input{leqno.clo}}
236 \DeclareOption{fleqn}{\input{fleqn.clo}}
237 % fleqn 用の eqnarray 環境の再定義
238 \def\eqnarray{%
239     \stepcounter{equation}%
240     \def\@currentlabel{\p@equation\theequation}%
241     \global\@eqnswtrue\m@th
242     \global\@eqcnt\z@
243     \tabskip\mathindent
244     \let\=\@eqnocr
245     \setlength\abovedisplayskip{\topsep}%
246     \ifvmode
247         \addtolength\abovedisplayskip{\partopsep}%
248     \fi
249     \addtolength\abovedisplayskip{\parskip}%
250     \setlength\belowdisplayskip{\abovedisplayskip}%
251     \setlength\belowdisplayshortskip{\abovedisplayskip}%
252     \setlength\abovedisplayshortskip{\abovedisplayskip}%
253     $$\everycr{}\halign to\linewidth% $$
254     \bgroup
255     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
256     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
257     &\global\@eqcnt\tw@
258     $\displaystyle{##}$\hfil \tabskip\@centering
259     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
260     \tabskip\z@skip\cr
261 }

```

■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。

```

262 % \DeclareOption{openbib}{%
263 %     \AtEndOfPackage{%
264 %         \renewcommand\@openbib@code{%

```

```

265 %      \advance\leftmargin\bibindent
266 %      \itemindent -\bibindent
267 %      \listparindent \itemindent
268 %      \parsep \z@}%
269 %      \renewcommand\newblock{\par}}

```

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

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

```

270 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。
271 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
272 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
273 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
274 \define@key{bxjs}{enablejfam}[true]{%
275   \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。

```

276 \DeclareOption{disablejfam}{\let\bxjs@enablejfam=f}

```

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

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

[2016-07-13] \ifdraft を定義するのをやめました。

\ifjsDraft 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。

```

277 \let\ifjsDraft\iffalse
278 \@onlypreamble\bxjs@draft
279 \def\bxjs@draft#1{%
280   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}

```

```

281 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
282 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
283 \AtBeginDocument{%
284   \expandafter\ifx\csname ifdraft\endcsname\relax
285     \expandafter\let\csname ifdraft\endcsname\expandafter\endcsname
286     \csname ifjsDraft\endcsname
287   \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。

```

288 \newif\ifbxjs@papersize
289 \bxjs@papersizetrue
290 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
291 \DeclareOption{papersize}{\bxjs@papersizetrue}

```

---

■英語化 オプション english を新設しました。

```

292 \newif\if@english
293 \@englishfalse
294 \DeclareOption{english}{\@englishtrue}

```

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

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

---

BXJS では当初から bxjsreport クラスが用意されている。

---

■jslogo パッケージの読み込み L<sup>A</sup>T<sub>E</sub>X 関連のロゴを再定義する jslogo パッケージを読み込まないオプション nojslogo を新設しました。jslogo オプションの指定で従来どおりの

動作となります。デフォルトは `jslogo` で、すなわちパッケージを読み込みます。

---

BXJS クラスでは、`nojslogo` を既定とする。

---

```
295 \newif\if@jslogo \@jslogofalse
296 \DeclareOption{jslogo}{\@jslogotrue}
297 \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 クラスを読込済として扱う／扱わない。
- `precisetext` / `noprecisetext`： $\text{\XeTeX}$  の “`generateactualtext`” を有効／無効にする。
- `simplejasetup` / `nosimplejasetup`： $\text{\XeTeX}$  の “`linebreaklocale`” を有効／無効にする。
- `bigcode` / `nobigcode`： $\text{\upTeX}$  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 10pt 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 (= 13Q/10pt)。
- `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  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  における「長さのスケール」の逆関数を求めるもの。例えば `\bxjs@invscale\dimX{1.3}` は `\dimX=1.3\dimX` の逆の演算を行う。

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

※元の長さが 128 pt 以上の場合でも動作するように修正した。

```
298 \mathchardef\bxjs@isc@ll=128
299 \mathchardef\bxjs@isc@sl=259
300 \def\bxjs@isc@sl@h{65539 }
301 \def\bxjs@invscale#1#2{%
302   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
303   \ifdim\@tempdima<\bxjs@isc@ll\p@
304     \@tempcnta\@tempdima \multiply\@tempcnta\@ccclvi
305     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@ccclvi
306   \else
307     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
308     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
309   \fi
310   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
311   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
312   \@tempdimb\@tempcnta\@ne
313   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
314   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
315   \@whiledim\@tempdimb<\@tempdimc\do{%
316     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
317     \advance\@tempcntb\@ne \divide\@tempcntb\tw@
318     \ifdim #2\@tempcntb>\@tempdima
319       \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
320     \else \@tempdimb=\@tempcntb\@ne \fi}%
321   \xdef\bxjs@gtmpa{\the\@tempdimb}%
322   \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性が高いと判断さ

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。

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

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

```
324 \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。

```
325 \def\bxjs@composite@proc{%
326   \bxjs@oldfontcommandstrue
327   \setkeys{bxjs}{ja=pandoc}%
328   \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…。

```
329 \def\bxjs@driver@opt{dvipdfmx}%
330 \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。

```
331 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
332 %\let\bxjs@engine@opt\@undefined
```

エンジン明示指定のオプションの処理。

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

```
333 \DeclareOption{autodetect-engine}{%
334   \let\bxjs@engine@given=*\%
335 \DeclareOption{latex}{%
336   \def\bxjs@engine@opt{latex}%
337   \let\bxjs@engine@given=n}%
338 \DeclareOption{platex}{%
339   \def\bxjs@engine@opt{platex}%
340   \let\bxjs@engine@given=j}%
341 \DeclareOption{uplatex}{%
342   \def\bxjs@engine@opt{uplatex}%
343   \let\bxjs@engine@given=u}%
344 \DeclareOption{xelatex}{%
345   \def\bxjs@engine@opt{xelatex}%
```

```

346 \let\bxjs@engine@given=x}
347 \DeclareOption{pdflatex}{%
348 \def\bxjs@engine@opt{pdflatex}%
349 \let\bxjs@engine@given=p}
350 \DeclareOption{lualatex}{%
351 \def\bxjs@engine@opt{lualatex}%
352 \let\bxjs@engine@given=l}
353 \DeclareOption{platex-ng}{%
354 \def\bxjs@engine@opt{platex-ng}%
355 \let\bxjs@engine@given=g}
356 \DeclareOption{platex-ng*}{%
357 \def\bxjs@engine@opt{platex-ng*}%
358 \let\bxjs@platexng@nodrv=t%
359 \let\bxjs@engine@given=g}

```

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

```

360 %\let\bxjs@driver@given\@undefined
361 \let\bxjs@driver@@dvimode=0
362 \let\bxjs@driver@@dvipdfmx=1
363 \let\bxjs@driver@@pdfmode=2
364 \let\bxjs@driver@@xetex=3
365 \let\bxjs@driver@@dvips=4
366 \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。

```

367 %\let\bxjs@driver@opt\@undefined

368 \DeclareOption{dvips}{%
369 \def\bxjs@driver@opt{dvips}%
370 \let\bxjs@driver@given\bxjs@driver@@dvips}
371 \DeclareOption{dviout}{%
372 \def\bxjs@driver@opt{dviout}%
373 \let\bxjs@driver@given\bxjs@driver@@dvimode}
374 \DeclareOption{xdvi}{%
375 \def\bxjs@driver@opt{xdvi}%
376 \let\bxjs@driver@given\bxjs@driver@@dvimode}
377 \DeclareOption{dvipdfmx}{%
378 \def\bxjs@driver@opt{dvipdfmx}%
379 \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
380 \DeclareOption{nodvidriver}{%
381 \def\bxjs@driver@opt{nodvidriver}%
382 \let\bxjs@driver@given\bxjs@driver@@none}
383 \DeclareOption{pdftex}{%
384 \def\bxjs@driver@opt{pdftex}%
385 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
386 \DeclareOption{luatex}{%
387 \def\bxjs@driver@opt{luatex}%
388 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
389 \DeclareOption{xetex}{%

```

```

390 \def\bxjs@driver@opt{xetex}%
391 \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。

```

392 \DeclareOption{dvipdfmx-if-dvi}{%
393 \setkeys{bxjs}{dvi=dvipdfmx}}

```

## ■その他の BXJS 独自オプション

`\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。

```

394 \newif\ifbxjs@usezw \bxjs@usezwtrue

```

`zw`、`noz` オプションの定義。

```

395 \DeclareOption{noz}{%
396 \bxjs@usezwfalse}
397 \DeclareOption{zw}{%
398 \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。

```

399 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue

```

`nojs`、`js` オプションの定義。

```

400 \DeclareOption{nojs}{%
401 \bxjs@disguise@jsfalse}
402 \DeclareOption{js}{%
403 \bxjs@disguise@jstrue}

```

`\ifbxjs@precisetext`  $\TeX$  の “generateactualtext” を有効にするか。既定は偽。

```

404 \newif\ifbxjs@precisetext

```

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

```

405 \DeclareOption{noprecisetext}{%
406 \bxjs@precisetextfalse}
407 \DeclareOption{precisetext}{%
408 \bxjs@precisetexttrue}

```

`\ifbxjs@simplejasetup`  $\TeX$  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）。

```

409 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

```

`nosimplejasetup` / `simplejasetup` オプションの定義。

```

410 \DeclareOption{nosimplejasetup}{%
411 \bxjs@simplejasetupfalse}
412 \DeclareOption{simplejasetup}{%
413 \bxjs@simplejasetuptrue}

```

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

```
414 \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。

```
415 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
```

```
416 \def\bxjs@tmpb#1 #2#3\@nil{%
```

```
417 \ifx1#2\bxjs@bigcodetrue \fi}
```

```
418 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil
```

`nobigcode` / `bigcode` オプションの定義。

```
419 \DeclareOption{nobigcode}{%
```

```
420 \bxjs@bigcodefalse}
```

```
421 \DeclareOption{bigcode}{%
```

```
422 \bxjs@bigcodetrue}
```

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

```
423 \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。

```
424 \DeclareOption{nooldfontcommands}{%
```

```
425 \bxjs@oldfontcommandsfalse}
```

```
426 \DeclareOption{oldfontcommands}{%
```

```
427 \bxjs@oldfontcommandstrue}
```

---

## ■keyval 型のオプション

---

```
428 \def\bxjs@setkey{%
```

```
429 \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
```

```
430 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
```

```
431 \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`。

```
432 \def\bxjs@safe@setkeys#1#2{%
```

```
433 \let\bxjs@KV@errx\KV@errx
```

```
434 \let\KV@errx\bxjs@safe@setkeys@a
```

```
435 \setkeys{#1}{#2}%
```

```
436 \let\KV@errx\bxjs@KV@errx}
```

```
437 \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。

```
438 \def\bxjs@set@keyval#1#2#3{%
439   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
440   \ifx\bxjs@next\relax
441     \bxjs@error@keyval{#1}-{#2}%
442     #3%
443   \else \bxjs@next
444   \fi}
445 \@onlypreamble\bxjs@error@keyval
446 \def\bxjs@error@keyval#1#2{%
447   \ClassError\bxjs@clsname
448   {Invalid value '#2' for option #1}\@ehc}
```

`\ifbxjs@scaleset` 和文スケール値が指定されたか。

```
449 \newif\ifbxjs@scaleset
```

`\jsScale` 〔実数値マクロ〕和文スケール値。

```
450 \def\jsScale{0.924715}
```

`base` オプションの処理。

```
451 \define@key{bxjs}{base}{\bxjs@setbasefontsize{#1}}
```

`jbase` オプションの処理。ここでは `\jsScale` の値を使用する。`scale` の処理との順序依存を消すため、`jbase` の処理の実行を遅延させている。

```
452 \@onlypreamble\bxjs@do@opt@jbase
453 \let\bxjs@do@opt@jbase\relax
454 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
455 \def\bxjs@setjbasefontsize#1{%
456   \def\bxjs@do@opt@jbase{%
457     \bxjs@setbasefontlength\@tempdima{#1}%
458     \bxjs@invscale\@tempdima\jsScale
459     \bxjs@setbasefontsize{\@tempdima}}}
```

`scale` オプションの処理。

```
460 \define@key{bxjs}{scale}{%
461   \bxjs@scalesettrue
462   \edef\jsScale{#1}}
```

`noscale` オプションの処理。

```
463 \DeclareOption{noscale}{%
464   \bxjs@scalesettrue
465   \def\jsScale{1}}
```

`\bxjs@param@mag` `mag` オプションの値。

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

`mag` オプションの処理。

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

`paper` オプションの処理。

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

`\bxjs@jadriver` 和文ドライバの名前。

```
469 \let\bxjs@jadriver\relax
470 %\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)。

```
471 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
472 \define@key{bxjs}{ja}[\relax]{%
473   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}
```

`\jsJaFont` 和文フォント設定の名前。

```
474 \let\jsJaFont\@empty
```

`jafont` オプションの処理。

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

`\jsJaParam` 和文ドライバパラメタの文字列。

```
476 \let\jsJaParam\@empty
```

`japaram` オプションの処理。

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

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

```
478 \let\bxjs@magstyle@mag=m
479 \let\bxjs@magstyle@real=r
480 \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。

```
481 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
482 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
483 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

`\bxjs@magstyle@default` は既定の値を表す。

```
484 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
485 \ifx l\jsEngine \ifnum\luatexversion>86
486   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
487 \fi\fi
488 \ifjsWithpTeXng
489   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
490 \fi
491 \let\bxjs@magstyle\bxjs@magstyle@default
```

`magstyle` オプションの処理。

```
492 \define@key{bxjs}{magstyle}{%
493   \expandafter\let\expandafter\bxjs@magstyle\csname
```

```

494   bxjs@magstyle@#1\endcsname
495   \ifx\bxjs@magstyle\relax
496     \ClassError\bxjs@clsname
497     {Invalid value '#1' for option magstyle}\@ehc
498     \let\bxjs@magstyle\bxjs@magstyle@default
499   \fi}

```

\bxjs@geometry geometry オプションの値。

```

500 \let\bxjs@geometry@class=c
501 \let\bxjs@geometry@user=u
502 \let\bxjs@geometry\bxjs@geometry@class

```

geometry オプションの処理。

```

503 \define@key{bxjs}{geometry}{%
504   \expandafter\let\expandafter\bxjs@geometry\csname
505     bxjs@geometry@#1\endcsname
506   \ifx\bxjs@geometry\relax
507     \ClassError\bxjs@clsname
508     {Invalid value '#1' for option geometry}\@ehc
509     \let\bxjs@geometry\bxjs@geometry@class
510   \fi}

```

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

```

511 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

```

fancyhdr オプションの処理。

```

512 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
513 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
514 \define@key{bxjs}{fancyhdr}[true]{%
515   \bxjs@set@keyval{fancyhdr}{#1}{}}

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

516 \newif\ifbxjs@dvi@opt

```

DVI モードのドライバとドライバ種別との対応。

```

517 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
518 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
519 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
520 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
521 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

```

dvi オプションの処理。

```

522 \define@key{bxjs}{dvi}{%
523   \expandafter\let\expandafter\bxjs@tmpa\csname
524     bxjs@dvidriver@@#1\endcsname
525   \ifx\bxjs@tmpa\relax
526     \ClassError\bxjs@clsname
527     {Invalid value '#1' for option dvi}\@ehc
528   \else

```



```

\bxjs@driver@given を未定義にしていることに注意。
529 \def\bxjs@driver@opt{#1}%
530 \let\bxjs@driver@given\@undefined
531 \bxjs@dvi@opttrue
532 \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。
533 \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。
534 \newif\ifbxjs@force@chapterabstract
535 %<book>\bxjs@force@chapterabstracttrue

layout オプションの処理。
536 \@namedef{bxjs@kv@layout@v1}{%
537 %<book>\bxjs@layout@buggyhmargintrue
538 %<report>\bxjs@force@chapterabstracttrue
539 }
540 \@namedef{bxjs@kv@layout@v2}{%
541 %<book>\bxjs@layout@buggyhmarginfalse
542 %<report>\bxjs@force@chapterabstractfalse
543 }
544 \define@key{bxjs}{layout}{%
545 \bxjs@set@keyval{layout}{#1}{}}

\bxjs@textwidth@limit textwidth-limit の指定値。 \textwidth の上限。
546 %\let\bxjs@textwidth@limit\@undefined
547 \define@key{bxjs}{textwidth-limit}{%
548 \edef\bxjs@textwidth@limit{#1}}

\bxjs@paragraph@mark paragraph-mark の指定値。 パラグラフのマーク。
549 %\let\bxjs@paragraph@mark\@undefined
550 \define@key{bxjs}{paragraph-mark}{%
551 \edef\bxjs@paragraph@mark{#1}}

\ifbxjs@whole@zw@lines [スイッチ] whole-zw-lines の指定値。
552 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
553 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
554 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
555 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}

\ifbxjs@jaspace@cmd [スイッチ] jaspacemcmd の指定値。
556 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
557 \let\bxjs@kv@jaspacemcmd@true\bxjs@jaspace@cmdtrue
558 \let\bxjs@kv@jaspacemcmd@false\bxjs@jaspace@cmdfalse
559 \define@key{bxjs}{jaspacemcmd}[true]{\bxjs@set@keyval{jaspacemcmd}{#1}{}}
560 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{jaspacemcmd}{#1}{}}

```

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

```
561 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
562 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
563 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
564 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}
```

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

```
565 \chardef\bxjs@everyparhook@none=0
566 \chardef\bxjs@everyparhook@compat=1
567 \chardef\bxjs@everyparhook@modern=2
568 \if j\jsEngine
569 \let\bxjs@everyparhook\bxjs@everyparhook@compat
570 \else
571 \let\bxjs@everyparhook\bxjs@everyparhook@modern
572 \fi
573 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
574 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
575 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
576 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}
```

---

## ■ オプションの実行

---

L<sup>A</sup>T<sub>E</sub>X の実装では、クラスやパッケージのオプションのトークン列の中に { } が含まれると正常に処理ができない。これに対処する為 `\@removeelement` の実装に少し手を加える(仕様は変わらない)。

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

```
577 \let\bxjs@ltx@removeelement\@removeelement
578 \def\@removeelement#1#2#3{%
579   \def\reserved@a{#2}%
580   \ifx\reserved@a\@empty \let#3\@empty
581   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
582   \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 dvout が対応しています。`multicols` や `url` を `\RequirePackage` するのはやめました。

```
583 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
584 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
585 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
586 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
587 \ProcessOptions\relax
588 \bxjs@composite@proc
```

後処理

```

589 \if@slide
590   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
591 \fi
592 \if@landscape
593   \setlength\@tempdima {\paperheight}
594   \setlength\paperheight{\paperwidth}
595   \setlength\paperwidth {\@tempdima}
596 \fi

```

## ■ グローバルオプションの整理

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

```

597 \@onlypreamble\bxjs@purge@brace@elts
598 \def\bxjs@purge@brace@elts{%
599   \def\bxjs@tmpa{\@gobble}%
600   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
601   \let\@classoptionslist\bxjs@tmpa}
602 \@onlypreamble\bxjs@purge@be@a
603 \def\bxjs@purge@be@a#1,{%
604   \ifx\@nil#1\relax\else
605     \bxjs@purge@be@b#1{\@nil}
606     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
607     \expandafter\bxjs@purge@be@a
608   \fi}
609 \@onlypreamble\bxjs@purge@be@b
610 \def\bxjs@purge@be@b#1#2{\bxjs@purge@be@c}
611 \@onlypreamble\bxjs@purge@be@c
612 \def\bxjs@purge@be@c#1\@nil{%
613   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
614 \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。

```

615 \@expandtwoargs\@removeelement
616   {papersize}\@classoptionslist\@classoptionslist
617 \@expandtwoargs\@removeelement
618   {10pt}\@classoptionslist\@classoptionslist
619 \@expandtwoargs\@removeelement
620   {noscale}\@classoptionslist\@classoptionslist

```

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

[2016-11-09] pL<sup>A</sup>T<sub>E</sub>X/ uP<sup>A</sup>L<sub>A</sub>T<sub>E</sub>X を自動判別するオプション autodetect-engine を新設しました。uP<sup>A</sup>L<sub>A</sub>T<sub>E</sub>X の場合は、グローバルオプションに uplatex を追加することで、自動

判定に応じて `otf` パッケージにも `uplatex` オプションが渡るようにします。

[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。

```
621 \ifx\bxjs@jadriver\relax\else
622   \let\bxjs@jadriver@given\bxjs@jadriver
623 \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。

```
624 \let\bxjs@tmpb\jsEngine
625 \ifx j\bxjs@tmpb\ifjsWithpTeXng
626   \let\bxjs@tmpb=g
627 \fi\fi
628 \ifx j\bxjs@tmpb\ifjsWithupTeX
629   \let\bxjs@tmpb=u
630 \fi\fi
631 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
632   \let\bxjs@tmpb=n
633 \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。）


```
634 \ifx *\bxjs@engine@given
635   \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。

```
636 \ifx j\bxjs@engine@given
637   \g@addto@macro\@classoptionslist{,platex}
638 \else\ifx u\bxjs@engine@given
639   \g@addto@macro\@classoptionslist{,uplatex}
640 \fi\fi
641 \fi
642 \ifx\bxjs@engine@given\@undefined\else
643   \ifx\bxjs@engine@given\bxjs@tmpb\else
644     \ClassError\bxjs@clsname
645       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
646   \fi
647 \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。

```
648 \ifjsWithpTeXng
649   \g@addto@macro\@classoptionslist{,uplatex}
650 \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。

```

651 \@tempwattrue
652 \ifx \bxjs@driver@given\@undefined\else
653   \ifjsInPdfMode
654     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
655       \@tempwafalse
656     \fi
657   \else\ifx x\jsEngine
658     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
659       \@tempwafalse
660     \fi
661   \else
662     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
663       \@tempwafalse
664     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
665       \@tempwafalse
666     \fi\fi
667   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
668     \@tempwafalse
669   \fi\fi
670 \fi\fi
671 \fi
672 \if@tempwa\else
673   \ClassError\bxjs@clsname
674   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
675 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

676 \ifjsInPdfMode \@tempwafalse
677 \else\ifx x\jsEngine \@tempwafalse
678 \else\ifjsWithpTeXng \@tempwafalse
679 \else \@tempwattrue
680 \fi\fi\fi
681 \if@tempwa

```

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)。

```

682 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
683   \ClassWarningNoLine\bxjs@clsname
684   {No driver option is given}
685 \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。)

```

686 \ifbxjs@dvi@opt
687   \edef\bxjs@nxt{%
688     \let\noexpand\bxjs@driver@given

```

```

689      \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
690      \noexpand\g@addto@macro\noexpand\@classoptionslist
691      {,\bxjs@driver@opt}%
692    }\bxjs@nxt
693  \fi
694 \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。

```

695 \ifjsWithpTeXng
696   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
697     \let\bxjs@platexng@nodrv\@undefined
698   \else\ifx t\bxjs@platexng@nodrv\else
699     \g@addto@macro\@classoptionslist{,dvipdfmx}
700   \fi\fi
701 \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。

```

702 \ifx\bxjs@driver@given\bxjs@driver@@none
703   \bxjs@papersizefalse
704 \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。

```

705 \def\bxjs@@minimal{minimal}
706 \ifx\bxjs@jadriver\relax
707   \ifx j\jsEngine
708     \def\bxjs@jadriver{standard}
709   \else
710     \let\bxjs@jadriver\bxjs@@minimal
711   \fi
712 \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。

```

713 \ifx\bxjs@jadriver@given\@undefined\else
714   \ifx\bxjs@engine@given\@undefined
715     \ClassError\bxjs@clsname
716     {An engine option must be explicitly given}%
717     {When you use a Japanese-driver you must specify a correct\MessageBreak
718       engine option.\MessageBreak\@ehc}
719 \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。)

```

720 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else

```

```

721 \ifx\bxjs@magstyle\bxjs@magstyle@mag
722 \let\bxjs@magstyle\bxjs@magstyle@default
723 \ClassError\bxjs@clsname
724 {The engine does not support 'magstyle=usemag'}%
725 {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
726 The default value 'nomag*' is used instead.\MessageBreak \@ehc}
727 \fi
728 \fi

```

オプション処理時に遅延させていた jbase の処理をここで実行する。

```

729 \bxjs@do@opt@jbase

```

`\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。

```

730 \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。）

```

731 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
732 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
733 \catcode\@tempcnta\active
734 \advance\@tempcnta\@ne
735 \repeat
736 \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。

```

737 \ifbxjs@disguise@js
738 %<book|report>\def\bxjs@js@clsname{jsbook}
739 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
740 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
741 \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。

```

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

```

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

```

743 \ifbxjs@oldfontcommands
744 \AtEndOfClass{\allowoldfontcommands}
745 \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 p<sub>La</sub>T<sub>E</sub>X 2<sub>ε</sub> はトンボ出力幅を両側に 1 インチとっていますので、dvips 使用時に

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

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

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

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

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

---

BXJS では出力用紙サイズ記録は `geometry` パッケージが行う。

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

---

```
746 \newdimen\stockwidth \newdimen\stockheight
747 \begingroup\expandafter\expandafter\expandafter\endgroup
748 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
749 % \newdimen\stockwidth \newdimen\stockheight
750 \setlength{\stockwidth}{\paperwidth}
751 \setlength{\stockheight}{\paperheight}
752 \advance \stockwidth 2in
753 \advance \stockheight 2in
754 \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。

```
755 %<slide>\def\n@baseline{13}%
756 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
757 %<!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。

```
758 \ifx\bxjs@magstyle\bxjs@magstyle@mag
759 \jsc@magtrue
760 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
761 \jsc@mag@xrealtrue
762 \fi\fi
```

---

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

[2008-12-26] 1000 / `\mag` に相当する `\inv@mag` を定義しました。`truein` を使っていた



ところを `\inv@mag in` に直しましたので、`geometry` パッケージと共存できると思います。  
なお、新ドキュメントクラス側で 10pt 以外にする場合の注意：

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

---

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

```
763 \ifx\bxjs@param@mag\relax
764   \@tempdima=\bxjs@param@basefontsize
765   \advance\@tempdima.001pt \multiply\@tempdima25
766   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
767   \edef\bxjs@param@mag{\the\@tempcnta}
768 \else
769 % mag 値が直接指定された場合
770   \let\c@bxjs@cnta\@tempcnta
771   \setcounter{bxjs@cnta}{\bxjs@param@mag}
772   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
773 % 有効な mag 値の範囲は 1--32768
774   \edef\bxjs@param@mag{\the\@tempcnta}
775   \advance\@tempcnta100000
776   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
777   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
778   \edef\bxjs@param@basefontsize{\the\@tempdima}
779 \fi
780 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
781 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
782 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
783 \edef\jsc@magscale{\strip@pt\@tempdima}
784 \let\jsBaseFontSize\bxjs@param@basefontsize
785 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
786 %\typeout{\string\jsEngine: \meaning\jsEngine}
787 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
788 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
789 %\typeout{\string\jsc@magscale: \jsc@magscale}
790 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
791 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}
```

---

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

---

`\mag` する場合（現状はこれが既定）にコードの変更を低減するために、以下では必要に応じて、`\jsc@mpt` を `\p@?` と書く。その上で、`\mag` する場合は `?` を無視して `\p@` と解釈させ、`\mag` しない場合は `?` を英字扱いにして `\p@?` という制御綴を `\jsc@mpt` と同値にする。

※（多分 2.0 版あたりで）JS クラスに合わせるため `\p@?` 表記を止める予定。

```

792 \newdimen\jsc@mpt
793 \newdimen\jsc@mmm
794 \ifjsc@mag
795   \jsc@mpt=1\p@
796   \jsc@mmm=1mm
797   \catcode`\?=9 % \p@? read as \p@
798 \else
799   \jsc@mpt=\jsc@magscale\p@
800   \jsc@mmm=\jsc@magscale mm
801   \catcode`\?=11 \let\p@?\jsc@mpt
802 \fi
803 \chardef\bxjs@qmc=\catcode`\?\relax

```

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

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

```

804 \newdimen\jsZw
805 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
806 \ifbxjs@usezw
807   \providecommand*\zw{\jsZw}
808 \fi

```

`\zwspace` 全角幅の水平空き。

```

809 \def\zwspace{\hspace{\jsZw}\relax}

```

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

```

810 \ifjsc@mag@xreal
811   \RequirePackage{type1cm}
812   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

---

```

813 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
814 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
815 \let\jsc@get@external@font\get@external@font
816 \def\get@external@font{%
817   \jsc@preadjust@extract@font
818   \jsc@get@external@font}
819 \def\jsc@fstrunc#1{%
820   \edef\jsc@tmpa{\strip@pt#1}%
821   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
822 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
823   \if#5*\else
824     \edef\jsc@tmpa{#1%
825       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
826   \fi}
827 \def\jsc@preadjust@extract@font{%
828   \let\jsc@req@size\f@size

```

```

829 \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
830 \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
831 \let\jsc@ref@size\jsc@tmpa
832 \let\f@size\jsc@ref@size}
833 \def\execute@size@function#1{%
834 \let\jsc@cref@size\f@size
835 \let\f@size\jsc@req@size
836 \csname s@fct@#1\endcsname}
837 \let\jsc@DeclareErrorFont\DeclareErrorFont
838 \def\DeclareErrorFont#1#2#3#4#5{%
839 \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
840 \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
841 \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
842 \def\gen@sfcnt{%
843 \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
844 \empty@sfcnt}
845 \def\genb@sfcnt{%
846 \edef\mandatory@arg{%
847 \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@@}%
848 \empty@sfcnt}
849 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
850 \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 851 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 852 %\def\jsc@medskip{\vspace\jsc@medskipamount}
853 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount 854 \newskip\jsc@smallskipamount
\jsc@bigskipamount 855 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
856 %\newskip\jsc@medskipamount
857 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
858 %\newskip\jsc@bigskipamount
859 %\jsc@bigskipamoun =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] トンボオブションが指定されているとき「だけ」`\stockwidth`, `\stockheight` が定義されています。

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

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

```

860 % \ifpapersize
861 %   \setlength{\@tempdima}{\paperwidth}
862 %   \setlength{\@tempdimb}{\paperheight}
863 %   \iftombow
864 %     \advance \@tempdima 2truein
865 %     \advance \@tempdimb 2truein
866 %   \fi
867 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
868 % \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。

```

869 \chardef\bxjs@periodchar=`\
870 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcode\bxjs@periodchar}
871 \def\@{\bxjs@SE{}}

```

### 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  $\text{\LaTeX}$  の内部命令 `\@xpt` を使っています。この `\@xpt` の類は次のものがあり、 $\text{\LaTeX}$  本体で定義されています。

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

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

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

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

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

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

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

```
872 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
873 \expandafter\bxjs@tmpa\expandafter{%
874   \set@fontsize{#1}{#2}{#3}%
875 % 末尾にコードを追加
876   \expandafter\def\expandafter\size@update\expandafter{%
877     \size@update
878     \jsFontSizeChanged}%
879 }
```

`\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。

```
880 \newcommand*\jsFontSizeChanged{%
881   \jsZw=\f@size\p@
882   \jsZw=\jsScale \jsZw
883   \ifdim\parindent>\z@
884     \if@english \parindent=1em
885     \else       \parindent=1\jsZw
886   \fi
887   \fi\relax
888   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
889 \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。

```
890 \ifjsc@mag
891   \let\jsc@setfontsize\@setfontsize
892 \else
```

```

893 \def\jsc@setfontsize#1#2#3{%
894   \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
895 \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。

---

```

896 \emergencystretch 3\jsZw

```

```

\ifnarrowbaselines 欧文用に行間を狭くする論理変数と、それを真・偽にするためのコマンドです。
\narrowbaselines   [2003-06-30] 数式に入るところで \narrowbaselines を実行しているので
\widebaselines     \abovedisplayskip 等が初期化されてしまうという shintok さんのご指摘に対し
                    て、しっぱ愛好家さんが次の修正を教えてくださいました。
                    [2008-02-18] english オプションで最初の段落のインデントをしないようにしました。
                    TODO: Hasumi さん [qa:54539] のご指摘は考慮中です。

```

---

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

---

```

897 \newif\ifnarrowbaselines
898 \if@english
899   \narrowbaselinestrue
900 \fi
901 \def\narrowbaselines{%
902   \narrowbaselinestrue
903   \skip0=\abovedisplayskip
904   \skip2=\abovedisplayshortskip
905   \skip4=\belowdisplayskip
906   \skip6=\belowdisplayshortskip
907 % 一時的に警告を無効化する
908   \let\bxjs@ltx@nomath\@nomath
909   \let\@nomath\@gobble
910   \@currsize\selectfont
911   \let\@nomath\bxjs@ltx@nomath
912   \abovedisplayskip=\skip0
913   \abovedisplayshortskip=\skip2
914   \belowdisplayskip=\skip4
915   \belowdisplayshortskip=\skip6\relax}
916 \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。

```
917 \def\bxjs@if@narrowbaselines{%
918   \ifnarrowbaselines\expandafter\@firstoftwo
919   \else \expandafter\@secondoftwo
920   \fi
921 }
```

---

`\normalsize` 標準のフォントサイズと行送りを選ぶコマンドです。

本文 10 ポイントのときの行送りは、欧文の標準クラスファイルでは 12 ポイント、アスキーの和文クラスファイルでは 15 ポイントになっていますが、ここでは 16 ポイントにしました。ただし `\narrowbaselines` で欧文用の 12 ポイントになります。

公称 10 ポイントの和文フォントが約 9.25 ポイント（アスキーのものの 0.961 倍）であることもあり、行送りがかなりゆったりとしたと思います。実際、 $16/9.25 \approx 1.73$  であり、和文の推奨値の一つ「二分四分」（1.75）に近づきました。

---

`microtype` 対策のため `if` 文を避ける。

---

```
922 \renewcommand{\normalsize}{%
923   \bxjs@if@narrowbaselines{%
924     \jsc@setfontsize\normalsize\@xpt\@xipt
925   }{%else
926     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
927   }%
```

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

[2003-02-16] ちょっと変えました。

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

```
928 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
929 \abovedisplayshortskip \z@ \@plus3\p@?
930 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
931 \belowdisplayshortskip \belowdisplayskip
```

最後に、リスト環境のトップレベルのパラメータ `\@listI` を、`\@listi` にコピーしておきます。`\@listI` の設定は後で出てきます。

```
932 \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。

```
933 \normalsize
```

`\Cht` 基準となる長さの設定をします。p<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 2<sub>ε</sub> カーネル (`plfonts.dtx`) で宣言されているパ  
`\Cdp` 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  
`\Cvs` ド `0x3441`) へ変更しました。

---

`\Chs`

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

```
934 \ifx\Cht\undefined \newdimen\Cht \fi
935 \ifx\Cdp\undefined \newdimen\Cdp \fi
936 \ifx\Cwd\undefined \newdimen\Cwd \fi
937 \ifx\Cvs\undefined \newdimen\Cvs \fi
938 \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。

---

```
939 \setlength\Cht{0.88\jsZw}
940 \setlength\Cdp{0.12\jsZw}
941 \setlength\Cwd{1\jsZw}
942 \setlength\Cvs{\baselineskip}
943 \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。

---

```
944 \newcommand{\small}{%
945   \bxjs@if@narrowbaselines{%
946     <!kiyou> \jsc@setfontsize\small\@ixpt{11}%
947     <kiyou> \jsc@setfontsize\small{8.8888}{11}%
948   }{%else
949     <!kiyou> \jsc@setfontsize\small\@ixpt{13}%
950     <kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
951   }%
952   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
953   \abovedisplayshortskip \z@ \@plus3\p@?
954   \belowdisplayskip \abovedisplayskip
955   \belowdisplayshortskip \belowdisplayskip
956   \def\@listi{\leftmargin\leftmargini
957     \topsep \z@
958     \parsep \z@
959     \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。

```

960 \newcommand{\footnotesize}{%
961   \bxjs@if@narrowbaselines{%
962     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
963     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
964   }{%else
965     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
966     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
967   }%
968   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
969   \abovedisplayshortskip \z@ \@plus2\p@?
970   \belowdisplayskip \abovedisplayskip
971   \belowdisplayshortskip \belowdisplayskip
972   \def\@listi{\leftmargin\leftmargini
973             \topsep \z@
974             \parsep \z@
975             \itemsep \parsep}}

```

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

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 976 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
977 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 978 \if@twocolumn
\HUGE 979 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
980 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
981 \else
982 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
983 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
984 \fi
985 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
986 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
987 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
988 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
989 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
990 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

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

```

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

```

しかし、このおかげで別行数式の上下のスペースが少し違っていました。とりあえず

amsmath の equation 関係は okumacro のほうで逃げていますが、もっとうまい逃げ道があれば教えてください。

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

```
992 % \newcommand{\headfont}{\bfseries}
993 \newcommand{\headfont}{\sffamily}
994 % \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。

```
995 %<!kiyou>\setlength\columnsep{2\Cwd}
996 %<kiyou>\setlength\columnsep{28truebp}
997 \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 でしたが 1pt に変更しました。normal... の付いた方は保存用です。

```
\lineskiplimit 998 \setlength\lineskip{1\jsc@mp}
\normallineskip 999 \setlength\normallineskip{1\jsc@mp}
\normallineskiplimit 1000 \setlength\lineskiplimit{1\jsc@mp}
1001 \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。

```
1002 \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。

```
1003 \setlength\parskip{0\jsc@mp}
1004 \if@slide
1005   \setlength\parindent{0\p@}
1006 \else
```

```
1007 \setlength\parindent{1\Cwd}
1008 \fi
```

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

```
\@highpenalty 1009 \@lowpenalty 51
1010 \@medpenalty 151
1011 \@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。

```
1012 % \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。

```
1013 % \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。

```
1014 \ifjsc@mag
1015 \mag=\bxjs@param@mag
1016 \fi
1017 \setlength{\topskip}{10\p@?}
```

`\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。

```
1018 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
1019 \def\bxjs@read@b{%
1020 \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
1021 \else \expandafter\bxjs@read@d \fi}
1022 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}%
1023 \def\bxjs@read@d#1\@nil{}
1024 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
1025 \edef\bxjs@layout@paper{%
1026 \ifjsc@mag truedimen,\fi
1027 \if@landscape landscape,\fi
1028 \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。

```

1029 %<*article|report>
1030 \def\bxjs@layout{%
1031   headheight=\topskip,footskip=0.03367\paperheight,%
1032   headsep=\footskip-\topskip,includeheadfoot,%
1033   hscale=0.76,hmarginratio=1:1,%
1034   vscale=0.83,vmarginratio=1:1,%
1035 }
1036 %</article|report>
1037 %<*book>
1038 \ifbxjs@layout@buggyhmargin    %---
1039 % アレ
1040 \def\bxjs@layout{%
1041   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1042   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1043   vscale=0.83,vmarginratio=1:1,%
1044 }
1045 \else    %---
1046 % 非アレ
1047 \def\bxjs@layout{%
1048   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1049   hmargin=18\jsc@mmm,%
1050   vscale=0.83,vmarginratio=1:1,%
1051 }
1052 \fi    %---
1053 %</book>
1054 %<*slide>
1055 \def\bxjs@layout{%
1056   noheadfoot,%
1057   hscale=0.9,hmarginratio=1:1,%
1058   vscale=0.944,vmarginratio=1:1,%
1059 }
1060 %</slide>

```

\fullwidth 〔寸法レジスタ〕 ヘッダ・フッタ領域の横幅。

```
1061 \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。

```

1062 %<*book>
1063 \newcommand\jsTextWidthLimit{40}
1064 \ifx\bxjs@textwidth@limit\@undefined\else
1065   \let\c@bxjs@canta\@tempcanta
1066   \setcounter{bxjs@canta}{\bxjs@textwidth@limit}
1067   \long\edef\jsTextWidthLimit{\the\@tempcanta}
1068 \fi
1069 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

1070 \def\bxjs@postproc@layout{%
1071 % ドライバ再設定

```

```

1072 \ifx\bxjs@geometry@driver\relax\else
1073 \let\Gm@driver\bxjs@geometry@driver
1074 \fi
1075 % textwidth 調整
1076 \ifbxjs@whole@zw@lines
1077 \@tempdimb=\textwidth
1078 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1079 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1080 \advance\@tempdimb-\textwidth
1081 \advance\oddsidemargin 0.5\@tempdimb
1082 \advance\evensidemargin 0.5\@tempdimb
1083 \fi
1084 \fullwidth=\textwidth
1085 %<*book>
1086 \ifdim\textwidth>\jsTextWidthLimit\Cwd
1087 \textwidth=\jsTextWidthLimit\Cwd
1088 \addtolength\evensidemargin{\fullwidth-\textwidth}
1089 \fi
1090 %</book>
1091 % textheight 調整
1092 \@tempdimb=\textheight
1093 \advance\textheight-\topskip
1094 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1095 \advance\textheight\topskip
1096 \advance\@tempdimb-\textheight
1097 \advance\topmargin0.5\@tempdimb
1098 % headheight 調整
1099 \@tempdima=\topskip
1100 \advance\headheight\@tempdima
1101 \advance\topmargin-\@tempdima
1102 % marginpar 設定
1103 \setlength\marginparsep{\columnsep}
1104 \setlength\marginparpush{\baselineskip}
1105 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1106 -\textwidth-10\jsc@mmm-\marginparsep}
1107 \ifbxjs@whole@zw@lines
1108 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1109 \fi
1110 % 連動する変数
1111 \maxdepth=.5\topskip
1112 \stockwidth=\paperwidth
1113 \stockheight=\paperheight
1114 }

```

\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。

```

1115 \edef\jsGeometryOptions{%
1116 \bxjs@layout@paper,\bxjs@layout}

```

---

## ■geometry パッケージ読込

---

geometry オプションの値に応じて分岐する。

まずは geometry=class の場合。

```
1117 \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。

```
1118 \@onlypreamble\bxjs@revert
```

```
1119 \let\bxjs@revert\@empty
```

```
1120 \edef\bxjs@tmpa{\the\ht\strutbox}
```

```
1121 \ht\strutbox=10\p@?
```

```
1122 \g@addto@macro\bxjs@revert{\ht\strutbox=\bxjs@tmpa\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。

```
1123 \ifbxjs@papersize
```

```
1124 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
```

```
1125 \PassOptionsToPackage{dvipdfm}{geometry}
```

```
1126 \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
```

```
1127 \PassOptionsToPackage{dvipdfm}{geometry}
```

```
1128 \fi\fi
```

```
1129 \let\bxPapersizeSpecialDone=t
```

```
1130 \else
```

```
1131 \PassOptionsToPackage{driver=none}{geometry}
```

```
1132 \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。

```
1133 \edef\bxjs@nxt{%
```

```
1134 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
```

```
1135 \AtBeginDocument{\bxjs@pre@geometry@hook}
```

```
1136 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
```

```
1137 \bxjs@nxt \bxjs@revert
```

```
1138 \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。

```
1139 \let\bxjs@geometry@driver\Gm@driver
```

```
1140 \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。

```
1141 \@onlypreamble\bxjs@pre@geometry@hook
1142 \def\bxjs@pre@geometry@hook{%
1143   \@ifpackageloaded{geometry}{%
1144     \@ifpackagelater{geometry}{2010/02/12}{%}%else
1145     \PackageWarningNoLine\bxjs@clsname
1146     {The 'geometry' package installed\MessageBreak
1147     is too old (< v5.0)}%
1148     \if x\jsEngine \ifnum\mag=\@m\else
1149     \def\bxjs@Gm@driver{pdftex}
1150     \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1151     \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1152     \fi\fi
1153   }%
1154   \ifjsWithpTeXng
1155     \ifx\Gm@driver\@empty
1156     \def\Gm@driver{pdftex}%
1157     \fi
1158   \fi
1159 }{}}
```

`\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。

```
1160 \def\setpagelayout{\@ifstar
1161   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1162 \def\bxjs@modify@layout#1{%
1163   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1164   \bxjs@nxt\bxjs@postproc@layout}
1165 \def\bxjs@reset@layout#1{%
1166   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1167   \bxjs@nxt\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。

```
1168 \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。

```
1169 \AtBeginDocument{\bxjs@check@page@layout}
1170 \@onlypreamble\bxjs@check@page@layout
1171 \def\bxjs@check@page@layout{%
1172   \ifdim\textwidth=.5\maxdimen
```

```

1173 \ClassError\bxjs@clsname
1174 {Page layout is not properly set}%
1175 {\@ehd}
1176 \fi}
1177 \def\jsUseMinimalPageLayout{%
1178 \setlength{\textwidth}{6.5in}%
1179 \setlength{\textheight}{8in}}
1180 \let\bxjs@geometry@driver\relax
1181 \def\setpagelayout{\@ifstar
1182 {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1183 \def\bxjs@pagelayout@a#1{%
1184 \ClassError\bxjs@clsname
1185 {Command '\string\setpagelayout' is not supported,\MessageBreak
1186 because 'geometry' value is not 'class'}\@eha}

```

geometry=user の場合の処理はここで終わり。

```

1187 \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。

---

<\*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。

```

1188 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1189 \if@slide
1190 \setlength\headheight{0\jsc@mpt}
1191 \else
1192 \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
    06-26)

```



1193 \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) になっていましたが、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。

```
1194 %<*article|kiyou>
1195 \if@slide
1196   \setlength\footskip{0pt}
1197 \else
1198   \setlength\footskip{0.03367\paperheight}
1199   \ifdim\footskip<\baselineskip
1200     \setlength\footskip{\baselineskip}
1201   \fi
1202 \fi
1203 %</article|kiyou>
1204 %<jspf>\setlength\footskip{9\jsc@mmm}
1205 %<*book>
1206 \if@report
1207   \setlength\footskip{0.03367\paperheight}
1208   \ifdim\footskip<\baselineskip
1209     \setlength\footskip{\baselineskip}
1210   \fi
1211 \else
1212   \setlength\footskip{0pt}
1213 \fi
1214 %</book>
1215 %<*report>
1216 \setlength\footskip{0.03367\paperheight}
1217 \ifdim\footskip<\baselineskip
1218   \setlength\footskip{\baselineskip}
1219 \fi
1220 %</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)。

```
1221 %<*article>
1222 \if@slide
1223   \setlength\headsep{0\jsc@empt}
1224   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1225   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1226 \else
1227   \setlength\headsep{\footskip}
1228   \addtolength\headsep{-\topskip}
```

```

1229 \fi
1230 %</article>
1231 %<*book>
1232 \if@report
1233   \setlength\headsep{\footskip}
1234   \addtolength\headsep{-\topskip}
1235 \else
1236   \setlength\headsep{6\jsc@mmm}
1237   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1238   \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1239 \fi
1240 %</book>
1241 %<*report>
1242 \setlength\headsep{\footskip}
1243 \addtolength\headsep{-\topskip}
1244 %</report>
1245 %<*jspf>
1246 \setlength\headsep{9\jsc@mmm}
1247 \addtolength\headsep{-\topskip}
1248 %</jspf>
1249 %<*kiyou>
1250 \setlength\headheight{0\jsc@mpt}
1251 \setlength\headsep{0\jsc@mpt}
1252 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1253 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1254 %</kiyou>

```

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

```

1255 \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。

```

1256 \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。

```

1257 %<*article>
1258 \if@slide
1259   \setlength\fullwidth{0.9\paperwidth}
1260 \else
1261   \setlength\fullwidth{0.76\paperwidth}
1262 \fi
1263 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1264 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1265 \setlength\textwidth{\fullwidth}
1266 %</article>
1267 %<*book>
1268 \if@report
1269   \setlength\fullwidth{0.76\paperwidth}
1270 \else
1271   \setlength\fullwidth{\paperwidth}
1272   \addtolength\fullwidth{-36\jsc@mmm}
1273 \fi
1274 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1275 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1276 \setlength\textwidth{\fullwidth}
1277 \if@report \else
1278   \if@twocolumn \else
1279     \ifdim \fullwidth>40zw
1280       \setlength\textwidth{40zw}
1281     \fi
1282   \fi
1283 \fi
1284 %</book>
1285 %<*report>
1286 \setlength\fullwidth{0.76\paperwidth}
1287 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1288 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1289 \setlength\textwidth{\fullwidth}
1290 %</report>
1291 %<*jspf>
1292 \setlength\fullwidth{50zw}
1293 \addtolength\fullwidth{8\jsc@mmm}
1294 \setlength\textwidth{\fullwidth}
1295 %</jspf>
1296 %<*kiyou>
1297 \setlength\fullwidth{48zw}
1298 \addtolength\fullwidth{\columnsep}
1299 \setlength\textwidth{\fullwidth}
1300 %</kiyou>

```

\textheight 紙の高さ \paperheight は、1 インチと \topmargin と \headheight と \headsep と

`\textheight` と `\footskip` とページ下部の余白を加えたものです。

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

某学会誌スタイルでは 44 行にします。

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

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

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

```
1301 %<*article|book|report>
1302 \if@slide
1303   \setlength{\textheight}{0.95\paperheight}
1304 \else
1305   \setlength{\textheight}{0.83\paperheight}
1306 \fi
1307 \addtolength{\textheight}{-10\jsc@mppt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1308 \addtolength{\textheight}{-\headsep}
1309 \addtolength{\textheight}{-\footskip}
1310 \addtolength{\textheight}{-\topskip}
1311 \divide\textheight\baselineskip
1312 \multiply\textheight\baselineskip
1313 %</article|book|report>
1314 %<jspf>\setlength{\textheight}{51\baselineskip}
1315 %<kiyou>\setlength{\textheight}{47\baselineskip}
1316 \addtolength{\textheight}{\topskip}
1317 \addtolength{\textheight}{0.1\jsc@mppt}
1318 %<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  $\text{\LaTeX 2}_\epsilon$  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。

```
1319 \def\flushbottom{%
1320   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mppt}%
1321   \let\@texttop\relax}
```

`\marginparsep` `\marginparsep` は欄外の書き込みと本文との間隔です。`\marginparpush` は欄外の書き込み  
`\marginparpush`

みどうしの最小の間隔です。

```
1322 \setlength\marginparsep{\columnsep}
1323 \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。TEX 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 pLATEX 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。

```
1324 \setlength{\oddsidemargin}{\paperwidth}
1325 \addtolength{\oddsidemargin}{-\fullwidth}
1326 \setlength{\oddsidemargin}{.5\oddsidemargin}
1327 \iftombow
1328   \addtolength{\oddsidemargin}{-1in}
1329 \else
1330   \addtolength{\oddsidemargin}{-\inv@mag in}
1331 \fi
1332 \setlength{\evensidemargin}{\oddsidemargin}
1333 \if@mparswitch
1334   \addtolength{\evensidemargin}{\fullwidth}
1335   \addtolength{\evensidemargin}{-\textwidth}
1336 \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。

```
1337 \setlength\marginparwidth{\paperwidth}
1338 \addtolength\marginparwidth{-\oddsidemargin}
1339 \addtolength\marginparwidth{-\inv@mag in}
1340 \addtolength\marginparwidth{-\textwidth}
1341 \addtolength\marginparwidth{-10\jsc@mmm}
1342 \addtolength\marginparwidth{-\marginparsep}
1343 \@tempdima=1zw
1344 \divide\marginparwidth\@tempdima
1345 \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つたので，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)。

```
1346 \setlength\topmargin{\paperheight}
1347 \addtolength\topmargin{-\textheight}
1348 \if@slide
1349   \addtolength\topmargin{-\headheight}
1350 \else
```

```

1351 \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1352 \fi
1353 \addtolength\topmargin{-\headsep}
1354 \addtolength\topmargin{-\footskip}
1355 \setlength\topmargin{0.5\topmargin}
1356 %<kiyou>\setlength\topmargin{81truebp}
1357 \iftombow
1358 \addtolength\topmargin{-1in}
1359 \else
1360 \addtolength\topmargin{-\inv@mag in}
1361 \fi

</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。

---

```
1362 \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。

```
1363 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

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

[2003-08-23] ちょっと増やしました。

```
1364 \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。

```
1365 \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。  
 1366 `\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。  
 1367 `\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。  
 1368 `\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。  
 1369 `\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。  
 1370 `\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。  
 1371 `\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。  
 1372 `\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。  
 1373 `\renewcommand{\dblfloatpagefraction}{.8}`

`\floatsep` `\floatsep` はページ上部・下部のフロート間の距離です。`\textfloatsep` はページ上部・  
`\textfloatsep` 下部のフロートと本文との距離です。`\intextsep` は本文の途中に出力されるフロートと本  
`\intextsep` 文との距離です。  
 1374 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`  
 1375 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`  
 1376 `\setlength\intextsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

`\dblfloatsep` 二段組のときの段抜きのフロートについての値です。  
`\dbltextfloatsep` 1377 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`  
 1378 `\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` 1379 `\setlength\@fptop{0\p@? \@plus 1fil}`  
 1380 `\setlength\@fpsep{8\p@? \@plus 2fil}`  
 1381 `\setlength\@fpbot{0\p@? \@plus 1fil}`

`\dblftop` 段抜きフロートについての値です。

`\dblfpsep` 1382 `\setlength\dblftop{0\p? \@plus 1fil}`

`\dblfpbot` 1383 `\setlength\dblfpsep{8\p? \@plus 2fil}`

1384 `\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 pT<sub>E</sub>X の標準クラス 2017/02/15 に合わせて、同じ命令を追  
`\pltx@cleartoleftpage` 加しました。

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

となっています。

```
1385 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1386 % \ifodd\c@page
1387 % \iftdir
1388 % \hbox{}\thispagestyle{empty}\newpage
1389 % \if@twocolumn\hbox{}\newpage\fi
1390 % \fi
1391 % \else
1392 % \ifydir
1393 % \hbox{}\thispagestyle{empty}\newpage
1394 % \if@twocolumn\hbox{}\newpage\fi
1395 % \fi
1396 % \fi\fi}
1397 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1398 % \ifodd\c@page
1399 % \ifydir
1400 % \hbox{}\thispagestyle{empty}\newpage
1401 % \if@twocolumn\hbox{}\newpage\fi
1402 % \fi
1403 % \else
1404 % \iftdir
1405 % \hbox{}\thispagestyle{empty}\newpage
1406 % \if@twocolumn\hbox{}\newpage\fi
1407 % \fi
1408 % \fi\fi}
1409 %\def\pltx@cleartooddpage{\clearpage\if@twoside
1410 % \ifodd\c@page\else
1411 % \hbox{}\thispagestyle{empty}\newpage
1412 % \if@twocolumn\hbox{}\newpage\fi
1413 % \fi\fi}
1414 %\def\pltx@cleartoevenpage{\clearpage\if@twoside
1415 % \ifodd\c@page
```



```

1416 \hbox{}\thispagestyle{empty}\newpage
1417 \if@twocolumn\hbox{}\newpage\fi
1418 \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。

```

1419 \let\pltx@cleartorightpage\pltx@cleartooddpage
1420 \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。

```

1421 %<*book|report>
1422 \if@openleft
1423 \let\cleardoublepage\pltx@cleartoleftpage
1424 \else\if@openright
1425 \let\cleardoublepage\pltx@cleartorightpage
1426 \fi\fi
1427 %</book|report>

```

## 7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

```

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

```

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

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

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

```
1428 % \def\ps@empty{%
1429 %   \let\mkboth\gobbletwo
1430 %   \let\oddhead\empty
1431 %   \let\oddfoot\empty
1432 %   \let\evenhead\empty
1433 %   \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。

```
1434 \def\ps@plainfoot{%
1435   \let\mkboth\gobbletwo
1436   \let\oddhead\empty
1437   \def\oddfoot{\normalfont\hfil\thepage\hfil}%
1438   \let\evenhead\empty
1439   \let\evenfoot\oddfoot}
1440 \def\ps@plainhead{%
1441   \let\mkboth\gobbletwo
1442   \let\oddfoot\empty
1443   \let\evenfoot\empty
1444   \def\evenhead{%
1445     \ifmparswitch \hss \fi
1446     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1447     \ifmparswitch\else \hss \fi}%
1448   \def\oddhead{%
1449     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1450 %<book>\let\ps@plain\ps@plainhead
1451 %<!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。

```
1452 %<*article|slide>
1453 \if@twoside
1454   \def\ps@headings{%
1455     \let\oddfoot\empty
1456     \let\evenfoot\empty
1457     \def\evenhead{\ifmparswitch \hss \fi
1458       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1459     \ifmparswitch\else \hss \fi}%
1460   \def\oddhead{%
1461     \underline{%
1462       \hbox to \fullwidth{\rightmark\hfil\textbf{\thepage}}\hss}%
1462     \hss \fi}%
1463   }
```

```

1463 \let\mkboth\markboth
1464 \def\sectionmark##1{\markboth{%
1465 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1466 ##1}{}}%
1467 \def\subsectionmark##1{\markright{%
1468 \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
1469 ##1}}%
1470 }
1471 \else % if not twoside
1472 \def\ps@headings{%
1473 \let\@oddfoot\@empty
1474 \def\@oddhead{%
1475 \underline{%
1476 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1477 \let\mkboth\markboth
1478 \def\sectionmark##1{\markright{%
1479 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1480 ##1}}%
1481 \fi
1482 %</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。

---

```

1483 %<*book|report>
1484 \def\bxjs@maybe@autoxspacing{%
1485 \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1486 \newif\if@omit@number
1487 \def\ps@headings{%
1488 \let\@oddfoot\@empty
1489 \let\@evenfoot\@empty
1490 \def\@evenhead{%
1491 \if@mparswitch \hss \fi
1492 \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1493 \textbf{\thepage}\hfil\leftmark}}}%
1494 \if@mparswitch\else \hss \fi}%
1495 \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1496 {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1497 \let\mkboth\markboth
1498 \def\chaptermark##1{\markboth{%
1499 \ifnum \c@secnumdepth >\m@ne
1500 \if@mainmatter
1501 \if@omit@number\else
1502 \@chapapp\thechapter\@chappos\hskip1\jsZw
1503 \fi

```

```

1504     \fi
1505     \fi
1506     ##1}{}}%
1507     \def\sectionmark##1{\markright{%
1508         \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1509         ##1}}}%
1510 %</book|report>

    最後は学会誌の場合です。

1511 %<*jspf>
1512 \def\ps@headings{%
1513     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1514     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1515     \def\@oddhead{\normalfont\hfil \@title \hfil}
1516     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1517 %</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。

```

1518 \def\ps@myheadings{%
1519     \let\@oddfoot\@empty\let\@evenfoot\@empty
1520     \def\@evenhead{%
1521         \if@mparswitch \hss \fi%
1522         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1523         \if@mparswitch\else \hss \fi}%
1524     \def\@oddhead{%
1525         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1526     \let\@mkboth\@gobbletwo
1527 %<book|report> \let\chaptermark\@gobble
1528 \let\sectionmark\@gobble
1529 %<!book&!report> \let\subsectionmark\@gobble
1530 }

```

## 8 文書のマークアップ

### 8.1 表題

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

```

\date 1531 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1532 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1533 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1534 % \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。

```
1535 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1536 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1537 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1538 \AtBeginDocument{\bxjs@decl@subtitle}
1539 \def\bxjs@decl@subtitle{%
1540   \global\let\bxjs@decl@subtitle\relax
1541   \ifx\subtitle\@undefined
1542     \global\let\subtitle\jsSubtitle
1543   \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。

```
1544 \def\bxjs@annihilate@subtitle{%
1545   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1546   \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 1547 %<*jspf>
\keywords 1548 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1549 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
1550 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1551 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1552 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1553 %</jspf>
```

`\plainifnotempty` 従来の標準クラスでは、文書全体のページスタイルを `empty` にしても表題のあるページだけ `plain` になってしまうことがありました。これは `\maketitle` の定義中に `\thispagestyle{plain}` が入っているためです。この問題を解決するために、「全体のページスタイルが `empty` でないならこのページのスタイルを `plain` にする」という次の命令を作ることになります。

```
1554 \def\plainifnotempty{%
1555   \ifx \@oddhead \@empty
1556     \ifx \@oddfoot \@empty
1557       \else
1558         \thispagestyle{plainfoot}%
1559       \fi
1560     \else
1561       \thispagestyle{plainhead}%
1562     \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。

```
1563 %<*article|book|report|slide>
1564 \if@titlepage
1565   \newcommand{\maketitle}{%
1566     \begin{titlepage}%
1567       \let\footnotesize\small
1568       \let\footnoterule\relax
1569       \let\footnote\thanks
1570       \null\vfil
1571       \if@slide
1572         {\footnotesize \@date}%
1573         \begin{center}
1574           \mbox{} \[\[1\jsZw]
1575           \large
1576           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1577           \jsc@smallskip
1578           \@title
1579           \ifx\bxjs@subtitle\@undefined\else
1580             \par\vskip\z@
1581             {\small \bxjs@subtitle\par}
1582           \fi
1583           \jsc@smallskip
1584           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1585           \vfill
1586           {\small \@author}%
1587         \end{center}
1588       \else
1589       \vskip 60\p@?
1590       \begin{center}%
1591         {\LARGE \@title \par}%
1592         \ifx\bxjs@subtitle\@undefined\else
1593           \vskip5\p@?
1594           {\normalsize \bxjs@subtitle\par}
1595         \fi
1596         \vskip 3em%
1597         {\large
1598           \lineskip .75em
1599           \begin{tabular}[t]{c}%
1600             \@author
1601           \end{tabular}\par}%
1602         \vskip 1.5em
1603         {\large \@date \par}%
1604       \end{center}%

```

```

1605     \fi
1606     \par
1607     \@thanks\vfil\null
1608 \end{titlepage}%
1609 \setcounter{footnote}{0}%
1610 \global\let\thanks\relax
1611 \global\let\maketitle\relax
1612 \global\let\@thanks\@empty
1613 \global\let\@author\@empty
1614 \global\let\@date\@empty
1615 \global\let\@title\@empty
1616 \global\let\title\relax
1617 \global\let\author\relax
1618 \global\let\date\relax
1619 \global\let\and\relax
1620 \bxjs@annihilate@subtitle
1621 }%
1622 \else
1623 \newcommand{\maketitle}{\par
1624 \begingroup
1625 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1626 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1627 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1628 \parindent 1\jsZw\noindent
1629 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1630 \if@twocolumn
1631 \ifnum \col@number=\@ne
1632 \maketitle
1633 \else
1634 \twocolumn[\maketitle]%
1635 \fi
1636 \else
1637 \newpage
1638 \global\@topnum\z@ % Prevents figures from going at top of page.
1639 \maketitle
1640 \fi
1641 \plainifnotempty
1642 \@thanks
1643 \endgroup
1644 \setcounter{footnote}{0}%
1645 \global\let\thanks\relax
1646 \global\let\maketitle\relax
1647 \global\let\@thanks\@empty
1648 \global\let\@author\@empty
1649 \global\let\@date\@empty
1650 \global\let\@title\@empty
1651 \global\let\title\relax
1652 \global\let\author\relax
1653 \global\let\date\relax

```

```

1654 \global\let\and\relax
1655 \bxjs@annihilate@subtitle
1656 }

```

`\@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。

```

1657 \def\@maketitle{%
1658 \newpage\null
1659 \vskip 2em
1660 \begin{center}%
1661 \let\footnote\thanks
1662 {\LARGE \@title \par}%
1663 \ifx\bxjs@subtitle\@undefined\else
1664 \vskip3\p@?
1665 {\normalsize \bxjs@subtitle\par}
1666 \fi
1667 \vskip 1.5em
1668 {\large
1669 \lineskip .5em
1670 \begin{tabular}[t]{c}%
1671 \@author
1672 \end{tabular}\par}%
1673 \vskip 1em
1674 {\large \@date}%
1675 \end{center}%
1676 \par\vskip 1.5em
1677 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1678 }
1679 \fi
1680 %</article|book|report|slide>
1681 %<*jspf>
1682 \newcommand{\maketitle}{\par
1683 \begingroup
1684 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1685 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1686 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1687 \parindent 1\jsZw\noindent
1688 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1689 \twocolumn[\@maketitle]%
1690 \plainifnotempty
1691 \@thanks
1692 \endgroup
1693 \setcounter{footnote}{0}%
1694 \global\let\thanks\relax
1695 \global\let\maketitle\relax
1696 \global\let\@thanks\@empty
1697 \global\let\@author\@empty
1698 \global\let\@date\@empty
1699 % \global\let\@title\@empty % \@title は柱に使う
1700 \global\let\title\relax

```



```

1701 \global\let\author\relax
1702 \global\let\date\relax
1703 \global\let\and\relax
1704 \ifx\authors@mail\@undefined\else{%
1705     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1706     \footnotetext[0]{\itshape\authors@mail}%
1707 } \fi
1708 \global\let\authors@mail\@undefined}
1709 \def\@maketitle{%
1710 \newpage\null
1711 \vskip 6em % used to be 2em
1712 \begin{center}
1713     \let\footnote\thanks
1714     \ifx\@title\@undefined\else{\LARGE\headfont\@title\par} \fi
1715     \lineskip .5em
1716     \ifx\@author\@undefined\else
1717         \vskip 1em
1718         \begin{tabular}[t]{c}%
1719             \@author
1720         \end{tabular}\par
1721     \fi
1722     \ifx\@etitle\@undefined\else
1723         \vskip 1em
1724         {\large \@etitle \par}%
1725     \fi
1726     \ifx\@eauthor\@undefined\else
1727         \vskip 1em
1728         \begin{tabular}[t]{c}%
1729             \@eauthor
1730         \end{tabular}\par
1731     \fi
1732     \vskip 1em
1733     \@date
1734 \end{center}
1735 \vskip 1.5em
1736 \centerline{\box\@abstractbox}
1737 \ifx\@keywords\@undefined\else
1738     \vskip 1.5em
1739     \centerline{\parbox{157\jscmmm}{\textsf{Keywords:}}\ \small\@keywords}}
1740 \fi
1741 \vskip 1.5em}
1742 %</jspf>

```

## 8.2 章・節

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

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

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```

1743 \def\@startsection#1#2#3#4#5#6{%
1744   \if@noskipsec \leavevmode \fi
1745   \par
1746 % 見出し上の空きを \@tempskipa にセットする
1747   \@tempskipa #4\relax
1748 % \@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
1749   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1750 % 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
1751   \ifdim \@tempskipa <\z@
1752     \@tempskipa -\@tempskipa \@afterindentfalse
1753   \fi
1754   \if@nobreak
1755 %   \everypar{\everyparhook}% これは間違い
1756     \everypar{}%
1757   \else
1758     \addpenalty\@secpenalty
1759 % 次の行は削除
1760 %   \addvspace\@tempskipa
1761 % 次の \noindent まで追加
1762     \ifdim \@tempskipa >\z@
1763       \if@slide\else
1764         \null
1765         \vspace*{-\baselineskip}%

```

```

1766     \fi
1767     \vskip\@tempskipa
1768     \fi
1769     \fi
1770     \noindent
1771 % 追加終わり
1772     \@ifstar
1773     {\@ssect{#3}{#4}{#5}{#6}}%
1774     {\@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。

```

1775 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1776     \let\bxjs@if@ceph\@firstofone
1777 \else \let\bxjs@if@ceph\@gobble
1778 \fi

```

---

```

1779 \def\@sect#1#2#3#4#5#6[#7]#8{%
1780     \ifnum #2>\c@secnumdepth
1781         \let\@svsec\@empty
1782     \else
1783         \refstepcounter{#1}%
1784         \protected@edef\@svsec{\@secntformat{#1}\relax}%
1785     \fi
1786 % 見出し後の空きを \@tempskipa にセット
1787     \@tempskipa #5\relax
1788 % 条件判断の順序を入れ替えました
1789     \ifdim \@tempskipa<\z@
1790         \def\@svsechd{%
1791             #6{\hskip #3\relax
1792                 \@svsec #8}%
1793             \csname #1mark\endcsname{#7}%
1794             \addcontentsline{toc}{#1}{%
1795                 \ifnum #2>\c@secnumdepth \else
1796                     \protect\numberline{\csname the#1\endcsname}%
1797                 \fi
1798                 #7}}% 目次にフルネームを載せるなら #8
1799     \else
1800         \begingroup
1801             \interlinepenalty \@M % 下から移動
1802             #6{%
1803                 \@hangfrom{\hskip #3\relax\@svsec}%
1804             % \interlinepenalty \@M % 上に移動
1805             #8\@@par}%

```

```

1806 \endgroup
1807 \csname #1mark\endcsname{#7}%
1808 \addcontentsline{toc}{#1}{%
1809 \ifnum #2>\c@secnumdepth \else
1810 \protect\numberline{\csname the#1\endcsname}%
1811 \fi
1812 #7}% 目次にフルネームを載せるならここは #8
1813 \fi
1814 \@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。

```

1815 \def\@xsect#1{%
1816 % 見出しの後ろの空きを \@tempskipa にセット
1817 \@tempskipa #1\relax
1818 % 条件判断の順序を変えました
1819 \ifdim \@tempskipa<\z@
1820 \@nbreakfalse
1821 \global\@noskipsectrue
1822 \everypar{%
1823 \if@noskipsec
1824 \global\@noskipsecfalse
1825 {\setbox\z@\lastbox}%
1826 \clubpenalty\@M
1827 \begingroup \@svsechd \endgroup
1828 \unskip
1829 \@tempskipa #1\relax
1830 \hskip -\@tempskipa
1831 \else
1832 \clubpenalty \@clubpenalty
1833 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1834 \fi\bxjs@if@ceph\everyparhook}%
1835 \else
1836 \par \nbreak
1837 \vskip \@tempskipa
1838 \@afterheading
1839 \fi
1840 \if@slide
1841 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1842 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1843 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1844 \fi
1845 \par % 2000-12-18
1846 \ignorespaces}
1847 \def\@ssect#1#2#3#4#5{%
1848 \@tempskipa #3\relax

```

```

1849 \ifdim \@tempskipa<\z@
1850 \def\@svsechd{#4{\hskip #1\relax #5}}%
1851 \else
1852 \begingroup
1853 #4{%
1854 \hangfrom{\hskip #1}%
1855 \interlinepenalty \@M #5\@@par}%
1856 \endgroup
1857 \fi
1858 \@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 1859 \newcommand*\chaptermark[1]{}
1860 % \newcommand*\sectionmark[1]{}
\subsubsectionmark 1861 % \newcommand*\subsectionmark[1]{}
1862 % \newcommand*\subsubsectionmark[1]{}
\paragraphmark 1863 % \newcommand*\paragraphmark[1]{}
1864 % \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。

```

1865 %<!book&!report>\setcounter{secnumdepth}{3}
1866 %<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 1867 \newcounter{part}
1868 %<book|report>\newcounter{chapter}
\c@subsubsection 1869 %<book|report>\newcounter{section}[chapter]
1870 %<!book&!report>\newcounter{section}
\c@paragraph 1871 \newcounter{subsection}[section]
1872 \newcounter{subsubsection}[subsection]
1873 \newcounter{paragraph}[subsubsection]
1874 \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。

```
1875 \renewcommand{\thepart}{\@Roman\c@part}
1876 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1877 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1878 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1879 %<*book|report>
1880 \renewcommand{\thechapter}{\@arabic\c@chapter}
1881 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1882 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1883 %</book|report>
1884 \renewcommand{\thesubsubsection}{%
1885   \thesubsection.\@arabic\c@subsubsection}
1886 \renewcommand{\theparagraph}{%
1887   \thesubsubsection.\@arabic\c@paragraph}
1888 \renewcommand{\thesubparagraph}{%
1889   \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。

```
1890 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1891 %<book|report>\newcommand{\@chappos}{\postchaptername}
```

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

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

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

```
1892 %<*book|report>
1893 \newcommand\frontmatter{%
1894   \pltx@cleartooddpage
1895   \@mainmatterfalse
1896   \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。

```
1897 \newcommand\mainmatter{%
1898   \pltx@cleartooddpage
1899   \@mainmattertrue
1900   \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。

```

1901 \newcommand\backmatter{%
1902   \if@openleft
1903     \cleardoublepage
1904   \else\if@openright
1905     \cleardoublepage
1906   \else
1907     \clearpage
1908   \fi\fi
1909   \@mainmatterfalse}
1910 %</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。

```

1911 %<!*book&!report>
1912 \newcommand\part{%
1913   \if@noskipsec \leavevmode \fi
1914   \par
1915   \addvspace{4ex}%
1916   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1917   \secdef\@part\@spart}
1918 %</!*book&!report>

```

`book` および `report` クラスの場合は、少し複雑です。

```

1919 %<*book|report>
1920 \newcommand\part{%
1921   \if@openleft
1922     \cleardoublepage
1923   \else\if@openright
1924     \cleardoublepage
1925   \else
1926     \clearpage
1927   \fi\fi
1928   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1929   \if@twocolumn
1930     \onecolumn

```

```

1931 \@restonecoltrue
1932 \else
1933 \@restonecolfalse
1934 \fi
1935 \null\vfil
1936 \secdef\@part\@spart}
1937 %</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。

```

1938 %<*&!book&!report>
1939 \def\@part[#1]#2{%
1940 \ifnum \c@secnumdepth >\m@ne
1941 \refstepcounter{part}%
1942 \addcontentsline{toc}{part}{%
1943 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1944 \else
1945 \addcontentsline{toc}{part}{#1}%
1946 \fi
1947 \markboth{}{}%
1948 {\parindent\z@
1949 \raggedright
1950 \interlinepenalty \@M
1951 \normalfont
1952 \ifnum \c@secnumdepth >\m@ne
1953 \Large\headfont\prepartname\thepart\postpartname
1954 \par\nobreak
1955 \fi
1956 \huge \headfont #2%
1957 \markboth{}{}\par}%
1958 \nobreak
1959 \vskip 3ex
1960 \@afterheading}
1961 %</!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。

```

1962 %<*&book|report>
1963 \def\@part[#1]#2{%
1964 \ifnum \c@secnumdepth >-2\relax
1965 \refstepcounter{part}%
1966 \addcontentsline{toc}{part}{%
1967 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1968 \else
1969 \addcontentsline{toc}{part}{#1}%
1970 \fi
1971 \markboth{}{}%
1972 {\centering
1973 \interlinepenalty \@M

```



```

1974 \normalfont
1975 \ifnum \c@secnumdepth >-2\relax
1976 \huge\headfont \prepartname\thepart\postpartname
1977 \par\vskip20\p@?
1978 \fi
1979 \Huge \headfont #2\par}%
1980 \@endpart}
1981 %</book|report>

```

\@spart 番号を付けない部です。

```

1982 %<*&book&!report>
1983 \def\@spart#1{%
1984 \parindent \z@ \raggedright
1985 \interlinepenalty \@M
1986 \normalfont
1987 \huge \headfont #1\par}%
1988 \nobreak
1989 \vskip 3ex
1990 \@afterheading}
1991 %</&book&!report>
1992 %<*&book|report>
1993 \def\@spart#1{%
1994 \centering
1995 \interlinepenalty \@M
1996 \normalfont
1997 \Huge \headfont #1\par}%
1998 \@endpart}
1999 %</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。

```

2000 %<*&book|report>
2001 \def\@endpart{\vfil\newpage
2002 \if@twoside
2003 \if@openleft %% added (2017/02/24)
2004 \null\thispagestyle{empty}\newpage
2005 \else\if@openright %% added (2016/12/13)
2006 \null\thispagestyle{empty}\newpage
2007 \fi\fi %% added (2016/12/13, 2017/02/24)
2008 \fi
2009 \if@restonecol
2010 \twocolumn
2011 \fi}
2012 %</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。

```

2013 %<*book|report>
2014 \newcommand{\chapter}{%
2015   \if@openleft\cleardoublepage\else
2016   \if@openright\cleardoublepage\else\clearpage\fi\fi
2017   \plainifnotempty % 元: \thispagestyle{plain}
2018   \global\@topnum\z@
2019   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2020   \secdef
2021     {\@omit@numberfalse\@chapter}%
2022     {\@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。

```

2023 \def\@chapter[#1]#2{%
2024   \ifnum \c@secnumdepth >\m@ne
2025     \if@mainmatter
2026       \refstepcounter{chapter}%
2027       \typeout{\@chapapp\thechapter\@chappos}%
2028       \addcontentsline{toc}{chapter}%
2029         {\protect\numberline
2030 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2031         {\@chapapp\thechapter\@chappos}%
2032         #1}%
2033     \else\addcontentsline{toc}{chapter}{#1}\fi
2034   \else
2035     \addcontentsline{toc}{chapter}{#1}%
2036   \fi
2037   \chaptermark{#1}%
2038   \addtocontents{lof}{\protect\addvspace{10\jsc@mp}}%
2039   \addtocontents{lot}{\protect\addvspace{10\jsc@mp}}%
2040   \if@twocolumn
2041     \@topnewpage[\@makechapterhead{#2}]%
2042   \else
2043     \@makechapterhead{#2}%
2044     \@afterheading
2045   \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。

```

2046 \def\@makechapterhead#1{%
2047   \vspace*{2\Cvs}% 欧文は 50pt
2048   {\parindent \z@ \raggedright \normalfont
2049     \ifnum \c@secnumdepth >\m@ne
2050       \if@mainmatter
2051         \huge\headfont \@chapapp\thechapter\@chappos

```

```

2052      \par\nobreak
2053      \vskip \Cvs % 欧文は 20pt
2054      \fi
2055      \fi
2056      \interlinepenalty\@M
2057      \Huge \headfont #1\par\nobreak
2058      \vskip 3\Cvs}} % 欧文は 40pt

```

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

```

2059 \def\@schapter#1{%
2060   \chaptermark{#1}%
2061   \if@twocolumn
2062     \@topnewpage[\@makeschapterhead{#1}]%
2063   \else
2064     \@makeschapterhead{#1}\@afterheading
2065   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2066 \def\@makeschapterhead#1{%
2067   \vspace*{2\Cvs}% 欧文は 50pt
2068   {\parindent \z@ \raggedright
2069     \normalfont
2070     \interlinepenalty\@M
2071     \Huge \headfont #1\par\nobreak
2072     \vskip 3\Cvs}} % 欧文は 40pt
2073 %</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。

```

2074 \if@twocolumn
2075   \newcommand{\section}{%
2076     %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2077     \@startsection{section}{1}{\z@}%
2078     %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2079     %<kiyou>  {\Cvs}{0.5\Cvs}%
2080     % {\normalfont\large\headfont\@secapp}}
2081     {\normalfont\large\headfont\raggedright}}
2082 \else
2083   \newcommand{\section}{%
2084     \if@slide\clearpage\fi
2085     \@startsection{section}{1}{\z@}%
2086     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2087     {.5\Cvs \@plus.3\Cdp}% 後アキ
2088     % {\normalfont\Large\headfont\@secapp}}
2089     {\normalfont\Large\headfont\raggedright}}

```

2090 \fi

\subsection 同上です。

```
2091 \if@twocolumn
2092   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2093     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2094     {\normalfont\normalsize\headfont}}
2095 \else
2096   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2097     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2098     {.5\Cvs \@plus.3\Cdp}% 後アキ
2099     {\normalfont\large\headfont}}
2100 \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)。

```
2101 \if@twocolumn
2102   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2103     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2104     {\normalfont\normalsize\headfont}}
2105 \else
2106   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2107     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2108     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2109     {\normalfont\normalsize\headfont}}
2110 \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。

---

```
2111 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2112 \let\bxjs@org@paragraph@mark\jsParagraphMark
2113 \ifx\bxjs@paragraph@mark\@empty
2114   \let\jsParagraphMark\@empty
2115 \else\ifx\bxjs@paragraph@mark\@undefined\else
```

```

2116 \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2117 \fi\fi
2118 \let\jsJaChar\empty
2119 \if@twocolumn
2120 \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2121   {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2122 %<jspf> {\normalfont\normalsize\headfont}}
2123 %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2124 \else
2125 \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2126   {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2127   {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2128 %<jspf> {\normalfont\normalsize\headfont}}
2129 %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2130 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2131 \if@twocolumn
2132 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2133   {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2134   {\normalfont\normalsize\headfont}}
2135 \else
2136 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2137   {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2138   {\normalfont\normalsize\headfont}}
2139 \fi

```

### 8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2140 \if@slide
2141 \setlength\leftmargini{1\jsZw}
2142 \else
2143 \if@twocolumn
2144 \setlength\leftmargini{2\jsZw}
2145 \else
2146 \setlength\leftmargini{3\jsZw}
2147 \fi
2148 \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。

```
2149 \if@slide
2150   \setlength\leftmarginii {1\jsZw}
2151   \setlength\leftmarginiii{1\jsZw}
2152   \setlength\leftmarginiv {1\jsZw}
2153   \setlength\leftmarginv  {1\jsZw}
2154   \setlength\leftmarginvi {1\jsZw}
2155 \else
2156   \setlength\leftmarginii {2\jsZw}
2157   \setlength\leftmarginiii{2\jsZw}
2158   \setlength\leftmarginiv {2\jsZw}
2159   \setlength\leftmarginv  {1\jsZw}
2160   \setlength\leftmarginvi {1\jsZw}
2161 \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。

```
2162 \setlength \labelsep {0.5\jsZw} % .5em
2163 \setlength \labelwidth{\leftmargini}
2164 \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。

```
2165 \setlength\partopsep{\z@} % {2\p@ \@plus 1\p@ \@minus 1\p@}
```

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

```
\@endparpenalty 2166 \@beginparpenalty -\@lowpenalty
\@itempenalty 2167 \@endparpenalty -\@lowpenalty
2168 \@itempenalty -\@lowpenalty
```

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

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

```
2169 \def\@listi{\leftmargin\leftmargini
2170   \parsep \z@
2171   \topsep 0.5\baselineskip
2172   \itemsep \z@ \relax}
2173 \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）。

```
2174 \@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 2175 \def\@listii{\leftmargin\leftmarginii
\@listiv 2176 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2177 \topsep \z@
\@listv 2178 \parsep \z@
\@listvi 2179 \itemsep\parsep}
2180 \def\@listiii{\leftmargin\leftmarginiii
2181 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2182 \topsep \z@
2183 \parsep \z@
2184 \itemsep\parsep}
2185 \def\@listiv {\leftmargin\leftmarginiv
2186 \labelwidth\leftmarginiv
2187 \advance\labelwidth-\labelsep}
2188 \def\@listv {\leftmargin\leftmarginv
2189 \labelwidth\leftmarginv
2190 \advance\labelwidth-\labelsep}
2191 \def\@listvi {\leftmargin\leftmarginvi
2192 \labelwidth\leftmarginvi
2193 \advance\labelwidth-\labelsep}
```

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

`\theenumi` 出力する番号の書式を設定します。これらは L<sup>A</sup>T<sub>E</sub>X 本体 (`ltlists.dtx` 参照) で定義済みですが、ここでは表し方を変えています。`\@arabic`, `\@alph`, `\@roman`, `\@Alph` はそれぞれ算用数字、小文字アルファベット、小文字ローマ数字、大文字アルファベットで番号を出力する命令です。

```
2194 \renewcommand{\theenumi}{\@arabic\c@enumi}
2195 \renewcommand{\theenumii}{\@alph\c@enumii}
2196 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2197 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

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

---

`\labelenumiv` 和文の括弧で囲むための補助命令 `\jsInJaParen` を定義して `\labelenumii` でそれを用いている。

※現状の `zxjatype` の `\inhibitglue` の実装には「前後のグルーを消してしまう」という不備があって、そのため `enumii` の出力が異常になるという不具合があった。`zxjatype` を修正するまでの回避策として、サイズがゼロの罫 (`\bxjs@dust`) でガードしておく。

---

```
2198 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2199 \newcommand*\{jsInJaParen}[1]{%
```

```

2200 \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2201 \newcommand{\labelenumi}{\theenumi.}
2202 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
2203 \newcommand{\labelenumiii}{\theenumiii.}
2204 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2205 \renewcommand{\p@enumii}{\theenumi}
2206 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
2207 \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 2208 \newcommand\labelitemi{\textbullet}
\labelitemiii 2209 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiv 2210 \newcommand\labelitemiii{\textasteriskcentered}
2211 \newcommand\labelitemiv{\textperiodcentered}

```

### ■description 環境

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

```

2212 \newenvironment{description}{%
2213 \list{}{%
2214 \labelwidth=\leftmargin
2215 \labelsep=1\jsZw
2216 \advance \labelwidth by -\labelsep
2217 \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。

```

2218 \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。

```
2219 %<*book|report>
2220 \newenvironment{chapterabstract}{%
2221   \begin{list}{}{%
2222     \listparindent=1\jsZw
2223     \itemindent=\listparindent
2224     \rightmargin=\z@
2225     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2226 %</book|report>
```

“普通の” abstract 環境の定義。

```
2227 %<*article|report|slide>
2228 \newbox\@abstractbox
2229 \if@titlepage
2230   \newenvironment{abstract}{%
2231     \titlepage
2232     \null\vfil
2233     \@beginparpenalty\@lowpenalty
2234     \begin{center}%
2235       \headfont \abstractname
2236       \@endparpenalty\@M
2237     \end{center}%
2238     \par}%
2239   {\par\vfil\@endtitlepage}
2240 \else
2241   \newenvironment{abstract}{%
2242     \if@twocolumn
2243       \ifx\maketitle\relax
2244         \section*{\abstractname}%
2245       \else
2246         \global\setbox\@abstractbox\hbox\bgroup
2247         \begin{minipage}[b]{\textwidth}
2248           \small\parindent1\jsZw
2249           \begin{center}%
2250             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2251           \end{center}%
2252           \list{}{%
2253             \listparindent\parindent
2254             \itemindent \listparindent
2255             \rightmargin \leftmargin}%
2256           \item\relax
2257         \fi
2258       \else
```

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。

```
2238   \par}%
2239   {\par\vfil\@endtitlepage}
2240 \else
2241   \newenvironment{abstract}{%
2242     \if@twocolumn
2243       \ifx\maketitle\relax
2244         \section*{\abstractname}%
2245       \else
2246         \global\setbox\@abstractbox\hbox\bgroup
2247         \begin{minipage}[b]{\textwidth}
2248           \small\parindent1\jsZw
2249           \begin{center}%
2250             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2251           \end{center}%
2252           \list{}{%
2253             \listparindent\parindent
2254             \itemindent \listparindent
2255             \rightmargin \leftmargin}%
2256           \item\relax
2257         \fi
2258       \else
```

```

2259 \small
2260 \begin{center}%
2261 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2262 \end{center}%
2263 \list{}{%
2264 \listparindent\parindent
2265 \itemindent \listparindent
2266 \rightmargin \leftmargin}%
2267 \item\relax
2268 \fi}{\if@twocolumn
2269 \ifx\maketitle\relax
2270 \else
2271 \endlist\end{minipage}\egroup
2272 \fi
2273 \else
2274 \endlist
2275 \fi}
2276 \fi
2277 %</article|report|slide>
2278 %<*jspf>
2279 \newbox\@abstractbox
2280 \newenvironment{abstract}{%
2281 \global\setbox\@abstractbox\hbox\bgroup
2282 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2283 \small
2284 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2285 {\end{minipage}\egroup}
2286 %</jspf>

```

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

```

2287 %<*book|report>
2288 \ifbxjs@force@chapterabstract
2289 \let\abstract\chapterabstract
2290 \let\endabstract\endchapterabstract
2291 \fi
2292 %</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。

```

2293 %<*jspf>
2294 %\newbox\@keywordsbox
2295 %\newenvironment{keywords}{%
2296 % \global\setbox\@keywordsbox\hbox\bgroup
2297 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2298 % \small\parindent0\jsZw}%

```

```

2299 % {\end{minipage}\egroup}
2300 %</jspf>

```

### ■verse 環境

**verse** 詩のための verse 環境です。

```

2301 \newenvironment{verse}{%
2302   \let \=\@centercr
2303   \list{}{%
2304     \itemsep \z@
2305     \itemindent -2\jsZw % 元: -1.5em
2306     \listparindent\itemindent
2307     \rightmargin \z@
2308     \advance\leftmargin 2\jsZw}% 元: 1.5em
2309   \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。

```

2310 \newenvironment{quotation}{%
2311   \list{}{%
2312     \listparindent\parindent
2313     \itemindent\listparindent
2314     \rightmargin \z@}%
2315   \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。

```

2316 \newenvironment{quote}%
2317   {\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。

```

2318 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2319   \item[\hspace \labelsep{\headfont #1\ #2}]}
2320 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2321   \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。

```
2322 \newenvironment{titlepage}{%
2323 %<book>      \pltx@cleartooddpage %% 2017-02-24
2324     \if@twocolumn
2325         \@restonecoltrue\onecolumn
2326     \else
2327         \@restonecolfalse\newpage
2328     \fi
2329     \thispagestyle{empty}%
2330     \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2331     24
2332 }%
2333 {\if@restonecol\twocolumn \else \newpage \fi
2334     \if@twoside\else
2335         \setcounter{page}\@ne
2336     \fi}
```

## ■付録

**\appendix** 本文と付録を分離するコマンドです。

```
2336 %<*&!book&!report>
2337 \newcommand{\appendix}{\par
2338     \setcounter{section}{0}%
2339     \setcounter{subsection}{0}%
2340     \gdef\presectionname{\appendixname}%
2341     \gdef\postsectionname{}}%
2342 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2343 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2344 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
2345 %</&!book&!report>
2346 %<*&book|report>
2347 \newcommand{\appendix}{\par
2348     \setcounter{chapter}{0}%
2349     \setcounter{section}{0}%
2350     \gdef\@chapapp{\appendixname}%
2351     \gdef\@chappos{}}%
2352 \gdef\thechapter{\@Alph\c@chapter}}
2353 %</book|report>
```

## 8.4 パラメータの設定

### ■array と tabular 環境

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

```
2354 \setlength\arraycolsep{5\p@?}
```

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

```
2355 \setlength\tabcolsep{6\p@?}
```

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

```
2356 \setlength\arrayrulewidth{.4\p@}
```

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

```
2357 \setlength\doublerulesep{2\p@}
```

### ■tabbing 環境

`\tabbingsep` `\'` コマンドで入るアキです。

```
2358 \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。

```
2359 \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。

```
2360 \setlength\fboxsep{3\p@?}
```

```
2361 \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。

```
2362 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```
2363 %<*book|report>
```

```
2364 \@addtoreset{equation}{chapter}
```

```
2365 \renewcommand\theequation
```

```
2366 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

```
2367 %</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。

```
2368 % \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。

```
2369 % \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。

```
2370 % \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。

```
2371 %<!*book&!report>
2372 \newcounter{figure}
2373 \renewcommand \thefigure {\@arabic\c@figure}
2374 %</!*book&!report>
2375 %<!*book|report>
2376 \newcounter{figure}[chapter]
2377 \renewcommand \thefigure
2378     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2379 %</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 2380 \def\fps@figure{tbp}
2381 \def\ftype@figure{1}
\fnum@figure 2382 \def\ext@figure{lof}
2383 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` \* 形式は段抜きのフロートです。

```
figure* 2384 \newenvironment{figure}%
2385     {\@float{figure}}%
2386     {\end@float}
```

```

2387 \newenvironment{figure*}%
2388         {\@dblfloat{figure}}%
2389         {\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。

```

2390 %<!*book&!report>
2391 \newcounter{table}
2392 \renewcommand\thetable{\@arabic\c@table}
2393 %<\/!book&!report>
2394 %<*book|report>
2395 \newcounter{table}[chapter]
2396 \renewcommand \thetable
2397     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2398 %<\/book|report>

```

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

```

\ext@table 2399 \def\fps@table{tbp}
\fnun@table 2400 \def\ftype@table{2}
2401 \def\ext@table{lot}
2402 \def\fnun@table{\tablename\nobreak\thetable}

```

`table` \* は段抜きのフロートです。

```

table* 2403 \newenvironment{table}%
2404         {\@float{table}}%
2405         {\end@float}
2406 \newenvironment{table*}%
2407         {\@dblfloat{table}}%
2408         {\end@dblfloat}

```

## 8.6 キャプション

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

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

```

2409 \newlength\abovecaptionskip
2410 \newlength\belowcaptionskip
2411 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2412 \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。

```
2413 %<*!jspf>
2414 % \long\def\@makecaption#1#2{\small
2415 %   \advance\leftskip 10\jsc@mmm
2416 %   \advance\rightskip 10\jsc@mmm
2417 %   \vskip\abovecaptionskip
2418 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2419 %   \ifdim \wd\@tempboxa >\hsize
2420 %     #1\hskip1\jsZw\relax #2\par
2421 %   \else
2422 %     \global \@minipagefalse
2423 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2424 %   \fi
2425 %   \vskip\belowcaptionskip}}
2426 \long\def\@makecaption#1#2{\small
2427   \advance\leftskip .0628\linewidth
2428   \advance\rightskip .0628\linewidth
2429   \vskip\abovecaptionskip
2430   \sbox\@tempboxa{#1\zwspace#2}%
2431   \ifdim \wd\@tempboxa <\hsize \centering \fi
2432   #1\zwspace#2\par
2433   \vskip\belowcaptionskip}}
2434 %</!jspf>
2435 %<*jspf>
2436 \long\def\@makecaption#1#2{%
2437   \vskip\abovecaptionskip
2438   \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2439   \ifdim \wd\@tempboxa >\hsize
2440     {\small\sffamily
2441       \list{#1}{%
2442         \renewcommand{\makelabel}[1]{##1\hfil}
2443         \itemsep \z@
2444         \itemindent \z@
2445         \labelsep \z@
2446         \labelwidth 11\jsc@mmm
2447         \listparindent\z@
2448         \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2449   \else
2450     \global \@minipagefalse
2451     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2452   \fi
2453   \vskip\belowcaptionskip}
2454 %</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。

---

```
2455 \newif\if@jsc@warnoldfontcmd
2456 \@jsc@warnoldfontcmdtrue
2457 \newif\if@jsc@warnoldfontcmdexception
2458 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand

2459 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2460   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2461   \DeclareOldFontCommand{#1}{%
2462     \bxjs@oldfontcmd{#1}#2%
2463   }{%
2464     \bxjs@oldfontcmd{#1}#3%
2465   }%
2466 }
2467 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2468   \ClassInfo\bxjs@clsname
2469   {Old font command '\string#1' is used!!\MessageBreak
2470     The first occurrence is}%
2471 }

\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。

2472 \newcommand*{\allowoldfontcommands}{%
2473   \@jsc@warnoldfontcmdexceptiontrue}
2474 \newcommand*{\disallowoldfontcommands}{%
2475   \@jsc@warnoldfontcmdexceptionfalse}
```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2476 \let\bxjs@oldfontcmd@list\@empty
2477 \def\bxjs@oldfontcmd#1{%
2478   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2479 \def\bxjs@oldfontcmd@a#1#2{%
2480   \if@jsc@warnoldfontcmdexception\else
2481     \global\@jsc@warnoldfontcmdfalse
2482     \ifx#1\relax
2483       \global\let#1=t%
2484       \jsc@warnoldfontcmd{#2}%
2485     \fi
2486   \fi}
2487 \def\bxjs@warnoldfontcmd@final{%
2488 % \par
2489 \global\let\bxjs@warnoldfontcmd@final\@empty
2490 \let\@tempa\@empty
2491 \def\do##1{%
2492   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2493     \edef\@tempa{\@tempa \space\string##1}\fi}
2494 \bxjs@oldfontcmd@list
2495 \ifx\@tempa\@empty\else
2496   \ClassWarningNoLine\bxjs@clsname
2497   {Some old font commands were used in text\MessageBreak
2498    (see the log file for detail)}%
2499   \ClassInfo\bxjs@clsname
2500   {Some old font commands were used in text:\MessageBreak
2501    \space\@tempa\MessageBreak
2502    You should note, that since 1994 LaTeX2e provides a\MessageBreak
2503    new font selection scheme called NFSS2 with several\MessageBreak
2504    new, combinable font commands. The
2505    class provides\MessageBreak
2506    the old font commands
2507    only for compatibility%
2508    \@gobble}%
2509 \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。

```

2510 \def\bxjs@warnoldfontcmd@kick@final{%
2511   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2512 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

---

`\mc` フォントファミリーを変更します。

```

\gt 2513 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
2514 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\rm 2515 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff

```

`\tt`

```
2516 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
2517 \jsc@DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
```

`\bf` ボールドシリーズにします。通常のミーディウムシリーズに戻るコマンドは `\mdseries` です。

```
2518 \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るコマンドは `\upshape` です。

```
2519 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2520 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2521 \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 2522 \DeclareRobustCommand*{\cal}{\@fontswitch\relax\mathcal}
2523 \DeclareRobustCommand*{\mit}{\@fontswitch\relax\mathnormal}
```

## 10 相互参照

### 10.1 目次の類

`\section` コマンドは `.toc` ファイルに次のような行を出力します。

```
\contentsline{section}{タイトル}{ページ}
```

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

となります。この「番号」は `\thesection` コマンドで生成された見出し番号です。

`figure` 環境の `\caption` コマンドは `.lof` ファイルに次のような行を出力します。

```
\contentsline{figure}{\numberline{番号}{キャプション}{ページ}}
```

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

`\contentsline{...}` は `\l@...` というコマンドを実行するので、あらかじめ `\l@chapter`, `\l@section`, `\l@figure`などを定義しておかなければなりません。これらの多くは `\@dottedtocline` コマンドを使って定義します。これは

```
\@dottedtocline{レベル}{インデント}{幅}{タイトル}{ページ}
```

という書式です。

**レベル** この値が `tocdepth` 以下のときだけ出力されます。`\chapter` はレベル 0, `\section` はレベル 1, 等々です。

インデント 左側の字下げ量です。

幅 「タイトル」に `\numberline` コマンドが含まれる場合、節番号が入る箱の幅です。

`\@pnumwidth` ページ番号の入る箱の幅です。

`\@tocrmarg` 右マージンです。`\@tocrmarg ≥ \@pnumwidth` とします。

`\@dotsep` 点の間隔です (単位 mu)。

`\c@tocdepth` 目次ページに出力する見出しレベルです。元は `article` で 3, その他で 2 でしたが, ここでは一つずつ減らしています。

```
2524 \newcommand\@pnumwidth{1.55em}
2525 \newcommand\@tocrmarg{2.55em}
2526 \newcommand\@dotsep{4.5}
2527 %<!book&!report>\setcounter{tocdepth}{2}
2528 %<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)

```
2529 \newdimen\jsc@tocl@width
2530 \newcommand{\tableofcontents}{%
2531 %<*book|report>
2532 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2533 \settowidth\@tempdima{\headfont\appendixname}%
2534 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2535 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2536 \if@twocolumn
2537 \@restonecoltrue\onecolumn
2538 \else
2539 \@restonecolfalse
2540 \fi
2541 \chapter*{\contentsname}%
2542 \@mkboth{\contentsname}{}%
2543 %</book|report>
2544 %<*!book&!report>
2545 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2546 \settowidth\@tempdima{\headfont\appendixname}%
2547 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2548 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2549 \section*{\contentsname}%
2550 \@mkboth{\contentsname}{\contentsname}%
2551 %</!book&!report>
2552 \@starttoc{toc}%
2553 %<book|report> \if@restonecol\twocolumn\fi
2554 }
```

`\l@part` 部の目次です。

```
2555 \newcommand*{\l@part}[2]{%
2556   \ifnum \c@tocdepth >-2\relax
2557   %<!book&!report>   \addpenalty\@secpenalty
2558   %<book|report>     \addpenalty{-\@highpenalty}%
2559   \addvspace{2.25em \@plus\p@?}%
2560   \begingroup
2561     \parindent \z@
2562   %   \@pnumwidth should be \@tocrmarg
2563   %   \rightskip \@pnumwidth
2564   \rightskip \@tocrmarg
2565   \parfillskip -\rightskip
2566   {\leavevmode
2567     \large \headfont
2568     \setlength\@lnumwidth{4\jsZw}%
2569     #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2570   \nobreak
2571   %<book|report>     \global\@nobreaktrue
2572   %<book|report>     \everypar{\global\@nobreakfalse\everypar{}}%
2573   \endgroup
2574   \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@toctocwidth` から決めるようにしてみました。(by ts)

```
2575 %<*book|report>
2576 \newcommand*{\l@chapter}[2]{%
2577   \ifnum \c@tocdepth >\m@ne
2578     \addpenalty{-\@highpenalty}%
2579     \addvspace{1.0em \@plus\p@?}
2580   %   \vskip 1.0em \@plus\p@   % book.cls では↑がこうなっている
2581   \begingroup
2582     \parindent\z@
2583   %   \rightskip\@pnumwidth
2584   \rightskip\@tocrmarg
2585   \parfillskip-\rightskip
2586   \leavevmode\headfont
2587   %   % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2588   \setlength\@lnumwidth{\jsc@toctocwidth}\advance\@lnumwidth 2.683\jsZw
2589   \advance\leftskip\@lnumwidth \hskip-\leftskip
2590   #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2591   \penalty\@highpenalty
2592   \endgroup
2593   \fi}
2594 %</book|report>
```

`\l@section` 節の目次です。

```
2595 %<*!book&!report>
```

```

2596 \newcommand*{\l@section}[2]{%
2597   \ifnum \c@tocdepth >\z@
2598     \addpenalty{\@secpenalty}%
2599     \addvspace{1.0em \@plus\p@?}%
2600     \begingroup
2601       \parindent\z@
2602 %     \rightskip\@pnumwidth
2603     \rightskip\@tocrmarg
2604     \parfillskip-\rightskip
2605     \leavevmode\headfont
2606 %     % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2607     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2608     \advance\leftskip\@lnumwidth \hskip-\leftskip
2609     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2610   \endgroup
2611   \fi}
2612 %</*!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。

```

2613 %<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 2614 %</*!book&!report>
2615 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2616 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2617 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2618 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2619 %
2620 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2621 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2622 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2623 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2624 %
2625 \newcommand*{\l@subsection}{%
2626   \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2627   \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2628 \newcommand*{\l@subsubsection}{%
2629   \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2630   \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2631 \newcommand*{\l@paragraph}{%
2632   \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2633   \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2634 \newcommand*{\l@subparagraph}{%
2635   \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2636   \@dottedtocline{5}{\@tempdima}{6\jsZw}}

```

```

2637 %</!book&!report>
2638 %<*book|report>
2639 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2640 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2641 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2642 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2643 \newcommand*{\l@section}{%
2644     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2645     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2646 \newcommand*{\l@subsubsection}{%
2647     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2648     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2649 \newcommand*{\l@subsubsection}{%
2650     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2651     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2652 \newcommand*{\l@paragraph}{%
2653     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2654     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2655 \newcommand*{\l@subparagraph}{%
2656     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2657     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2658 %</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。

```

2659 \newdimen\@lnumwidth
2660 \def\numberline#1{\hbext@{\@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。

```

2661 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2662     \vskip \z@ \@plus.2\p@?
2663     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2664         \parindent #2\relax\@afterindenttrue
2665         \interlinepenalty\@M
2666         \leavevmode
2667         \@lnumwidth #3\relax
2668         \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2669         {#4}\nobreak
2670         \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2671             mu$}\hfill \nobreak\hbext@{\@pnumwidth{#5}}
2672         \hfil\normalfont \normalcolor #5}\par}\fi}

```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2673 \newcommand{\listoffigures}{%
2674 %<*book|report>
2675 \if@twocolumn\@restonecoltrue\onecolumn
2676 \else\@restonecolfalse\fi
2677 \chapter*{\listfigurename}%
2678 \mkboth{\listfigurename}{}%
2679 %</book|report>
2680 %<*&!book&!report>
2681 \section*{\listfigurename}%
2682 \mkboth{\listfigurename}{\listfigurename}%
2683 %<*&!book&!report>
2684 \@starttoc{lof}%
2685 %<book|report> \if@restonecol\twocolumn\fi
2686 }

```

`\l@figure` 図目次の項目を出力します。

```
2687 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```

2688 \newcommand{\listoftables}{%
2689 %<*book|report>
2690 \if@twocolumn\@restonecoltrue\onecolumn
2691 \else\@restonecolfalse\fi
2692 \chapter*{\listtablename}%
2693 \mkboth{\listtablename}{}%
2694 %</book|report>
2695 %<*&!book&!report>
2696 \section*{\listtablename}%
2697 \mkboth{\listtablename}{\listtablename}%
2698 %<*&!book&!report>
2699 \@starttoc{lot}%
2700 %<book|report> \if@restonecol\twocolumn\fi
2701 }

```

`\l@table` 表目次は図目次と同じです。

```
2702 \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。

```

2703 \newdimen\bibindent
2704 \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。



```

2705 \newenvironment{thebibliography}[1]{%
2706   \@jsc@warnoldfontcmdexceptiontrue
2707   \global\let\presectionname\relax
2708   \global\let\postsectionname\relax
2709   %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2710   %<*kiyou>
2711   \vspace{1.5\baselineskip}
2712   \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2713   \vspace{0.5\baselineskip}
2714   %</kiyou>
2715   %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2716   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2717   \list{\@biblabel{\@arabic{c@enumiv}}}%
2718     {\settowidth\labelwidth{\@biblabel{#1}}%
2719     \leftmargin\labelwidth
2720     \advance\leftmargin\labelsep
2721     \@openbib@code
2722     \usecounter{enumiv}%
2723     \let\p@enumiv\@empty
2724     \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2725   %<kiyou> \small
2726   \sloppy
2727   \clubpenalty4000
2728   \@clubpenalty\clubpenalty
2729   \widowpenalty4000%
2730   \sfcode`.\@m}
2731   {\def\@noitemerr
2732     {\@latex@warning{Empty `thebibliography' environment}}}%
2733   \endlist}

```

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

```

2734 \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。

```

2735 \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。

```

2736 % \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。かつこの前後に入るグルーを `\jsInhibitGlue` で取って  
 いますので、オリジナル同様、Knuth-`\cite{knu}` のように半角空白で囲んでください。

```

2737 % \def\@citex[#1]#2{\leavevmode
2738 %   \let\@citea\@empty

```

```

2739 % \cite{\for\citeb:=#2\do
2740 %   {\citea\def\citea{, \inhibitglue\penalty\@m\ }%
2741 %   \edef\citeb{\expandafter\@firstofone\citeb\@empty}%
2742 %   \if@filesw\immediate\write\@auxout{\string\citation{\citeb}}\fi
2743 %   \ifundefined{b@\citeb}{\mbox{\normalfont\bfseries ?}%
2744 %   \G@refundefinedtrue
2745 %   \@latex@warning
2746 %     {Citation `'\citeb' on page \thepage \space undefined}}%
2747 %   {\cite\ofmt{\csname b@\citeb\endcsname}}}{#1}}
2748 % \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。

```

2749 % \DeclareRobustCommand\cite{\unskip
2750 %   \ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2751 %   \def\cite#1#2{\$~\hbox{\scriptsize{#1\if@tempswa
2752 %     , \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)。

```

2753 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2754   \if@twocolumn
2755     \onecolumn\@restonecolfalse
2756   \else
2757     \clearpage\@restonecoltrue
2758   \fi
2759   \columnseprule.4pt \columnsep 2\jsZw
2760   \ifx\multicols\@undefined
2761 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}]%
2762 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2763 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2764 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2765   \else
2766     \ifdim\textwidth<\fullwidth
2767       \setlength{\evensidemargin}{\oddsidemargin}
2768       \setlength{\textwidth}{\fullwidth}
2769       \setlength{\linewidth}{\fullwidth}
2770 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}]%
2771 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2772 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2773 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2774   \else
2775 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}]%
2776 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2777 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2778 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%

```

```

2779     \fi
2780     \fi
2781 %<book|report>      \@mkboth{\indexname}{}%
2782 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2783     \plainifnotempty % \thispagestyle{plain}
2784     \parindent\z@
2785     \parskip\z@ \@plus .3\p@?\relax
2786     \let\item\@idxitem
2787     \raggedright
2788     \footnotesize\narrowbaselines
2789   }{
2790     \ifx\multicols\@undefined
2791       \if@restonecol\onecolumn\fi
2792     \else
2793       \end{multicols}
2794     \fi
2795     \clearpage
2796   }

```

`\@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 2797 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2798 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2799 \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。

```

2800 \newcommand{\indexspace}{\par \vskip 10\p@? \@plus5\p@? \@minus3\p@?\relax}

```

`\seenname` 索引の `\see`, `\seealso` コマンドで出力されるものです。デフォルトはそれぞれ *see*, *see also*

`\alsoname` という英語ですが, ここではとりあえず両方とも「→」に変えました。⇒ ( $\rightarrow$ )  
などでもいいでしょう。

```

2801 \newcommand\seenname{\if@english see\else →\fi}
2802 \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` のかわりに `\jsInhibitGlue` を使う。

---

```

2803 \ifx\pltx@foot@penalty\@undefined
2804   \let\footnotes@ve=\footnote
2805   \def\footnote{\jsInhibitGlue\footnotes@ve}
2806   \let\footnotemarks@ve=\footnotemark

```

```
2807 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2808 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

[2016-07-11] コミュニティ版 p<sub>L</sub>A<sub>T</sub>E<sub>X</sub> の変更に従いました (Thanks: 角藤さん)。p<sub>L</sub>A<sub>T</sub>E<sub>X</sub> の日付が 2016/04/17 より新しい場合は、このパッチが不要なのであてません。

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

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

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

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

```
2809 \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。

```
2810 % \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。

```
2811 \renewcommand{\footnoterule}{%
2812 \kern-2.6\p@? \kern-.4\p@
2813 \hrule width .4\columnwidth
2814 \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2815 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

```

2816 \long\def\@footnotetext{%
2817   \insert\footins\bgroup
2818     \normalfont\footnotesize
2819     \interlinepenalty\interfootnotelinepenalty
2820     \splittopskip\footnotesep
2821     \splitmaxdepth \dp\strutbox \floatingpenalty \MM
2822     \hsize\columnwidth \@parboxrestore
2823     \protected@edef\@currentlabel{%
2824       \csname p@footnote\endcsname\@thefnmark
2825     }%
2826     \color@begingroup
2827     \@makefntext{%
2828       \rule{z@\footnotesep}{\ignorespaces}%
2829       \futurelet\next\fo@t}
2830 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
2831           \else \let\next\fo@t\fi \next}
2832 \def\fo@@t{\bgroup\aftergroup\@foot\let\next}
2833 \def\fo@t#1{#1\@foot}
2834 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2835   \ifx\pltx@foot@penalty\@undefined\else
2836     \ifhmode\null\fi
2837     \ifnum\pltx@foot@penalty=z@\else
2838       \penalty\pltx@foot@penalty
2839       \pltx@foot@penalty\z@
2840     \fi
2841   \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。

```

2842 \newcommand\@makefntext[1]{%
2843   \advance\leftskip 3\jsZw
2844   \parindent 1\jsZw
2845   \noindent
2846   \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。

```

2847 % \def\@xfootnotenext[#1]{%
2848 %   \begingroup
2849 %     \ifnum#1>z@
2850 %       \csname c@\@mpfn\endcsname #1\relax
2851 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2852 %     \else
2853 %       \unrestored@protected@xdef\@thefnmark{}%

```

```

2854 %      \fi
2855 %      \endgroup
2856 %      \@footnotetext}

```

---

ここまでのコードは JS クラスを踏襲する。

---

## 11 段落の頭へのグルー挿入禁止

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

---

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

```
2857 \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。

```

2858 \def\everyparhook{\jsInhibitGlueAtParTop}
2859 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2860 \AtBeginDocument{\everypar{\everyparhook}}
2861 \fi

```

---

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

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

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

---

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

---

```

2862 \ifx j\jsEngine
2863 \def\@inhibitglue{%
2864   \futurelet\@let@token\@@inhibitglue}
2865 \begingroup
2866 \let\GDEF=\gdef
2867 \let\CATCODE=\catcode
2868 \let\ENDGROUP=\endgroup
2869 \CATCODE`k=12
2870 \CATCODE`a=12

```

```

2871 \CATCODE`n=12
2872 \CATCODE`j=12
2873 \CATCODE`i=12
2874 \CATCODE`c=12
2875 \CATCODE`h=12
2876 \CATCODE`r=12
2877 \CATCODE`t=12
2878 \CATCODE`e=12
2879 \GDEF\KANJI@CHARACTER{kanji character }
2880 \ENDGROUP
2881 \def\@@inhibitglue{%
2882   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
2883   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
2884   \def\jsc@ig@temp{#1}%
2885   \ifx\jsc@ig@temp\@empty
2886     \ifnum\the\inhibitxspcode`#2=2\relax
2887       \inhibitglue
2888     \fi
2889   \fi}
2890 \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。

---

```

2891 \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。

```

2892 \def\@doendpe{%
2893   \@endpetrue
2894   \def\par{%
2895     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2896   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

2897 \def\@setminipage{%
2898   \@minipagetrue
2899   \everypar{\@minipagefalse\everypar{\everyparhook}}%
2900 }

```

`\item` 命令の直後です。

```

2901 \def\@item[#1]{%
2902   \if@noperitem
2903     \@donoperitem
2904   \else

```

```

2905 \if@inlabel
2906 \indent \par
2907 \fi
2908 \ifhmode
2909 \unskip\unskip \par
2910 \fi
2911 \if@newlist
2912 \if@nobreak
2913 \@nbitem
2914 \else
2915 \addpenalty\@beginparpenalty
2916 \addvspace\@topsep
2917 \addvspace{-\parskip}%
2918 \fi
2919 \else
2920 \addpenalty\@itempenalty
2921 \addvspace\itemsep
2922 \fi
2923 \global\@inlabeltrue
2924 \fi
2925 \everypar{%
2926 \@minipagefalse
2927 \global\@newlistfalse
2928 \if@inlabel
2929 \global\@inlabelfalse
2930 {\setbox\z@\lastbox
2931 \ifvoid\z@
2932 \kern-\itemindent
2933 \fi}%
2934 \box\@labels
2935 \penalty\z@
2936 \fi
2937 \if@nobreak
2938 \@nobreakfalse
2939 \clubpenalty \@M
2940 \else
2941 \clubpenalty \@clubpenalty
2942 \everypar{\everyparhook}%
2943 \fi
2944 \everyparhook}%
2945 \if@noitemarg
2946 \@noitemargfalse
2947 \if@nmbrlist
2948 \refstepcounter\@listctr
2949 \fi
2950 \fi
2951 \sbox\@tempboxa{\makelabel{#1}}%
2952 \global\setbox\@labels\hbox{%
2953 \unhbox\@labels

```



```

2954 \hskip \itemindent
2955 \hskip -\labelwidth
2956 \hskip -\labelsep
2957 \ifdim \wd\@tempboxa >\labelwidth
2958 \box\@tempboxa
2959 \else
2960 \hbox to\labelwidth {\unhbox\@tempboxa}%
2961 \fi
2962 \hskip \labelsep}%
2963 \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。

```

2964 \def\@afterheading{%
2965 \@nbreaktrue
2966 \everypar{%
2967 \if@nbreak
2968 \@nbreakfalse
2969 \clubpenalty \@M
2970 \if@afterindent \else
2971 {\setbox\z@\lastbox}%
2972 \fi
2973 \else
2974 \clubpenalty \@clubpenalty
2975 \everypar{\everyparhook}%
2976 \fi\everyparhook}}

```

---

「`\everyparhook` 用のパッチ処理」はここまで。

---

```

2977 \fi

```

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

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

---

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

---

```

2978 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
2979 \AtEndOfPackage{%
2980 \def\@gnewline #1{%
2981 \ifvmode
2982 \@nolnerr
2983 \else

```

```

2984 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
2985 \jsInhibitGlue \ignorespaces
2986 \fi}
2987 }
2988 \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。

---

```

2989 \if@jslogo
2990 \IfFileExists{jslogo.sty}{%
2991 \RequirePackage{jslogo}%
2992 }{%
2993 \ClassWarningNoLine{bxjs@clsname}
2994 {The package 'jslogo' is not installed.\MessageBreak
2995 It is included in the recent release of\MessageBreak
2996 the 'jsclasses' bundle}
2997 }
2998 \fi

```

## 13 amsmath との衝突の回避

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

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

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

```

2999 %\let\ltx@ifnextchar\@ifnextchar
3000 %\def\ProvidesFile#1{%
3001 % \begingroup
3002 % \catcode\ 10 %
3003 % \ifnum \endlinechar<256 %

```

```

3004 %      \ifnum \endlinechar>\m@ne
3005 %      \catcode\endlinechar 10 %
3006 %      \fi
3007 %      \fi
3008 %      \@makeother\/%
3009 %      \@makeother\&%
3010 %      \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 3011 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3012 \newcommand{\postpartname}{\if@english\else 部\fi}
3013 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3014 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3015 \newcommand{\presectionname}{}% 第
\postsectionname 3016 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3017 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3018 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3019 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3020 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3021 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3022 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3023 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3024 %<jspf>\newcommand{\figurename}{Fig.~}
3025 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3026 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3027 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3028 \newcommand{\appendixname}{\if@english \else 付録\fi}
3029 %<!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_TEX_PRIMITIVES` が設定されている場合は“今日”が過去の日付になる可能性があるが、その場合、和暦表記は平成 2 年（1990 年）以降でのみサポートする。

※ “新元号” への対応？

---

```
\today
3030 \@tempswafalse
3031 \if p\jsEngine \@tempswatrue \fi
3032 \if n\jsEngine \@tempswatrue \fi
3033 \if@tempswa \expandafter\@firstoftwo
3034 \else \expandafter\@secondoftwo
3035 \fi
3036 {%
3037 % 欧文 8bitTeX の場合
3038 \newif\ifjsSeireki \jsSeirekitrue
3039 \def\bxjs@decl@Seireki@cmds{%
3040   \def\西暦{\jsSeirekitrue}%
3041   \def\和暦{\jsSeirekifalse}}
3042 \def\Seireki{\jsSeirekitrue}
3043 \def\Wareki{\jsSeirekifalse}
3044 \def\bxjs@if@use@seireki{%
3045   \ifjsSeireki \expandafter\@firstoftwo
3046   \else \expandafter\@secondoftwo \fi}
3047 }{%
3048 \newif\if 西暦 \西暦 true
3049 \def\bxjs@decl@Seireki@cmds{%
3050   \def\西暦{\西暦 true}%
3051   \def\和暦{\西暦 false}}
3052 \def\Seireki{\西暦 true}
3053 \def\Wareki{\西暦 false}
3054 \def\bxjs@if@use@seireki{%
3055   \if 西暦 \expandafter\@firstoftwo
3056   \else \expandafter\@secondoftwo \fi}
3057 }
3058 \bxjs@decl@Seireki@cmds
3059 % \bxjs@unxp
3060 \let\bxjs@unxp\@firstofone
3061 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
3062 % \bxjs@iai
3063 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3064   \def\bxjs@iai{\noexpand~}
3065 \else \def\bxjs@iai{}
3066 \fi
3067 % \heisei
3068 \newcount\heisei \heisei\year \advance\heisei-1988\relax
3069 % \today
3070 \edef\bxjs@today{%
3071   \if@english
3072     \ifcase\month\or
3073       January\or February\or March\or April\or May\or June\or
3074       July\or August\or September\or October\or November\or December\fi
```

```

3075     \space\number\day, \number\year
3076 \else
3077     \ifnum\heisei>\@ne
3078         \expandafter\noexpand\expandafter\bxjs@if@use@seireki
3079     \else \expandafter\@firstoftwo
3080     \fi {%
3081         \number\year\bxjs@iai\bxjs@unxp{年}%
3082         \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3083         \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3084     }{%
3085         \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
3086         \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3087         \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3088     }%
3089 \fi}
3090 \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。

```

3091 \AtBeginDocument{%
3092     \ifx\bbl@jpn@Seirekitrue\@undefined\else
3093         \bxjs@decl@Seireki@cmds
3094         \g@addto@macro\datejapanese{%
3095             \let\today\bxjs@today}%
3096     \fi}

```

---

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

```

3097 \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。

```

3098 %<slide>\pagestyle{empty}%
3099 %<article|report>\pagestyle{plain}%
3100 %<book>\pagestyle{headings}%
3101 \pagenumbering{arabic}
3102 \if@twocolumn
3103     \twocolumn
3104     \sloppy
3105     \flushbottom
3106 \else
3107     \onecolumn
3108     \raggedbottom
3109 \fi
3110 %<*slide>

```

```
3111 \renewcommand\familydefault{\sfdefault}
3112 \raggedright
3113 %</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。

```
3114 \catcode\?=12
3115 \ifx\bxjs@jadriver\relax\else
3116 \input{bxjsja-\bxjs@jadriver.def}
3117 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
3118 \bxjs@restore@jltrcc
3119 %</cls>
```

---

以上です。

## 付録 A 和文ドライバの仕様

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

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

用いられる「和文フォント」のメトリックに基づくのではなく、機械的に `\jsScale` × (フォントサイズ) であると定められている (フォントサイズ変更の度に再設定される)。従って、「和文コンポーネント」はこの設定と辻褄が合うように和文フォントサイズを調整する必要がある。ほとんどの場合、和文フォントを NFSS で規定する際に `\jsScale` の値をスケール値として与えれば上手くいく。

- `\jsFontSizeChanged` [マクロ] フォントサイズが変更された時に必ず呼び出されるマクロ。
- `\jsResetDimen` [マクロ] ★ 上記 `\jsFontSizeChanged` の中で呼び出される、ユーザ (和文モジュール) 用のフック。フォントサイズに依存するパラメタをここで設定することができる。既定の定義は空。

以下で標準で用意されている和文ドライバの実装を示す。

```
3120 %<*drv>
```

## 付録 B 和文ドライバ : minimal

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

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

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

### B.1 補助マクロ

```
3121 %<*minimal>
```

```
3122 %% 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.
```

```
3123 \def\DeclareJaTextFontCommand#1#2{%
```

```
3124   \DeclareRobustCommand#1[1]{%
```

```
3125     \relax
```

```
3126     \ifmmode \expandafter\nfss@text \fi
```

```
3127     {#2##1}}%
```

```
3128 }
```

```
\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。
```

```
3129 \def\DeclareJaMathFontCommand#1#2{%
```

```
3130   \DeclareRobustCommand#1[1]{%
```

```
3131     \relax
```

```
3132     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3133     \nfss@text{\fontfamily\familydefault
```

```
3134       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3135       #2##1}}%
```



```

3136 }%
3137 }

```

`\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。

```

3138 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3139 \@onlypreamble\bxjs@if@sf@default
3140 \def\bxjs@if@sf@default#1{%
3141   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3142   \AtBeginDocument{%
3143     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3144 }

```

`\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。

```

3145 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3146 \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。

```

3147 \def\jsLetHeadChar#1#2{%
3148   \begingroup
3149   \escapechar=`\ %
3150   \let\bxjs@tmpa={% brace-match-hack
3151   \bxjs@let@hchar@exp#2}%
3152   \endgroup
3153   \let#1\bxjs@g@tmpa}
3154 \def\bxjs@let@hchar@exp{%
3155   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3156 \def\bxjs@let@hchar@exp@a{%
3157   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3158   \bxjs@let@hchar@out\let\relax
3159   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3160   \bxjs@let@hchar@out\let\space%
3161   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3162   \bxjs@let@hchar@out\let\@backslashchar
3163   }{\bxjs@let@hchar@exp@b}}}%
3164 \def\bxjs@let@hchar@exp@b#1{%
3165   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3166 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3167   %\message{<#1#2>}%
3168   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3169 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3170 \bxjs@let@hchar@out\let\relax
3171 }{%else
3172 \expandafter\bxjs@let@hchar@exp
3173 }%
3174 }{%else
3175 \bxjs@let@hchar@chr#1%
3176 }}
3177 \def\bxjs@let@hchar@chr#1{%
3178 \bxjs@let@hchar@out\def{{#1}}}}
3179 \def\bxjs@let@hchar@out#1#2{%
3180 \global#1\bxjs@g@tmpa#2\relax
3181 \toks@{bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3182 \chardef\bxjs@let@hchar@csta=128
3183 \chardef\bxjs@let@hchar@cstb=192
3184 \chardef\bxjs@let@hchar@cstc=224
3185 \chardef\bxjs@let@hchar@cstd=240
3186 \chardef\bxjs@let@hchar@cste=248
3187 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3188 \def\bxjs@let@hchar@chr@ue#1{%
3189 \@tempcnta=#1\relax
3190 %\message{\the\@tempcnta}%
3191 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3192 \bxjs@let@hchar@chr@ue@a#1%
3193 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3194 \bxjs@let@hchar@out\let\relax
3195 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3196 \bxjs@let@hchar@chr@ue@b
3197 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3198 \bxjs@let@hchar@chr@ue@c
3199 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3200 \bxjs@let@hchar@chr@ue@d
3201 }{%else
3202 \bxjs@let@hchar@out\let\relax
3203 }}}}}
3204 \def\bxjs@let@hchar@chr@ue@a#1{%
3205 \bxjs@let@hchar@out\def{{#1}}}}
3206 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3207 \bxjs@let@hchar@out\def{{#1#2}}}}
3208 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3209 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3210 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3211 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

## B.2 (u)pTeX 用の設定

```
3212 \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。

```
3213 \def\bxjs@let@hchar@chr@pp#1{%
3214   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
3215 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
3216   \%message{(#1)}%
3217   \bxjs@cond\if#1t\fi{%
3218     \bxjs@let@hchar@chr@ue#3%
3219   }{%else
3220     \bxjs@let@hchar@out\def{#3}}%
3221   }}
3222 \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。

```
3223 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3224 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3225 \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。

```
3226 \@onlypreamble\bxjs@declarefontshape
3227 \ifjsWithupTeX
3228 \def\bxjs@declarefontshape{%
3229   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3230   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3231   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3232   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3233 }
3234 \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。

```
3235 \else
3236 \def\bxjs@declarefontshape{%
3237   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3238   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3239   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3240 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3241 }
3242 \def\bxjs@sizereference{jis}
3243 \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。

```

3244 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
3245   \def\bxjs@y{#5}}
3246 \ifjsWithpTeXng \def\bxjs@y{10}%
3247 \else
3248 \expandafter\expandafter\expandafter\bxjs@tmpa
3249 \expandafter\string\the\jfont\relax
3250 \fi
3251 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3252               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3253   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
3254    \expandafter\let\csname\bxjs@x/\bxjs@y\endcsname=\@undefined}

```

■和文フォントスケールの補正 実は、pTeX の標準的な和文フォント（JFM のこと、例えば jis）では、指定された `\jsScale`（この値を  $s$  とする）をそのまま使って定義すると期待通りの大きさにならない。これらの JFM では 1zw の大きさが指定されたサイズではなく既にスケール（この値を  $f$  とする；jis では 0.962216 倍）が掛けられた値になっているからである。そのため、ここでは  $s/f$  を求めてその値をマクロ `\bxjs@scale` に保存する。

```

3255 \begingroup
3256 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3257 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3258 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3259 % 幅が丁度 10pt なら補正は不要
3260 \ifdim\wd\z@=10pt
3261   \global\let\bxjs@scale\jsScale
3262 \else
3263 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3264   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3265   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3266   \bxjs@invscale\@tempdima\bxjs@tmpa
3267   \xdef\bxjs@scale{\strip@pt\@tempdima}
3268 \fi
3269 \endgroup
3270 %\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。

```

3271 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3272 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3273 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3274 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3275 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3276 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3277 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3278 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3279 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3280 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3281 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3282 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3283 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3284 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3285 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}

```

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

```

3286 \DeclareRobustCommand\rmfamily
3287   {\not@math@alphabet\rmfamily\mathrm
3288     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3289 \DeclareRobustCommand\sffamily
3290   {\not@math@alphabet\sffamily\mathsf
3291     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3292 \DeclareRobustCommand\ttfamily
3293   {\not@math@alphabet\ttfamily\mathtt
3294     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3295 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3296 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3297 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3298 \fi
3299 \bxjs@if@sf@default{%
3300   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3301 \selectfont

```

## ■パラメタの設定

```

3302 \prebreakpenalty\jis"2147=10000
3303 \postbreakpenalty\jis"2148=10000
3304 \prebreakpenalty\jis"2149=10000
3305 \inhibitxspcode`!=1
3306 \inhibitxspcode`〒=2
3307 \xspcode`+=3
3308 \xspcode`%=3

```

"80～"FF の範囲の \xspcode を 3 に変更。

```

3309 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3310   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3311 \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。

```
3312 \begingroup
3313 \catcode`\!=0
```

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

```
3314 \gdef\bxjs@ptex@dir{%
3315   !iftdir t%
3316   !else!ifydir y%
3317   !else ?%
3318   !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。

```
3319 % 古い \@makefnmark の定義
3320 \long\def\bxjs@tmpa{\hbox{%
3321   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3322   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3323 \ifx\@makefnmark\bxjs@tmpa
3324 \long\gdef\@makefnmark{%
3325   !ifydir \hbox{}\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}%
3326   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3327 \fi
3328 \endgroup
```

### B.3 pdfTeX 用の処理

```
3329 \else\ifx p\jsEngine
3330 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3331 \@onlypreamble\bxjs@cjk@loaded
3332 \def\bxjs@cjk@loaded{%
3333   \def\@footnotemark{%
3334     \leavevmode
3335     \ifhmode
3336       \edef\@x@sf{\the\spacefactor}%
3337       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3338         \unkern\unkern
3339         \ifdim\lastskip>\z@ \unskip \fi
3340       \fi\fi
3341       \nobreak
3342     \fi
3343     \@makefnmark
3344     \ifhmode \spacefactor\@x@sf \fi
3345     \relax}%
```

```

3346 \let\bxjs@CJK@loaded\relax
3347 }
3348 \AtBeginDocument{%
3349 \ifpackageloaded{CJK}{%
3350 \bxjs@CJK@loaded
3351 }{}%
3352 }

```

## B.4 XeTeX 用の処理

```

3353 \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。

```

3354 \def\bxjs@let@hchar@chr#1{%
3355 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3356 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3357 \bxjs@let@hchar@chr@xe
3358 }{\bxjs@let@hchar@out\def{{#1}}}}
3359 \def\bxjs@let@hchar@chr@xe#1{%
3360 \lccode`0=`#1\relax
3361 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3362 \ifx\XeTeXgenerateactualtext\@undefined\else
3363 \def\bxjs@do@precisetext{%
3364 \XeTeXgenerateactualtext=\@ne}
3365 \fi

```

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

```

3366 \@onlypreamble\bxjs@do@simplejasetup
3367 \def\bxjs@do@simplejasetup{%
3368 \ifnum\XeTeXinterchartokenstate>\z@
3369 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3370 \jsSimpleJaSetup
3371 \ClassInfo\bxjs@clsname
3372 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3373 \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

3374 \newcommand*{\jsSimpleJaSetup}{%
3375 \XeTeXlinebreaklocale "ja"\relax
3376 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3377 \XeTeXlinebreakpenalty=0\relax}

```

## B.5 後処理（エンジン共通）

```

3378 \fi\fi\fi

```

simplejasetup オプションの処理。

```

3379 \ifx\bxjs@do@simplejasetup\@undefined\else
3380   \AtBeginDocument{%
3381     \ifbxjs@simplejasetup
3382       \bxjs@do@simplejasetup
3383     \fi}
3384 \fi

precisetext オプションの処理。
3385 \ifbxjs@precisetext
3386   \ifx\bxjs@do@precisetext\@undefined
3387     \ClassWarning\bxjs@clsname
3388       {The current engine does not support the\MessageBreak
3389         'precisetext' option\@gobble}
3390   \else
3391     \bxjs@do@precisetext
3392   \fi
3393 \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。

```

3394 \@onlypreamble\bxjs@check@everyparhook
3395 \def\bxjs@check@everyparhook{%
3396   \ifx\jsInhibitGlueAtParTop\@empty
3397     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3398     \ifx\everyparhook\bxjs@tmpa
3399       \let\everyparhook\@empty
3400     \fi
3401   \fi}
3402 \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。

```

3403 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3404 \let\bxjs@everypar\everypar
3405 \newtoks\everypar
3406 \everypar\bxjs@everypar

```

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

```

3407 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3408 \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。



3409 \ifbxjs@fancyhdr

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

3410 \@onlypreamble\bxjs@adjust@fancyhdr

3411 \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。

3412 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%

3413 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%

3414 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3415 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3416 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3417 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3418 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%

3419 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%

3420 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3421 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3422 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3423 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3424 \def\bxjs@tmpa{\rm\thepage\strut}%

3425 \def\bxjs@tmpb{\thepage\strut}%

3426 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi

3427 \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。

3428 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth

3429 \setlength{\@tempdima}{\fullwidth-\textwidth}%

3430 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%

3431 }\bxjs@tmpa

3432 \fi\fi

3433 \PackageInfo\bxjs@clsname

3434 {Patch to fancyhdr is applied\@gobble}}

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

3435 \def\bxjs@pagestyle@hook{%

3436 \@ifpackageloaded{fancyhdr}{%

3437 \bxjs@adjust@fancyhdr

3438 \global\let\bxjs@adjust@fancyhdr\relax

3439 }{}}

\pagestyle にフックを入れ込む。

3440 \let\bxjs@org@pagestyle\pagestyle

3441 \def\pagestyle{%

3442 \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。

```
3443 \AtBeginDocument{%
3444   \bxjs@pagestyle@hook
3445   \global\let\bxjs@pagestyle@hook\relax}
```

#### ■和文空白命令

```
3446 \ifbxjs@jaspace@cmd
```

`\jaenspace` 半角幅の水平空き。

```
3447   \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。

```
3448   \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。

```
3449   \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3450   \DeclareRobustCommand*{\jaspace}[1]{%
3451     \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3452     \ClassError\bxjs@clsname
3453     {Unknown jaspacespace: #1}{\@eha}%
3454     \else
3455     \csname bxjs@jaspace@@#1\endcsname
3456     \fi}
3457   \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3458   \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3459   \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3460 \fi
```

終わり。

```
3461 \fi
```

以上で終わり。

```
3462 %</minimal>
```

## 付録 C 和文ドライバ : standard

標準のドライバ。

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

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

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

## C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
3463 %<*standard>
3464 %% このファイルは日本語文字を含みます
3465 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3466 \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。

`\bxjs@jp@jismmiv` 2004JIS 字形を優先させるか。

```
3467 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3468 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3469 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3470 \define@key{bxjsStd}{jis2004}[true]{%
3471   \bxjs@set@keyval{jis2004}{#1}{}}
```

`\bxjs@jp@font` フォントパッケージの追加オプション。

```
3472 \let\bxjs@jp@font\@empty

jis2004 オプションの処理。
3473 \define@key{bxjsStd}{font}{%
3474   \edef\bxjs@jp@font{#1}}
```

実際の `japaram` の値を適用する。

```
3475 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3476 \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。

```
3477 \ifbxjs@jp@jismmiv
3478 \g@addto@macro\@classoptionslist{,jis2004}
3479 % \ifpackagewith 判定への対策
3480 \PassOptionsToPackage{jis2004}{otf}
3481 \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。

```
3482 \ifjsWitheTeX
```

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

\jQ \jQ と \jH はともに 0.25 mm に等しい。

```
\jH 3483 \@tempdima=0.25mm
3484 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3485 \let\jH\jQ
```

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

```
\trueH 3486 \ifjsc@mag
3487 \@tempdimb=\jsBaseFontSize\relax
3488 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3489 \@tempdima=2.5mm
3490 \bxjs@invscale\@tempdima\bxjs@tmpa
3491 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3492 \@tempdima=10pt
3493 \bxjs@invscale\@tempdima\bxjs@tmpa
3494 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3495 \else \let\trueQ\jQ \let\bxjs@truept\p@
3496 \fi
3497 \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。

```
3498 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3499 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3500 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3501 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3502 \fi
```

\jsSetQHLengh \jsSetQHLengh\CS{〈長さ式〉} : \setlength の変種で、通常の calc の長さ式の代わりに、〈実数〉Q・〈実数〉H・〈実数〉trueQ・〈実数〉trueH が記述できる。

```
3503 \def\jsSetQHLengh#1#2{%
```

```

3504 \begingroup
3505   \bxjs@parse@qh{#2}%
3506   \ifx\bxjs@tmpb\relax \setlength\@tempdima{#2}%
3507   \else \@tempdima\bxjs@tmpb\relax
3508   \fi
3509   \xdef\bxjs@g@tmpa{\the\@tempdima}%
3510 \endgroup
3511 #1=\bxjs@g@tmpa\relax}

```

`\bxjs@parse@qh` `jsSetQHLength` の下請け。#1 が Q/H/trueQ/trueH で終わる場合、`\bxjs@tmpb` にそれに等しい寸法の表現を返す。それ以外では、`\bxjs@tmpb` は `\relax` になる。

```

3512 \if j\jsEngine

```

(u)pL<sup>A</sup>T<sub>E</sub>X ではこの処理は不要なので無効化する。

```

3513 \def\bxjs@parse@qh#1{%
3514   \let\bxjs@tmpb\relax}
3515 \else
3516 \def\bxjs@parse@qh#1{%
3517   \def\bxjs@tmpa{#1}\let\bxjs@tmpb\relax
3518   \bxjs@parse@qh@a{trueQ}\trueQ
3519   \bxjs@parse@qh@a{trueH}\trueH
3520   \bxjs@parse@qh@a{Q}\jQ
3521   \bxjs@parse@qh@a{H}\jH}
3522 \def\bxjs@parse@qh@a#1#2{%
3523   \ifx\bxjs@tmpb\relax
3524     \def\bxjs@next##1#1\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}}%
3525     \expandafter\bxjs@next\bxjs@tmpa\@nil#1\@nil\@nnil{#2}%
3526     \fi}
3527 \def\bxjs@parse@qh@b#1#2#3{%
3528   \ifx\@nnil#2\@nnil\else \def\bxjs@tmpb{#1#3}\fi}
3529 \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。

```

3530 \def\jafontsize#1#2{%
3531   \begingroup
3532     \bxjs@jafontsize@a{#1}%
3533     \@tempdimb\jsInverseScale\@tempdima
3534     \bxjs@jafontsize@a{#2}%
3535     \xdef\bxjs@g@tmpa{%
3536       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3537   \endgroup\bxjs@g@tmpa}
3538 \def\bxjs@jafontsize@a#1{%
3539   \bxjs@parse@qh{#1}%
3540   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3541   \@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。

```
3542 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3543 \newcommand*\setkanjiskip[1]{%
```

```
3544   \edef\bxjs@kanjiskip{#1}%
```

```
3545   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3546 \newcommand*\getkanjiskip{%
```

```
3547   \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)autospaceing` での制御を用いるのでこの変数は常に真とする。

```
3548 \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 3549 \bxjs@robust@def\bxjs@enable@kanjiskip{%
```

```
3550   \bxjs@kanjiskip@enabledtrue
```

```
3551   \bxjs@reset@kanjiskip}
```

```
3552 \bxjs@robust@def\bxjs@disable@kanjiskip{%
```

```
3553   \bxjs@kanjiskip@enabledfalse
```

```
3554   \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。

```
3555 \bxjs@robust@def\bxjs@reset@kanjiskip{%
```

```
3556   \ifbxjs@kanjiskip@enabled
```

```
3557     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
```

```
3558   \else \@tempskipa\z@
```

```
3559   \fi
```

```
3560   \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 3561 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3562 \newcommand*\setxkanjiskip[1]{%
```

```
3563   \edef\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3564   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3565 \newcommand*\getxkanjiskip{%
```

```
3566   \bxjs@xkanjiskip}
```

```
\bxjs@disable@xkanjiskip 3567 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3568 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3569   \bxjs@xkanjiskip@enabledtrue
```

```
3570   \bxjs@reset@xkanjiskip}
```

```
3571 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3572   \bxjs@xkanjiskip@enabledfalse
```

```
3573   \bxjs@reset@xkanjiskip}
```

```
3574 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```
3575   \ifbxjs@xkanjiskip@enabled
```

```
3576     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
```

```

3577 \else \@tempskipa\z@
3578 \fi
3579 \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。

```

3580 \g@addto@macro\jsResetDimen{%
3581 \bxjs@reset@kanjiskip
3582 \bxjs@reset@xkanjiskip}
3583 \let\bxjs@apply@kanjiskip\relax
3584 \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。

```

3585 \@onlypreamble\bxjs@adjust@jafont
3586 \def\bxjs@adjust@jafont#1{%
3587 \ifx\jsJaFont\bxjs@@auto
3588 \bxjs@get@kanjiEmbed
3589 \ifx\bxjs@kanjiEmbed\relax
3590 \let\bxjs@tmpa\@empty
3591 \else
3592 \let\bxjs@tmpa\bxjs@kanjiEmbed
3593 \fi
3594 \else
3595 \let\bxjs@tmpa\jsJaFont
3596 \fi
3597 \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3598 \ClassWarningNoLine\bxjs@clsname
3599 {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3600 not available on the current situation}%
3601 \let\bxjs@tmpa\@empty
3602 \fi\fi
3603 }
3604 \def\bxjs@@auto{auto}
3605 \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。

```

3606 \let\bxjs@kanjiEmbed\relax

```

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

```

3607 \@onlypreamble\bxjs@get@kanjiEmbed
3608 \def\bxjs@get@kanjiEmbed{%
3609 \begingroup\setbox\z@=\hbox{%
3610 \global\let\bxjs@g@tmpa\relax

```

```

3611 \endlinechar\m@ne
3612 \let\do\@makeother\dospecials
3613 \catcode32=10 \catcode12=10 %form-feed
3614 \let\bxjs@tmpa\@empty
3615 \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3616 \ifeof\@inputcheck\else
3617 \read\@inputcheck to\bxjs@tmpa
3618 \closein\@inputcheck
3619 \fi
3620 \ifx\bxjs@tmpa\@empty\else
3621 \openin\@inputcheck="\bxjs@tmpa"\relax
3622 \@tempswatrue
3623 \loop\if@tempswa
3624 \read\@inputcheck to\bxjs@tmpa
3625 \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3626 \ifx\bxjs@tmpb\relax\else
3627 \global\let\bxjs@g@tmpa\bxjs@tmpb
3628 \@tempswafalse
3629 \fi
3630 \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3631 \ifx\bxjs@tmpb\relax\else
3632 \global\let\bxjs@g@tmpa\bxjs@tmpb
3633 \@tempswafalse
3634 \fi
3635 \ifeof\@inputcheck \@tempswafalse \fi
3636 \repeat
3637 \fi
3638 }\endgroup
3639 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3640 }
3641 \@onlypreamble\bxjs@get@ke@a
3642 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3643 \ifx$#1$\def\bxjs@tmpb{#2}%
3644 \else \let\bxjs@tmpb\relax
3645 \fi}
3646 \@onlypreamble\bxjs@get@ke@b
3647 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3648 \ifx$#1$\def\bxjs@tmpb{#2}%
3649 \else \let\bxjs@tmpb\relax
3650 \fi}

```

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

```

3651 \newcommand*\jachar[1]{%
3652 \begingroup

```

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

```

3653 \jsLetHeadChar\bxjs@tmpa{#1}%
3654 \ifx\bxjs@tmpa\relax
3655 \ClassWarningNoLine\bxjs@clsname

```



```

3656      {Illegal argument given to \string\jachar}%
3657      \else
3658      \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3659      \fi
3660      \endgroup}

```

\jsJaChar を \jachar と等価にする。

```

3661 \let\jsJaChar\jachar

```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```

3662 \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。

```

3663 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3664 \@onlypreamble\bxjs@fix@hyperref@unicode
3665 \def\bxjs@fix@hyperref@unicode#1{%
3666   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3667   \@namedef{KV@Hyp@bxjs/hook}##1{%
3668     \KV@Hyp@unicode{##1}%
3669     \def\KV@Hyp@unicode####1{%
3670       \expandafter\ifx\csname if##1\expandafter\endcsname
3671       \csname if####1\endcsname\else
3672       \ClassWarningNoLine{bxjs@clsname
3673       {Blcoked hyperref option 'unicode=####1'}%
3674       \fi
3675     }%
3676   }%
3677 }

```

\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。

```

3678 \@onlypreamble\jsCheckHyperrefUnicode
3679 \let\jsCheckHyperrefUnicode\@empty
3680 \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。

```

3681 \@onlypreamble\bxjs@check@hyperref@unicode
3682 \def\bxjs@check@hyperref@unicode#1{%
3683   \g@addto@macro\jsCheckHyperrefUnicode{%
3684     \@tempwafalse
3685     \begingroup
3686       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3687       \aftergroup\@tempwattrue \fi
3688       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3689       \csname if#1\endcsname
3690       \aftergroup\@tempwattrue \fi
3691     \endgroup

```

```

3692 \if@tempswa\else
3693 \ClassError\bxjs@clsname
3694 {The value of hyperref 'unicode' key is not suitable\MessageBreak
3695 for the present engine (must be #1)}%
3696 {\@ehc}%
3697 \fi}}

```

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

```

3698 \onlypreamble\bxjs@urgent@special
3699 \def\bxjs@urgent@special#1{%
3700 \AtBeginDvi{\special{#1}}%
3701 \AtBeginDocument{%
3702 \ifpackageloaded{atbegshi}{%
3703 \begingroup
3704 \toks\z@\special{#1}}%
3705 \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3706 \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3707 \endgroup
3708 }{}%
3709 }%
3710 }

```

## C.2 pTeX 用設定

```
3711 \if j\jsEngine
```

### ■ 共通命令の実装

```

3712 \def\bxjs@apply@kanjiskip{%
3713 \kanjiskip\@tempkipa}
3714 \def\bxjs@apply@xkanjiskip{%
3715 \xkanjiskip\@tempkipa}

```

`\jaJaChar` のサブマクロ。

```

3716 \def\bxjs@jachar#1{%
3717 \bxjs@jachar@a#1...\@nil}
3718 \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。

```
3719 \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思なし、そのスカラー値を `\@tempcnta` に代入する。

```

3720 \else\ifx.#3%
3721 \@tempcnta`#1 \multiply\@tempcnta64
3722 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3723 \bxjs@jachar@b
3724 \else\ifx.#4%
3725 \@tempcnta`#1 \multiply\@tempcnta64
3726 \advance\@tempcnta`#2 \multiply\@tempcnta64

```

```

3727 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3728 \bxjs@jachar@b
3729 \else
3730 \@tempcnta`#1 \multiply\@tempcnta64
3731 \advance\@tempcnta`#2 \multiply\@tempcnta64
3732 \advance\@tempcnta`#3 \multiply\@tempcnta64
3733 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3734 \bxjs@jachar@b
3735 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3736 \ifjsWithupTeX
3737 \def\bxjs@jachar@b{\kchar\@tempcnta}
3738 \else
3739 \def\bxjs@jachar@b{%
3740 \ifx\bxUInt\@undefined\else
3741 \bxUInt{\@tempcnta}%
3742 \fi}
3743 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

3744 \ifbxjs@jaspace@cmd
3745 \def\jathinspace{\hskip\xkanjiskip}
3746 \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。

```

3747 \ifbxjs@jp@jismmiv
3748 \PassOptionsToPackage{prefer2004jis}{pxchfon}
3749 \PassOptionsToPackage{prefer2004jis}{pxbabel}
3750 \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。

```

3751 \let\bxjs@tmpa\jsJaFont
3752 \ifx\bxjs@tmpa\bxjs@@auto
3753 \let\bxjs@tmpa\empty
3754 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3755 \def\bxjs@tmpa{noembed}
3756 \fi\fi
3757 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3758 \ifx\bxjs@tmpa\empty\else
3759 \edef\bxjs@next{%
3760 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3761 }\bxjs@next
3762 \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。

※ 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……。)

```

3763 \begingroup
3764   \global\let\@gtempa\relax
3765   \catcode`\|=0 \catcode`\|=12
3766   |def|bxjs@check#1|@nil{%
3767     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3768   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3769     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3770   |catcode`\|=0 \catcode`\|=12
3771   \def\bxjs@check@b#1keyval#2\@nnil{%
3772     \ifx$#2$ \else
3773       \xdef\@gtempa{%
3774         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3775       \fi}
3776 \@firstofone{%
3777   \catcode10=12 \endlinechar\m@ne
3778   \let\do\@makeother \dospecials \catcode32=10
3779   \openin\@inputcheck=otf.sty\relax
3780   \@tempwattrue
3781   \loop\if@tempwa
3782     \ifeof\@inputcheck \@tempwafalse \fi
3783     \if@tempwa
3784       \read\@inputcheck to\bxjs@line
3785       \expandafter\bxjs@check\bxjs@line\@nil
3786     \fi
3787     \repeat
3788   \closein\@inputcheck
3789 \endgroup}
3790 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3791 \ifbxjs@hyperref@enc
3792   \bxjs@check@hyperref@unicode{false}
3793 \fi

```

tounicode special 命令を出力する。

```

3794 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3795   \else\ifjsWithpTeXng T\else F\fi\fi T%
3796   \ifnum\jis"2121="A1A1 %euc
3797     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3798   \else\ifnum\jis"2121="8140 %sjis
3799     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3800   \else\ifnum\jis"2121="3000 %uptex
3801     \ifbxjs@bigcode

```

```

3802     \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3803     \PassOptionsToPackage{bigcode}{pxjahyper}
3804     \else
3805     \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3806     \fi
3807     \fi\fi\fi
3808     \let\bxToUnicodeSpecialDone=t
3809 \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。

```

3810 \ifx f\bxjs@enablejfam\else
3811     \@enablejfamtrue
3812 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

3813 \if@enablejfam
3814     \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3815     \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3816     \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3817     \jfam\symmincho
3818     \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3819     \AtBeginDocument{%
3820         \ifx\reDeclareMathAlphabet\undefined\else
3821             \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3822             \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3823             \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3824         \fi}
3825 \fi

```

### C.3 pdfTeX 用設定 : CJK + bxcjkatype

```

3826 \else\if p\jsEngine

```

■`bxcjkatype` パッケージの読込 `\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。

```

3827 \bxjs@adjust@jafont{f}
3828 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3829 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
3830 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
3831     \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
3832 \fi
3833 \edef\bxjs@next{%
3834     \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
3835 }\bxjs@next

```

```
3836 \bxjs@CJK@loaded
```

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

※取りあえず固定はしない。

```
3837 \ifbxjs@hyperref@enc
```

```
3838 \PassOptionsToPackage{unicode}{hyperref}
```

```
3839 \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。

※ bxcjkjatype 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。

```
3840 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
```

```
3841 \begingroup
```

```
3842 \CJK@input{UTF8.bdg}
```

```
3843 \endgroup
```

```
3844 \g@addto@macro\pdfstringdefPreHook{%
```

```
3845 \@nameuse{CJK@UTF8Binding}%
```

```
3846 }
```

```
3847 \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。

```
3848 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
```

```
3849 \g@addto@macro\pdfstringdefPreHook{%
```

```
3850 \ifx~\bxjs@@CJKtilde
```

```
3851 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
```

```
3852 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
```

```
3853 \let~\@empty
```

```
3854 \fi
```

```
3855 }
```

```
3856 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
```

```
3857 \def\bxjs@@tildecmd{~}
```

```
3858 \def\bxjs@LetUnexpandableSpace#1{%
```

```
3859 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
```

```
3860 \bxjs@org@LetUnexpandableSpace#1%
```

```
3861 \fi}
```

```
3862 \fi
```

■共通命令の実装

```
3863 \newskip\jsKanjiSkip
```

```
3864 \newskip\jsXKanjiSkip
```

```
3865 \ifx\CJKecglue\@undefined
```

```
3866 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
```

```
3867 \fi
```

```
3868 \let\autospacing\bxjs@enable@kanjiskip
```

```
3869 \let\noautospacing\bxjs@disable@kanjiskip
```

```
3870 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
```

```

3871 \def\bxjs@apply@kanjiskip{%
3872   \jsKanjSkip\@tempskipa
3873   \let\CJKglue\bxjs@CJKglue}
3874 \let\autoxspacing\bxjs@enable@xkanjiskip
3875 \let\noautoxspacing\bxjs@disable@xkanjiskip
3876 \protected\def\bxjs@CJKEcglue{\hskip\jsXKanjSkip}
3877 \def\bxjs@apply@xkanjiskip{%
3878   \jsXKanjSkip\@tempskipa
3879   \let\CJKEcglue\bxjs@CJKEcglue}

```

\jachar のサブマクロの実装。

```

3880 \def\bxjs@jachar#1{%
3881   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

3882 \ifbxjs@jaspace@cmd
3883   \protected\def\jathinspace{\CJKecglue}
3884 \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。

```

3885 \ifx t\bxjs@enablejfam
3886   \ClassWarningNoLine\bxjs@clsname
3887     {You cannot use 'enablejfam=true', since the\MessageBreak
3888      CJK package does not support Japanese math}
3889 \fi

```

## C.4 Xe<sub>La</sub>TeX 用設定 : xeCJK + zxjatype

```

3890 \else\if x\jsEngine

```

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

```

3891 \RequirePackage{zxjatype}
3892 \PassOptionsToPackage{no-math}{fontspec}%!
3893 \PassOptionsToPackage{xetex}{graphicx}%!
3894 \PassOptionsToPackage{xetex}{graphics}%!
3895 \ifx\zxJaFamilyName\@undefined
3896   \ClassError\bxjs@clsname
3897     {xeCJK or zxjatype is too old}\@ehc
3898 \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。

```

3899 \bxjs@adjust@jafont{f}
3900 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3901 \ifx\bxjs@tmpa\@empty
3902   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3903   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}

```

```

3904 \else
3905   \edef\bxjs@next{%
3906     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
3907   }\bxjs@next
3908 \fi

```

#### ■hyperref 対策 unicode オプションの指定に関する話。

X<sub>Ǝ</sub>T<sub>Ǝ</sub>X 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>Ǝ</sub>T<sub>Ǝ</sub>X の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「X<sub>Ǝ</sub>T<sub>Ǝ</sub>X 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。

```

3909 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3910   \ifbxjs@hyperref@enc
3911     \PassOptionsToPackage{unicode}{hyperref}
3912   \fi
3913 \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。

```

3914 \AtEndOfPackage{%
3915 \def\@gnewline #1{%
3916   \ifvmode \@nolnerr
3917   \else
3918     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3919     \nobreak \hskip-1sp\hskip1sp\relax
3920     \ignorespaces
3921   \fi}
3922 }

```

#### ■共通命令の実装

```

3923 \newskip\jsKanjiSkip
3924 \newskip\jsXKanjiSkip
3925 \ifx\CJKecglue\@undefined
3926   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3927 \fi
3928 \let\autospacing\bxjs@enable@kanjiskip

```



```

3929 \let\noautospacing\bxjs@disable@kanjiskip
3930 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3931 \def\bxjs@apply@kanjiskip{%
3932   \jsKanjiSkip\@tempskipa
3933   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3934 \let\autoxspacing\bxjs@enable@xkanjiskip
3935 \let\noautoxspacing\bxjs@disable@xkanjiskip
3936 \protected\def\bxjs@CJKe glue{\hskip\jsXKanjiSkip}
3937 \def\bxjs@apply@xkanjiskip{%
3938   \jsXKanjiSkip\@tempskipa
3939   \xeCJKsetup{CJKe glue={\bxjs@CJKe glue}}}

```

`\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。

```

3940 \ifx\mcfamily\@undefined
3941   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3942   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3943 \fi

```

`\jachar` のサブマクロの実装。

```

3944 \def\bxjs@jachar#1{%
3945   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3946   #1}

```

`\jathinspace` の実装。

```

3947 \ifbxjs@jaspace@cmd
3948   \protected\def\jathinspace{\CJKe glue}
3949 \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。

```

3950 \ifx t\bxjs@enablejfam
3951   \@enablejfamtrue
3952 \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。

```

3953 \if@enablejfam
3954   \xeCJKsetup{CJKmath=true}
3955 \fi

```

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

```

3956 \else\if 1\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。

```
3957 \let\zw\@undefined
3958 \RequirePackage{luatexja}
3959 \RequirePackage{luatexja-fontspec}
```

■和文フォント定義 `luatexja-fontspec` で使用する和文スケール値を `\jsScale` と合致させたいのだが……もっと良い方法はないのか？

```
3960 \ExplSyntaxOn
3961 \fp_gset:Nn \g_ltj_fontspec_scale_fp { \jsScale }
3962 \ExplSyntaxOff
```

`\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）。

```
3963 \bxjs@adjust@jafont{t}
3964 \ifx\bxjs@tmpa\bxjs@@noEmbed
3965   \def\bxjs@tmpa{noembed}
3966 \fi
3967 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3968 \ifx\bxjs@tmpa\@empty
3969   \defaultjfontfeatures{ Kerning=Off }
3970   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3971   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3972 \else
3973   \edef\bxjs@next{%
3974     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
3975   }\bxjs@next
3976 \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。

```
3977 \DeclareRobustCommand\rmfamily
3978   {\not@math@alphabet\rmfamily\mathrm}
3979   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3980 \DeclareRobustCommand\sffamily
3981   {\not@math@alphabet\sffamily\mathsf}
3982   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3983 \DeclareRobustCommand\ttfamily
3984   {\not@math@alphabet\ttfamily\mathtt}
3985   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3986 \AtBeginDocument{%
3987   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3988   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
3989   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
3990 \bxjs@if@sf@default{%
3991   \renewcommand\kanjifamilydefault{\gtdefault}}
```

## ■和文パラメタの設定

```

3992 % 次の3つは既定値の通り
3993 %\ltjsetparameter{prebreakpenalty={`',10000}}
3994 %\ltjsetparameter{postbreakpenalty={`",10000}}
3995 %\ltjsetparameter{prebreakpenalty={`",10000}}
3996 \ltjsetparameter{jaxspmode={`!',1}}
3997 \ltjsetparameter{jaxspmode={`〒,2}}
3998 \ltjsetparameter{alxspmode={`+,3}}
3999 \ltjsetparameter{alxspmode={`\%,3}}

```

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

※\jsInhibitGlueAtParTop は使わない。

`\ltjfakeparbegin` 現在の LuaTeX-jā 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。

```

4000 \ifx\ltjfakeparbegin\@undefined
4001   \protected\def\ltjfakeparbegin{%
4002     \ifhmode
4003       \relax\directlua{%
4004         luatexja.jfmglue.create_beginpar_node()}}
4005   \fi}
4006 \fi

```

ltjs\* クラスの定義と同等になるようにパッチを当てる。

```

4007 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4008 \begingroup
4009   \let\% \@percentchar \def\@#1[[[\detokenize{#1}]]]
4010   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4011   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4012     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4013   \directlua{
4014     local function patchcmd(cs, code, from, to)
4015       tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4016         :gsub("macro:", "\@gdef..cs, 1):gsub("->", "{", 1).."")
4017     end
4018     patchcmd(\@xsect, [[\meaning\xsect]],
4019       \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4020     patchcmd(\@item, [[\meaning\@item]],
4021       \bxjs@tmpa, \@{\ltjfakeparbegin})}
4022 \endgroup
4023 \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。

```

4024 \ifbxjs@hyperref@enc
4025   \PassOptionsToPackage{unicode}{hyperref}
4026   \bxjs@check@hyperref@unicode{true}

```

4027 \fi

#### ■ 共通命令の実装

```
4028 \protected\def\autospacing{%
4029   \ltjsetparameter{autospacing=true}}
4030 \protected\def\noautospacing{%
4031   \ltjsetparameter{autospacing=false}}
4032 \protected\def\autoxspacing{%
4033   \ltjsetparameter{autoxspacing=true}}
4034 \protected\def\noautoxspacing{%
4035   \ltjsetparameter{autoxspacing=false}}
4036 \def\bxjs@apply@kanjiskip{%
4037   \ltjsetparameter{kanjiskip={\@tempskipa}}}}
4038 \def\bxjs@apply@xkanjiskip{%
4039   \ltjsetparameter{xkanjiskip={\@tempskipa}}}}
```

\jachar のサブマクロの実装。

```
4040 \def\bxjs@jachar#1{%
4041   \ltjjachar`#1\relax}
```

\jathinspace の実装。

```
4042 \ifbxjs@jaspace@cmd
4043   \protected\def\jathinspace{%
4044     \hskip\ltjgetparameter{xkanjiskip}\relax}
4045 \fi
```

■ 和文数式ファミリ Lua<sub>T</sub><sub>E</sub>X-j<sub>a</sub> では和文数式ファミリは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablej<sub>f</sub>am は常に真になる。

```
4046 \ifx f\bxjs@enablejfam
4047   \ClassWarningNoLine\bxjs@clsname
4048     {You cannot use 'enablejfam=false', since the\MessageBreak
4049     LuaTEX-ja always provides Japanese math families}
4050 \fi
```

## C.6 共通処理 (2)

4051 \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 4052 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4053   \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4054   \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4055 \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。

```
4056 \ifx\mathmc\@undefined
```

```

4057 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4058 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4059 \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)。

```

4060 \ifbxjs@jaspace@cmd
4061 \bxjs@protected\def\bxjs@choice@jathinspace{%
4062   \relax\ifmmode \mskip\medmuskip
4063   \else \jathinspace\ignorespaces
4064   \fi}
4065 \jsAtEndOfClass{%
4066   \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4067   \else \def\>{\protect\bxjs@choice@jathinspace}%
4068   \fi}
4069 \fi

```

## ■和文・和欧文間空白の初期値

```

4070 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4071 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4072 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4073 \fi

```

以上で終わり。

```

4074 %</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。

```

4075 %<*modern>
4076 \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。

```

4077 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4078 \def\encodingdefault{T1}%
4079 \input{t1enc.def}%
4080 \fontencoding\encodingdefault\selectfont
4081 \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。

```
4082 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4083 \renewcommand{\rmdefault}{lmr}
4084 \renewcommand{\sfdefault}{lmss}
4085 \renewcommand{\ttdefault}{lmtt}
4086 \fi
```

大型演算子用の数式フォントの設定。

※ `amsfonts` パッケージと同等にする。

```
4087 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4088   <-7.5>cmex7<7.5-8.5>cmex8%
4089   <8.5-9.5>cmex9<9.5->cmex10}{}%
4090 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
4091 \def\cmex@opt{10}

amsmath 読込時に上書きされるのを防ぐ。
4091 \def\cmex@opt{10}
```

## D.2 fixltx2e 読込

※ `fixltx2e` 廃止前の  $\text{\LaTeX}$  カーネルの場合。

```
4092 \ifx\@IncludeInRelease\@undefined
4093 \RequirePackage{fixltx2e}
4094 \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。

```
4095 \RequirePackage{bxjscssjcat}
```

## D.4 完了

おしまい。

```
4096 %</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。

```
4097 %<*pandoc>
4098 \input{bxjsja-standard.def}
4099 \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。

```
4100 \@onlypreamble\bxjs@set@dupload@proc
4101 \def\bxjs@set@dupload@proc#1{%
4102   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
4103 \@onlypreamble\bxjs@set@dupload@proc@a
4104 \def\bxjs@set@dupload@proc@a#1{%
4105   \@onlypreamble#1\def#1##1}
```

`\@if@options` `\@if@options` の再定義。

```
4106 \@onlypreamble\bxjs@org@if@options
4107 \let\bxjs@org@if@options\@if@options
4108 \newif\ifbxjs@dlp
4109 \def\@if@options#1#2#3{%
4110   \bxjs@dlpfalse
4111   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4112   \ifx\bxjs@tmpa\bxjs@tmpb
4113     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4114     \bxjs@dlptrue \fi
4115   \fi
4116   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
4117   \else \expandafter\bxjs@org@if@options
4118   \fi {#1}{#2}{#3}}
4119 \AtBeginDocument{%
4120   \let\@if@options\bxjs@org@if@options}
4121 \@onlypreamble\bxjs@do@dupload@proc
4122 \def\bxjs@do@dupload@proc#1#2#3{%
4123   \csname bxjs@dlp/#2.#1\endcsname{#3}%
4124   \@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。

```
4125 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0
```

Polyglossia について。

```
4126 \pandocSkipLoadPackage{polyglossia}
```

```

4127 \bxjs@set@dupload@proc{polyglossia.sty}{%
4128   \ClassWarning\bxjs@clsname
4129     {Loading of polyglossia is blocked}}
4130 \ifx\setmainlanguage\@undefined
4131 \newcommand*\setmainlanguage[2] [] {}
4132 \newcommand*\setotherlanguage[2] [] {%
4133   \ifcat_#2_\else
4134     \expandafter\let\csname #2\endcsname\@empty
4135     \expandafter\let\csname end#2\endcsname\@empty
4136     \expandafter\let\csname text#2\endcsname\@firstofone
4137   \fi}
4138 \newcommand*\setotherlanguages[2] [] {%
4139   \@for\bxjs@tmpa:={#2}\do{%
4140     \setotherlangauge{\bxjs@tmpa}}}
4141 \fi
4142 \else

```

Babel について。

```

4143 \pandocSkipLoadPackage{babel}
4144 \bxjs@set@dupload@proc{babel.sty}{%
4145   \ClassWarning\bxjs@clsname
4146     {Loading of babel is blocked}}
4147 \let\foreignlanguage\@secondoftwo
4148 \let\otherlanguage\@gobble
4149 \let\endotherlanguage\@empty
4150 \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。

```

4151 \bxjs@set@dupload@proc{geometry.sty}{%
4152   \setpagelayout*{#1}}

```

### E.4 CJKmainfont 変数

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

```

4153 \if 1\jsEngine
4154   \pandocSkipLoadPackage{xeCJK}
4155   \providecommand*\setCJKmainfont{\setmainjfont}
4156 \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 がプレアンプ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。

```
4157 \AtBeginDocument{%
4158   \@tempswafalse
```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```
4159   \ifx\oldparagraph\@undefined\else
4160     \@tempswatruetrue
4161   \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4162   \ifnum\c@secnumdepth>3
4163     \@tempswatruetrue
4164   \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。

```
4165   \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4166     \let\jsParagraphMark\@empty
4167   \fi\fi}
```

## E.6 全角空白文字

```
4168 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4169   \catcode"3000=\active
4170   \begingroup \catcode`\!=7
4171   \protected\gdef!!!!3000{\zwspace}
4172   \endgroup
4173 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4174   \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4175   \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4176 \fi\fi
```

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

```
4177 \if j\jsEngine
4178   \bxjs@fix@hyperref@unicode{false}
4179 \else
```

```
4180 \bxjs@fix@hyperref@unicode{true}
4181 \fi
```

## E.7 完了

おしまい。

```
4182 %</pandoc>
```

和文ドライバ実装はここまで。

```
4183 %</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。

```
4184 %<*anc>
```

## 付録 G 補助パッケージ : bxjscompat

ムニャムニャムニャ……。

### G.1 準備

```
4185 %<*compat>
```

```
4186 \def\bxac@pkgname{bxjscompat}
```

\bxjx@engine エンジンの種別。

```
4187 \let\bxac@engine=n
```

```
4188 \def\bxac@do#1#2{%
```

```
4189 \edef\bxac@tmpa{\string#1}%
```

```
4190 \edef\bxac@tmpb{\meaning#1}%
```

```
4191 \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
4192 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
4193 \bxac@do\luatexversion{\let\bxac@engine=l}
```

\bxac@delayed@if@bxjs もし BXJS クラスの読込中でこのパッケージが読み込まれているならば、BXJS のクラスの終わりまで実行を遅延する。

```
4194 \ifx\jsAtEndOfClass\undefined
```

```
4195 \let\bxac@delayed@if@bxjs\@firstofone
```

```
4196 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
4197 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4198 \newif\ifbxac@in@old@behavior
4199 \let\ImposeOldLuaTeXBehavior\relax
4200 \let\RevokeOldLuaTeXBehavior\relax
```

## G.2 XeTeX 部分

```
4201 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。
4202 \@onlypreamble\bxac@adjust@charclass
4203 \bxac@delayed@if@bxjs{%
4204   \@ifpackageloaded{xCJK}{}{%else
4205     \ifx\XeTeX@alloc@intercharclass\undefined\else
4206       \ifnum\XeTeX@alloc@intercharclass=\z@
4207         \PackageInfo\bxac@pkgname
4208           {Setting up interchar class for CJK...\@gobble}%
4209         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4210           \XeTeX@alloc@intercharclass=3
4211         }{%else
4212           \PackageWarning\bxac@pkgname
4213             {Cannot find file 'load-unicode-xetex-classes.tex'%
4214             \@gobble}%
4215         }%
4216       \fi\fi
4217     \ifnum\XeTeXcharclass"3041=\z@
4218       \PackageInfo\bxac@pkgname
4219         {Adjusting interchar class for CJK...\@gobble}%
4220       \@for\bxac@x:={%
4221         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4222         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4223         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4224         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4225         31FF%
4226       }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4227     \fi
4228   }%
4229 }

以上。
4230 \fi
```

## G.3 LuaTeX 部分

```
4231 \ifx l\bxac@engine

ムニャムニャ。
4232 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4233   \chardef\pdfTeXversion=200
4234   \def\pdfTeXrevision{0}
```

```

4235 \let\pdfptxheader\luatexheader
4236 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

\RevokeOldLuaTeXBehavior 4237 \begingroup\expandafter\expandafter\expandafter\endgroup
4238 \expandafter\ifx\csname outputmode\endcsname\relax\else
4239 \def\bxac@ob@list{%
4240 \do{\let}\pdfoutput{\outputmode}%
4241 \do{\let}\pdfpagewidth{\pagewidth}%
4242 \do{\let}\pdfpageheight{\pageheight}%
4243 \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
4244 \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}
4245 \def\bxac@ob@do#1#2{\begingroup
4246 \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4247 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4248 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4249 \else \let#3#1\relax \let#1\@undefined
4250 \fi}
4251 \protected\def\ImposeOldLuaTeXBehavior{%
4252 \unless\ifbxac@in@old@behavior
4253 \bxac@in@old@behaviortrue
4254 \let\do\bxac@ob@do \bxac@ob@list
4255 \fi}
4256 \protected\def\RevokeOldLuaTeXBehavior{%
4257 \ifbxac@in@old@behavior
4258 \bxac@in@old@behaviorfalse
4259 \let\do\bxac@ob@do \bxac@ob@list
4260 \fi}
4261 \fi

```

漢字および完成形ハングルのカテゴリコードのムニヤムニヤ。

```

4262 \ifnum\luatexversion>64 \directlua{
4263 local function range(cs, ce, cc, ff)
4264 if ff or not tex.getcatcode(cs) == cc then
4265 local setcc = tex.setcatcode
4266 for c = cs, ce do setcc(c, cc) end
4267 end
4268 end
4269 range(0x3400, 0x4DB5, 11, false)
4270 range(0x4DB5, 0x4DBF, 11, true)
4271 range(0x4E00, 0x9FCC, 11, false)
4272 range(0x9FCD, 0x9FFF, 11, true)
4273 range(0xAC00, 0xD7A3, 11, false)
4274 range(0x20000, 0x2A6D6, 11, false)
4275 range(0x2A6D7, 0x2A6FF, 11, true)
4276 range(0x2A700, 0x2B734, 11, false)
4277 range(0x2B735, 0x2B73F, 11, true)
4278 range(0x2B740, 0x2B81D, 11, false)
4279 range(0x2B81E, 0x2B81F, 11, true)

```

```

4280     range(0x2B820, 0x2CEA1, 11, false)
4281     range(0x2CEA2, 0x2FFFD, 11, true)
4282 } \fi

```

以上。

```

4283 \fi

```

## G.4 完了

おしまい。

```

4284 %</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

```

4285 %<*cjkcat>
4286 \def\bxjx@pkgname{bxjscjkat}
4287 \newcount\bxjx@cmta

\bxjx@engine エンジンの種別。

4288 \let\bxjx@engine=n
4289 \def\bxjx@do#1#2{%
4290   \edef\bxjx@tmpa{\string#1}%
4291   \edef\bxjx@tmpb{\meaning#1}%
4292   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4293 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
4294 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
4295 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
4296 \bxjx@do\pdfTeXversion{\let\bxjx@engine=p}
4297 \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。

```

4298 \def\bxjx@do#1#2{%
4299   \if#1\bxjx@engine
4300     \@ifpackageloaded{#2}{\}%else
4301     \PackageError\bxjx@pkgname
4302       {Package '#2' must be loaded}%
4303       {Package loading is aborted.\MessageBreak\@ehc}%
4304     \endinput}
4305 \fi}
4306 \bxjx@do{p}{bxjcjkatype}
4307 \bxjx@do{x}{xeCJK}
4308 \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。

```
4309 \ifx\TextOrMath\@undefined
4310   \RequirePackage{fixltx2e}
4311 \fi
```

## H.2 和文カテゴリコードの設定

upL<sup>A</sup>T<sub>E</sub>X の場合、和文カテゴリコードの設定を LuaT<sub>E</sub>X-j<sub>a</sub> と（ほぼ）等価なものに変更する。

※ LuaT<sub>E</sub>X-j<sub>a</sub> との相違点：A830、A960、1B000。

```
4312 \if u\bxjx@engine
4313 \@for\bxjx@x:={%
4314 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4315 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
4316 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
4317 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
4318 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
4319 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
4320 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
4321 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
4322 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
4323 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
4324 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
4325 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4326 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4327 10480,10500,10530,10600,10800,10840,10860,10880,%
4328 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
4329 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
4330 11000,11080,110D0,11100,11150,11180,111E0,11200,%
4331 11280,112B0,11300,11480,11580,11600,11680,11700,%
4332 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
4333 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4334 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
4335 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
4336 1F700,1F780,1F800,1F900,E0000,F0000,100000%
4337 }\do{\kcatcode"\bxjx@x=15 }
4338 \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。

```
4339 \def\bxjx@grkcyr@list{%
4340 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4341 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
```

4342	$\backslash\mathrm{do}\{0393\}\{\mathrm{LGR}\}\{\backslash\mathrm{textGamma}\}\{\backslash\mathrm{Gamma}\}\%$	% GR. C. L. GAMMA
4343	$\backslash\mathrm{do}\{0394\}\{\mathrm{LGR}\}\{\backslash\mathrm{textDelta}\}\{\backslash\mathrm{Delta}\}\%$	% GR. C. L. DELTA
4344	$\backslash\mathrm{do}\{0395\}\{\mathrm{LGR}\}\{\backslash\mathrm{textEpsilon}\}\{\mathrm{E}\}\%$	% GR. C. L. EPSILON
4345	$\backslash\mathrm{do}\{0396\}\{\mathrm{LGR}\}\{\backslash\mathrm{textZeta}\}\{\mathrm{Z}\}\%$	% GR. C. L. ZETA
4346	$\backslash\mathrm{do}\{0397\}\{\mathrm{LGR}\}\{\backslash\mathrm{textEta}\}\{\mathrm{H}\}\%$	% GR. C. L. ETA
4347	$\backslash\mathrm{do}\{0398\}\{\mathrm{LGR}\}\{\backslash\mathrm{textTheta}\}\{\backslash\mathrm{Theta}\}\%$	% GR. C. L. THETA
4348	$\backslash\mathrm{do}\{0399\}\{\mathrm{LGR}\}\{\backslash\mathrm{textIota}\}\{\mathrm{I}\}\%$	% GR. C. L. IOTA
4349	$\backslash\mathrm{do}\{039A\}\{\mathrm{LGR}\}\{\backslash\mathrm{textKappa}\}\{\mathrm{K}\}\%$	% GR. C. L. KAPPA
4350	$\backslash\mathrm{do}\{039B\}\{\mathrm{LGR}\}\{\backslash\mathrm{textLambda}\}\{\backslash\mathrm{Lambda}\}\%$	% GR. C. L. LAMDA
4351	$\backslash\mathrm{do}\{039C\}\{\mathrm{LGR}\}\{\backslash\mathrm{textMu}\}\{\mathrm{M}\}\%$	% GR. C. L. MU
4352	$\backslash\mathrm{do}\{039D\}\{\mathrm{LGR}\}\{\backslash\mathrm{textNu}\}\{\mathrm{N}\}\%$	% GR. C. L. NU
4353	$\backslash\mathrm{do}\{039E\}\{\mathrm{LGR}\}\{\backslash\mathrm{textXi}\}\{\backslash\mathrm{Xi}\}\%$	% GR. C. L. XI
4354	$\backslash\mathrm{do}\{039F\}\{\mathrm{LGR}\}\{\backslash\mathrm{textOmicron}\}\{\mathrm{O}\}\%$	% GR. C. L. OMICRON
4355	$\backslash\mathrm{do}\{03A0\}\{\mathrm{LGR}\}\{\backslash\mathrm{textPi}\}\{\backslash\mathrm{Pi}\}\%$	% GR. C. L. PI
4356	$\backslash\mathrm{do}\{03A1\}\{\mathrm{LGR}\}\{\backslash\mathrm{textRho}\}\{\mathrm{P}\}\%$	% GR. C. L. RHO
4357	$\backslash\mathrm{do}\{03A3\}\{\mathrm{LGR}\}\{\backslash\mathrm{textSigma}\}\{\backslash\mathrm{Sigma}\}\%$	% GR. C. L. SIGMA
4358	$\backslash\mathrm{do}\{03A4\}\{\mathrm{LGR}\}\{\backslash\mathrm{textTau}\}\{\mathrm{T}\}\%$	% GR. C. L. TAU
4359	$\backslash\mathrm{do}\{03A5\}\{\mathrm{LGR}\}\{\backslash\mathrm{textUpsilon}\}\{\backslash\mathrm{Upsilon}\}\%$	% GR. C. L. UPSILON
4360	$\backslash\mathrm{do}\{03A6\}\{\mathrm{LGR}\}\{\backslash\mathrm{textPhi}\}\{\backslash\mathrm{Phi}\}\%$	% GR. C. L. PHI
4361	$\backslash\mathrm{do}\{03A7\}\{\mathrm{LGR}\}\{\backslash\mathrm{textChi}\}\{\backslash\mathrm{Chi}\}\%$	% GR. C. L. CHI
4362	$\backslash\mathrm{do}\{03A8\}\{\mathrm{LGR}\}\{\backslash\mathrm{textPsi}\}\{\backslash\mathrm{Psi}\}\%$	% GR. C. L. PSI
4363	$\backslash\mathrm{do}\{03A9\}\{\mathrm{LGR}\}\{\backslash\mathrm{textOmega}\}\{\backslash\mathrm{Omega}\}\%$	% GR. C. L. OMEGA
4364	$\backslash\mathrm{do}\{03B1\}\{\mathrm{LGR}\}\{\backslash\mathrm{textalpha}\}\{\backslash\mathrm{alpha}\}\%$	% GR. S. L. ALPHA
4365	$\backslash\mathrm{do}\{03B2\}\{\mathrm{LGR}\}\{\backslash\mathrm{textbeta}\}\{\backslash\mathrm{beta}\}\%$	% GR. S. L. BETA
4366	$\backslash\mathrm{do}\{03B3\}\{\mathrm{LGR}\}\{\backslash\mathrm{textgamma}\}\{\backslash\mathrm{gamma}\}\%$	% GR. S. L. GAMMA
4367	$\backslash\mathrm{do}\{03B4\}\{\mathrm{LGR}\}\{\backslash\mathrm{textdelta}\}\{\backslash\mathrm{delta}\}\%$	% GR. S. L. DELTA
4368	$\backslash\mathrm{do}\{03B5\}\{\mathrm{LGR}\}\{\backslash\mathrm{textepsilon}\}\{\backslash\mathrm{epsilon}\}\%$	% GR. S. L. EPSILON
4369	$\backslash\mathrm{do}\{03B6\}\{\mathrm{LGR}\}\{\backslash\mathrm{textzeta}\}\{\backslash\mathrm{zeta}\}\%$	% GR. S. L. ZETA
4370	$\backslash\mathrm{do}\{03B7\}\{\mathrm{LGR}\}\{\backslash\mathrm{texteta}\}\{\backslash\mathrm{eta}\}\%$	% GR. S. L. ETA
4371	$\backslash\mathrm{do}\{03B8\}\{\mathrm{LGR}\}\{\backslash\mathrm{texttheta}\}\{\backslash\mathrm{theta}\}\%$	% GR. S. L. THETA
4372	$\backslash\mathrm{do}\{03B9\}\{\mathrm{LGR}\}\{\backslash\mathrm{textiota}\}\{\backslash\mathrm{iota}\}\%$	% GR. S. L. IOTA
4373	$\backslash\mathrm{do}\{03BA\}\{\mathrm{LGR}\}\{\backslash\mathrm{textkappa}\}\{\backslash\mathrm{kappa}\}\%$	% GR. S. L. KAPPA
4374	$\backslash\mathrm{do}\{03BB\}\{\mathrm{LGR}\}\{\backslash\mathrm{textlambda}\}\{\backslash\mathrm{lambda}\}\%$	% GR. S. L. LAMDA
4375	$\backslash\mathrm{do}\{03BC\}\{\mathrm{LGR}\}\{\backslash\mathrm{textmu}\}\{\backslash\mathrm{mu}\}\%$	% GR. S. L. MU
4376	$\backslash\mathrm{do}\{03BD\}\{\mathrm{LGR}\}\{\backslash\mathrm{textnu}\}\{\backslash\mathrm{nu}\}\%$	% GR. S. L. NU
4377	$\backslash\mathrm{do}\{03BE\}\{\mathrm{LGR}\}\{\backslash\mathrm{textxi}\}\{\backslash\mathrm{xi}\}\%$	% GR. S. L. XI
4378	$\backslash\mathrm{do}\{03BF\}\{\mathrm{LGR}\}\{\backslash\mathrm{textomicron}\}\{\mathrm{o}\}\%$	% GR. S. L. OMICRON
4379	$\backslash\mathrm{do}\{03C0\}\{\mathrm{LGR}\}\{\backslash\mathrm{textpi}\}\{\backslash\mathrm{pi}\}\%$	% GR. S. L. PI
4380	$\backslash\mathrm{do}\{03C1\}\{\mathrm{LGR}\}\{\backslash\mathrm{textrho}\}\{\backslash\mathrm{rho}\}\%$	% GR. S. L. RHO
4381	$\backslash\mathrm{do}\{03C2\}\{\mathrm{LGR}\}\{\backslash\mathrm{textvarsigma}\}\{\backslash\mathrm{varsigma}\}\%$	% GR. S. L. FINAL SIGMA
4382	$\backslash\mathrm{do}\{03C3\}\{\mathrm{LGR}\}\{\backslash\mathrm{textsigma}\}\{\backslash\mathrm{sigma}\}\%$	% GR. S. L. SIGMA
4383	$\backslash\mathrm{do}\{03C4\}\{\mathrm{LGR}\}\{\backslash\mathrm{texttau}\}\{\backslash\mathrm{tau}\}\%$	% GR. S. L. TAU
4384	$\backslash\mathrm{do}\{03C5\}\{\mathrm{LGR}\}\{\backslash\mathrm{textupsilon}\}\{\backslash\mathrm{upsilon}\}\%$	% GR. S. L. UPSILON
4385	$\backslash\mathrm{do}\{03C6\}\{\mathrm{LGR}\}\{\backslash\mathrm{textphi}\}\{\backslash\mathrm{phi}\}\%$	% GR. S. L. PHI
4386	$\backslash\mathrm{do}\{03C7\}\{\mathrm{LGR}\}\{\backslash\mathrm{textchi}\}\{\backslash\mathrm{chi}\}\%$	% GR. S. L. CHI
4387	$\backslash\mathrm{do}\{03C8\}\{\mathrm{LGR}\}\{\backslash\mathrm{textpsi}\}\{\backslash\mathrm{psi}\}\%$	% GR. S. L. PSI
4388	$\backslash\mathrm{do}\{03C9\}\{\mathrm{LGR}\}\{\backslash\mathrm{textomega}\}\{\backslash\mathrm{omega}\}\%$	% GR. S. L. OMEGA
4389	$\backslash\mathrm{do}\{0401\}\{\mathrm{T2A}\}\{\backslash\mathrm{CYRYO}\}\{\}\%$	% CY. C. L. IO
4390	$\backslash\mathrm{do}\{0410\}\{\mathrm{T2A}\}\{\backslash\mathrm{CYRA}\}\{\}\%$	% CY. C. L. A

4391 \do{0411}{T2A}{\CYRB}{}%	% CY. C. L. BE
4392 \do{0412}{T2A}{\CYRV}{}%	% CY. C. L. VE
4393 \do{0413}{T2A}{\CYRG}{}%	% CY. C. L. GHE
4394 \do{0414}{T2A}{\CYRD}{}%	% CY. C. L. DE
4395 \do{0415}{T2A}{\CYRE}{}%	% CY. C. L. IE
4396 \do{0416}{T2A}{\CYRZH}{}%	% CY. C. L. ZHE
4397 \do{0417}{T2A}{\CYRZ}{}%	% CY. C. L. ZE
4398 \do{0418}{T2A}{\CYRI}{}%	% CY. C. L. I
4399 \do{0419}{T2A}{\CYRISHRT}{}%	% CY. C. L. SHORT I
4400 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
4401 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
4402 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
4403 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
4404 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
4405 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
4406 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
4407 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
4408 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
4409 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
4410 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
4411 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
4412 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
4413 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
4414 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
4415 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
4416 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
4417 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
4418 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
4419 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
4420 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
4421 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
4422 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
4423 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
4424 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
4425 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
4426 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
4427 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
4428 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
4429 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
4430 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
4431 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
4432 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
4433 \do{043B}{T2A}{\cyrl}{}%	% CY. S. L. EL
4434 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
4435 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
4436 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
4437 \do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
4438 \do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
4439 \do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES



```

4440 \do{0442}{T2A}{\cyrt}{}%           % CY. S. L. TE
4441 \do{0443}{T2A}{\cyru}{}%           % CY. S. L. U
4442 \do{0444}{T2A}{\cyrf}{}%           % CY. S. L. EF
4443 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
4444 \do{0446}{T2A}{\cyrc}{}%           % CY. S. L. TSE
4445 \do{0447}{T2A}{\cyrch}{}%          % CY. S. L. CHE
4446 \do{0448}{T2A}{\cyrsh}{}%          % CY. S. L. SHA
4447 \do{0449}{T2A}{\cyrshch}{}%        % CY. S. L. SHCHA
4448 \do{044A}{T2A}{\cyrhrdsn}{}%       % CY. S. L. HARD SIGN
4449 \do{044B}{T2A}{\cyrrery}{}%        % CY. S. L. YERU
4450 \do{044C}{T2A}{\cyrsftsn}{}%       % CY. S. L. SOFT SIGN
4451 \do{044D}{T2A}{\cyrrerev}{}%       % CY. S. L. E
4452 \do{044E}{T2A}{\cyryu}{}%          % CY. S. L. YU
4453 \do{044F}{T2A}{\cyrya}{}%          % CY. S. L. YA
4454 \do{0451}{T2A}{\cyryo}{}%          % CY. S. L. IO
4455 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4456 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4457 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4458 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
4459 \do{00B4}{TS1}{\textasciicute}{}%  % ACUTE ACCENT
4460 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4461 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4462 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
4463 }

4464 \providecommand*{\mathdegree}{\circ}

```

`\ifbxjx@gcc@cjkl` [スイッチ] ギリシャ・キリル文字を和文扱いにするか。

```
4465 \newif\ifbxjx@gcc@cjkl
```

`\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。

```

4466 \newcommand*\greekasCJK{%
4467   \bxjx@gcc@cjkltrue}
4468 \newcommand*\nogreekasCJK{%
4469   \bxjx@gcc@cjklfalse}

```

`\bx@fake@grk` `\bx@fake@grk{<出力文字>}{<基準文字>}` :

```

4470 \def\bxjx@do#1\relax{%
4471   \def\bxjx@fake@grk##1##2{%
4472     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4473   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4474     \ifx\##1\%
4475       \bxjx@canta##4\divide\bxjx@canta\@cclvi
4476       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
4477       \mathchar\bxjx@canta
4478     \else ##3\fi}
4479 } \expandafter\bxjx@do\string\mathchar\relax

```

## ■pdfLaTeX・upLaTeX の場合

```
4480 \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。

```
4481 \@ifpackageloaded{inputenc}{\fi}{%else
4482   \RequirePackage[utf8]{inputenc}}
4483 \def\bxjx@tmpa{utf8}
4484 \ifx\bxjx@tmpa\inputencdoingname
4485   \PackageWarningNoLine\bxjx@pkgname
4486     {Input encoding changed to utf8}%
4487   \inputencoding{utf8}%
4488 \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。

```
4489 \if u\bxjx@engine
4490 \kcatcode"0370=15
4491 \kcatcode"0400=15
4492 \kcatcode"0500=15
4493 \fi
```

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

```
4494 \def\do#1{%
4495   \@tempcnta="#1\relax
4496   \@tempcntb=\@tempcnta \divide\@tempcntb256
4497   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4498   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4499 \def\bxjx@do@a#1#2#3#4#5{%
4500   \ifx\#5\%
4501     \def\bxjx@tmpa{\@inmathwarn#4}%
4502   \else\ifcat A\noexpand#5%
4503     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4504       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4505   \else \def\bxjx@tmpa{#5}%
4506   \fi\fi
4507   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4508   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
4509 \if u\bxjx@engine
4510 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
4511 \def\bxjx@do@b#1#2#3#4#5{%
4512   \kchardef#1=\@tempcnta
4513   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4514   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
4515 \else\if p\bxjx@engine
4516 \def\bxjx@do@b#1#2#3#4#5{%
4517   \mathchardef#1=\@tempcnta
4518   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4519   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
4520 \fi\fi
4521 \bxjx@grkcyr@list
```

```

4522 \let\bxjx@do@a\undefined
4523 \let\bxjx@do@b\undefined

```

`\bxjx@DeclareUnicodeCharacter` `\bxjx@DeclareUnicodeCharacter` を改変して、ギリシャ・キリル文字の場合に再定義を抑制したもの。

```

4524 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4525 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4526 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4527 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4528   \count@="#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
4529   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
4530     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4531   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4532     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4533   \else
4534     \wlog{ \space\space skipped defining Unicode char U+#1}%
4535   \fi\fi}

```

`\bxjx@ja@or@not` `\bxjx@ja@or@not`

```

4536 \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。

```

```

4537 \ifbxjx@gcc@cjkk #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。

```

4538 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4539 \else \UseTextSymbol{#2}{#3}%
4540 \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。

```

4541 \begingroup
4542 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4543 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4544   \noexpand\bxjx@swap@DUC@cmd
4545   \the\toks@
4546   \noexpand\bxjx@swap@DUC@cmd}}
4547 \endgroup\next
4548 \def\bxjx@swap@DUC@cmd{%
4549   \let\bxjx@tmpa\DeclareUnicodeCharacter
4550   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4551   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

#### ■ Xe<sub>La</sub>TeX・Lua<sub>La</sub>TeX の場合

```

4552 \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。

```

4553 \def\do#1{%
4554   \bxjx@cmta="#1\relax
4555   \begingroup
4556     \lccode`~=\bxjx@cmta
4557   \lowercase{\endgroup
4558     \bxjx@do@a{~}}{#1}}
4559 \def\bxjx@do@a#1#2#3#4#5{%
4560   \ifx\#5\\\let\bxjx@tmpa\relax
4561   \else\ifcat A\noexpand#5%
4562     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4563       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4564   \else \def\bxjx@tmpa{#5}%
4565   \fi\fi
4566   \ifx\bxjx@tmpa\relax\else
4567     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4568   \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。

```

4569 \mathchardef\bxjx@tmpa="119
4570 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4571 \let\bxjx@do@a\undefined

```

Lua<sub>T</sub><sub>E</sub>X における \(\no)greekasCJK の定義。jacharrange の設定を変更する。

```

4572 \if l\bxjx@engine
4573   \protected\def\greekasCJK{%
4574     \bxjx@gcc@cjctrue
4575     \ltjsetparameter{jacharrange={+2, +8}}
4576   \protected\def\nogreekasCJK{%
4577     \bxjx@gcc@cjkfalse
4578     \ltjsetparameter{jacharrange={-2, -8}}
4579 \fi

```

X<sub>Y</sub><sub>L</sub><sub>A</sub>T<sub>E</sub>X における \(\no)greekasCJK の定義。

```

4580 \if x\bxjx@engine
4581   \protected\def\greekasCJK{%
4582     \bxjx@gcc@cjctrue
4583     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4584     \bxjx@grkcyr@list}
4585   \protected\def\nogreekasCJK{%
4586     \bxjx@gcc@cjkfalse
4587     \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4588     \bxjx@grkcyr@list}
4589 \fi

```

以上。

```

4590 \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。

```
4591 \nogreekasCJK
```

## H.5 完了

おしまい。

```
4592 %</cjkat>
```

## 付録 I 補助パッケージ : bxjspandoc

Pandoc の L<sup>A</sup>T<sub>E</sub>X 用標準テンプレートをより幸せに使うための設定。BXJS クラスの pandoc ドライバのコードの中の、“汎用的”に使える部分を切り出したもの。つまり現在の pandoc ドライバはこのパッケージを読みこむ。

※テンプレートの T<sub>E</sub>X コードより前に読み込む必要があるため、専ら文書クラス内での読込に限られる。

### I.1 準備

```
4593 %<*ancpandoc>
```

```
4594 %% このファイルは日本語文字を含みます.
```

```
4595 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```
4596 \let\bxjsp@engine=n
```

```
4597 \@onlypreamble\bxjsp@do
```

```
4598 \def\bxjsp@do#1#2{%
```

```
4599   \edef\bxjsp@tmpa{\string#1}%
```

```
4600   \edef\bxjsp@tmpb{\meaning#1}%
```

```
4601   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
```

```
4602 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
```

```
4603 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
```

```
4604 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
```

```
4605 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

### I.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。

```
4606 \newcommand*\pandocSkipLoadFile[1]{%
```

```
4607   \expandafter\bxjsp@skip@load@file@a\csize ver@#1\endcsize{#1}}
```

```
4608 \def\bxjsp@skip@load@file@a#1#2{%
```

```
4609   \ifx#1\relax
```

```

4610 \def#1{2001/01/01}%
4611 \PackageInfo{bxjisp}{pkgname}
4612 {File '#2' marked as loaded\@gobble}%
4613 \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。

```

4614 \newcommand*\pandocSkipLoadPackage[1]{%
4615 \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。

```

4616 \ifx\@IncludeInRelease\@undefined\else
4617 \pandocSkipLoadPackage{fixltx2e}
4618 \fi

```

### 1.4 cmap パッケージ

エンジンが `(u)pLATEX` 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。）

```

4619 \if j\bxjisp@engine
4620 \pandocSkipLoadPackage{cmap}
4621 \fi

```

### 1.5 microtype パッケージ

警告が多すぎなので消す。

```

4622 \if j\bxjisp@engine \else
4623 \PassOptionsToPackage{verbose=silent}{microtype}
4624 \fi

```

エンジンが `(u)pLATEX` 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。

```

4625 \if j\bxjisp@engine
4626 \pandocSkipLoadPackage{microtype}
4627 \newcommand*\UseMicrotypeSet[2] [] {}
4628 \fi

```

## I.6 Unicode 文字変換対策

Pandoc で  $\LaTeX$  形式に書き出す場合は、元データ中の一部の Unicode 文字を「 $\LaTeX$  の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

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

日本語  $\LaTeX$  では「 $\LaTeX$  の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「 $\cdots$ 」の置換を抑止する機能はないようである。そこで、「 $\backslash\ldots\{$ 」を『 $\cdots$ 』に戻す」という処置を行う。

`\pandocLdots`   Pandoc 用の  $\backslash\ldots$  の実装。非数式でありかつ後続が  $\{$  の場合は代わりに  $\cdots$  を実行する。

```
4629 \DeclareRobustCommand{\pandocLdots}{%
4630   \relax\ifmmode \expandafter\bxjsp@org@ldots
4631   \else \expandafter\bxjsp@ldots@a
4632   \fi}
4633 \def\bxjsp@ja@ellipsis{…}
4634 \let\bxjsp@org@ldots\ldots
4635 \def\bxjsp@ldots@a{%
4636   \futurelet\bxjsp@tok\bxjsp@ldots@b}
4637 \def\bxjsp@ldots@b{%
4638   \ifx\bxjsp@tok\bgroup \expandafter\bxjsp@ldots@c
4639   \else \expandafter\bxjsp@org@ldots
4640   \fi}
4641 \def\bxjsp@ldots@c{%
4642   \afterassignment\bxjsp@ldots@d \let\bxjsp@tok=}
4643 \def\bxjsp@ldots@d{%
4644   \futurelet\bxjsp@tok\bxjsp@ldots@e}
4645 \def\bxjsp@ldots@e{%
4646   \ifx\bxjsp@tok\egroup \expandafter\bxjsp@ldots@f
4647   \else \expandafter\bxjsp@ldots@g
4648   \fi}
4649 \def\bxjsp@ldots@f{%
4650   \bxjsp@ja@ellipsis \let\bxjsp@tok=}
4651 \def\bxjsp@ldots@g{%
4652   \expandafter\bxjsp@org@ldots\expandafter{\romannumeral-`} }
```

$\backslash\ldots$  の実装を置き換える。

```
4653 \AtBeginDocument{%
4654   \let\bxjsp@org@ldots\ldots
4655   \let\ldots\pandocLdots}
```

## I.7 PandoLa モジュール

インストール済であれば読み込む。

```
4656 \IfFileExists{bxpandola.sty}{%  
4657   \RequirePackage{bxpandola}\relax  
4658   \PackageInfo{bxjzp@pkgname  
4659     {PandoLa module is loaded\@gobble}  
4660 }{}
```

## I.8 完了

おしまい。

```
4661 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
4662 %</anc>
```