

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”)





v1.7a [2017/12/09]

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	37
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	61
8.1	表題	61
8.2	章・節	66
8.3	リスト環境	78
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	88
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	97
10.3	索引	98
10.4	脚注	100
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	107
14	初期設定	107
付録 A	和文ドライバの仕様 罨	112
付録 B	和文ドライバ : minimal 罨	113
B.1	補助マクロ	113
B.2	(u)pTeX 用の設定	115
B.3	pdfTeX 用の処理	119
B.4	X _Y TeX 用の処理	120
B.5	後処理 (エンジン共通)	120
付録 C	和文ドライバ : standard 罨	123
C.1	共通処理 (1)	124
C.2	pTeX 用設定	131
C.3	pdfTeX 用設定 : CJK + bxcjkatype	134
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	136
C.5	LuaTeX 用設定 : LuaTeX-ja	138
C.6	共通処理 (2)	141
付録 D	和文ドライバ : modern 罨	142
D.1	フォント設定	142
D.2	fixltx2e 読込	143
D.3	和文カテゴリコード	143
D.4	完了	143
付録 E	和文ドライバ : pandoc 罨	143
E.1	dupload システム	144
E.2	lang 変数	144
E.3	geometry 変数	145
E.4	CJKmainfont 変数	145
E.5	paragraph のマーク	146

E.6	全角空白文字	146
E.7	完了	147
付録 F	補助パッケージ一覧 	147
付録 G	補助パッケージ : bxjscompat 	147
G.1	準備	147
G.2	X _Y TeX 部分	148
G.3	LuaTeX 部分	148
G.4	完了	150
付録 H	補助パッケージ : bxjscjkat 	150
H.1	準備	150
H.2	和文カテゴリコードの設定	151
H.3	ギリシャ・キリル文字の扱い	151
H.4	初期設定	158
H.5	完了	158
付録 I	補助パッケージ : bxjspandoc 	158
I.1	準備	158
I.2	パッケージ読込の阻止	158
I.3	fixltx2e パッケージ	159
I.4	cmap パッケージ	159
I.5	microtype パッケージ	159
I.6	Unicode 文字変換対策	160
I.7	PandoLa モジュール	161
I.8	完了	161

1 はじめに

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

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

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「p_{La}TeX 2_ε 新ドキュメントクラス」に改変を加えたものである。本ドキュメントクラスに関する説明は全てこの形式の枠の中に記す。枠の外にあるものは原版著者による原版に対する解説である。

これは L^AT_EX3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいて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^AT_EX 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^AT_EX や L^AT_EX の不都合な点に対して、クラスファイル内で独自に対策を施していました。しかし、2016 年以降、コミュニティ版 pL^AT_EX が次第に対策コードをカーネル内に取り込むようになりました。そこで、新しい pL^AT_EX カーネルと衝突しないように、日付が古い場合だけパッチをあてる場合があります。この処理に使用するフラグを定義します。

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

■ BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため `calc` を読み込む。

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

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

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

クラスの本体ではこの他に `geometry` パッケージが読み込まれる。

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

```
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 m^2 、縦横比 $1:\sqrt{2}$ の長方形の辺の長さを mm 単位に切り捨てたものです。これを基準として順に半截しては mm 単位に切り捨てたものが A1, A2, …です。

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

デフォルトは `a4paper` です。

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

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

```
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、`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_{La}T_EX 2_ε 本体で行います (plcore.dtx 参照)。オプション `tombow` で日付付きのトンボ、オプション `tombo` で日付なしのトンボを出力します。これらはアスキー版のままです。カウンタ `\hour`, `\minute` は p_{La}T_EX 2_ε 本体で宣言されています。

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

```

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 \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` \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^AT_EX 関連のロゴを再定義する jslogo パッケージを読み込まないオプション nojslogo を新設しました。jslogo オプションの指定で従来どおりの

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

BXJS クラスでは、`nojslogo` を既定とする。

```
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`： $\text{X}_{\text{Y}}\text{T}_{\text{E}}\text{X}$ の “`generateactualtext`” を有効／無効にする。
- `simplejasetup` / `nosimplejasetup`： $\text{X}_{\text{Y}}\text{T}_{\text{E}}\text{X}$ の “`linebreaklocale`” を有効／無効にする。
- `bigcode` / `nobigcode`： $\text{upT}_{\text{E}}\text{X}$ で CMap として UTF8-UCS2 の代わりに UTF8-UTF16 を使う／使わない。
- `oldfontcommands` / `nooldfontcommands`：古い “二文字フォント命令” に対する警告を抑止する／しない。
- `base=<dimen>`：基底フォントサイズを直接指定する。（`xxpt` オプションの代用なので、既定値は 10 pt である。）
- `jbase=<dimen>`：基底フォントサイズを “和文規準で” 直接指定する。
- `scale=<real>`：和文フォントのスケールを表すマクロ `\jsScale` の値を設定する。もちろんこの値を何らかの方法で和文処理モジュールに渡さないと意味を成さない。既定値は 0.924715 (= 13 Q/10 pt)。
- `noscale`：`scale=1` と等価。
- `mag=<int>`：`\mag` 値の直接設定。既定は `base` から算出する。
- `paper={<dimen:width>}{<dimen:height>}`：用紙サイズ設定。用紙サイズオプションの代用で、既定値は `a4paper` 相当。
- `ja=<name>`：使用する和文ドライバの指定。
- `jafont=<name>`：和文フォントプリセットの指定。
- `japaram=<name>`：和文フォントパラメタの指定。
- `magstyle=<name>`：“版面拡大” の実現方法の選択。
- `dvi=<name>`：DVI モードの時のみに参照されるドライバ指定。
- `geometry={class | user}`：`geometry` パッケージの読込を自動的に行うかユーザに

任せるか。

- `fancyhdr=<bool>` : `fancyhdr` パッケージ用の調整を行うか。
- `layout=<name>` : レイアウト変種の指定。
- `textwidth-limit=<number>` : `bxjsbook` における、`\textwidth` の上限の全角単位での値。
- `paragraph-mark=<char>` : パラグラフのマーク。
- `whole-zw-lines=<bool>` : 行長を全角単位に丸めるか。
- `hyperref-enc=<bool>` : `hyperref` の文字コード指定補正を行うか。

`\bxjs@invscale` `\bxjs@invscale` は $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 \LaTeX 用の既定テンプレートを用いて他形式から \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 \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@fix@at@cmd` [スイッチ] `fix-at-cmd` の指定値。

```
561 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
562 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
563 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
564 \define@key{bxjs}{fix-at-cmd}[true]{\bxjs@set@keyval{fixatcmd}{#1}{}}
```

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

```
565 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
566 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
567 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
568 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}
```

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

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

■ オプションの実行

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

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

```
581 \let\bxjs@ltx@removeelement\@removeelement
582 \def\@removeelement#1#2#3{%
583 \def\reserved@a{#2}%
584 \ifx\reserved@a\@empty \let#3\@empty
585 \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
586 \fi}
```

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

```
587 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
```

```

588 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
589 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
590 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
591 \ProcessOptions\relax
592 \bxjs@composite@proc

  後処理

593 \if@slide
594   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
595 \fi
596 \if@landscape
597   \setlength\@tempdima {\paperheight}
598   \setlength\paperheight{\paperwidth}
599   \setlength\paperwidth {\@tempdima}
600 \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。

```

601 \@onlypreamble\bxjs@purge@brace@elts
602 \def\bxjs@purge@brace@elts{%
603   \def\bxjs@tmpa{\gobble}%
604   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
605   \let\@classoptionslist\bxjs@tmpa}
606 \@onlypreamble\bxjs@purge@be@a
607 \def\bxjs@purge@be@a#1,{%
608   \ifx\@nil#1\relax\else
609     \bxjs@purge@be@b#1{\@nil
610       \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
611       \expandafter\bxjs@purge@be@a
612       \fi}
613 \@onlypreamble\bxjs@purge@be@b
614 \def\bxjs@purge@be@b#1#{\bxjs@purge@be@c}
615 \@onlypreamble\bxjs@purge@be@c
616 \def\bxjs@purge@be@c#1\@nil{%
617   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
618 \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。

```

619 \@expandtwoargs\@removeelement
620 {papersize}\@classoptionslist\@classoptionslist
621 \@expandtwoargs\@removeelement
622 {10pt}\@classoptionslist\@classoptionslist
623 \@expandtwoargs\@removeelement
624 {noscale}\@classoptionslist\@classoptionslist

```

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

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

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

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

```
625 \ifx\bxjs@jadriver\relax\else
626   \let\bxjs@jadriver@given\bxjs@jadriver
627 \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。

```
628 \let\bxjs@tmpb\jsEngine
629 \ifx j\bxjs@tmpb\ifjsWithpTeXng
630   \let\bxjs@tmpb=g
631 \fi\fi
632 \ifx j\bxjs@tmpb\ifjsWithupTeX
633   \let\bxjs@tmpb=u
634 \fi\fi
635 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
636   \let\bxjs@tmpb=n
637 \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。）

```
638 \ifx *\bxjs@engine@given
639   \let\bxjs@engine@given\bxjs@tmpb
```

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

```
640 \ifx j\bxjs@engine@given
641   \g@addto@macro\@classoptionslist{,latex}
642 \else\ifx u\bxjs@engine@given
643   \g@addto@macro\@classoptionslist{,uplatex}
644 \fi\fi
645 \fi
646 \ifx\bxjs@engine@given\@undefined\else
647   \ifx\bxjs@engine@given\bxjs@tmpb\else
648     \ClassError\bxjs@clsname
649     {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
650 \fi
```


```
651 \fi
```

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

```
652 \ifjsWithpTeXng
```

```
653   \g@addto@macro\@classoptionslist{,uplatex}
```

```
654 \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。

```
655 \@tempswatrue
```

```
656 \ifx \bxjs@driver@given\@undefined\else
```

```
657   \ifjsInPdfMode
```

```
658     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
```

```
659       \@tempswafalse
```

```
660     \fi
```

```
661   \else\ifx x\jsEngine
```

```
662     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
```

```
663       \@tempswafalse
```

```
664     \fi
```

```
665   \else
```

```
666     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
```

```
667       \@tempswafalse
```

```
668     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
```

```
669       \@tempswafalse
```

```
670     \fi\fi
```

```
671   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
```

```
672     \@tempswafalse
```

```
673   \fi\fi
```

```
674 \fi\fi
```

```
675 \fi
```

```
676 \if@tempswa\else
```

```
677   \ClassError\bxjs@clsname
```

```
678     {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
```

```
679 \fi
```

DVI 出力のエンジンである場合の追加処理。

```
680 \ifjsInPdfMode \@tempswafalse
```

```
681 \else\ifx x\jsEngine \@tempswafalse
```

```
682 \else\ifjsWithpTeXng \@tempswafalse
```

```
683 \else \@tempswatrue
```

```
684 \fi\fi\fi
```

```
685 \if@tempswa
```

ドライバオプションがない場合は警告を出す。

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

```
686 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
```

```
687   \ClassWarningNoLine\bxjs@clsname
```

```
688     {No driver option is given}
```

```
689 \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。)

```

690 \ifbxjs@dvi@opt
691   \edef\bxjs@nxt{%
692     \let\noexpand\bxjs@driver@given
693     \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
694     \noexpand\g@addto@macro\noexpand\@classoptionslist
695     {,\bxjs@driver@opt}%
696   }\bxjs@nxt
697 \fi
698 \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 latex-ng* (*付) の場合、および既に dvipdfmx が指定されている場合を除く。

```

699 \ifjsWithpTeXng
700   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
701     \let\bxjs@platexng@nodrv\@undefined
702   \else\ifx t\bxjs@platexng@nodrv\else
703     \g@addto@macro\@classoptionslist{,dvipdfmx}
704   \fi\fi
705 \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。

```

706 \ifx\bxjs@driver@given\bxjs@driver@@none
707   \bxjs@papersizefalse
708 \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。

```

709 \def\bxjs@@minimal{minimal}
710 \ifx\bxjs@jadriver\relax
711   \ifx j\jsEngine
712     \def\bxjs@jadriver{standard}
713   \else
714     \let\bxjs@jadriver\bxjs@@minimal
715   \fi
716 \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。

```

717 \ifx\bxjs@jadriver@given\@undefined\else
718   \ifx\bxjs@engine@given\@undefined
719     \ClassError\bxjs@clsname
720     {An engine option must be explicitly given}%
721     {When you use a Japanese-driver you must specify a correct\MessageBreak
722       engine option.\MessageBreak\@ehc}
723 \fi\fi

```

新しい Lua_T_EX (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。)

```

724 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
725   \ifx\bxjs@magstyle\bxjs@magstyle@mag
726     \let\bxjs@magstyle\bxjs@magstyle@default
727     \ClassError\bxjs@clsname
728     {The engine does not support 'magstyle=usemag'}%
729     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
730       The default value 'nomag*' is used instead.\MessageBreak \@ehc}
731   \fi
732 \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。

```

733 \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。

```

734 \let\Cjascale\jsScale

```

8bit 欧文 T_EX 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。)

```

735 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
736   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
737     \catcode\@tempcnta\active
738     \advance\@tempcnta\@ne
739   \repeat
740 \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。

```

741 \ifbxjs@disguise@js
742 %<book|report>\def\bxjs@js@clsname{jsbook}
743 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
744 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
745 \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。

```

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

```

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

```

747 \ifbxjs@oldfontcommands
748   \AtEndOfClass{\allowoldfontcommands}
749 \fi

```

■papersize スペシャルの出力 dvips の papersize スペシャルを出力します。これで dvips や新しい dviout で出力領域が自動設定できます。トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが pL^AT_EX 2_ε 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。

```
750 \newdimen\stockwidth \newdimen\stockheight
751 \begingroup\expandafter\expandafter\expandafter\endgroup
752 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
753 % \newdimen\stockwidth \newdimen\stockheight
754 \setlength{\stockwidth}{\paperwidth}
755 \setlength{\stockheight}{\paperheight}
756 \advance \stockwidth 2in
757 \advance \stockheight 2in
758 \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。

```
759 %<slide>\def\n@baseline{13}%
760 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
761 %<!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。

```
762 \ifx\bxjs@magstyle\bxjs@magstyle@mag
763 \jsc@magtrue
764 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
765 \jsc@mag@xrealtrue
766 \fi\fi
```


サイズの変更は \TeX 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) \times 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。

```
767 \ifx\bxjs@param@mag\relax
768   \@tempdima=\bxjs@param@basefontsize
769   \advance\@tempdima.001pt \multiply\@tempdima25
770   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
771   \edef\bxjs@param@mag{\the\@tempcnta}
772 \else
773 % mag 値が直接指定された場合
774   \let\c@bxjs@cnta\@tempcnta
775   \setcounter{bxjs@cnta}{\bxjs@param@mag}
776   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
777 % 有効な mag 値の範囲は 1--32768
778   \edef\bxjs@param@mag{\the\@tempcnta}
779   \advance\@tempcnta100000
780   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
781   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
782   \edef\bxjs@param@basefontsize{\the\@tempdima}
783 \fi
784 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
785 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
786 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
787 \edef\jsc@magscale{\strip@pt\@tempdima}
788 \let\jsBaseFontSize\bxjs@param@basefontsize
789 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
790 %\typeout{\string\jsEngine: \meaning\jsEngine}
791 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
792 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
793 %\typeout{\string\jsc@magscale: \jsc@magscale}
794 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
795 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}
```

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

\mag する場合 (現状はこれが既定) にコードの変更を低減するために、以下では必要に応

じて、`\jsc@mp` 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@mp` 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。

```

796 \newdimen\jsc@mp
797 \newdimen\jsc@mmm
798 \ifjsc@mag
799   \jsc@mp=1\p@
800   \jsc@mmm=1mm
801   \catcode\?=9 % \p@? read as \p@
802 \else
803   \jsc@mp=\jsc@magscale\p@
804   \jsc@mmm=\jsc@magscale mm
805   \catcode\?=11 \let\p@?\jsc@mp
806 \fi
807 \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。

```

808 \newdimen\jsZw
809 \jsZw=10\jsc@mp \jsZw=\jsScale\jsZw
810 \ifbxjs@usezw
811   \providecommand*\zw{\jsZw}
812 \fi

```

`\zwspace` 全角幅の水平空き。

```

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

```

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

```

814 \ifjsc@mag@xreal
815   \RequirePackage{type1cm}
816   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

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

```

```

830 \fi}
831 \def\jsc@preadjust@extract@font{%
832 \let\jsc@req@size\f@size
833 \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
834 \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
835 \let\jsc@ref@size\jsc@tmpa
836 \let\f@size\jsc@ref@size}
837 \def\execute@size@function#1{%
838 \let\jsc@cref@size\f@size
839 \let\f@size\jsc@req@size
840 \csname s@fct@#1\endcsname}
841 \let\jsc@DeclareErrorFont\DeclareErrorFont
842 \def\DeclareErrorFont#1#2#3#4#5{%
843 \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
844 \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
845 \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
846 \def\gen@sfcnt{%
847 \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
848 \empty@sfcnt}
849 \def\genb@sfcnt{%
850 \edef\mandatory@arg{%
851 \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@}%
852 \empty@sfcnt}
853 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
854 \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 855 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 856 %\def\jsc@medskip{\vspace\jsc@medskipamount}
857 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 858 \newskip\jsc@smallskipamount
859 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
\jsc@bigskipamount 860 %\newskip\jsc@medskipamount
861 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
862 %\newskip\jsc@bigskipamount
863 %\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。

```
864 % \ifpapersize
865 %   \setlength{\@tempdima}{\paperwidth}
866 %   \setlength{\@tempdimb}{\paperheight}
867 %   \iftombow
868 %     \advance \@tempdima 2truein
869 %     \advance \@tempdimb 2truein
870 %   \fi
871 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
872 % \fi
```

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

\@ 欧文といえば、 \LaTeX の `\def\@{\spacefactor\@m}` という定義 ($\@m$ は 1000) では `I watch TV\@.` と書くと `V` とピリオドの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 \LaTeX で、auxiliary files に書き出されたときにスペースが食われないようにする修正が入りました。これに合わせて `{}` を補いました。

BXJS クラスでの変更点：

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

```
873 \chardef\bxjs@periodchar=`\ .
874 \bxjs@protected\def\bxjs@SE{%
875   \ifnum\spacefactor<\@m \spacefactor\@m
876   \else \spacefactor\sfcode\bxjs@periodchar
877   \fi}
878 \ifbxjs@fix@at@cmd
879   \def\@{\bxjs@SE{}}
880 \fi
```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

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

```
881 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
882 \expandafter\bxjs@tmpa\expandafter{%
883   \set@fontsize{#1}{#2}{#3}%
884 % 末尾にコードを追加
885   \expandafter\def\expandafter\size@update\expandafter{%
886     \size@update
887     \jsFontSizeChanged}%
888 }
```

`\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。

```
889 \newcommand*\jsFontSizeChanged{%
890   \jsZw=\f@size\p@
891   \jsZw=\jsScale \jsZw
892   \ifdim\parindent>\z@
893     \if@english \parindent=1em
894     \else       \parindent=1\jsZw
895   \fi
896   \fi\relax
897   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
898 \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。

```
899 \ifjsc@mag
900   \let\jsc@setfontsize\@setfontsize
901 \else
902   \def\jsc@setfontsize#1#2#3{%
903     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
904 \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。

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

```

906 \newif\ifnarrowbaselines
907 \if@english
908   \narrowbaselinestrue
909 \fi
910 \def\narrowbaselines{%
911   \narrowbaselinestrue
912   \skip0=\abovedisplayskip
913   \skip2=\abovedisplayshortskip
914   \skip4=\belowdisplayskip
915   \skip6=\belowdisplayshortskip
916 % 一時的に警告を無効化する
917   \let\bxjs@ltx@nomath\@nomath
918   \let\@nomath\@gobble
919   \@currsize\selectfont
920   \let\@nomath\bxjs@ltx@nomath
921   \abovedisplayskip=\skip0
922   \abovedisplayshortskip=\skip2
923   \belowdisplayskip=\skip4
924   \belowdisplayshortskip=\skip6\relax}
925 \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。

```

926 \def\bxjs@if@narrowbaselines{%
927   \ifnarrowbaselines\expandafter\@firstoftwo
928   \else \expandafter\@secondoftwo
929   \fi
930 }

```

`\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。

```

931 \renewcommand{\normalsize}{%
932   \bxjs@if@narrowbaselines{%
933     \jsc@setfontsize\normalsize\@xpt\@xipt

```

```

934 }{%else
935   \jsc@setfontsize\normalsize\xpt{\n@baseline}%
936 }%

```

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

[2003-02-16] ちょっと変えました。

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

```

937 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
938 \abovedisplayshortskip \z@ \@plus3\p@?
939 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
940 \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。

```

941 \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。

```

942 \normalsize

```

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

`\Chs`

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

```

943 \ifx\Cht\@undefined \newdimen\Cht \fi
944 \ifx\Cdp\@undefined \newdimen\Cdp \fi
945 \ifx\Cwd\@undefined \newdimen\Cwd \fi
946 \ifx\Cvs\@undefined \newdimen\Cvs \fi
947 \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。

```

948 \setlength\Cht{0.88\jsZw}
949 \setlength\Cdp{0.12\jsZw}
950 \setlength\Cwd{1\jsZw}
951 \setlength\Cvs{\baselineskip}
952 \setlength\Chs{1\jsZw}

```

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

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

```

953 \newcommand{\small}{%
954   \bxjs@if@narrowbaselines{%
955     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
956     %<kiyou>    \jsc@setfontsize\small{8.888}{11}%
957   }{%else
958     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
959     %<kiyou>    \jsc@setfontsize\small{8.888}{13.2418}%
960   }%
961   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
962   \abovedisplayshortskip \z@ \@plus3\p@?
963   \belowdisplayskip \abovedisplayskip
964   \belowdisplayshortskip \belowdisplayskip
965   \def\@listi{\leftmargin\leftmargini
966             \topsep \z@
967             \parsep \z@
968             \itemsep \parsep}}

```

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

```

969 \newcommand{\footnotesize}{%
970   \bxjs@if@narrowbaselines{%
971     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
972     %<kiyou>    \jsc@setfontsize\footnotesize{8.888}{11}%
973   }{%else
974     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
975     %<kiyou>    \jsc@setfontsize\footnotesize{8.888}{13.2418}%
976   }%
977   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
978   \abovedisplayshortskip \z@ \@plus2\p@?
979   \belowdisplayskip \abovedisplayskip
980   \belowdisplayshortskip \belowdisplayskip
981   \def\@listi{\leftmargin\leftmargini
982             \topsep \z@
983             \parsep \z@
984             \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 985 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
986 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 987 \if@twocolumn
988 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}

```

```

989 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
990 \else
991 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large\xi\pt{17}}
992 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
993 \fi
994 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\xiv\pt{21}}
995 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
996 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\xv\pt{25}}
997 \newcommand{\huge}{\jsc@setfontsize\huge\xx\pt{28}}
998 \newcommand{\Huge}{\jsc@setfontsize\Huge\xxv\pt{33}}
999 \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。

```

1000 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}

```

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

見出し用のフォントは `\bfseries` 固定ではなく、`\headfont` という命令で定めることにします。これは太ゴシックが使えるときは `\sffamily \bfseries` でいいと思いますが、通常の中ゴシックでは単に `\sffamily` だけのほうがよさそうです。『`LaTeX 2ε` 美文書作成入門』(1997 年)では `\sffamily \fontseries{sbc}` として新ゴ M と合わせましたが、`\fontseries{sbc}` はちょっと幅が狭いように感じました。

```

1001 % \newcommand{\headfont}{\bfseries}
1002 \newcommand{\headfont}{\sffamily}
1003 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

5 レイアウト

■二段組

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

```

1004 %<kiyou>\setlength\columnsep{2\Cwd}
1005 %<kiyou>\setlength\columnsep{28truebp}
1006 \setlength\columnseprule{0\jsc@empt}

```

■段落

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

```

\lineskiplimit

```

```

\normallineskiplimit

```

```

1007 \setlength\lineskip{1\jsc@empt}
1008 \setlength\normallineskip{1\jsc@empt}
1009 \setlength\lineskiplimit{1\jsc@empt}
1010 \setlength\normallineskiplimit{1\jsc@empt}

```

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

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

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

```
1011 \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。

```

1012 \setlength\parskip{0\jsc@empt}
1013 \if@slide
1014   \setlength\parindent{0\p@}
1015 \else
1016   \setlength\parindent{1\Cwd}
1017 \fi

```

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

```

\@highpenalty 1018 \@lowpenalty 51
1019 \@medpenalty 151
1020 \@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。

```
1021 % \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。

```
1022 % \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。

```
1023 \ifjsc@mag
```

```

1024 \mag=\bxjs@param@mag
1025 \fi
1026 \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。

1027 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
1028 \def\bxjs@read@b{%
1029 \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
1030 \else \expandafter\bxjs@read@d \fi}
1031 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
1032 \def\bxjs@read@d#1\@nil{}
1033 \expandafter\bxjs@read@a\bxjs@param@paper\@nil

```

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

```

1034 \edef\bxjs@layout@paper{%
1035 \ifjsc@mag truedimen,\fi
1036 \if@landscape landscape,\fi
1037 \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。

```

1038 %<*article|report>
1039 \def\bxjs@layout{%
1040 headheight=\topskip,footskip=0.03367\paperheight,%
1041 headsep=\footskip-\topskip,includeheadfoot,%
1042 hscale=0.76,hmarginratio=1:1,%
1043 vscale=0.83,vmarginratio=1:1,%
1044 }
1045 %</article|report>
1046 %<*book>
1047 \ifbxjs@layout@buggyhmargin %---
1048 % アレ
1049 \def\bxjs@layout{%
1050 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1051 hmargin=36\jsc@mmm,hmarginratio=1:1,%
1052 vscale=0.83,vmarginratio=1:1,%
1053 }
1054 \else %---
1055 % 非アレ
1056 \def\bxjs@layout{%
1057 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1058 hmargin=18\jsc@mmm,%
1059 vscale=0.83,vmarginratio=1:1,%
1060 }
1061 \fi %---
1062 %</book>
1063 %<*slide>
1064 \def\bxjs@layout{%
1065 noheadfoot,%

```

```

1066 hscale=0.9,hmarginratio=1:1,%
1067 vscale=0.944,vmarginratio=1:1,%
1068 }
1069 %</slide>

```

`\fullwidth` 〔寸法レジスタ〕 ヘッダ・フッタ領域の横幅。

```

1070 \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。

```

1071 %<*book>
1072 \newcommand\jsTextWidthLimit{40}
1073 \ifx\bxjs@textwidth@limit\undefined\else
1074   \let\c@bxjs@cmta\@tempcmta
1075   \setcounter{bxjs@cmta}{\bxjs@textwidth@limit}
1076   \long\edef\jsTextWidthLimit{\the\@tempcmta}
1077 \fi
1078 %</book>

```

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

```

1079 \def\bxjs@postproc@layout{%
1080 % ドライバ再設定
1081   \ifx\bxjs@geometry@driver\relax\else
1082     \let\Gm@driver\bxjs@geometry@driver
1083   \fi
1084 % textwidth 調整
1085   \ifbxjs@whole@zw@lines
1086     \@tempdimb=\textwidth
1087     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1088     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1089     \advance\@tempdimb-\textwidth
1090     \advance\oddsidemargin 0.5\@tempdimb
1091     \advance\evensidemargin 0.5\@tempdimb
1092   \fi
1093   \fullwidth=\textwidth
1094 %<*book>
1095   \ifdim\textwidth>\jsTextWidthLimit\Cwd
1096     \textwidth=\jsTextWidthLimit\Cwd
1097     \addtolength\evensidemargin{\fullwidth-\textwidth}
1098   \fi
1099 %</book>
1100 % textheight 調整
1101   \@tempdimb=\textheight
1102   \advance\textheight-\topskip
1103   \divide\textheight\baselineskip \multiply\textheight\baselineskip
1104   \advance\textheight\topskip
1105   \advance\@tempdimb-\textheight
1106   \advance\topmargin0.5\@tempdimb
1107 % headheight 調整
1108   \@tempdima=\topskip

```

```

1109 \advance\headheight\@tempdima
1110 \advance\topmargin-\@tempdima
1111 % marginpar 設定
1112 \setlength\marginparsep{\columnsep}
1113 \setlength\marginparpush{\baselineskip}
1114 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1115     -\textwidth-10\jsc@mmm-\marginparsep}
1116 \ifbxjs@whole@zw@lines
1117     \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1118 \fi
1119 % 連動する変数
1120 \maxdepth=.5\topskip
1121 \stockwidth=\paperwidth
1122 \stockheight=\paperheight
1123 }

```

`\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。

```

1124 \edef\jsGeometryOptions{%
1125     \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。

```

1126 \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。

```

1127 \@onlypreamble\bxjs@revert
1128 \let\bxjs@revert\@empty
1129 \edef\bxjs@tmpa{\the\ht\strutbox}
1130 \ht\strutbox=10\p@?
1131 \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。

```

1132 \ifbxjs@papersize
1133     \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1134         \PassOptionsToPackage{dvipdfm}{geometry}
1135     \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1136         \PassOptionsToPackage{dvipdfm}{geometry}
1137     \fi\fi

```

```

1138 \let\bxPapersizeSpecialDone=t
1139 \else
1140 \PassOptionsToPackage{driver=none}{geometry}
1141 \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。

```

1142 \edef\bxjs@nxt{%
1143 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1144 \AtBeginDocument{\bxjs@pre@geometry@hook}
1145 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1146 \bxjs@nxt \bxjs@revert
1147 \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。

```

1148 \let\bxjs@geometry@driver\Gm@driver
1149 \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。

```

1150 \@onlypreamble\bxjs@pre@geometry@hook
1151 \def\bxjs@pre@geometry@hook{%
1152 \ifpackageloaded{geometry}{%
1153 \ifpackagelater{geometry}{2010/02/12}{%}%else
1154 \PackageWarningNoLine\bxjs@clsname
1155 {The 'geometry' package installed\MessageBreak
1156 is too old (< v5.0)}%
1157 \if x\jsEngine \ifnum\mag=\@m\else
1158 \def\bxjs@Gm@driver{pdftex}
1159 \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1160 \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1161 \fi\fi
1162 }%
1163 \ifjsWithpTeXng
1164 \ifx\Gm@driver\@empty
1165 \def\Gm@driver{pdftex}%
1166 \fi
1167 \fi
1168 }{}}

```

`\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。

```

1169 \def\setpagelayout{\@ifstar
1170   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1171 \def\bxjs@modify@layout#1{%
1172   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1173   \bxjs@nxt\bxjs@postproc@layout}
1174 \def\bxjs@reset@layout#1{%
1175   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1176   \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。

```

1177 \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。

```

1178 \AtBeginDocument{\bxjs@check@page@layout}
1179 \@onlypreamble\bxjs@check@page@layout
1180 \def\bxjs@check@page@layout{%
1181   \ifdim\textwidth=.5\maxdimen
1182     \ClassError\bxjs@clsname
1183       {Page layout is not properly set}%
1184       {\@ehd}
1185   \fi}
1186 \def\jsUseMinimalPageLayout{%
1187   \setlength{\textwidth}{6.5in}%
1188   \setlength{\textheight}{8in}}

```

\setpagelayout はとりあえず無効にしておく。

```

1189 \let\bxjs@geometry@driver\relax
1190 \def\setpagelayout{\@ifstar
1191   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1192 \def\bxjs@pagelayout@a#1{%
1193   \ClassError\bxjs@clsname
1194     {Command '\string\setpagelayout' is not supported,\MessageBreak
1195     because 'geometry' value is not 'class'}\@eha}

```

geometry=user の場合の処理はここで終わり。

```

1196 \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。

```
1197 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1198 \if@slide
1199   \setlength\headheight{0\jsc@empt}
1200 \else
1201   \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1202 \fi
```

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

```
1203 %<*article|kiyou>
1204 \if@slide
1205   \setlength\footskip{0pt}
1206 \else
1207   \setlength\footskip{0.03367\paperheight}
1208   \ifdim\footskip<\baselineskip
1209     \setlength\footskip{\baselineskip}
1210   \fi
1211 \fi
1212 %</article|kiyou>
1213 %<jspf>\setlength\footskip{9\jsc@mmm}
1214 %<*book>
1215 \if@report
1216   \setlength\footskip{0.03367\paperheight}
1217   \ifdim\footskip<\baselineskip
1218     \setlength\footskip{\baselineskip}
1219   \fi
1220 \else
1221   \setlength\footskip{0pt}
1222 \fi
```

```

1223 %</book>
1224 %<*report>
1225 \setlength\footskip{0.03367\paperheight}
1226 \ifdim\footskip<\baselineskip
1227   \setlength\footskip{\baselineskip}
1228 \fi
1229 %</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)。

```

1230 %<*article>
1231 \if@slide
1232   \setlength\headsep{0\jsc@empt}
1233   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1234   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1235 \else
1236   \setlength\headsep{\footskip}
1237   \addtolength\headsep{-\topskip}
1238 \fi
1239 %</article>
1240 %<*book>
1241 \if@report
1242   \setlength\headsep{\footskip}
1243   \addtolength\headsep{-\topskip}
1244 \else
1245   \setlength\headsep{6\jsc@mmm}
1246   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1247   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1248 \fi
1249 %</book>
1250 %<*report>
1251 \setlength\headsep{\footskip}
1252 \addtolength\headsep{-\topskip}
1253 %</report>
1254 %<*jspf>
1255 \setlength\headsep{9\jsc@mmm}
1256 \addtolength\headsep{-\topskip}
1257 %</jspf>
1258 %<*kiyou>
1259 \setlength\headheight{0\jsc@empt}
1260 \setlength\headsep{0\jsc@empt}
1261 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1262 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1263 %</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。

```
1264 \setlength\maxdepth{.5\topskip}
```

■本文の幅と高さ

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

```
1265 \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。

```
1266 %<*article>
1267 \if@slide
1268   \setlength\fullwidth{0.9\paperwidth}
1269 \else
1270   \setlength\fullwidth{0.76\paperwidth}
1271 \fi
1272 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1273 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1274 \setlength\textwidth{\fullwidth}
1275 %</article>
1276 %<*book>
1277 \if@report
1278   \setlength\fullwidth{0.76\paperwidth}
1279 \else
1280   \setlength\fullwidth{\paperwidth}
1281   \addtolength\fullwidth{-36\jsc@mmm}
1282 \fi
1283 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1284 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1285 \setlength\textwidth{\fullwidth}
1286 \if@report \else
1287   \if@twocolumn \else
1288     \ifdim \fullwidth>40zw
1289       \setlength\textwidth{40zw}
```

```

1290 \fi
1291 \fi
1292 \fi
1293 %</book>
1294 %<*report>
1295 \setlength\fullwidth{0.76\paperwidth}
1296 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1297 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1298 \setlength\textwidth{\fullwidth}
1299 %</report>
1300 %<*jspf>
1301 \setlength\fullwidth{50zw}
1302 \addtolength\fullwidth{8\jsc@mmm}
1303 \setlength\textwidth{\fullwidth}
1304 %</jspf>
1305 %<*kiyou>
1306 \setlength\fullwidth{48zw}
1307 \addtolength\fullwidth{\columnsep}
1308 \setlength\textwidth{\fullwidth}
1309 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

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

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

```

1310 %<*article|book|report>
1311 \if@slide
1312 \setlength{\textheight}{0.95\paperheight}
1313 \else
1314 \setlength{\textheight}{0.83\paperheight}
1315 \fi
1316 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1317 \addtolength{\textheight}{-\headsep}
1318 \addtolength{\textheight}{-\footskip}
1319 \addtolength{\textheight}{-\topskip}

```

```

1320 \divide\textheight\baselineskip
1321 \multiply\textheight\baselineskip
1322 %</article|book|report>
1323 %<jspf>\setlength{\textheight}{51\baselineskip}
1324 %<kiyou>\setlength{\textheight}{47\baselineskip}
1325 \addtolength{\textheight}{\topskip}
1326 \addtolength{\textheight}{0.1\jsc@mpt}
1327 %<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 \LaTeX 2_ϵ での完全な `\flushbottom` の定義は

```

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

```

ですが、次のようにします。

```

1328 \def\flushbottom{%
1329   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1330   \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。

```

1331 \setlength\marginparsep{\columnsep}
1332 \setlength\marginparpush{\baselineskip}

```

`\oddsidemargin` それぞれ奇数ページ、偶数ページの左マージンから 1 インチ引いた値です。片面印刷では `\evensidemargin` `\oddsidemargin` が使われます。 \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 $\text{\pLaTeX 2}_\epsilon$ (`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。

```

1333 \setlength{\oddsidemargin}{\paperwidth}
1334 \addtolength{\oddsidemargin}{-\fullwidth}
1335 \setlength{\oddsidemargin}{.5\oddsidemargin}
1336 \iftombow
1337   \addtolength{\oddsidemargin}{-1in}
1338 \else
1339   \addtolength{\oddsidemargin}{-\inv@mag in}
1340 \fi
1341 \setlength{\evensidemargin}{\oddsidemargin}
1342 \if@mparswitch
1343   \addtolength{\evensidemargin}{\fullwidth}
1344   \addtolength{\evensidemargin}{-\textwidth}
1345 \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。

```

1346 \setlength\marginparwidth{\paperwidth}

```

```

1347 \addtolength\marginparwidth{-\oddsidemargin}
1348 \addtolength\marginparwidth{-\inv@mag in}
1349 \addtolength\marginparwidth{-\textwidth}
1350 \addtolength\marginparwidth{-10\jsc@mmm}
1351 \addtolength\marginparwidth{-\marginparsep}
1352 \@tempdima=1zw
1353 \divide\marginparwidth\@tempdima
1354 \multiply\marginparwidth\@tempdima

```

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

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

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

```

1355 \setlength\topmargin{\paperheight}
1356 \addtolength\topmargin{-\textheight}
1357 \if@slide
1358   \addtolength\topmargin{-\headheight}
1359 \else
1360   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1361 \fi
1362 \addtolength\topmargin{-\headsep}
1363 \addtolength\topmargin{-\footskip}
1364 \setlength\topmargin{0.5\topmargin}
1365 %<kiyou>\setlength\topmargin{81truebp}
1366 \iftombow
1367   \addtolength\topmargin{-1in}
1368 \else
1369   \addtolength\topmargin{-\inv@mag in}
1370 \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。

1371 \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。

1372 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}

■フロート関連 フロート（図、表）関連のパラメータは L^AT_EX 2_ε 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。

1373 \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。

1374 \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。

1375 \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。

1376 \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。

1377 \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。

1378 \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。

1379 \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。

1380 \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。

1381 \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。

```
1382 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1383 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1384 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1385 \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 1386 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1387 \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 1388 \setlength\@fptop{0\p@? \@plus 1fil}
```

```
1389 \setlength\@fpsep{8\p@? \@plus 2fil}
```

```
1390 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpptop` 段抜きフロートについての値です。

```
\@dblfpsep 1391 \setlength\@dblfpptop{0\p@? \@plus 1fil}
```

```
1392 \setlength\@dblfpsep{8\p@? \@plus 2fil}
```

```
\@dblfpbot 1393 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

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

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

となっています。

```
1394 %\def\pltx@cleartorightpage{\clearpage\if@twoside
```

```
1395 % \ifodd\c@page
```

```
1396 % \iftdir
```

```
1397 % \hbox{}\thispagestyle{empty}\newpage
```

```
1398 % \if@twocolumn\hbox{}\newpage\fi
```

```
1399 % \fi
```

```
1400 % \else
```

```
1401 % \ifydir
```

```
1402 % \hbox{}\thispagestyle{empty}\newpage
```



```

1403 %      \if@twocolumn\hbox{}\newpage\fi
1404 %      \fi
1405 %    \fi\fi}
1406 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1407 %  \ifodd\c@page
1408 %    \ifydir
1409 %      \hbox{}\thispagestyle{empty}\newpage
1410 %    \if@twocolumn\hbox{}\newpage\fi
1411 %    \fi
1412 %  \else
1413 %    \iftdir
1414 %      \hbox{}\thispagestyle{empty}\newpage
1415 %    \if@twocolumn\hbox{}\newpage\fi
1416 %    \fi
1417 %  \fi\fi}
1418 %\def\pltx@cleartooddpage{\clearpage\if@twoside
1419 %  \ifodd\c@page\else
1420 %    \hbox{}\thispagestyle{empty}\newpage
1421 %  \if@twocolumn\hbox{}\newpage\fi
1422 %  \fi\fi}
1423 %\def\pltx@cleartoevenpage{\clearpage\if@twoside
1424 %  \ifodd\c@page
1425 %    \hbox{}\thispagestyle{empty}\newpage
1426 %  \if@twocolumn\hbox{}\newpage\fi
1427 %  \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。

```

1428 \let\pltx@cleartorightpage\pltx@cleartooddpage
1429 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

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

```

1430 %<*book|report>
1431 \if@openleft
1432   \let\cleardoublepage\pltx@cleartoleftpage
1433 \else\if@openright
1434   \let\cleardoublepage\pltx@cleartorightpage
1435 \fi\fi
1436 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

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

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

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

`\ps@empty` `empty` ページスタイルの定義です。L^AT_EX 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。

```
1437 % \def\ps@empty{%
1438 %   \let\@mkboth\@gobbletwo
1439 %   \let\@oddhead\@empty
1440 %   \let\@oddfoot\@empty
1441 %   \let\@evenhead\@empty
1442 %   \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。

```
1443 \def\ps@plainfoot{%
1444   \let\@mkboth\@gobbletwo
1445   \let\@oddhead\@empty
1446   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1447   \let\@evenhead\@empty
1448   \let\@evenfoot\@oddfoot}
1449 \def\ps@plainhead{%
1450   \let\@mkboth\@gobbletwo
1451   \let\@oddfoot\@empty
1452   \let\@evenfoot\@empty
1453   \def\@evenhead{%
1454     \if@mparswitch \hss \fi
```

```

1455 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1456 \if@mparswitch\else \hss \fi}%
1457 \def\@oddhead{%
1458 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}%
1459 %<book>\let\ps@plain\ps@plainhead
1460 %<!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。

```

1461 %<*article|slide>
1462 \if@twoside
1463 \def\ps@headings{%
1464 \let\@oddfoot\@empty
1465 \let\@evenfoot\@empty
1466 \def\@evenhead{\if@mparswitch \hss \fi
1467 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1468 \if@mparswitch\else \hss \fi}%
1469 \def\@oddhead{%
1470 \underline{%
1471 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1472 \let\@mkboth\markboth
1473 \def\sectionmark##1{\markboth{%
1474 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1475 ##1}\}}%
1476 \def\subsectionmark##1{\markright{%
1477 \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
1478 ##1}\}}%
1479 }
1480 \else % if not twoside
1481 \def\ps@headings{%
1482 \let\@oddfoot\@empty
1483 \def\@oddhead{%
1484 \underline{%
1485 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1486 \let\@mkboth\markboth
1487 \def\sectionmark##1{\markright{%
1488 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1489 ##1}\}}%
1490 \fi
1491 %</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。

```

1492 %<*book|report>
1493 \def\bxjs@maybe@autoxspacing{%
1494   \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1495 \newif\if@omit@number
1496 \def\ps@headings{%
1497   \let\@oddfoot\@empty
1498   \let\@evenfoot\@empty
1499   \def\@evenhead{%
1500     \if@mparswitch \hss \fi
1501     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1502       \textbf{\thepage}\hfil\leftmark}}}%
1503     \if@mparswitch\else \hss \fi}%
1504   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1505     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1506   \let\@mkboth\markboth
1507   \def\chaptermark##1{\markboth{%
1508     \ifnum \c@secnumdepth >\m@ne
1509       \if@mainmatter
1510         \if@omit@number\else
1511           \@chapapp\thechapter\@chappos\hskip1\jsZw
1512         \fi
1513       \fi
1514     \fi
1515     ##1}\{}}%
1516   \def\sectionmark##1{\markright{%
1517     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1518     ##1}\{}}%
1519 %</book|report>

```

最後は学会誌の場合です。

```

1520 %<*jspf>
1521 \def\ps@headings{%
1522   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1523   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1524   \def\@oddhead{\normalfont\hfil \@title \hfil}
1525   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1526 %</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。

```

1527 \def\ps@myheadings{%
1528   \let\@oddfoot\@empty\let\@evenfoot\@empty
1529   \def\@evenhead{%
1530     \if@mparswitch \hss \fi%
1531     \hbox to \fullwidth{\thepage\hfil\leftmark}}%
1532     \if@mparswitch\else \hss \fi}%
1533   \def\@oddhead{%
1534     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%

```

```

1535 \let\mkboth\@gobbletwo
1536 %<book|report> \let\chaptermark\@gobble
1537 \let\sectionmark\@gobble
1538 %<!book&!report> \let\subsectionmark\@gobble
1539 }

```

8 文書のマークアップ

8.1 表題

`\title` これらは L^AT_EX 本体で次のように定義されています。ここではコメントアウトした形で示します。

```

\date 1540 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1541 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1542 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1543 % \date{\today}

```

`\subtitle` 副題を設定する。

`\jsSubtitle` ※プレアンプルにおいて `\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。

```

1544 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1545 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1546 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1547 \AtBeginDocument{\bxjs@decl@subtitle}
1548 \def\bxjs@decl@subtitle{%
1549   \global\let\bxjs@decl@subtitle\relax
1550   \ifx\subtitle\@undefined
1551     \global\let\subtitle\jsSubtitle
1552   \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。

```

1553 \def\bxjs@annihilate@subtitle{%
1554   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1555   \global\let\jsSubtitle\relax}

```

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

```

\author 1556 %<*jspf>
\keywords 1557 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}

```

```

1558 \newcommand*\eauthor}[1]{\gdef\@eauthor{#1}}
1559 \newcommand*\keywords}[1]{\gdef\@keywords{#1}}
1560 \newcommand*\email}[1]{\gdef\authors@mail{#1}}
1561 \newcommand*\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1562 %</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。

```

1563 \def\plainifnotempty{%
1564   \ifx \@oddhead \@empty
1565     \ifx \@oddfoot \@empty
1566       \else
1567         \thispagestyle{plainfoot}%
1568       \fi
1569     \else
1570       \thispagestyle{plainhead}%
1571     \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。

```

1572 %<*article|book|report|slide>
1573 \if@titlepage
1574   \newcommand{\maketitle}{%
1575     \begin{titlepage}%
1576       \let\footnotesize\small
1577       \let\footnoterule\relax
1578       \let\footnote\thanks
1579       \null\vfil
1580       \if@slide
1581         {\footnotesize \@date}%
1582         \begin{center}
1583           \mbox{} \[\[1\jsZw]
1584           \large
1585           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1586           \jsc@smallskip
1587           \@title
1588           \ifx\bxjs@subtitle\@undefined\else
1589             \par\vskip\z@
1590             {\small \bxjs@subtitle\par}
1591           \fi
1592           \jsc@smallskip

```

```

1593         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1594         \vfill
1595         {\small \@author}%
1596     \end{center}
1597 \else
1598 \vskip 60\p@?
1599 \begin{center}%
1600     {\LARGE \@title \par}%
1601     \ifx\bxjs@subtitle\@undefined\else
1602         \vskip5\p@?
1603         {\normalsize \bxjs@subtitle\par}
1604     \fi
1605     \vskip 3em%
1606     {\large
1607         \lineskip .75em
1608         \begin{tabular}[t]{c}%
1609             \@author
1610         \end{tabular}\par}%
1611     \vskip 1.5em
1612     {\large \@date \par}%
1613 \end{center}%
1614 \fi
1615 \par
1616 \@thanks\vfil\null
1617 \end{titlepage}%
1618 \setcounter{footnote}{0}%
1619 \global\let\thanks\relax
1620 \global\let\maketitle\relax
1621 \global\let\@thanks\@empty
1622 \global\let\@author\@empty
1623 \global\let\@date\@empty
1624 \global\let\@title\@empty
1625 \global\let\title\relax
1626 \global\let\author\relax
1627 \global\let\date\relax
1628 \global\let\and\relax
1629 \bxjs@annihilate@subtitle
1630 }%
1631 \else
1632 \newcommand{\maketitle}{\par
1633     \begingroup
1634         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1635         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1636         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1637             \parindent 1\jsZw\noindent
1638             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1639         \if@twocolumn
1640             \ifnum \col@number=\@ne
1641                 \@maketitle

```

```

1642     \else
1643     \twocolumn[\@maketitle]%
1644     \fi
1645     \else
1646     \newpage
1647     \global\@topnum\z@ % Prevents figures from going at top of page.
1648     \@maketitle
1649     \fi
1650     \plainifnotempty
1651     \@thanks
1652 \endgroup
1653 \setcounter{footnote}{0}%
1654 \global\let\thanks\relax
1655 \global\let\maketitle\relax
1656 \global\let\@thanks\@empty
1657 \global\let\@author\@empty
1658 \global\let\@date\@empty
1659 \global\let\@title\@empty
1660 \global\let\title\relax
1661 \global\let\author\relax
1662 \global\let\date\relax
1663 \global\let\and\relax
1664 \bxjs@annihilate@subtitle
1665 }

```

`\@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。

```

1666 \def\@maketitle{%
1667   \newpage\null
1668   \vskip 2em
1669   \begin{center}%
1670     \let\footnote\thanks
1671     {\LARGE \@title \par}%
1672     \ifx\bxjs@subtitle\@undefined\else
1673       \vskip3\p@?
1674       {\normalsize \bxjs@subtitle\par}
1675     \fi
1676     \vskip 1.5em
1677     {\large
1678       \lineskip .5em
1679       \begin{tabular}[t]{c}%
1680         \@author
1681         \end{tabular}\par}%
1682     \vskip 1em
1683     {\large \@date}%
1684   \end{center}%
1685   \par\vskip 1.5em
1686 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1687 }
1688 \fi

```



```

1689 %</article|book|report|slide>
1690 %<*jspf>
1691 \newcommand{\maketitle}{\par
1692   \begin{group}
1693     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1694     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1695     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1696       \parindent 1\jsZw\noindent
1697       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1698     \twocolumn[\@maketitle]%
1699     \plainifnotempty
1700     \@thanks
1701   \end{group}
1702   \setcounter{footnote}{0}%
1703   \global\let\thanks\relax
1704   \global\let\maketitle\relax
1705   \global\let\@thanks\@empty
1706   \global\let\@author\@empty
1707   \global\let\@date\@empty
1708 % \global\let\@title\@empty % \@title は柱に使う
1709   \global\let\title\relax
1710   \global\let\author\relax
1711   \global\let\date\relax
1712   \global\let\and\relax
1713   \ifx\authors@mail\@undefined\else{%
1714     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1715     \footnotetext[0]{\itshape\authors@mail}%
1716   }\fi
1717   \global\let\authors@mail\@undefined}
1718 \def\@maketitle{%
1719   \newpage\null
1720   \vskip 6em % used to be 2em
1721   \begin{center}
1722     \let\footnote\thanks
1723     \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1724     \lineskip .5em
1725     \ifx\@author\@undefined\else
1726       \vskip 1em
1727       \begin{tabular}[t]{c}%
1728         \@author
1729       \end{tabular}\par
1730     \fi
1731     \ifx\@etitle\@undefined\else
1732       \vskip 1em
1733       {\large \@etitle \par}%
1734     \fi
1735     \ifx\@eauthor\@undefined\else
1736       \vskip 1em
1737       \begin{tabular}[t]{c}%

```

```

1738      \@eauthor
1739      \end{tabular}\par
1740      \fi
1741      \vskip 1em
1742      \@date
1743      \end{center}
1744      \vskip 1.5em
1745      \centerline{\box\@abstractbox}
1746      \ifx\@keywords\@undefined\else
1747      \vskip 1.5em
1748      \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1749      \fi
1750      \vskip 1.5em}
1751 %</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。

```

1752 \def\startsection#1#2#3#4#5#6{%
1753   \if@noskipsec \leavevmode \fi
1754   \par

```

```

1755 % 見出し上の空きを \@tempskipa にセットする
1756 \@tempskipa #4\relax
1757 % \@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
1758 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1759 % 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
1760 \ifdim \@tempskipa <\z@
1761   \@tempskipa -\@tempskipa \@afterindentfalse
1762 \fi
1763 \if@nobreak
1764 %   \everypar{\everyparhook}% これは間違い
1765   \everypar{}%
1766 \else
1767   \addpenalty\@secpenalty
1768 % 次の行は削除
1769 %   \addvspace\@tempskipa
1770 % 次の \noindent まで追加
1771   \ifdim \@tempskipa >\z@
1772     \if@slide\else
1773       \null
1774       \vspace*{-\baselineskip}%
1775     \fi
1776     \vskip\@tempskipa
1777   \fi
1778 \fi
1779 \noindent
1780 % 追加終わり
1781 \@ifstar
1782   {\@ssect{#3}{#4}{#5}{#6}}%
1783   {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

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

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

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

```

1784 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1785   \let\bxjs@if@ceph\firstofone
1786 \else \let\bxjs@if@ceph\@gobble
1787 \fi

```

```

1788 \def\@sect#1#2#3#4#5#6[#7]#8{%
1789   \ifnum #2>\c@secnumdepth
1790     \let\@svsec\@empty
1791   \else
1792     \refstepcounter{#1}%
1793     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1794   \fi

```

```

1795 % 見出し後の空きを \@tempskipa にセット
1796 \@tempskipa #5\relax
1797 % 条件判断の順序を入れ替えました
1798 \ifdim \@tempskipa<\z@
1799     \def\@svsechd{%
1800         #6{\hskip #3\relax
1801             \@svsec #8}%
1802         \csname #1mark\endcsname{#7}%
1803         \addcontentsline{toc}{#1}{%
1804             \ifnum #2>\c@secnumdepth \else
1805                 \protect\numberline{\csname the#1\endcsname}%
1806             \fi
1807             #7}}% 目次にフルネームを載せるなら #8
1808     \else
1809         \begingroup
1810             \interlinepenalty \@M % 下から移動
1811             #6{%
1812                 \@hangfrom{\hskip #3\relax\@svsec}%
1813             % \interlinepenalty \@M % 上に移動
1814             #8\@@par}%
1815         \endgroup
1816         \csname #1mark\endcsname{#7}%
1817         \addcontentsline{toc}{#1}{%
1818             \ifnum #2>\c@secnumdepth \else
1819                 \protect\numberline{\csname the#1\endcsname}%
1820             \fi
1821             #7}}% 目次にフルネームを載せるならここは #8
1822     \fi
1823     \@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的位置を微調整しました。

```

1824 \def\@xsect#1{%
1825 % 見出しの後ろの空きを \@tempskipa にセット
1826 \@tempskipa #1\relax
1827 % 条件判断の順序を変えました
1828 \ifdim \@tempskipa<\z@
1829     \@nbreakfalse
1830     \global\@noskipsecttrue
1831     \everypar{%
1832         \if@noskipsec
1833             \global\@noskipsecfalse
1834             {\setbox\z@\lastbox}%
1835             \clubpenalty\@M
1836         \begingroup \@svsechd \endgroup
1837         \unskip

```

```

1838      \@tempskipa #1\relax
1839      \hskip -\@tempskipa
1840      \else
1841      \clubpenalty \@clubpenalty
1842      \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1843      \fi\bxjs@if@ceph\everyparhook}%
1844      \else
1845      \par \nobreak
1846      \vskip \@tempskipa
1847      \@afterheading
1848      \fi
1849      \if@slide
1850      {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1851      \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1852      \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1853      \fi
1854      \par % 2000-12-18
1855      \ignorespaces}
1856 \def\@ssect#1#2#3#4#5{%
1857   \@tempskipa #3\relax
1858   \ifdim \@tempskipa<\z@
1859     \def\@svsechd{#4{\hskip #1\relax #5}}%
1860   \else
1861     \begingroup
1862     #4{%
1863       \@hangfrom{\hskip #1}%
1864       \interlinepenalty \@M #5\@@par}%
1865     \endgroup
1866   \fi
1867   \@xsect{#3}}

```

■柱関係の命令

`\chaptermark` `\...mark` の形の命令を初期化します (第 7 節参照)。`\chaptermark` 以外は L^AT_EX 本体で定義済みです。

```

\subsectionmark 1868 \newcommand*\chaptermark[1]{}
1869 % \newcommand*\sectionmark[1]{}
\subsubsectionmark 1870 % \newcommand*\subsectionmark[1]{}
1871 % \newcommand*\subsubsectionmark[1]{}
\paragraphmark 1872 % \newcommand*\paragraphmark[1]{}
\subparagraphmark 1873 % \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。

```

1874 %<!book&!report>\setcounter{secnumdepth}{3}
1875 %<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`

`\c@subsection`

`\c@subsubsection`

`\c@paragraph`

`\c@subparagraph`

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。

```
1876 \newcounter{part}
1877 %<book|report>\newcounter{chapter}
1878 %<book|report>\newcounter{section}[chapter]
1879 %<!book&!report>\newcounter{section}
1880 \newcounter{subsection}[section]
1881 \newcounter{subsubsection}[subsection]
1882 \newcounter{paragraph}[subsubsection]
1883 \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。

```
1884 \renewcommand{\thepart}{\@Roman\c@part}
1885 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1886 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1887 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1888 %<*book|report>
1889 \renewcommand{\thechapter}{\@arabic\c@chapter}
1890 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1891 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1892 %</book|report>
1893 \renewcommand{\thesubsubsection}{%
1894   \thesubsection.\@arabic\c@subsubsection}
1895 \renewcommand{\theparagraph}{%
1896   \thesubsubsection.\@arabic\c@paragraph}
1897 \renewcommand{\thesubparagraph}{%
1898   \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。

```
1899 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1900 %<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、 \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 \LaTeX の標準クラス 2017/03/05 も参照）。

```
1901 %<*book|report>
1902 \newcommand\frontmatter{%
1903   \pltx@cleartooddpage
1904   \@mainmatterfalse
1905   \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。

```
1906 \newcommand\mainmatter{%
1907   \pltx@cleartooddpage
1908   \@mainmattertrue
1909   \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。

```
1910 \newcommand\backmatter{%
1911   \if@openleft
1912     \cleardoublepage
1913   \else\if@openright
1914     \cleardoublepage
1915   \else
1916     \clearpage
1917   \fi\fi
1918   \@mainmatterfalse}
1919 %</book|report>
```

■部

`\part` 新しい部を始めます。

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

```
\secdef{星なし}{星あり}
```

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

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

まず `book` と `report` のクラス以外です。

```
1920 %<*&book&!report>
```

```

1921 \newcommand\part{%
1922   \if@noskipsec \leavevmode \fi
1923   \par
1924   \addvspace{4ex}%
1925   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1926   \secdef\@part\@spart}
1927 %<*/!book&!report>

```

book および report クラスの場合は、少し複雑です。

```

1928 %<*book|report>
1929 \newcommand\part{%
1930   \if@openleft
1931     \cleardoublepage
1932   \else\if@openright
1933     \cleardoublepage
1934   \else
1935     \clearpage
1936   \fi\fi
1937   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1938   \if@twocolumn
1939     \onecolumn
1940     \@restonecoltrue
1941   \else
1942     \@restonecolfalse
1943   \fi
1944   \null\vfil
1945   \secdef\@part\@spart}
1946 %</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。

```

1947 %<*!book&!report>
1948 \def\@part[#1]#2{%
1949   \ifnum \c@secnumdepth >\m@ne
1950     \refstepcounter{part}%
1951     \addcontentsline{toc}{part}{%
1952       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1953   \else
1954     \addcontentsline{toc}{part}{#1}%
1955   \fi
1956   \markboth{}{}%
1957   {\parindent\z@
1958     \raggedright
1959     \interlinepenalty \@M
1960     \normalfont
1961     \ifnum \c@secnumdepth >\m@ne
1962       \Large\headfont\prepartname\thepart\postpartname
1963       \par\nobreak

```



```

1964 \fi
1965 \huge \headfont #2%
1966 \markboth{}{}\par}%
1967 \nobreak
1968 \vskip 3ex
1969 \@afterheading}
1970 %</!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。

```

1971 %<*book|report>
1972 \def\@part[#1]#2{%
1973 \ifnum \c@secnumdepth >-2\relax
1974 \refstepcounter{part}%
1975 \addcontentsline{toc}{part}{%
1976 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1977 \else
1978 \addcontentsline{toc}{part}{#1}%
1979 \fi
1980 \markboth{}{}\%
1981 {\centering
1982 \interlinepenalty \@M
1983 \normalfont
1984 \ifnum \c@secnumdepth >-2\relax
1985 \huge\headfont \prepartname\thepart\postpartname
1986 \par\vskip20\p@?
1987 \fi
1988 \Huge \headfont #2\par}%
1989 \@endpart}
1990 %</book|report>

```

\@spart 番号を付けない部です。

```

1991 %<*!book&!report>
1992 \def\@spart#1{%
1993 \parindent \z@ \raggedright
1994 \interlinepenalty \@M
1995 \normalfont
1996 \huge \headfont #1\par}%
1997 \nobreak
1998 \vskip 3ex
1999 \@afterheading}
2000 %</!book&!report>
2001 %<*book|report>
2002 \def\@spart#1{%
2003 \centering
2004 \interlinepenalty \@M
2005 \normalfont
2006 \Huge \headfont #1\par}%
2007 \@endpart}
2008 %</book|report>

```

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

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

```
2009 %<*book|report>
2010 \def\@endpart{\vfil\newpage
2011   \if@twoside
2012     \if@openleft %% added (2017/02/24)
2013       \null\thispagestyle{empty}\newpage
2014     \else\if@openright %% added (2016/12/13)
2015       \null\thispagestyle{empty}\newpage
2016     \fi\fi %% added (2016/12/13, 2017/02/24)
2017   \fi
2018   \if@restonecol
2019     \twocolumn
2020   \fi}
2021 %</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。

```
2022 %<*book|report>
2023 \newcommand{\chapter}{%
2024   \if@openleft\cleardoublepage\else
2025   \if@openright\cleardoublepage\else\clearpage\fi\fi
2026   \plainifnotempty % 元: \thispagestyle{plain}
2027   \global\@topnum\z@
2028   \if@english \afterindentfalse \else \afterindenttrue \fi
2029   \secdef
2030     {\@omit@numberfalse\@chapter}%
2031     {\@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。

```
2032 \def\@chapter[#1]#2{%
2033   \ifnum \c@secnumdepth >\m@ne
2034     \if@mainmatter
2035       \refstepcounter{chapter}%
2036       \typeout{\@chapapp\thechapter\@chappos}%
2037       \addcontentsline{toc}{chapter}%
2038         {\protect\numberline
2039 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2040         {\@chapapp\thechapter\@chappos}%
2041         #1}%
2042     \else\addcontentsline{toc}{chapter}{#1}\fi
```

```

2043 \else
2044 \addcontentsline{toc}{chapter}{#1}%
2045 \fi
2046 \chaptermark{#1}%
2047 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
2048 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
2049 \if@twocolumn
2050 \@topnewpage[\@makechapterhead{#2}]%
2051 \else
2052 \@makechapterhead{#2}%
2053 \@afterheading
2054 \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。

```

2055 \def\@makechapterhead#1{%
2056 \vspace*{2\Cvs}% 欧文は 50pt
2057 {\parindent \z@ \raggedright \normalfont
2058 \ifnum \c@secnumdepth >\m@ne
2059 \if@mainmatter
2060 \huge\headfont \@chapapp\thechapter\@chappos
2061 \par\nobreak
2062 \vskip \Cvs % 欧文は 20pt
2063 \fi
2064 \fi
2065 \interlinepenalty\@M
2066 \Huge \headfont #1\par\nobreak
2067 \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。

```

2068 \def\@schapter#1{%
2069 \chaptermark{#1}%
2070 \if@twocolumn
2071 \@topnewpage[\@makeschapterhead{#1}]%
2072 \else
2073 \@makeschapterhead{#1}\@afterheading
2074 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2075 \def\@makeschapterhead#1{%
2076 \vspace*{2\Cvs}% 欧文は 50pt
2077 {\parindent \z@ \raggedright
2078 \normalfont
2079 \interlinepenalty\@M
2080 \Huge \headfont #1\par\nobreak
2081 \vskip 3\Cvs}} % 欧文は 40pt
2082 %</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。

```
2083 \if@twocolumn
2084   \newcommand{\section}{%
2085     %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2086     \@startsection{section}{1}{\z@}%
2087     %<kiyou>    {0.6\Cvs}{0.4\Cvs}%
2088     %<kiyou>    {\Cvs}{0.5\Cvs}%
2089     %    {\normalfont\large\headfont\@secapp}%
2090     {\normalfont\large\headfont\raggedright}}
2091 \else
2092   \newcommand{\section}{%
2093     \if@slide\clearpage\fi
2094     \@startsection{section}{1}{\z@}%
2095     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2096     {.5\Cvs \@plus.3\Cdp}% 後アキ
2097     %    {\normalfont\Large\headfont\@secapp}%
2098     {\normalfont\Large\headfont\raggedright}}
2099 \fi
```

`\subsection` 同上です。

```
2100 \if@twocolumn
2101   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2102     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2103     {\normalfont\normalsize\headfont}}
2104 \else
2105   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2106     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2107     {.5\Cvs \@plus.3\Cdp}% 後アキ
2108     {\normalfont\large\headfont}}
2109 \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)。

```
2110 \if@twocolumn
2111   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2112     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2113     {\normalfont\normalsize\headfont}}
2114 \else
2115   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2116     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2117     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2118     {\normalfont\normalsize\headfont}}
2119 \fi
```

`\paragraph` 見出しの後ろで改行されません。

[2016-11-16] 従来は `\paragraph` の最初に出るマークを「■」に固定していましたが、こ

のマークを変更可能にするため `\jsParagraphMark` というマクロに切り出しました。これで、たとえば

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

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

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

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

```
2120 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2121 \let\bxjs@org@paragraph@mark\jsParagraphMark
2122 \ifx\bxjs@paragraph@mark@empty
2123   \let\jsParagraphMark@empty
2124 \else\ifx\bxjs@paragraph@mark@undefined\else
2125   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2126 \fi\fi
2127 \let\jsJaChar@empty
2128 \if@twocolumn
2129   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2130     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2131     {\normalfont\normalsize\headfont}}
2132 %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
2133 \else
2134   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2135     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2136     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2137     {\normalfont\normalsize\headfont}}
2138 %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
2139 \fi
```

`\subparagraph` 見出しの後ろで改行されません。

```
2140 \if@twocolumn
2141   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2142     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2143     {\normalfont\normalsize\headfont}}
2144 \else
2145   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2146     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2147     {\normalfont\normalsize\headfont}}
2148 \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。

```
2149 \if@slide
2150   \setlength\leftmargini{1\jsZw}
2151 \else
2152   \if@twocolumn
2153     \setlength\leftmargini{2\jsZw}
2154   \else
2155     \setlength\leftmargini{3\jsZw}
2156   \fi
2157 \fi
```

`\leftmarginii` `ii`, `iii`, `iv` は `\labelsep` とそれぞれ ‘(m)’, ‘vii.’, ‘M.’ の幅との和より大きくすること
`\leftmarginiii` になっています。ここでは全角幅の整数倍に丸めました。

```
\leftmarginiv 2158 \if@slide
\leftmarginv 2159   \setlength\leftmarginii {1\jsZw}
2160   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2161   \setlength\leftmarginiv {1\jsZw}
2162   \setlength\leftmarginv {1\jsZw}
2163   \setlength\leftmarginvi {1\jsZw}
2164 \else
2165   \setlength\leftmarginii {2\jsZw}
2166   \setlength\leftmarginiii{2\jsZw}
2167   \setlength\leftmarginiv {2\jsZw}
2168   \setlength\leftmarginv {1\jsZw}
2169   \setlength\leftmarginvi {1\jsZw}
2170 \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。

```
2171 \setlength \labelsep {0.5\jsZw} % .5em
2172 \setlength \labelwidth{\leftmargini}
2173 \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。

```
2174 \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`

`\@itempenalty`

```

2175 \@beginparpenalty -\@lowpenalty
2176 \@endparpenalty -\@lowpenalty
2177 \@itempenalty -\@lowpenalty

```

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

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

```

2178 \def\@listi{\leftmargin\leftmarginI
2179   \parsep \z@
2180   \topsep 0.5\baselineskip
2181   \itemsep \z@ \relax}
2182 \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）。

```

2183 \@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 2184 \def\@listii{\leftmargin\leftmarginii
\@listiv 2185   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2186   \topsep \z@
\@listv 2187   \parsep \z@
\@listvi 2188   \itemsep\parsep}
2189 \def\@listiii{\leftmargin\leftmarginiii
2190   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2191   \topsep \z@
2192   \parsep \z@
2193   \itemsep\parsep}
2194 \def\@listiv {\leftmargin\leftmarginiv
2195   \labelwidth\leftmarginiv
2196   \advance\labelwidth-\labelsep}
2197 \def\@listv {\leftmargin\leftmarginv
2198   \labelwidth\leftmarginv
2199   \advance\labelwidth-\labelsep}
2200 \def\@listvi {\leftmargin\leftmarginvi
2201   \labelwidth\leftmarginvi
2202   \advance\labelwidth-\labelsep}

```

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

`\theenumi` 出力する番号の書式を設定します。これらは L^AT_EX 本体 (`ltlists.dtx` 参照) で定義済みですが、ここでは表し方を変えています。`\@arabic`, `\@alph`, `\@roman`, `\@Alph` はそれぞれ `\theenumiii` `\theenumiv`

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。

```
2203 \renewcommand{\theenumi}{\@arabic\c@enumi}
2204 \renewcommand{\theenumii}{\@alph\c@enumii}
2205 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2206 \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
`\labelenumii` 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。

```
2207 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2208 \newcommand*{\jsInJaParen}[1]{%
2209   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2210 \newcommand{\labelenumi}{\theenumi.}
2211 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
2212 \newcommand{\labelenumiii}{\theenumiii.}
2213 \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
`\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 2214 \renewcommand{\p@enumii}{\theenumi}
2215 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
2216 \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 2217 \newcommand\labelitemi{\textbullet}
\labelitemiii 2218 \newcommand\labelitemii{\normalfont\bfseries \textendash}
2219 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2220 \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。

```
2221 \newenvironment{description}{%
2222   \list{}{%
```



```

2223 \labelwidth=\leftmargin
2224 \labelsep=1\jsZw
2225 \advance \labelwidth by -\labelsep
2226 \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。

```

2227 \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 (+titlepage)` 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。

```

2228 %<*book|report>
2229 \newenvironment{chapterabstract}{%
2230 \begin{list}{%
2231 \listparindent=1\jsZw
2232 \itemindent=\listparindent
2233 \rightmargin=\z@
2234 \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2235 %</book|report>

```

“普通の” `abstract` 環境の定義。

```

2236 %<*article|report|slide>
2237 \newbox\@abstractbox
2238 \if@titlepage
2239 \newenvironment{abstract}{%
2240 \titlepage
2241 \null\vfil
2242 \@beginparpenalty\@lowpenalty
2243 \begin{center}%
2244 \headfont \abstractname
2245 \@endparpenalty\@M
2246 \end{center}%

```

BXJS クラスでは、概要の最初の段落に段落下げが入るようにする。

```

2247 \par}%
2248 {\par\vfil\null\endtitlepage}
2249 \else
2250 \newenvironment{abstract}{%
2251 \if@twocolumn
2252 \ifx\maketitle\relax
2253 \section*{\abstractname}%
2254 \else
2255 \global\setbox\@abstractbox\hbox\bgroup
2256 \begin{minipage}[b]{\textwidth}
2257 \small\parindent1\jsZw
2258 \begin{center}%
2259 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2260 \end{center}%
2261 \list{}{%
2262 \listparindent\parindent
2263 \itemindent \listparindent
2264 \rightmargin \leftmargin}%
2265 \item\relax
2266 \fi
2267 \else
2268 \small
2269 \begin{center}%
2270 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2271 \end{center}%
2272 \list{}{%
2273 \listparindent\parindent
2274 \itemindent \listparindent
2275 \rightmargin \leftmargin}%
2276 \item\relax
2277 \fi}{\if@twocolumn
2278 \ifx\maketitle\relax
2279 \else
2280 \endlist\end{minipage}\egroup
2281 \fi
2282 \else
2283 \endlist
2284 \fi}
2285 \fi
2286 %</article|report|slide>
2287 %<*jspf>
2288 \newbox\@abstractbox
2289 \newenvironment{abstract}{%
2290 \global\setbox\@abstractbox\hbox\bgroup
2291 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2292 \small
2293 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%

```

```

2294 {\end{minipage}\egroup}
2295 %</jspf>

```

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

```

2296 %<*book|report>
2297 \ifbxjs@force@chapterabstract
2298 \let\abstract\chapterabstract
2299 \let\endabstract\endchapterabstract
2300 \fi
2301 %</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。

```

2302 %<*jspf>
2303 %\newbox\@keywordsbox
2304 %\newenvironment{keywords}{%
2305 % \global\setbox\@keywordsbox\hbox\bgroup
2306 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2307 % \small\parindent0\jsZw}%
2308 % {\end{minipage}\egroup}
2309 %</jspf>

```

■verse 環境

`verse` 詩のための `verse` 環境です。

```

2310 \newenvironment{verse}{%
2311 \let \=\@centercr
2312 \list{}{%
2313 \itemsep \z@
2314 \itemindent -2\jsZw % 元: -1.5em
2315 \listparindent\itemindent
2316 \rightmargin \z@
2317 \advance\leftmargin 2\jsZw}% 元: 1.5em
2318 \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。

```

2319 \newenvironment{quotation}{%
2320 \list{}{%
2321 \listparindent\parindent
2322 \itemindent\listparindent
2323 \rightmargin \z@}%
2324 \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。

```
2325 \newenvironment{quote}%  
2326 {\list{}{\rightmargin\z0}\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。

```
2327 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw  
2328 \item[\hskip \labelsep{\headfont #1\ #2}]}  
2329 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw  
2330 \item[\hskip \labelsep{\headfont #1\ #2 (#3)}]}
```

`titlepage` タイトルを独立のページに出力するのに使われます。

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

```
2331 \newenvironment{titlepage}{%  
2332 %<book> \pltx@cleartooddpage %% 2017-02-24  
2333 \if@twocolumn  
2334 \@restonecoltrue\onecolumn  
2335 \else  
2336 \@restonecolfalse\newpage  
2337 \fi  
2338 \thispagestyle{empty}%  
2339 \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-  
24  
2340 }%  
2341 {\if@restonecol\twocolumn \else \newpage \fi  
2342 \if@twoside\else  
2343 \setcounter{page}\@ne  
2344 \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2345 %<*!book&!report>
2346 \newcommand{\appendix}{\par
2347   \setcounter{section}{0}%
2348   \setcounter{subsection}{0}%
2349   \gdef\presectionname{\appendixname}%
2350   \gdef\postsectionname{}%
2351   % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2352   \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2353   \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
2354 %</!book&!report>
2355 %<*book|report>
2356 \newcommand{\appendix}{\par
2357   \setcounter{chapter}{0}%
2358   \setcounter{section}{0}%
2359   \gdef\@chapapp{\appendixname}%
2360   \gdef\@chappos{}%
2361   \gdef\thechapter{\@Alph\c@chapter}}
2362 %</book|report>
```

8.4 パラメータの設定

■array と tabular 環境

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

```
2363 \setlength\arraycolsep{5\p@?}
```

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

```
2364 \setlength\tabcolsep{6\p@?}
```

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

```
2365 \setlength\arrayrulewidth{.4\p@}
```

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

```
2366 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2367 \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。

```
2368 \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。

```
2369 \setlength\fboxsep{3\p@?}
2370 \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。

```
2371 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2372 %<*book|report>
2373 \@addtoreset{equation}{chapter}
2374 \renewcommand\theequation
2375   {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2376 %</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。

```
2377 % \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。

```
2378 % \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。

```
2379 % \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。

```
2380 %<!*book&!report>
2381 \newcounter{figure}
2382 \renewcommand \thefigure {\@arabic\c@figure}
2383 %</!*book&!report>
2384 %<*book|report>
2385 \newcounter{figure}[chapter]
2386 \renewcommand \thefigure
2387     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2388 %</book|report>
```

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

```
\ext@figure 2389 \def\fps@figure{tbp}
2390 \def\ftype@figure{1}
\fnun@figure 2391 \def\ext@figure{lof}
2392 \def\fnun@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2393 \newenvironment{figure}%
2394         {\@float{figure}}%
2395         {\end@float}
2396 \newenvironment{figure*}%
2397         {\@dblfloat{figure}}%
2398         {\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。

```
2399 %<!*book&!report>
2400 \newcounter{table}
2401 \renewcommand\thetable{\@arabic\c@table}
2402 %</!*book&!report>
2403 %<*book|report>
2404 \newcounter{table}[chapter]
2405 \renewcommand \thetable
2406     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2407 %</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 2408 \def\fps@table{tbp}
2409 \def\ftype@table{2}
\fnun@table 2410 \def\ext@table{lot}
2411 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` * は段抜きのフロートです。

`table*`

```

2412 \newenvironment{table}%
2413         {\@float{table}}%
2414         {\end@float}
2415 \newenvironment{table*}%
2416         {\@dblfloat{table}}%
2417         {\end@dblfloat}

```

8.6 キャプション

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

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

```

2418 \newlength\abovecaptionskip
2419 \newlength\belowcaptionskip
2420 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2421 \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。

```

2422 %<!*jspf>
2423 % \long\def\@makecaption#1#2{{\small
2424 % \advance\leftskip 10\jsc@mmm
2425 % \advance\rightskip 10\jsc@mmm
2426 % \vskip\abovecaptionskip
2427 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2428 % \ifdim \wd\@tempboxa >\hsize
2429 % #1\hskip1\jsZw\relax #2\par
2430 % \else
2431 % \global \minipagefalse
2432 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2433 % \fi
2434 % \vskip\belowcaptionskip}}
2435 \long\def\@makecaption#1#2{{\small
2436 \advance\leftskip .0628\linewidth
2437 \advance\rightskip .0628\linewidth
2438 \vskip\abovecaptionskip
2439 \sbox\@tempboxa{#1\zwspace#2}%
2440 \ifdim \wd\@tempboxa <\hsize \centering \fi
2441 #1\zwspace#2\par
2442 \vskip\belowcaptionskip}}
2443 %</!*jspf>
2444 %<!*jspf>
2445 \long\def\@makecaption#1#2{%

```



```

2446 \vskip\abovecaptionskip
2447 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2448 \ifdim \wd\@tempboxa >\hsize
2449   {\small\sffamily
2450     \list{#1}{%
2451       \renewcommand{\makelabel}[1]{##1\hfil}
2452       \itemsep \z@
2453       \itemindent \z@
2454       \labelsep \z@
2455       \labelwidth 11\jsc@mmm
2456       \listparindent\z@
2457       \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2458 \else
2459   \global \@minipagefalse
2460   \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2461 \fi
2462 \vskip\belowcaptionskip}
2463 %</jspf>

```

9 フォントコマンド

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

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

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

```

\if@jsc@warnoldfontcmd


---


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


---


2464 \newif\if@jsc@warnoldfontcmd
2465 \@jsc@warnoldfontcmdtrue
2466 \newif\if@jsc@warnoldfontcmdexception
2467 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand
2468 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2469   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2470   \DeclareOldFontCommand{#1}{%
2471     \bxjs@oldfontcmd{#1}#2%
2472   }{%

```

```

2473     \bxjs@oldfontcmd{#1}#3%
2474 }%
2475 }
2476 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2477   \ClassInfo\bxjs@clsname
2478   {Old font command '\string#1' is used!!\MessageBreak
2479     The first occurrence is}%
2480 }

```

`\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。

```

2481 \newcommand*{\allowoldfontcommands}{%
2482   \@jsc@warnoldfontcmdexceptiontrue}
2483 \newcommand*{\disallowoldfontcommands}{%
2484   \@jsc@warnoldfontcmdexceptionfalse}

```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2485 \let\bxjs@oldfontcmd@list\@empty
2486 \def\bxjs@oldfontcmd#1{%
2487   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2488 \def\bxjs@oldfontcmd@a#1#2{%
2489   \if@jsc@warnoldfontcmdexception\else
2490     \global\@jsc@warnoldfontcmdfalse
2491     \ifx#1\relax
2492       \global\let#1=t%
2493       \jsc@warnoldfontcmd{#2}%
2494     \fi
2495   \fi}
2496 \def\bxjs@warnoldfontcmd@final{%
2497 % \par
2498 \global\let\bxjs@warnoldfontcmd@final\@empty
2499 \let\@tempa\@empty
2500 \def\do##1{%
2501   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2502     \edef\@tempa{\@tempa \space\string##1}\fi}
2503 \bxjs@oldfontcmd@list
2504 \ifx\@tempa\@empty\else
2505   \ClassWarningNoLine\bxjs@clsname
2506     {Some old font commands were used in text\MessageBreak
2507       (see the log file for detail)}%
2508   \ClassInfo\bxjs@clsname
2509     {Some old font commands were used in text:\MessageBreak
2510       \space\@tempa\MessageBreak
2511       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2512       new font selection scheme called NFSS2 with several\MessageBreak
2513       new, combinable font commands. The

```

```

2514     class provides\MessageBreak
2515     the old font commands
2516     only for compatibility%
2517     \@gobble}%
2518 \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。

```

2519 \def\bxjs@warnoldfontcmd@kick@final{%
2520   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2521 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2522 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2523 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2524 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2525 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2526 \jsc@DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

```

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

```

2527 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2528 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2529 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2530 \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 2531 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2532 \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。

```
2533 \newcommand\@pnumwidth{1.55em}
2534 \newcommand\@tocrmarg{2.55em}
2535 \newcommand\@dotsep{4.5}
2536 %<!book&!report>\setcounter{tocdepth}{2}
2537 %<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)

```
2538 \newdimen\jsc@tocl@width
2539 \newcommand{\tableofcontents}{%
2540 %<*book|report>
2541 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2542 \settowidth\@tempdima{\headfont\appendixname}%
2543 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2544 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
```

```

2545 \if@twocolumn
2546 \@restonecoltrue\onecolumn
2547 \else
2548 \@restonecolfalse
2549 \fi
2550 \chapter*{\contentsname}%
2551 \mkboth{\contentsname}{}%
2552 %</book|report>
2553 %<*&book&!report>
2554 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2555 \settowidth\@tempdima{\headfont\appendixname}%
2556 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2557 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2558 \section*{\contentsname}%
2559 \mkboth{\contentsname}{\contentsname}%
2560 %</!book&!report>
2561 \@starttoc{toc}%
2562 %<book|report> \if@restonecol\twocolumn\fi
2563 }

```

\l@part 部の目次です。

```

2564 \newcommand*{\l@part}[2]{%
2565 \ifnum \c@tocdepth >-2\relax
2566 %<!book&!report> \addpenalty\@secpenalty
2567 %<book|report> \addpenalty{-\@highpenalty}%
2568 \addvspace{2.25em \@plus\p@}%
2569 \begingroup
2570 \parindent \z@
2571 % \@pnumwidth should be \@tocrmarg
2572 % \rightskip \@pnumwidth
2573 \rightskip \@tocrmarg
2574 \parfillskip -\rightskip
2575 {\leavevmode
2576 \large \headfont
2577 \setlength\@lnumwidth{4\jsZw}%
2578 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2579 \nobreak
2580 %<book|report> \global\@nobreaktrue
2581 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2582 \endgroup
2583 \fi}

```

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

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

```

2584 %<*&book|report>
2585 \newcommand*{\l@chapter}[2]{%
2586 \ifnum \c@tocdepth >\m@ne
2587 \addpenalty{-\@highpenalty}%

```

```

2588      \addvspace{1.0em \@plus\p@?}
2589 %    \vskip 1.0em \@plus\p@    % book.cls では↑がこうなっている
2590      \begingroup
2591          \parindent\z@
2592 %    \rightskip\@pnumwidth
2593          \rightskip\@tocrmarg
2594          \parfillskip-\rightskip
2595          \leavevmode\headfont
2596 %    % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2597          \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2598          \advance\leftskip\@lnumwidth \hskip-\leftskip
2599          #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2600          \penalty\@highpenalty
2601      \endgroup
2602  \fi}
2603 %</book|report>

```

\l@section 節の目次です。

```

2604 %<*!book&!report>
2605 \newcommand*{\l@section}[2]{%
2606     \ifnum \c@tocdepth >\z@
2607         \addpenalty{\@secpenalty}%
2608         \addvspace{1.0em \@plus\p@?}%
2609         \begingroup
2610             \parindent\z@
2611 %         \rightskip\@pnumwidth
2612             \rightskip\@tocrmarg
2613             \parfillskip-\rightskip
2614             \leavevmode\headfont
2615 %         % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2616             \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2617             \advance\leftskip\@lnumwidth \hskip-\leftskip
2618             #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2619         \endgroup
2620     \fi}
2621 %</!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。

```

2622 %<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 2623 %<*!book&!report>
2624 % \newcommand*{\l@subsection}{\@dottedtocline{2}{1.5em}{2.3em}}
2625 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2626 % \newcommand*{\l@paragraph}{\@dottedtocline{4}{7.0em}{4.1em}}

```

```

2627 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2628 %
2629 % \newcommand*{\l@section} {\@dottedtocline{2}{1zw}{3zw}}
2630 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2631 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2632 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2633 %
2634 \newcommand*{\l@subsection}{%
2635     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2636     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2637 \newcommand*{\l@subsubsection}{%
2638     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2639     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2640 \newcommand*{\l@paragraph}{%
2641     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2642     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2643 \newcommand*{\l@subparagraph}{%
2644     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2645     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2646 %<!/book&!report>
2647 %<*book|report>
2648 % \newcommand*{\l@subsection} {\@dottedtocline{2}{3.8em}{3.2em}}
2649 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2650 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2651 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2652 \newcommand*{\l@section}{%
2653     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2654     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2655 \newcommand*{\l@subsection}{%
2656     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2657     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2658 \newcommand*{\l@subsubsection}{%
2659     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2660     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2661 \newcommand*{\l@paragraph}{%
2662     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2663     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2664 \newcommand*{\l@subparagraph}{%
2665     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2666     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2667 %</book|report>

```

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

```

2668 \newdimen\@lnumwidth
2669 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

`\@dottedtocline` L^AT_EX 本体 (ltsect.dtx 参照) での定義と同じですが, `\@tempdima` を `\@lnumwidth` に変えています。

```

2670 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2671   \vskip \z@ \@plus.2\p@?
2672   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2673    \parindent #2\relax\@afterindenttrue
2674    \interlinepenalty\@M
2675    \leavevmode
2676    \@lnumwidth #3\relax
2677    \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2678    {#4}\nobreak
2679    \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2680     mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2681     \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2682 \newcommand{\listoffigures}{%
2683 %<*book|report>
2684 \if@twocolumn\@restonecoltrue\onecolumn
2685 \else\@restonecolfalse\fi
2686 \chapter*{\listfigurename}%
2687 \@mkboth{\listfigurename}{}%
2688 %</book|report>
2689 %<*&!book&!report>
2690 \section*{\listfigurename}%
2691 \@mkboth{\listfigurename}{\listfigurename}%
2692 %</&!book&!report>
2693 \@starttoc{lof}%
2694 %<book|report> \if@restonecol\twocolumn\fi
2695 }

```

`\l@figure` 図目次の項目を出力します。

```

2696 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2697 \newcommand{\listoftables}{%
2698 %<*book|report>
2699 \if@twocolumn\@restonecoltrue\onecolumn
2700 \else\@restonecolfalse\fi
2701 \chapter*{\listtablename}%
2702 \@mkboth{\listtablename}{}%
2703 %</book|report>
2704 %<*&!book&!report>
2705 \section*{\listtablename}%
2706 \@mkboth{\listtablename}{\listtablename}%
2707 %</&!book&!report>

```



```

2708 \starttoc{lot}%
2709 %<book|report> \if@restonecol\twocolumn\fi
2710 }

```

`\l@table` 表目次は図目次と同じです。

```

2711 \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。

```

2712 \newdimen\bibindent
2713 \setlength\bibindent{2\jsZw}

```

`thebibliography` 参考文献リストを出力します。

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

```

2714 \newenvironment{thebibliography}[1]{%
2715 \jsc@warnoldfontcmdexceptiontrue
2716 \global\let\presectionname\relax
2717 \global\let\postsectionname\relax
2718 %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2719 %<*kiyou>
2720 \vspace{1.5\baselineskip}
2721 \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2722 \vspace{0.5\baselineskip}
2723 %</kiyou>
2724 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2725 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2726 \list{\@biblabel{\@arabic\c@enumiv}}%
2727 {\settowidth\labelwidth{\@biblabel{#1}}%
2728 \leftmargin\labelwidth
2729 \advance\leftmargin\labelsep
2730 \@openbib@code
2731 \usecounter{enumiv}%
2732 \let\p@enumiv\empty
2733 \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2734 %<kiyou> \small
2735 \sloppy
2736 \clubpenalty4000
2737 \@clubpenalty\clubpenalty
2738 \widowpenalty4000%
2739 \sfcode`.\@m}
2740 {\def\@noitemerr
2741 {\@latex@warning{Empty `thebibliography' environment}}}%
2742 \endlist}

```

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

```
2743 \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。

```
2744 \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。

```
2745 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

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

```
2746 % \def\@citex[#1]#2{\leavevmode
2747 %   \let\@citea\@empty
2748 %   \@cite{\@for\@citeb:=#2\do
2749 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2750 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2751 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2752 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2753 %       \G@refundefinedtrue
2754 %       \@latex@warning
2755 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2756 %       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2757 % \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。

```
2758 % \DeclareRobustCommand\cite{\unskip
2759 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2760 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2761 %   , \jsInhibitGlue\ #2\fi}}}$}
```

10.3 索引

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

```
2762 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2763   \if@twocolumn
2764     \onecolumn\@restonecolfalse
2765   \else
2766     \clearpage\@restonecoltrue
2767   \fi}
```

```

2768 \columnseprule.4pt \columnsep 2\jsZw
2769 \ifx\multicols\@undefined
2770 %<book|report> \twocolumn[\@makeschapterhead{\indexname}%
2771 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2772 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2773 %<!book&!report> \twocolumn[\section*{\indexname}]]%
2774 \else
2775 \ifdim\textwidth<\fullwidth
2776 \setlength{\evensidemargin}{\oddsidemargin}
2777 \setlength{\textwidth}{\fullwidth}
2778 \setlength{\linewidth}{\fullwidth}
2779 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}%
2780 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2781 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2782 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]]%
2783 \else
2784 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}%
2785 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2786 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2787 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]]%
2788 \fi
2789 \fi
2790 %<book|report> \mkboth{\indexname}{}%
2791 %<!book&!report> \mkboth{\indexname}{\indexname}%
2792 \plainifnotempty % \thispagestyle{plain}
2793 \parindent\z@
2794 \parskip\z@ \@plus .3\p@?\relax
2795 \let\item\@idxitem
2796 \raggedright
2797 \footnotesize\narrowbaselines
2798 }{
2799 \ifx\multicols\@undefined
2800 \if@restonecol\onecolumn\fi
2801 \else
2802 \end{multicols}
2803 \fi
2804 \clearpage
2805 }

```

\@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 2806 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2807 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2808 \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。

```
2809 \newcommand{\indexspace}{\par \vskip 10\p@? \@plus5\p@? \@minus3\p@?\relax}
```

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

\alsoname という英語ですが、ここではとりあえず両方とも「→」に変えました。⇒ (\rightarrow)

などでもいいでしょう。

```
2810 \newcommand\seename{\if@english see\else →\fi}  
2811 \newcommand\alsoname{\if@english see also\else →\fi}
```

10.4 脚注

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

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

```
2812 \ifx\pltx@foot@penalty\@undefined  
2813 \let\footnotes@ve=\footnote  
2814 \def\footnote{\jsInhibitGlue\footnotes@ve}  
2815 \let\footnotemarks@ve=\footnotemark  
2816 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}  
2817 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

[2016-07-11] コミュニティ版 p_{La}T_EX の変更に追随しました (Thanks: 角藤さん)。p_{La}T_EX の日付が 2016/04/17 より新しい場合は、このパッチが不要なのであてません。

p_TE_X 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

[2016-10-08] TODO: 脚注番号が `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)。

```
2818 \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。

```
2819 % \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。

```
2820 \renewcommand{\footnoterule}{%
2821   \kern-2.6\p@? \kern-.4\p@
2822   \hrule width .4\columnwidth
2823   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2824 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

```
2825 \long\def\@footnotetext{%
2826   \insert\footins\bgroup
2827     \normalfont\footnotesize
2828     \interlinepenalty\interfootnotelinepenalty
2829     \splittopskip\footnotesep
2830     \splitmaxdepth \dp\strutbox \floatingpenalty \MM
2831     \hsize\columnwidth \@parboxrestore
2832     \protected@edef\@currentlabel{%
2833       \csname p@footnote\endcsname\@thefnmark
2834     }%
2835     \color@begingroup
2836       \@makefnmark{%
2837         \rule{\z@\footnotesep}{\ignorespaces}%
2838         \futurelet\next\fo@t}
2839 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@@
2840                                     \else \let\next\fo@t\fi \next}
2841 \def\fo@@{\bgroup\aftergroup\@foot\let\next}
2842 \def\fo@t#1{#1\@foot}
2843 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2844   \ifx\pltx@foot@penalty\undefined\else
2845     \ifhmode\null\fi
2846     \ifnum\pltx@foot@penalty=\z@\else
2847       \penalty\pltx@foot@penalty
2848       \pltx@foot@penalty\z@
2849     \fi
2850   \fi}
```

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

```

2851 \newcommand\@makefntext[1]{%
2852   \advance\leftskip 3\jsZw
2853   \parindent 1\jsZw
2854   \noindent
2855   \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。

```

2856 % \def\@xfootnotenext[#1]{%
2857 %   \begingroup
2858 %     \ifnum#1>\z@
2859 %       \csname c@\@mpfn\endcsname #1\relax
2860 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2861 %     \else
2862 %       \unrestored@protected@xdef\@thefnmark{}%
2863 %     \fi
2864 %   \endgroup
2865 %   \@footnotetext}

```

ここまでのコードは JS クラスを踏襲する。

11 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

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

```

2866 \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。

```

2867 \def\everyparhook{\jsInhibitGlueAtParTop}
2868 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2869 \AtBeginDocument{\everypar{\everyparhook}}
2870 \fi

```

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

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

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

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

```
2871 \ifx j\jsEngine
2872 \def\@inhibitglue{%
2873   \futurelet\@let@token\@inhibitglue}
2874 \begingroup
2875 \let\GDEF=\gdef
2876 \let\CATCODE=\catcode
2877 \let\ENDGROUP=\endgroup
2878 \CATCODE`k=12
2879 \CATCODE`a=12
2880 \CATCODE`n=12
2881 \CATCODE`j=12
2882 \CATCODE`i=12
2883 \CATCODE`c=12
2884 \CATCODE`h=12
2885 \CATCODE`r=12
2886 \CATCODE`t=12
2887 \CATCODE`e=12
2888 \GDEF\KANJI@CHARACTER{kanji character }
2889 \ENDGROUP
2890 \def\@inhibitglue{%
2891   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
2892   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
2893   \def\jsc@ig@temp{#1}%
2894   \ifx\jsc@ig@temp\empty
2895     \ifnum\the\inhibitxspcode`#2=2\relax
2896       \inhibitglue
2897     \fi
2898   \fi}
2899 \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。

```
2900 \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。

```
2901 \def\@doendpe{%
2902   \@endpetrue
2903   \def\par{%
2904     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2905   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] minipage 環境にも対策します。

```
2906 \def\@setminipage{%
2907   \@minipagetrue
2908   \everypar{\@minipagefalse\everypar{\everyparhook}}%
2909 }
```

`\item` 命令の直後です。

```
2910 \def\@item[#1]{%
2911   \if@noperitem
2912     \@donoperitem
2913   \else
2914     \if@inlabel
2915       \indent \par
2916     \fi
2917     \ifhmode
2918       \unskip\unskip \par
2919     \fi
2920     \if@newlist
2921       \if@nobreak
2922         \@nbitem
2923       \else
2924         \addpenalty\@beginparpenalty
2925         \addvspace\@topsep
2926         \addvspace{-\parskip}%
2927       \fi
2928     \else
2929       \addpenalty\@itempenalty
2930       \addvspace\itemsep
2931     \fi
2932     \global\@inlabeltrue
2933   \fi
2934   \everypar{%
2935     \@minipagefalse
2936     \global\@newlistfalse
2937     \if@inlabel
2938       \global\@inlabelfalse
2939     {\setbox\z@\lastbox
2940      \ifvoid\z@
2941        \kern-\itemindent
2942      \fi}%
```



```

2943     \box\@labels
2944     \penalty\z@
2945     \fi
2946     \if@nobreak
2947         \@nobreakfalse
2948         \clubpenalty \@M
2949     \else
2950         \clubpenalty \@clubpenalty
2951         \everypar{\everyparhook}%
2952     \fi
2953     \everyparhook}%
2954     \if@noitemarg
2955         \@noitemargfalse
2956     \if@nmbrlist
2957         \refstepcounter\@listctr
2958     \fi
2959     \fi
2960     \sbox\@tempboxa{\makelabel{#1}}}%
2961     \global\setbox\@labels\hbox{%
2962         \unhbox\@labels
2963         \hskip \itemindent
2964         \hskip -\labelwidth
2965         \hskip -\labelsep
2966         \ifdim \wd\@tempboxa >\labelwidth
2967             \box\@tempboxa
2968         \else
2969             \hbox to\labelwidth {\unhbox\@tempboxa}%
2970         \fi
2971         \hskip \labelsep}%
2972     \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。

```

2973 \def\@afterheading{%
2974     \@nobreaktrue
2975     \everypar{%
2976         \if@nobreak
2977             \@nobreakfalse
2978             \clubpenalty \@M
2979             \if@afterindent \else
2980                 {\setbox\z@\lastbox}%
2981             \fi
2982         \else
2983             \clubpenalty \@clubpenalty
2984             \everypar{\everyparhook}%
2985         \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

2986 \fi

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

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

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

```
2987 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
2988 \AtEndOfPackage{%
2989 \def\@gnewline #1{%
2990   \ifvmode
2991     \@nolnerr
2992   \else
2993     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
2994     \jsInhibitGlue \ignorespaces
2995   \fi}
2996 }
2997 \fi
```

12 いろいろなロゴ

`IATEX` 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。

```
2998 \if@jslogo
2999   \IfFileExists{jslogo.sty}{%
3000     \RequirePackage{jslogo}%
3001   }{%
3002     \ClassWarningNoLine\bxjs@clsname
3003       {The package 'jslogo' is not installed.\MessageBreak
3004        It is included in the recent release of\MessageBreak
3005        the 'jsclasses' bundle}
3006   }
```

13 amsmath との衝突の回避

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

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

[2016-11-19] 本家の 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。

```

3008 %\let\ltx@ifnextchar\@ifnextchar
3009 %\def\ProvidesFile#1{%
3010 %   \begingroup
3011 %     \catcode\ 10 %
3012 %     \ifnum \endlinechar<256 %
3013 %       \ifnum \endlinechar>\m@ne
3014 %         \catcode\endlinechar 10 %
3015 %       \fi
3016 %     \fi
3017 %     \@makeother\/%
3018 %     \@makeother\&%
3019 %     \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 3020 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3021 \newcommand{\postpartname}{\if@english\else 部\fi}
3022 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3023 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3024 \newcommand{\presectionname}{}% 第
\postsectionname 3025 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 3026 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3027 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3028 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3029 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3030 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}

```

```

3031 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3032 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3033 %<jspf>\newcommand{\figurename}{Fig.~}
3034 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3035 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3036 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3037 \newcommand{\appendixname}{\if@english \else 付録\fi}
3038 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

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

```

3039 \@tempwafalse
3040 \if p\jsEngine \@tempwattrue \fi
3041 \if n\jsEngine \@tempwattrue \fi
3042 \if@tempswa \expandafter\@firstoftwo
3043 \else \expandafter\@secondoftwo
3044 \fi
3045 {%
3046 % 欧文 8bitTeX の場合
3047 \newif\ifjsSeireki \jsSeirekitrue
3048 \def\bxjs@decl@Seireki@cmds{%
3049 \def\西暦{\jsSeirekitrue}%
3050 \def\和暦{\jsSeirekifalse}}
3051 \def\Seireki{\jsSeirekitrue}
3052 \def\Wareki{\jsSeirekifalse}
3053 \def\bxjs@if@use@seireki{%
3054 \ifjsSeireki \expandafter\@firstoftwo
3055 \else \expandafter\@secondoftwo \fi}
3056 }{%
3057 \newif\if 西暦 \西暦 true
3058 \def\bxjs@decl@Seireki@cmds{%
3059 \def\西暦{\西暦 true}%
3060 \def\和暦{\西暦 false}}
3061 \def\Seireki{\西暦 true}
3062 \def\Wareki{\西暦 false}
3063 \def\bxjs@if@use@seireki{%
3064 \if 西暦 \expandafter\@firstoftwo

```

```

3065 \else \expandafter\@secondoftwo \fi}
3066 }
3067 \bxjs@decl@Seireki@cmds
3068 \let\bxjs@unxp@firstofone \let\bxjs@onxp@firstofone
3069 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded
3070 \def\bxjs@onxp{\unexpanded\expandafter}}
3071 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3072 \def\bxjs@iai{\noexpand~}
3073 \else \def\bxjs@iai{}
3074 \fi

```

`\jyear` 和暦における年の表記（元号 + 年数）。

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

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

```

3075 \newcount\bxjs@jyear
3076 \let\bxjs@gengo@empty \bxjs@jyear\year
3077 \tempcnta=\the\year\two@digits\month\two@digits\day\relax
3078 \ifnum\tempcnta<19700101
3079 \else\ifnum\tempcnta<19890108
3080 \def\bxjs@gengo{昭和}\advance\bxjs@jyear-1925\relax
3081 \let\heisei\bxjs@jyear
3082 \else\ifnum\tempcnta<20190501
3083 \def\bxjs@gengo{平成}\advance\bxjs@jyear-1988\relax
3084 \let\heisei\bxjs@jyear
3085 \else
3086 % 新元号対応
3087 % \def\bxjs@gengo{新元号}\advance\bxjs@jyear-2018\relax
3088 \fi\fi\fi
3089 \edef\jyear{%
3090 \ifx\bxjs@gengo@empty \the\bxjs@jyear\bxjs@iai
3091 \else \bxjs@onxp{\bxjs@gengo}%
3092 \ifnum\bxjs@jyear=\@ne \bxjs@unxp{元}%
3093 \else \bxjs@iai\the\bxjs@jyear\bxjs@iai
3094 \fi
3095 \fi}

```

`\today` 英語、西暦、和暦で場合分けをする。

```

3096 \edef\bxjs@today{%
3097 \if@english
3098 \ifcase\month\or
3099 January\or February\or March\or April\or May\or June\or
3100 July\or August\or September\or October\or November\or December\fi
3101 \space\number\day, \number\year
3102 \else
3103 \noexpand\bxjs@if@use@seireki{%
3104 \number\year\bxjs@iai\bxjs@unxp{年}%
3105 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3106 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%

```

```

3107 }{%
3108   \bxjs@onxp{\jayear}\bxjs@unxp{年}%
3109   \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3110   \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3111 }%
3112 \fi}
3113 \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。

```

3114 \AtBeginDocument{%
3115   \ifx\bbj@jpn@Seirekitrue\@undefined\else
3116     \bxjs@decl@Seireki@cmds
3117     \g@addto@macro\datejapanese{%
3118       \let\today\bxjs@today}%
3119   \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)

```

3120 \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。

```

3121 %<slide>\pagestyle{empty}%
3122 %<article|report>\pagestyle{plain}%
3123 %<book>\pagestyle{headings}%
3124 \pagenumbering{arabic}
3125 \if@twocolumn
3126   \twocolumn
3127   \sloppy
3128   \flushbottom
3129 \else
3130   \onecolumn
3131   \raggedbottom
3132 \fi
3133 %<*slide>
3134 \renewcommand\familydefault{\sfdefault}
3135 \raggedright
3136 %</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。

```

3137 \catcode`\?=12
3138 \ifx\bxjs@jadriver\relax\else

```

```
3139 \input{bxjsja-\bxjs@jadriver.def}
```

```
3140 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
3141 \bxjs@restore@jltrcc
```

```
3142 %</cls>
```

以上です。

付録 A 和文ドライバの仕様

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

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

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

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

以下で標準で用意されている和文ドライバの実装を示す。

```
3143 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

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

B.1 補助マクロ

```
3144 %<*minimal>
```

```
3145 %% 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.
```

```
3146 \def\DeclareJaTextFontCommand#1#2{%
```

```
3147   \DeclareRobustCommand#1[1]{%
```

```
3148     \relax
```

```
3149     \ifmmode \expandafter\nfss@text \fi
```

```
3150     {#2##1}}%
```

```
3151 }
```

```
\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。
```

```
3152 \def\DeclareJaMathFontCommand#1#2{%
```

```
3153   \DeclareRobustCommand#1[1]{%
```

```
3154     \relax
```

```
3155     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3156     \nfss@text{\fontfamily\familydefault
```

```
3157       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3158       #2##1}}%
```

```

3159 }%
3160 }

```

`\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。

```

3161 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3162 \@onlypreamble\bxjs@if@sf@default
3163 \def\bxjs@if@sf@default#1{%
3164   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3165   \AtBeginDocument{%
3166     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3167 }

```

`\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。

```

3168 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3169 \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。

```

3170 \def\jsLetHeadChar#1#2{%
3171   \begingroup
3172   \escapechar=`\ %
3173   \let\bxjs@tmpa={% brace-match-hack
3174   \bxjs@let@hchar@exp#2}%
3175   \endgroup
3176   \let#1\bxjs@g@tmpa}
3177 \def\bxjs@let@hchar@exp{%
3178   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3179 \def\bxjs@let@hchar@exp@a{%
3180   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3181   \bxjs@let@hchar@out\let\relax
3182   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3183   \bxjs@let@hchar@out\let\space%
3184   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3185   \bxjs@let@hchar@out\let\@backslashchar
3186   }{\bxjs@let@hchar@exp@b}}}%
3187 \def\bxjs@let@hchar@exp@b#1{%
3188   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3189 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3190   %\message{<#1#2>}%
3191   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3192 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3193 \bxjs@let@hchar@out\let\relax
3194 }{%else
3195 \expandafter\bxjs@let@hchar@exp
3196 }%
3197 }{%else
3198 \bxjs@let@hchar@chr#1%
3199 }}
3200 \def\bxjs@let@hchar@chr#1{%
3201 \bxjs@let@hchar@out\def{{#1}}}}
3202 \def\bxjs@let@hchar@out#1#2{%
3203 \global#1\bxjs@g@tmpa#2\relax
3204 \toks@{bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3205 \chardef\bxjs@let@hchar@csta=128
3206 \chardef\bxjs@let@hchar@cstb=192
3207 \chardef\bxjs@let@hchar@cstc=224
3208 \chardef\bxjs@let@hchar@cstd=240
3209 \chardef\bxjs@let@hchar@cste=248
3210 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3211 \def\bxjs@let@hchar@chr@ue#1{%
3212 \@tempcnta=#1\relax
3213 %\message{\the\@tempcnta}%
3214 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3215 \bxjs@let@hchar@chr@ue@a#1%
3216 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3217 \bxjs@let@hchar@out\let\relax
3218 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3219 \bxjs@let@hchar@chr@ue@b
3220 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3221 \bxjs@let@hchar@chr@ue@c
3222 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3223 \bxjs@let@hchar@chr@ue@d
3224 }{%else
3225 \bxjs@let@hchar@out\let\relax
3226 }}}}
3227 \def\bxjs@let@hchar@chr@ue@a#1{%
3228 \bxjs@let@hchar@out\def{{#1}}}}
3229 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3230 \bxjs@let@hchar@out\def{{#1#2}}}}
3231 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3232 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3233 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3234 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
3235 \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。

```
3236 \def\bxjs@let@hchar@chr@pp#1{%
3237   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
3238 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
3239   \%message{(#1)}%
3240   \bxjs@cond\if#1t\fi{%
3241     \bxjs@let@hchar@chr@ue#3%
3242   }{%else
3243     \bxjs@let@hchar@out\def{#3}}%
3244   }}
3245 \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。

```
3246 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3247 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3248 \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。

```
3249 \@onlypreamble\bxjs@declarefontshape
3250 \ifjsWithupTeX
3251 \def\bxjs@declarefontshape{%
3252   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3253   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3254   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3255   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3256 }
3257 \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。

```
3258 \else
3259 \def\bxjs@declarefontshape{%
3260   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3261   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3262   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3263 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3264 }
3265 \def\bxjs@sizereference{jis}
3266 \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。

```

3267 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
3268   \def\bxjs@y{#5}}
3269 \ifjsWithpTeXng \def\bxjs@y{10}%
3270 \else
3271 \expandafter\expandafter\expandafter\bxjs@tmpa
3272 \expandafter\string\the\jfont\relax
3273 \fi
3274 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3275               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3276   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
3277    \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。

```

3278 \begingroup
3279 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3280 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3281 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3282 % 幅が丁度 10pt なら補正は不要
3283 \ifdim\wd\z@=10pt
3284   \global\let\bxjs@scale\jsScale
3285 \else
3286 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3287   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3288   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3289   \bxjs@invscale\@tempdima\bxjs@tmpa
3290   \xdef\bxjs@scale{\strip@pt\@tempdima}
3291 \fi
3292 \endgroup
3293 %\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。

```

3294 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3295 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3296 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3297 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3298 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3299 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3300 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3301 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3302 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3303 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3304 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3305 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3306 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3307 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3308 \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。

```

3309 \DeclareRobustCommand\rmfamily
3310 {\not@math@alphabet\rmfamily\mathrm
3311 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3312 \DeclareRobustCommand\sffamily
3313 {\not@math@alphabet\sffamily\mathsf
3314 \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3315 \DeclareRobustCommand\ttfamily
3316 {\not@math@alphabet\ttfamily\mathtt
3317 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3318 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3319 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3320 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3321 \fi
3322 \bxjs@if@sf@default{%
3323 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3324 \selectfont

```

■パラメタの設定

```

3325 \prebreakpenalty\jis"2147=10000
3326 \postbreakpenalty\jis"2148=10000
3327 \prebreakpenalty\jis"2149=10000
3328 \inhibitxspcode`!=1
3329 \inhibitxspcode`〒=2
3330 \xspcode`+=3
3331 \xspcode`%=3

```

"80～"FF の範囲の \xspcode を 3 に変更。

```

3332 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3333 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3334 \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。

```
3335 \begingroup
3336 \catcode`\!=0
```

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

```
3337 \gdef\bxjs@ptex@dir{%
3338   !iftdir t%
3339   !else!ifydir y%
3340   !else ?%
3341   !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。

```
3342 % 古い \@makefnmark の定義
3343 \long\def\bxjs@tmpa{\hbox{%
3344   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3345   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3346 \ifx\@makefnmark\bxjs@tmpa
3347 \long\gdef\@makefnmark{%
3348   !ifydir \hbox{ }\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{ }%
3349   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3350 \fi
3351 \endgroup
```

B.3 pdfTeX 用の処理

```
3352 \else\ifx p\jsEngine
3353 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3354 \@onlypreamble\bxjs@cjk@loaded
3355 \def\bxjs@cjk@loaded{%
3356   \def\@footnotemark{%
3357     \leavevmode
3358     \ifhmode
3359       \edef\@x@sf{\the\spacefactor}%
3360       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3361         \unkern\unkern
3362         \ifdim\lastskip>\z@ \unskip \fi
3363       \fi\fi
3364       \nobreak
3365     \fi
3366     \@makefnmark
3367     \ifhmode \spacefactor\@x@sf \fi
3368     \relax}%
```

```

3369 \let\bxjs@CJK@loaded\relax
3370 }
3371 \AtBeginDocument{%
3372 \ifpackageloaded{CJK}{%
3373 \bxjs@CJK@loaded
3374 }{}%
3375 }

```

B.4 XeTeX 用の処理

```

3376 \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。

```

3377 \def\bxjs@let@hchar@chr#1{%
3378 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3379 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3380 \bxjs@let@hchar@chr@xe
3381 }{\bxjs@let@hchar@out\def{{#1}}}}
3382 \def\bxjs@let@hchar@chr@xe#1{%
3383 \lccode`0=`#1\relax
3384 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3385 \ifx\XeTeXgenerateactualtext\@undefined\else
3386 \def\bxjs@do@precisetext{%
3387 \XeTeXgenerateactualtext=\@one}
3388 \fi

```

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

```

3389 \@onlypreamble\bxjs@do@simplejasetup
3390 \def\bxjs@do@simplejasetup{%
3391 \ifnum\XeTeXinterchartokenstate>\z@
3392 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3393 \jsSimpleJaSetup
3394 \ClassInfo\bxjs@clsname
3395 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3396 \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

3397 \newcommand*{\jsSimpleJaSetup}{%
3398 \XeTeXlinebreaklocale "ja"\relax
3399 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3400 \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理（エンジン共通）

```

3401 \fi\fi\fi

```

simplejasetup オプションの処理。


```

3402 \ifx\bxjs@do@simplejasetup\@undefined\else
3403   \AtBeginDocument{%
3404     \ifbxjs@simplejasetup
3405       \bxjs@do@simplejasetup
3406     \fi}
3407 \fi

    precisetext オプションの処理。
3408 \ifbxjs@precisetext
3409   \ifx\bxjs@do@precisetext\@undefined
3410     \ClassWarning\bxjs@clsname
3411       {The current engine does not support the\MessageBreak
3412         'precisetext' option\@gobble}
3413   \else
3414     \bxjs@do@precisetext
3415   \fi
3416 \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。

```

3417 \@onlypreamble\bxjs@check@everyparhook
3418 \def\bxjs@check@everyparhook{%
3419   \ifx\jsInhibitGlueAtParTop\@empty
3420     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3421     \ifx\everyparhook\bxjs@tmpa
3422       \let\everyparhook\@empty
3423     \fi
3424   \fi}
3425 \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。

```

3426 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3427 \let\bxjs@everypar\everypar
3428 \newtoks\everypar
3429 \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。

```

3430 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3431 \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。

3432 \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。

3433 \@onlypreamble\bxjs@adjust@fancyhdr

3434 \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。

3435 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%

3436 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%

3437 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3438 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3439 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3440 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3441 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%

3442 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%

3443 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3444 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3445 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3446 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3447 \def\bxjs@tmpa{\rm\thepage\strut}%

3448 \def\bxjs@tmpb{\thepage\strut}%

3449 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi

3450 \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。

3451 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth

3452 \setlength{\@tempdima}{\fullwidth-\textwidth}%

3453 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%

3454 }\bxjs@tmpa

3455 \fi\fi

3456 \PackageInfo\bxjs@clsname

3457 {Patch to fancyhdr is applied\@gobble}}

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

3458 \def\bxjs@pagestyle@hook{%

3459 \@ifpackageloaded{fancyhdr}{%

3460 \bxjs@adjust@fancyhdr

3461 \global\let\bxjs@adjust@fancyhdr\relax

3462 }{}}

\pagestyle にフックを入れ込む。

3463 \let\bxjs@org@pagestyle\pagestyle

3464 \def\pagestyle{%

3465 \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。

```
3466 \AtBeginDocument{%
3467   \bxjs@pagestyle@hook
3468   \global\let\bxjs@pagestyle@hook\relax}
```

■和文空白命令

```
3469 \ifbxjs@jaspace@cmd
```

`\jaenspace` 半角幅の水平空き。

```
3470   \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。

```
3471   \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。

```
3472   \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3473   \DeclareRobustCommand*\jaspace[1]{%
3474     \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3475     \ClassError\bxjs@clsname
3476       {Unknown jaspacer: #1}{\@eha}%
3477     \else
3478       \csname bxjs@jaspace@@#1\endcsname
3479     \fi}
3480   \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3481   \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3482   \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3483 \fi
```

終わり。

```
3484 \fi
```

以上で終わり。

```
3485 %</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、T_EX 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。

```
3486 %<*standard>
3487 %% このファイルは日本語文字を含みます
3488 \input{bxjsja-minimal.def}
```

simplejasetup は standard では無効になる。

```
3489 \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。

```
3490 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3491 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3492 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3493 \define@key{bxjsStd}{jis2004}[true]{%
3494   \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。

```
3495 \let\bxjs@jp@font\@empty

jis2004 オプションの処理。
3496 \define@key{bxjsStd}{font}{%
3497   \edef\bxjs@jp@font{#1}}
```

実際の japaram の値を適用する。

```
3498 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3499 \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。

```
3500 \ifbxjs@jp@jismmiv
3501 \g@addto@macro\@classoptionslist{,jis2004}
3502 % \ifpackagewith 判定への対策
3503 \PassOptionsToPackage{jis2004}{otf}
3504 \fi
```

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

```
3505 \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 3506 \@tempdima=0.25mm
3507 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3508 \let\jH\jQ
```

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

```
\trueH 3509 \ifjsc@mag
3510 \@tempdimb=\jsBaseFontSize\relax
3511 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3512 \@tempdima=2.5mm
3513 \bxjs@invscale\@tempdima\bxjs@tmpa
3514 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3515 \@tempdima=10pt
3516 \bxjs@invscale\@tempdima\bxjs@tmpa
3517 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3518 \else \let\trueQ\jQ \let\bxjs@truept\p@
3519 \fi
3520 \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。

```
3521 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3522 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3523 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3524 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3525 \fi
```

\jsSetQHLLength \jsSetQHLLength\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。

```
3526 \def\jsSetQHLLength#1#2{%
```

```

3527 \begingroup
3528   \bxjs@parse@qh{#2}%
3529   \ifx\bxjs@tmpb\relax \setlength\@tempdima{#2}%
3530   \else \@tempdima\bxjs@tmpb\relax
3531   \fi
3532   \xdef\bxjs@g@tmpa{\the\@tempdima}%
3533 \endgroup
3534 #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。

```

3535 \if j\jsEngine

```

(u)pL^AT_EX ではこの処理は不要なので無効化する。

```

3536 \def\bxjs@parse@qh#1{%
3537   \let\bxjs@tmpb\relax}
3538 \else
3539 \def\bxjs@parse@qh#1{%
3540   \def\bxjs@tmpa{#1}\let\bxjs@tmpb\relax
3541   \bxjs@parse@qh@a{trueQ}\trueQ
3542   \bxjs@parse@qh@a{trueH}\trueH
3543   \bxjs@parse@qh@a{Q}\jQ
3544   \bxjs@parse@qh@a{H}\jH}
3545 \def\bxjs@parse@qh@a#1#2{%
3546   \ifx\bxjs@tmpb\relax
3547     \def\bxjs@next##1#1\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}}%
3548     \expandafter\bxjs@next\bxjs@tmpa\@nil#1\@nil\@nnil{#2}%
3549     \fi}
3550 \def\bxjs@parse@qh@b#1#2#3{%
3551   \ifx\@nnil#2\@nnil\else \def\bxjs@tmpb{#1#3}\fi}
3552 \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。

```

3553 \def\jafontsize#1#2{%
3554   \begingroup
3555     \bxjs@jafontsize@a{#1}%
3556     \@tempdimb\jsInverseScale\@tempdima
3557     \bxjs@jafontsize@a{#2}%
3558     \xdef\bxjs@g@tmpa{%
3559       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3560   \endgroup\bxjs@g@tmpa}
3561 \def\bxjs@jafontsize@a#1{%
3562   \bxjs@parse@qh{#1}%
3563   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3564   \@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。

```
3565 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3566 \newcommand*\setkanjiskip[1]{%
```

```
3567   \edef\bxjs@kanjiskip{#1}%
```

```
3568   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3569 \newcommand*\getkanjiskip{%
```

```
3570   \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。

```
3571 \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 3572 \bxjs@robust@def\bxjs@enable@kanjiskip{%
```

```
3573   \bxjs@kanjiskip@enabledtrue
```

```
3574   \bxjs@reset@kanjiskip}
```

```
3575 \bxjs@robust@def\bxjs@disable@kanjiskip{%
```

```
3576   \bxjs@kanjiskip@enabledfalse
```

```
3577   \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。

```
3578 \bxjs@robust@def\bxjs@reset@kanjiskip{%
```

```
3579   \ifbxjs@kanjiskip@enabled
```

```
3580     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
```

```
3581   \else \@tempskipa\z@
```

```
3582   \fi
```

```
3583   \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 3584 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3585 \newcommand*\setxkanjiskip[1]{%
```

```
3586   \edef\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3587   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3588 \newcommand*\getxkanjiskip{%
```

```
3589   \bxjs@xkanjiskip}
```

```
\bxjs@disable@xkanjiskip 3590 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3591 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3592   \bxjs@xkanjiskip@enabledtrue
```

```
3593   \bxjs@reset@xkanjiskip}
```

```
3594 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3595   \bxjs@xkanjiskip@enabledfalse
```

```
3596   \bxjs@reset@xkanjiskip}
```

```
3597 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```
3598   \ifbxjs@xkanjiskip@enabled
```

```
3599     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
```

```

3600 \else \@tempskipa\z@
3601 \fi
3602 \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。

```

3603 \g@addto@macro\jsResetDimen{%
3604 \bxjs@reset@kanjiskip
3605 \bxjs@reset@xkanjiskip}
3606 \let\bxjs@apply@kanjiskip\relax
3607 \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。

```

3608 \@onlypreamble\bxjs@adjust@jafont
3609 \def\bxjs@adjust@jafont#1{%
3610 \ifx\jsJaFont\bxjs@@auto
3611 \bxjs@get@kanjiEmbed
3612 \ifx\bxjs@kanjiEmbed\relax
3613 \let\bxjs@tmpa\@empty
3614 \else
3615 \let\bxjs@tmpa\bxjs@kanjiEmbed
3616 \fi
3617 \else
3618 \let\bxjs@tmpa\jsJaFont
3619 \fi
3620 \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3621 \ClassWarningNoLine\bxjs@clsname
3622 {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3623 not available on the current situation}%
3624 \let\bxjs@tmpa\@empty
3625 \fi\fi
3626 }
3627 \def\bxjs@@auto{auto}
3628 \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。

```

3629 \let\bxjs@kanjiEmbed\relax

```

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

```

3630 \@onlypreamble\bxjs@get@kanjiEmbed
3631 \def\bxjs@get@kanjiEmbed{%
3632 \begingroup\setbox\z@=\hbox{%
3633 \global\let\bxjs@g@tmpa\relax

```



```

3634 \endlinechar\m@ne
3635 \let\do\@makeother\dospecials
3636 \catcode32=10 \catcode12=10 %form-feed
3637 \let\bxjs@tmpa\@empty
3638 \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3639 \ifeof\@inputcheck\else
3640 \read\@inputcheck to\bxjs@tmpa
3641 \closein\@inputcheck
3642 \fi
3643 \ifx\bxjs@tmpa\@empty\else
3644 \openin\@inputcheck="\bxjs@tmpa"\relax
3645 \@tempswatrue
3646 \loop\if@tempswa
3647 \read\@inputcheck to\bxjs@tmpa
3648 \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3649 \ifx\bxjs@tmpb\relax\else
3650 \global\let\bxjs@g@tmpa\bxjs@tmpb
3651 \@tempswafalse
3652 \fi
3653 \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3654 \ifx\bxjs@tmpb\relax\else
3655 \global\let\bxjs@g@tmpa\bxjs@tmpb
3656 \@tempswafalse
3657 \fi
3658 \ifeof\@inputcheck \@tempswafalse \fi
3659 \repeat
3660 \fi
3661 }\endgroup
3662 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3663 }
3664 \@onlypreamble\bxjs@get@ke@a
3665 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3666 \ifx$#1$\def\bxjs@tmpb{#2}%
3667 \else \let\bxjs@tmpb\relax
3668 \fi}
3669 \@onlypreamble\bxjs@get@ke@b
3670 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3671 \ifx$#1$\def\bxjs@tmpb{#2}%
3672 \else \let\bxjs@tmpb\relax
3673 \fi}

```

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

```

3674 \newcommand*\jachar[1]{%
3675 \begingroup

```

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

```

3676 \jsLetHeadChar\bxjs@tmpa{#1}%
3677 \ifx\bxjs@tmpa\relax
3678 \ClassWarningNoLine\bxjs@clsname

```

```

3679      {Illegal argument given to \string\jachar}%
3680    \else
3681      \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3682    \fi
3683  \endgroup}

```

\jsJaChar を \jachar と等価にする。

```

3684 \let\jsJaChar\jachar

```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```

3685 \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。

```

3686 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3687 \@onlypreamble\bxjs@fix@hyperref@unicode
3688 \def\bxjs@fix@hyperref@unicode#1{%
3689   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3690   \@namedef{KV@Hyp@bxjs/hook}##1{%
3691     \KV@Hyp@unicode{##1}%
3692     \def\KV@Hyp@unicode####1{%
3693       \expandafter\ifx\csname if##1\expandafter\endcsname
3694         \csname if####1\endcsname\else
3695         \ClassWarningNoLine\bxjs@cclsname
3696         {Blcoked hyperref option 'unicode=####1'}%
3697       \fi
3698     }%
3699   }%
3700 }

```

\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。

```

3701 \@onlypreamble\jsCheckHyperrefUnicode
3702 \let\jsCheckHyperrefUnicode\@empty
3703 \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。

```

3704 \@onlypreamble\bxjs@check@hyperref@unicode
3705 \def\bxjs@check@hyperref@unicode#1{%
3706   \g@addto@macro\jsCheckHyperrefUnicode{%
3707     \@tempwafalse
3708     \begingroup
3709       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3710         \aftergroup\@tempwattrue \fi
3711       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3712         \csname if#1\endcsname
3713       \aftergroup\@tempwattrue \fi
3714     \endgroup

```

```

3715 \if@tempswa\else
3716 \ClassError\bxjs@clsname
3717 {The value of hyperref 'unicode' key is not suitable\MessageBreak
3718 for the present engine (must be #1)}%
3719 {\@ehc}%
3720 \fi}}

```

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

```

3721 \onlypreamble\bxjs@urgent@special
3722 \def\bxjs@urgent@special#1{%
3723 \AtBeginDvi{\special{#1}}%
3724 \AtBeginDocument{%
3725 \@ifpackageloaded{atbegshi}{%
3726 \begingroup
3727 \toks\z{\special{#1}}%
3728 \toks\tw{\expandafter\AtBegShi@HookFirst}%
3729 \xdef\AtBegShi@HookFirst{\the\toks@the\toks\tw}%
3730 \endgroup
3731 }{}%
3732 }%
3733 }

```

C.2 pTeX 用設定

```
3734 \if j\jsEngine
```

■ 共通命令の実装

```

3735 \def\bxjs@apply@kanjiskip{%
3736 \kanjiskip\@tempskipa}
3737 \def\bxjs@apply@xkanjiskip{%
3738 \xkanjiskip\@tempskipa}

```

`\jaJaChar` のサブマクロ。

```

3739 \def\bxjs@jachar#1{%
3740 \bxjs@jachar@a#1...\@nil}
3741 \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。

```
3742 \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。

```

3743 \else\ifx.#3%
3744 \@tempcnta`#1 \multiply\@tempcnta64
3745 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3746 \bxjs@jachar@b
3747 \else\ifx.#4%
3748 \@tempcnta`#1 \multiply\@tempcnta64
3749 \advance\@tempcnta`#2 \multiply\@tempcnta64

```

```

3750 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3751 \bxjs@jachar@b
3752 \else
3753 \@tempcnta`#1 \multiply\@tempcnta64
3754 \advance\@tempcnta`#2 \multiply\@tempcnta64
3755 \advance\@tempcnta`#3 \multiply\@tempcnta64
3756 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3757 \bxjs@jachar@b
3758 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3759 \ifjsWithupTeX
3760 \def\bxjs@jachar@b{\kchar\@tempcnta}
3761 \else
3762 \def\bxjs@jachar@b{%
3763 \ifx\bxUInt\@undefined\else
3764 \bxUInt{\@tempcnta}%
3765 \fi}
3766 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

3767 \ifbxjs@jaspace@cmd
3768 \def\jathinspace{\hskip\xkanjiskip}
3769 \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。

```

3770 \ifbxjs@jp@jismmiv
3771 \PassOptionsToPackage{prefer2004jis}{pxchfon}
3772 \PassOptionsToPackage{prefer2004jis}{pxbabel}
3773 \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。

```

3774 \let\bxjs@tmpa\jsJaFont
3775 \ifx\bxjs@tmpa\bxjs@@auto
3776 \let\bxjs@tmpa\@empty
3777 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3778 \def\bxjs@tmpa{noembed}
3779 \fi\fi
3780 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3781 \ifx\bxjs@tmpa\@empty\else
3782 \edef\bxjs@next{%
3783 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3784 }\bxjs@next
3785 \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……。)

```

3786 \begingroup
3787   \global\let\@gtempa\relax
3788   \catcode`\|=0 \catcode`\|=12
3789   |def|bxjs@check#1|@nil{%
3790     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3791   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3792     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3793   |catcode`\|=0 \catcode`\|=12
3794   \def\bxjs@check@b#1keyval#2\@nnil{%
3795     \ifx$#2$ \else
3796       \xdef\@gtempa{%
3797         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3798       \fi}
3799   \@firstofone{%
3800     \catcode10=12 \endlinechar\m@ne
3801     \let\do\@makeother \dospecials \catcode32=10
3802     \openin\@inputcheck=otf.sty\relax
3803     \@tempwattrue
3804     \loop\if@tempwa
3805       \ifeof\@inputcheck \@tempwafalse \fi
3806       \if@tempwa
3807         \read\@inputcheck to\bxjs@line
3808         \expandafter\bxjs@check\bxjs@line\@nil
3809       \fi
3810     \repeat
3811     \closein\@inputcheck
3812   \endgroup}
3813   \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3814 \ifbxjs@hyperref@enc
3815   \bxjs@check@hyperref@unicode{false}
3816 \fi

```

tounicode special 命令を出力する。

```

3817 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3818   \else\ifjsWithpTeXng T\else F\fi\fi T%
3819   \ifnum\jis"2121="A1A1 %euc
3820     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3821   \else\ifnum\jis"2121="8140 %sjis
3822     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3823   \else\ifnum\jis"2121="3000 %uptex
3824     \ifbxjs@bigcode

```

```

3825     \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3826     \PassOptionsToPackage{bigcode}{pxjahyper}
3827     \else
3828     \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3829     \fi
3830 \fi\fi\fi
3831 \let\bxToUnicodeSpecialDone=t
3832 \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。

```

3833 \ifx f\bxjs@enablejfam\else
3834     \@enablejfamtrue
3835 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

3836 \if@enablejfam
3837     \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3838     \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3839     \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3840     \jfam\symmincho
3841     \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3842     \AtBeginDocument{%
3843         \ifx\reDeclareMathAlphabet\undefined\else
3844             \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3845             \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3846             \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3847         \fi}
3848 \fi

```

C.3 pdf_TE_X 用設定 : CJK + bxcjkatype

```

3849 \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。

```

3850 \bxjs@adjust@jafont{f}
3851 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3852 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
3853 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
3854     \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
3855 \fi
3856 \edef\bxjs@next{%
3857     \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
3858 }\bxjs@next

```

```
3859 \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。

```
3860 \ifbxjs@hyperref@enc
3861 \PassOptionsToPackage{unicode}{hyperref}
3862 \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。

```
3863 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3864 \begingroup
3865 \CJK@input{UTF8.bdg}
3866 \endgroup
3867 \g@addto@macro\pdfstringdefPreHook{%
3868 \@nameuse{CJK@UTF8Binding}%
3869 }
3870 \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。

```
3871 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3872 \g@addto@macro\pdfstringdefPreHook{%
3873 \ifx~\bxjs@@CJKtilde
3874 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3875 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3876 \let~\@empty
3877 \fi
3878 }
3879 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3880 \def\bxjs@@tildecmd{~}
3881 \def\bxjs@LetUnexpandableSpace#1{%
3882 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
3883 \bxjs@org@LetUnexpandableSpace#1%
3884 \fi}
3885 \fi
```

■共通命令の実装

```
3886 \newskip\jsKanjiSkip
3887 \newskip\jsXKanjiSkip
3888 \ifx\CJKecglue\@undefined
3889 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3890 \fi
3891 \let\autospacing\bxjs@enable@kanjiskip
3892 \let\noautospacing\bxjs@disable@kanjiskip
3893 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
```

```

3894 \def\bxjs@apply@kanjiskip{%
3895   \jsKanjSkip\@tempskipa
3896   \let\CJKglue\bxjs@CJKglue}
3897 \let\autoxspacing\bxjs@enable@xkanjiskip
3898 \let\noautoxspacing\bxjs@disable@xkanjiskip
3899 \protected\def\bxjs@CJKe glue{\hskip\jsXKanjSkip}
3900 \def\bxjs@apply@xkanjiskip{%
3901   \jsXKanjSkip\@tempskipa
3902   \let\CJKe glue\bxjs@CJKe glue}

```

\jachar のサブマクロの実装。

```

3903 \def\bxjs@jachar#1{%
3904   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

3905 \ifbxjs@jaspace@cmd
3906   \protected\def\jathinspace{\CJKe glue}
3907 \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。

```

3908 \ifx t\bxjs@enablejfam
3909   \ClassWarningNoLine\bxjs@clsname
3910     {You cannot use 'enablejfam=true', since the\MessageBreak
3911      CJK package does not support Japanese math}
3912 \fi

```

C.4 X₃TEX 用設定 : xeCJK + zxjatype

```

3913 \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。

```

3914 \RequirePackage{zxjatype}
3915 \PassOptionsToPackage{no-math}{fontspec}%!
3916 \PassOptionsToPackage{xetex}{graphicx}%!
3917 \PassOptionsToPackage{xetex}{graphics}%!
3918 \ifx\zxJaFamilyName\@undefined
3919   \ClassError\bxjs@clsname
3920     {xeCJK or zxjatype is too old}\@ehc
3921 \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。

```

3922 \bxjs@adjust@jafont{f}
3923 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3924 \ifx\bxjs@tmpa\@empty
3925   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3926   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}

```



```

3927 \else
3928   \edef\bxjs@next{%
3929     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
3930   }\bxjs@next
3931 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X_ƎT_ƎX 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_ƎT_ƎX 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_ƎT_ƎX 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。

```

3932 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3933   \ifbxjs@hyperref@enc
3934     \PassOptionsToPackage{unicode}{hyperref}
3935   \fi
3936 \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。

```

3937 \AtEndOfPackage{%
3938 \def\@gnewline #1{%
3939   \ifvmode \@nolnerr
3940   \else
3941     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3942     \nobreak \hskip-1sp\hskip1sp\relax
3943     \ignorespaces
3944   \fi}
3945 }

```

■共通命令の実装

```

3946 \newskip\jsKanjiSkip
3947 \newskip\jsXKanjiSkip
3948 \ifx\CJKecglue\undefined
3949   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3950 \fi
3951 \let\autospacing\bxjs@enable@kanjiskip

```

```

3952 \let\noautospacing\bxjs@disable@kanjiskip
3953 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3954 \def\bxjs@apply@kanjiskip{%
3955   \jsKanjiSkip\@tempskipa
3956   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3957 \let\autoxspacing\bxjs@enable@xkanjiskip
3958 \let\noautoxspacing\bxjs@disable@xkanjiskip
3959 \protected\def\bxjs@CJKe glue{\hskip\jsXKanjiSkip}
3960 \def\bxjs@apply@xkanjiskip{%
3961   \jsXKanjiSkip\@tempskipa
3962   \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。

```

3963 \ifx\mcfamily\@undefined
3964   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3965   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3966 \fi

```

`\jachar` のサブマクロの実装。

```

3967 \def\bxjs@jachar#1{%
3968   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3969   #1}

```

`\jathinspace` の実装。

```

3970 \ifbxjs@jaspace@cmd
3971   \protected\def\jathinspace{\CJKe glue}
3972 \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。

```

3973 \ifx t\bxjs@enablejfam
3974   \@enablejfamtrue
3975 \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。

```

3976 \if@enablejfam
3977   \xeCJKsetup{CJKmath=true}
3978 \fi

```

C.5 Lua_TE_X 用設定 : Lua_TE_X-ja

```

3979 \else\if 1\jsEngine

```

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

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

`\jsZw` であることに注意が必要。

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

```
3980 \let\zw\@undefined
3981 \RequirePackage{luatexja}
3982 \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？

```
3983 \ExplSyntaxOn
3984 \fp_gset:Nn \g_ltj_fontspec_scale_fp { \jsScale }
3985 \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)。

```
3986 \bxjs@adjust@jafont{t}
3987 \ifx\bxjs@tmpa\bxjs@@noEmbed
3988   \def\bxjs@tmpa{noembed}
3989 \fi
3990 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3991 \ifx\bxjs@tmpa\@empty
3992   \defaultjfontfeatures{ Kerning=Off }
3993   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3994   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3995 \else
3996   \edef\bxjs@next{%
3997     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
3998   }\bxjs@next
3999 \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。

```
4000 \DeclareRobustCommand\rmfamily
4001   {\not@math@alphabet\rmfamily\mathrm}
4002   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4003 \DeclareRobustCommand\sffamily
4004   {\not@math@alphabet\sffamily\mathsf}
4005   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4006 \DeclareRobustCommand\ttfamily
4007   {\not@math@alphabet\ttfamily\mathtt}
4008   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4009 \AtBeginDocument{%
4010   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4011   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4012   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4013 \bxjs@if@sf@default{%
4014   \renewcommand\kanjifamilydefault{\gtdefault}}
```

■和文パラメタの設定

```
4015 % 次の3つは既定値の通り
4016 %\ltjsetparameter{prebreakpenalty={`',10000}}
4017 %\ltjsetparameter{postbreakpenalty={`",10000}}
4018 %\ltjsetparameter{prebreakpenalty={`",10000}}
4019 \ltjsetparameter{jaxspmode={`!',1}}
4020 \ltjsetparameter{jaxspmode={`〒,2}}
4021 \ltjsetparameter{alxspmode={`+,3}}
4022 \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。

```
4023 \ifx\ltjfakeparbegin\@undefined
4024   \protected\def\ltjfakeparbegin{%
4025     \ifhmode
4026       \relax\directlua{%
4027         luatexja.jfmglue.create_beginpar_node()}}
4028   \fi
4029 \fi
```

ltjs* クラスの定義と同等になるようにパッチを当てる。

```
4030 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4031 \begingroup
4032   \let\% \@percentchar \def\@#1{[[\detokenize{#1}]]}
4033   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4034   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4035     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4036   \directlua{
4037     local function patchcmd(cs, code, from, to)
4038       tex.sprint(code:gsub(from:gsub("%W", "\\%\\%\\%0"), "\\%0"..to)
4039         :gsub("macro:", "\\@\\gdef..cs, 1):gsub("->", "{", 1).."")
4040     end
4041     patchcmd(\@\\xsect, [[\meaning\\xsect]],
4042       \@{\hskip-\@tempskipa}, \@\\ltjfakeparbegin)
4043     patchcmd(\@\\item, [[\meaning\\item]],
4044       \bxjs@tmpa, \@\\ltjfakeparbegin)}
4045 \endgroup
4046 \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。

```
4047 \ifbxjs@hyperref@enc
4048   \PassOptionsToPackage{unicode}{hyperref}
4049   \bxjs@check@hyperref@unicode{true}
```

```
4050 \fi
```

■ 共通命令の実装

```
4051 \protected\def\autospacing{%
4052   \ltjsetparameter{autospacing=true}}
4053 \protected\def\noautospacing{%
4054   \ltjsetparameter{autospacing=false}}
4055 \protected\def\autoxspacing{%
4056   \ltjsetparameter{autoxspacing=true}}
4057 \protected\def\noautoxspacing{%
4058   \ltjsetparameter{autoxspacing=false}}
4059 \def\bxjs@apply@kanjiskip{%
4060   \ltjsetparameter{kanjiskip={\@tempskipa}}}
4061 \def\bxjs@apply@xkanjiskip{%
4062   \ltjsetparameter{xkanjiskip={\@tempskipa}}}
```

`\jachar` のサブマクロの実装。

```
4063 \def\bxjs@jachar#1{%
4064   \ltjjachar`#1\relax}
```

`\jathinspace` の実装。

```
4065 \ifbxjs@jaspace@cmd
4066   \protected\def\jathinspace{%
4067     \hskip\ltjgetparameter{xkanjiskip}\relax}
4068 \fi
```

■ 和文数式ファミリー `LuaTeX-j` では和文数式ファミリーは常に有効で、既にこの時点で必要な設定は済んでいる。従って `@enablejfam` は常に真になる。

```
4069 \ifx f\bxjs@enablejfam
4070   \ClassWarningNoLine\bxjs@clsname
4071     {You cannot use 'enablejfam=false', since the\MessageBreak
4072     LuaTeX-j always provides Japanese math families}
4073 \fi
```

C.6 共通処理 (2)

```
4074 \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 4075 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4076   \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4077   \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4078 \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。

```
4079 \ifx\mathmc\@undefined
```

```

4080 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4081 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4082 \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)。

```

4083 \ifbxjs@jaspace@cmd
4084 \bxjs@protected\def\bxjs@choice@jathinspace{%
4085   \relax\ifmmode \mskip\medmuskip
4086   \else \jathinspace\ignorespaces
4087   \fi}
4088 \jsAtEndOfClass{%
4089   \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4090   \else \def\>{\protect\bxjs@choice@jathinspace}%
4091   \fi}
4092 \fi

```

■和文・和欧文間空白の初期値

```

4093 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4094 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4095 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4096 \fi

```

以上で終わり。

```

4097 %</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。

```

4098 %<*modern>
4099 \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。

```

4100 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4101 \def\encodingdefault{T1}%
4102 \input{t1enc.def}%
4103 \fontencoding\encodingdefault\selectfont
4104 \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。

```
4105 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4106 \renewcommand{\rmdefault}{lmr}
4107 \renewcommand{\sfdefault}{lms}
4108 \renewcommand{\ttdefault}{lmtt}
4109 \fi
```

大型演算子用の数式フォントの設定。

※ `amsmath` パッケージと同等にする。

```
4110 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4111   <-7.5>cmex7<7.5-8.5>cmex8%
4112   <8.5-9.5>cmex9<9.5->cmex10}{}%
4113 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

amsmath 読込時に上書きされるのを防ぐ。
4114 \def\cmex@opt{10}
```

D.2 fixltx2e 読込

※ `fixltx2e` 廃止前の \LaTeX カーネルの場合。

```
4115 \ifx\@IncludeInRelease\@undefined
4116 \RequirePackage{fixltx2e}
4117 \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。

```
4118 \RequirePackage{bxjcsjkat}
```

D.4 完了

おしまい。

```
4119 %</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。

```
4120 %<*pandoc>
4121 \input{bxjsja-standard.def}
4122 \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。

4123 \@onlypreamble\bxjs@set@dupload@proc
4124 \def\bxjs@set@dupload@proc#1{%
4125   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
4126 \@onlypreamble\bxjs@set@dupload@proc@a
4127 \def\bxjs@set@dupload@proc@a#1{%
4128   \@onlypreamble#1\def#1##1}
```

\@if@options \@if@options の再定義。

```
4129 \@onlypreamble\bxjs@org@if@options
4130 \let\bxjs@org@if@options\@if@options
4131 \newif\ifbxjs@dlp
4132 \def\@if@options#1#2#3{%
4133   \bxjs@dlpfalse
4134   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4135   \ifx\bxjs@tmpa\bxjs@tmpb
4136     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4137       \bxjs@dlptrue \fi
4138   \fi
4139   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
4140   \else \expandafter\bxjs@org@if@options
4141   \fi {#1}{#2}{#3}}
4142 \AtBeginDocument{%
4143   \let\@if@options\bxjs@org@if@options}
4144 \@onlypreamble\bxjs@do@dupload@proc
4145 \def\bxjs@do@dupload@proc#1#2#3{%
4146   \csname bxjs@dlp/#2.#1\endcsname{#3}%
4147   \@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。

```
4148 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0
```

Polyglossia について。

```
4149 \pandocSkipLoadPackage{polyglossia}
```



```

4150 \bxjs@set@dupload@proc{polyglossia.sty}{%
4151   \ClassWarning\bxjs@clsname
4152   {Loading of polyglossia is blocked}}
4153 \ifx\setmainlanguage\@undefined
4154 \newcommand*\setmainlanguage[2] [] {}
4155 \newcommand*\setotherlanguage[2] [] {%
4156   \ifcat_#2_\else
4157     \expandafter\let\csname #2\endcsname\@empty
4158     \expandafter\let\csname end#2\endcsname\@empty
4159     \expandafter\let\csname text#2\endcsname\@firstofone
4160   \fi}
4161 \newcommand*\setotherlanguages[2] [] {%
4162   \@for\bxjs@tmpa:={#2}\do{%
4163     \setotherlangauge{\bxjs@tmpa}}}
4164 \fi
4165 \else

```

Babel について。

```

4166 \pandocSkipLoadPackage{babel}
4167 \bxjs@set@dupload@proc{babel.sty}{%
4168   \ClassWarning\bxjs@clsname
4169   {Loading of babel is blocked}}
4170 \let\foreignlanguage\@secondoftwo
4171 \let\otherlanguage\@gobble
4172 \let\endotherlanguage\@empty
4173 \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。

```

4174 \bxjs@set@dupload@proc{geometry.sty}{%
4175   \setpagelayout*{#1}}

```

E.4 CJKmainfont 変数

Lua_T_EX (+ Lua_T_EX-ja) の場合に CJKmainfont 変数が指定された場合は \setmainjfont の指定にまわす。

```

4176 \if 1\jsEngine
4177   \pandocSkipLoadPackage{xeCJK}
4178   \providecommand*\setCJKmainfont{\setmainjfont}
4179 \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。

```
4180 \AtBeginDocument{%
4181   \@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。

```
4182   \ifx\oldparagraph\@undefined\else
4183     \@tempswattrue
4184   \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4185   \ifnum\c@secnumdepth>3
4186     \@tempswattrue
4187   \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。

```
4188   \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4189     \let\jsParagraphMark\@empty
4190   \fi\fi}
```

E.6 全角空白文字

```
4191 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4192   \catcode"3000=\active
4193   \begingroup \catcode`\!=7
4194   \protected\gdef!!!!3000{\zwspace}
4195   \endgroup
4196 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4197   \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4198   \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4199 \fi\fi
```

■hyperref 対策 hyperref の unicode オプションの固定を行う。

```
4200 \if j\jsEngine
4201   \bxjs@fix@hyperref@unicode{false}
4202 \else
```

```
4203 \bxjs@fix@hyperref@unicode{true}
4204 \fi
```

E.7 完了

おしまい。

```
4205 %</pandoc>
```

和文ドライバ実装はここまで。

```
4206 %</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。

```
4207 %<*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

```
4208 %<*compat>
```

```
4209 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
4210 \let\bxac@engine=n
```

```
4211 \def\bxac@do#1#2{%
```

```
4212 \edef\bxac@tmpa{\string#1}%
```

```
4213 \edef\bxac@tmpb{\meaning#1}%
```

```
4214 \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
4215 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
4216 \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。

```
4217 \ifx\jsAtEndOfClass\@undefined
```

```
4218 \let\bxac@delayed@if@bxjs\@firstofone
```

```
4219 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
4220 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4221 \newif\ifbxac@in@old@behavior
4222 \let\ImposeOldLuaTeXBehavior\relax
4223 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 XeTeX 部分

```
4224 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。
4225 \@onlypreamble\bxac@adjust@charclass
4226 \bxac@delayed@if@bxjs{%
4227   \@ifpackageloaded{xCJK}{}{%else
4228     \ifx\XeTeX@alloc@intercharclass\undefined\else
4229       \ifnum\XeTeX@alloc@intercharclass=\z@
4230         \PackageInfo\bxac@pkgname
4231           {Setting up interchar class for CJK...\@gobble}%
4232         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4233           \XeTeX@alloc@intercharclass=3
4234         }{%else
4235           \PackageWarning\bxac@pkgname
4236             {Cannot find file 'load-unicode-xetex-classes.tex'%
4237             \@gobble}%
4238         }%
4239       \fi\fi
4240     \ifnum\XeTeXcharclass"3041=\z@
4241       \PackageInfo\bxac@pkgname
4242         {Adjusting interchar class for CJK...\@gobble}%
4243       \@for\bxac@x:={%
4244         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4245         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4246         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4247         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4248         31FF%
4249       }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4250     \fi
4251   }%
4252 }

以上。
4253 \fi
```

G.3 LuaTeX 部分

```
4254 \ifx l\bxac@engine

ムニャムニャ。
4255 \unless\ifnum\luaTeXversion<80 \ifnum\luaTeXversion<85
4256   \chardef\pdfTeXversion=200
4257   \def\pdfTeXrevision{0}
```

```

4258 \let\pdftexbanner\luatexbanner
4259 \fi\fi

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 4260 \begingroup\expandafter\expandafter\expandafter\endgroup
4261 \expandafter\ifx\csname outputmode\endcsname\relax\else
4262 \def\bxac@ob@list{%
4263   \do{\let}\pdfoutput{\outputmode}%
4264   \do{\let}\pdfpagewidth{\pagewidth}%
4265   \do{\let}\pdfpageheight{\pageheight}%
4266   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
4267   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
4268 \def\bxac@ob@do#1#2{\begingroup
4269   \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4270 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4271   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4272   \else \let#3#1\relax \let#1\@undefined
4273   \fi}
4274 \protected\def\ImposeOldLuaTeXBehavior{%
4275   \unless\ifbxac@in@old@behavior
4276     \bxac@in@old@behaviortrue
4277     \let\do\bxac@ob@do \bxac@ob@list
4278   \fi}
4279 \protected\def\RevokeOldLuaTeXBehavior{%
4280   \ifbxac@in@old@behavior
4281     \bxac@in@old@behaviorfalse
4282     \let\do\bxac@ob@do \bxac@ob@list
4283   \fi}
4284 \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。

```

4285 \ifnum\luatexversion>64 \directlua{
4286   local function range(cs, ce, cc, ff)
4287     if ff or not tex.getcatcode(cs) == cc then
4288       local setcc = tex.setcatcode
4289       for c = cs, ce do setcc(c, cc) end
4290     end
4291   end
4292   range(0x3400, 0x4DB5, 11, false)
4293   range(0x4DB5, 0x4DBF, 11, true)
4294   range(0x4E00, 0x9FCC, 11, false)
4295   range(0x9FCD, 0x9FFF, 11, true)
4296   range(0xAC00, 0xD7A3, 11, false)
4297   range(0x20000, 0x2A6D6, 11, false)
4298   range(0x2A6D7, 0x2A6FF, 11, true)
4299   range(0x2A700, 0x2B734, 11, false)
4300   range(0x2B735, 0x2B73F, 11, true)
4301   range(0x2B740, 0x2B81D, 11, false)
4302   range(0x2B81E, 0x2B81F, 11, true)

```

```

4303     range(0x2B820, 0x2CEA1, 11, false)
4304     range(0x2CEA2, 0x2FFFD, 11, true)
4305 } \fi

```

以上。

```

4306 \fi

```

G.4 完了

おしまい。

```

4307 %</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

```

4308 %<*cjkcat>
4309 \def\bxjx@pkgnam{bxjscjkat}
4310 \newcount\bxjx@cna

```

\bxjx@engine エンジンの種別。

```

4311 \let\bxjx@engine=n
4312 \def\bxjx@do#1#2{%
4313   \edef\bxjx@tmpa{\string#1}%
4314   \edef\bxjx@tmpb{\meaning#1}%
4315   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4316 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
4317 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
4318 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
4319 \bxjx@do\pdftexversion{\let\bxjx@engine=p}
4320 \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。

```

4321 \def\bxjx@do#1#2{%
4322   \if#1\bxjx@engine
4323     \@ifpackageloaded{#2}{\}%else
4324     \PackageError\bxjx@pkgnam
4325       {Package '#2' must be loaded}%
4326       {Package loading is aborted.\MessageBreak\@ehc}%
4327     \endinput}
4328 \fi}
4329 \bxjx@do{p}{\bxjcjkatype}
4330 \bxjx@do{x}{\xeCJK}
4331 \bxjx@do{l}{\luatexja}

```

古い L^AT_EX の場合、`\TextOrMath` は `fixltx2e` パッケージで提供される。

```
4332 \ifx\TextOrMath\@undefined
4333   \RequirePackage{fixltx2e}
4334 \fi
```

H.2 和文カテゴリコードの設定

upL^AT_EX の場合、和文カテゴリコードの設定を LuaT_EX-j_a と（ほぼ）等価なものに変更する。

※ LuaT_EX-j_a との相違点：A830、A960、1B000。

```
4335 \if u\bxjx@engine
4336 \@for\bxjx@x:={%
4337 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4338 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
4339 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
4340 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
4341 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
4342 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
4343 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
4344 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
4345 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
4346 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
4347 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
4348 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4349 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4350 10480,10500,10530,10600,10800,10840,10860,10880,%
4351 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
4352 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
4353 11000,11080,110D0,11100,11150,11180,111E0,11200,%
4354 11280,112B0,11300,11480,11580,11600,11680,11700,%
4355 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
4356 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4357 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
4358 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
4359 1F700,1F780,1F800,1F900,E0000,F0000,100000%
4360 }\do{\kcatcode"\bxjx@x=15 }
4361 \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。

```
4362 \def\bxjx@grkcyr@list{%
4363 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4364 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
```

4365	\do{0393}{LGR}{\textGamma}{\Gamma}%	% GR. C. L. GAMMA
4366	\do{0394}{LGR}{\textDelta}{\Delta}%	% GR. C. L. DELTA
4367	\do{0395}{LGR}{\textEpsilon}{\epsilon}%	% GR. C. L. EPSILON
4368	\do{0396}{LGR}{\textZeta}{\zeta}%	% GR. C. L. ZETA
4369	\do{0397}{LGR}{\textEta}{\eta}%	% GR. C. L. ETA
4370	\do{0398}{LGR}{\textTheta}{\theta}%	% GR. C. L. THETA
4371	\do{0399}{LGR}{\textIota}{\iota}%	% GR. C. L. IOTA
4372	\do{039A}{LGR}{\textKappa}{\kappa}%	% GR. C. L. KAPPA
4373	\do{039B}{LGR}{\textLambda}{\lambda}%	% GR. C. L. LAMDA
4374	\do{039C}{LGR}{\textMu}{\mu}%	% GR. C. L. MU
4375	\do{039D}{LGR}{\textNu}{\nu}%	% GR. C. L. NU
4376	\do{039E}{LGR}{\textXi}{\xi}%	% GR. C. L. XI
4377	\do{039F}{LGR}{\textOmicron}{\omicron}%	% GR. C. L. OMICRON
4378	\do{03A0}{LGR}{\textPi}{\pi}%	% GR. C. L. PI
4379	\do{03A1}{LGR}{\textRho}{\rho}%	% GR. C. L. RHO
4380	\do{03A3}{LGR}{\textSigma}{\sigma}%	% GR. C. L. SIGMA
4381	\do{03A4}{LGR}{\textTau}{\tau}%	% GR. C. L. TAU
4382	\do{03A5}{LGR}{\textUpsilon}{\Upsilon}%	% GR. C. L. UPSILON
4383	\do{03A6}{LGR}{\textPhi}{\phi}%	% GR. C. L. PHI
4384	\do{03A7}{LGR}{\textChi}{\chi}%	% GR. C. L. CHI
4385	\do{03A8}{LGR}{\textPsi}{\psi}%	% GR. C. L. PSI
4386	\do{03A9}{LGR}{\textOmega}{\omega}%	% GR. C. L. OMEGA
4387	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
4388	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
4389	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
4390	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4391	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4392	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4393	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4394	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4395	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4396	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4397	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4398	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4399	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4400	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4401	\do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
4402	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4403	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4404	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4405	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4406	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4407	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4408	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4409	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4410	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4411	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4412	\do{0401}{T2A}{\CYR0}{\cyr0}%	% CY. C. L. IO
4413	\do{0410}{T2A}{\CYRA}{\cyra}%	% CY. C. L. A

4414 \do{0411}{T2A}{\CYRB}{}%	% CY. C. L. BE
4415 \do{0412}{T2A}{\CYRV}{}%	% CY. C. L. VE
4416 \do{0413}{T2A}{\CYRG}{}%	% CY. C. L. GHE
4417 \do{0414}{T2A}{\CYRD}{}%	% CY. C. L. DE
4418 \do{0415}{T2A}{\CYRE}{}%	% CY. C. L. IE
4419 \do{0416}{T2A}{\CYRZH}{}%	% CY. C. L. ZHE
4420 \do{0417}{T2A}{\CYRZ}{}%	% CY. C. L. ZE
4421 \do{0418}{T2A}{\CYRI}{}%	% CY. C. L. I
4422 \do{0419}{T2A}{\CYRISHRT}{}%	% CY. C. L. SHORT I
4423 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
4424 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
4425 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
4426 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
4427 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
4428 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
4429 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
4430 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
4431 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
4432 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
4433 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
4434 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
4435 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
4436 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
4437 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
4438 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
4439 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
4440 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
4441 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
4442 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
4443 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
4444 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
4445 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
4446 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
4447 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
4448 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
4449 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
4450 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
4451 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
4452 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
4453 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
4454 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
4455 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
4456 \do{043B}{T2A}{\cyrl}{}%	% CY. S. L. EL
4457 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
4458 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
4459 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
4460 \do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
4461 \do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
4462 \do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES

```

4463 \do{0442}{T2A}{\cyrt}{}%           % CY. S. L. TE
4464 \do{0443}{T2A}{\cyru}{}%           % CY. S. L. U
4465 \do{0444}{T2A}{\cyrf}{}%           % CY. S. L. EF
4466 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
4467 \do{0446}{T2A}{\cyrc}{}%           % CY. S. L. TSE
4468 \do{0447}{T2A}{\cyrch}{}%          % CY. S. L. CHE
4469 \do{0448}{T2A}{\cyrsh}{}%          % CY. S. L. SHA
4470 \do{0449}{T2A}{\cyrshch}{}%        % CY. S. L. SHCHA
4471 \do{044A}{T2A}{\cyrhrdsn}{}%       % CY. S. L. HARD SIGN
4472 \do{044B}{T2A}{\cyrrery}{}%        % CY. S. L. YERU
4473 \do{044C}{T2A}{\cyrsftsn}{}%       % CY. S. L. SOFT SIGN
4474 \do{044D}{T2A}{\cyrrerev}{}%       % CY. S. L. E
4475 \do{044E}{T2A}{\cyryu}{}%          % CY. S. L. YU
4476 \do{044F}{T2A}{\cyrya}{}%          % CY. S. L. YA
4477 \do{0451}{T2A}{\cyryo}{}%          % CY. S. L. IO
4478 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4479 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4480 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4481 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
4482 \do{00B4}{TS1}{\textasciicute}{}% % ACUTE ACCENT
4483 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4484 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4485 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
4486 }

4487 \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。

```
4488 \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。

```

4489 \newcommand*\greekasCJK{%
4490   \bxjx@gcc@cjkltrue}
4491 \newcommand*\nogreekasCJK{%
4492   \bxjx@gcc@cjklfalse}

```

\bx@fake@grkl \bx@fake@grkl{(出力文字)}{(基準文字)} :

```

4493 \def\bxjx@do#1\relax{%
4494   \def\bxjx@fake@grkl##1##2{%
4495     \expandafter\bxjx@fake@grkl@a\meaning##2#1\@nil{##1}{##2}}%
4496   \def\bxjx@fake@grkl@a##1#1##2\@nil##3##4{%
4497     \ifx\##1\%
4498       \bxjx@canta##4\divide\bxjx@canta\@cclvi
4499       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
4500       \mathchar\bxjx@canta
4501     \else ##3\fi}
4502 } \expandafter\bxjx@do\string\mathchar\relax

```

■pdfLaTeX・upLaTeX の場合

4503 \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。

```
4504 \@ifpackageloaded{inputenc}{\%}{%else
4505   \RequirePackage[utf8]{inputenc}}
4506 \def\bxjx@tmpa{utf8}
4507 \ifx\bxjx@tmpa\inputencdoingname
4508   \PackageWarningNoLine\bxjx@pkgname
4509     {Input encoding changed to utf8}%
4510   \inputencoding{utf8}%
4511 \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。

```
4512 \if u\bxjx@engine
4513 \kcatcode"0370=15
4514 \kcatcode"0400=15
4515 \kcatcode"0500=15
4516 \fi
```

各文字について \DeclareUnicodeCharacter を実行する。

```
4517 \def\do#1{%
4518   \@tempcnta="#1\relax
4519   \@tempcntb=\@tempcnta \divide\@tempcntb256
4520   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4521   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4522 \def\bxjx@do@a#1#2#3#4#5{%
4523   \ifx\#5\%
4524     \def\bxjx@tmpa{\@inmathwarn#4}%
4525   \else\ifcat A\noexpand#5%
4526     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4527       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4528   \else \def\bxjx@tmpa{#5}%
4529   \fi\fi
4530   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4531   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
4532 \if u\bxjx@engine
4533 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
4534 \def\bxjx@do@b#1#2#3#4#5{%
4535   \kchardef#1=\@tempcnta
4536   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4537   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
4538 \else\if p\bxjx@engine
4539 \def\bxjx@do@b#1#2#3#4#5{%
4540   \mathchardef#1=\@tempcnta
4541   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4542   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
4543 \fi\fi
4544 \bxjx@grkcyr@list
```

```

4545 \let\bxjx@do@a\undefined
4546 \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。

```

4547 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4548 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4549 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4550 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4551   \count@="#1\relax \bxjx@cnta\count@ \divide\bxjx@cnta256
4552   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cnta\endcsname\relax
4553     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4554   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4555     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4556   \else
4557     \wlog{ \space\space skipped defining Unicode char U+#1}%
4558   \fi\fi}

```

`\bxjx@ja@or@not` `\bxjx@ja@or@not`

```

4559 \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。

```

```

4560 \ifbxjx@gcc@CJK #1%

```

`\nogreekasCJK` の場合は、エンコーディングを固定して欧文用のコードを実行するが、そのエンコーディングが未定義の場合は（フォールバックとして）和文用コードを使う。

```

4561 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4562 \else \UseTextSymbol{#2}{#3}%
4563 \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。

```

4564 \begingroup
4565 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4566 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4567   \noexpand\bxjx@swap@DUC@cmd
4568   \the\toks@
4569   \noexpand\bxjx@swap@DUC@cmd}}
4570 \endgroup\next
4571 \def\bxjx@swap@DUC@cmd{%
4572   \let\bxjx@tmpa\DeclareUnicodeCharacter
4573   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4574   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■ Xe_{La}TeX・Lua_{La}TeX の場合

```

4575 \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。

```

4576 \def\do#1{%
4577   \bxjx@cmta="#1\relax
4578   \begingroup
4579     \lccode`~=\bxjx@cmta
4580   \lowercase{\endgroup
4581     \bxjx@do@a{~}}{#1}}
4582 \def\bxjx@do@a#1#2#3#4#5{%
4583   \ifx\#5\\\let\bxjx@tmpa\relax
4584   \else\ifcat A\noexpand#5%
4585     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4586       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4587   \else \def\bxjx@tmpa{#5}%
4588   \fi\fi
4589   \ifx\bxjx@tmpa\relax\else
4590     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4591   \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。

```

4592 \mathchardef\bxjx@tmpa="119
4593 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4594 \let\bxjx@do@a\undefined

```

Lua_T_EX における \(\text{no}\)greekasCJK の定義。jacharrange の設定を変更する。

```

4595 \if l\bxjx@engine
4596   \protected\def\greekasCJK{%
4597     \bxjx@gcc@cjctrue
4598     \ltjsetparameter{jacharrange={+2, +8}}
4599   \protected\def\nogreekasCJK{%
4600     \bxjx@gcc@cjkfalse
4601     \ltjsetparameter{jacharrange={-2, -8}}
4602 \fi

```

X_YTeX における \(\text{no}\)greekasCJK の定義。

```

4603 \if x\bxjx@engine
4604   \protected\def\greekasCJK{%
4605     \bxjx@gcc@cjctrue
4606     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4607     \bxjx@grkcyr@list}
4608   \protected\def\nogreekasCJK{%
4609     \bxjx@gcc@cjkfalse
4610     \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4611     \bxjx@grkcyr@list}
4612 \fi

```

以上。

```

4613 \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。

```
4614 \nogreekasCJK
```

H.5 完了

おしまい。

```
4615 %</cjkat>
```

付録 I 補助パッケージ : bxjspandoc

Pandoc の L^AT_EX 用標準テンプレートをより幸せに使うための設定。BXJS クラスの pandoc ドライバのコードの中の、“汎用的”に使える部分を切り出したもの。つまり現在の pandoc ドライバはこのパッケージを読みこむ。

※テンプレートの T_EX コードより前に読み込む必要があるため、専ら文書クラス内での読込に限られる。

I.1 準備

```
4616 %<*ancpandoc>
```

```
4617 %% このファイルは日本語文字を含みます.
```

```
4618 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```
4619 \let\bxjsp@engine=n
```

```
4620 \@onlypreamble\bxjsp@do
```

```
4621 \def\bxjsp@do#1#2{%
```

```
4622   \edef\bxjsp@tmpa{\string#1}%
```

```
4623   \edef\bxjsp@tmpb{\meaning#1}%
```

```
4624   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
```

```
4625 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
```

```
4626 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
```

```
4627 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
```

```
4628 \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。

```
4629 \newcommand*\pandocSkipLoadFile[1]{%
```

```
4630   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
```

```
4631 \def\bxjsp@skip@load@file@a#1#2{%
```

```
4632   \ifx#1\relax
```

```

4633 \def#1{2001/01/01}%
4634 \PackageInfo{bxjisp}{\pkgname
4635 {File '#2' marked as loaded\@gobble}%
4636 \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。

```

4637 \newcommand*\pandocSkipLoadPackage[1]{%
4638 \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。

```

4639 \ifx\@IncludeInRelease\@undefined\else
4640 \pandocSkipLoadPackage{fixltx2e}
4641 \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。）

```

4642 \if j\bxjisp@engine
4643 \pandocSkipLoadPackage{cmap}
4644 \fi

```

1.5 microtype パッケージ

警告が多すぎなので消す。

```

4645 \if j\bxjisp@engine \else
4646 \PassOptionsToPackage{verbose=silent}{microtype}
4647 \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。

```

4648 \if j\bxjisp@engine
4649 \pandocSkipLoadPackage{microtype}
4650 \newcommand*\UseMicrotypeSet[2] [] {}
4651 \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。

```
4652 \DeclareRobustCommand{\pandocLdots}{%
4653   \relax\ifmmode \expandafter\bxjsp@org@ldots
4654   \else \expandafter\bxjsp@ldots@a
4655   \fi}
4656 \def\bxjsp@ja@ellipsis{…}
4657 \let\bxjsp@org@ldots\ldots
4658 \def\bxjsp@ldots@a{%
4659   \futurelet\bxjsp@tok\bxjsp@ldots@b}
4660 \def\bxjsp@ldots@b{%
4661   \ifx\bxjsp@tok\bgroup \expandafter\bxjsp@ldots@c
4662   \else \expandafter\bxjsp@org@ldots
4663   \fi}
4664 \def\bxjsp@ldots@c{%
4665   \afterassignment\bxjsp@ldots@d \let\bxjsp@tok=}
4666 \def\bxjsp@ldots@d{%
4667   \futurelet\bxjsp@tok\bxjsp@ldots@e}
4668 \def\bxjsp@ldots@e{%
4669   \ifx\bxjsp@tok\egroup \expandafter\bxjsp@ldots@f
4670   \else \expandafter\bxjsp@ldots@g
4671   \fi}
4672 \def\bxjsp@ldots@f{%
4673   \bxjsp@ja@ellipsis \let\bxjsp@tok=}
4674 \def\bxjsp@ldots@g{%
4675   \expandafter\bxjsp@org@ldots\expandafter{\romannumeral-`} }
```

$\backslash\ldots$ の実装を置き換える。

```
4676 \AtBeginDocument{%
4677   \let\bxjsp@org@ldots\ldots
4678   \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。

```
4679 \IfFileExists{bxpandola.sty}{%  
4680   \RequirePackage{bxpandola}\relax  
4681   \PackageInfo{bxjsp@pkgname  
4682     {PandoLa module is loaded\@gobble}  
4683 }{}
```

I.8 完了

おしまい。

```
4684 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
4685 %</anc>
```