

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”)





v1.8b [2018/04/14]

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

目次

1	はじめに	3
2	オプション	9
3	和文フォントの変更	37
4	フォントサイズ	38
5	レイアウト	44
5.1	ページレイアウト	45
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	59
7	ページスタイル	60
8	文書のマークアップ	63
8.1	表題	63
8.2	章・節	69
8.3	リスト環境	81
8.4	パラメータの設定	88
8.5	フロート	90
8.6	キャプション	91
9	フォントコマンド	92

10	相互参照	95
10.1	目次の類	95
10.2	参考文献	100
10.3	索引	102
10.4	脚注	103
11	段落の頭へのグルー挿入禁止	105
12	いろいろなロゴ	109
13	amsmath との衝突の回避	110
14	初期設定	110
付録 A	和文ドライバの仕様 罨	115
付録 B	和文ドライバ : minimal 罨	116
B.1	補助マクロ	116
B.2	(u)pTeX 用の設定	118
B.3	pdfTeX 用の処理	122
B.4	X _Y TeX 用の処理	123
B.5	後処理 (エンジン共通)	124
付録 C	和文ドライバ : standard 罨	126
C.1	共通処理 (1)	127
C.2	pTeX 用設定	135
C.3	pdfTeX 用設定 : CJK + bxcjkatype	138
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	140
C.5	LuaTeX 用設定 : LuaTeX-ja	142
C.6	共通処理 (2)	145
付録 D	和文ドライバ : modern 罨	146
D.1	フォント設定	146
D.2	fixltx2e 読込	147
D.3	和文カテゴリコード	147
D.4	完了	147
付録 E	和文ドライバ : pandoc 罨	147
E.1	dupload システム	147
E.2	lang 変数	148
E.3	geometry 変数	149
E.4	CJKmainfont 変数	149
E.5	paragraph のマーク	149

E.6	全角空白文字	150
E.7	完了	150
付録 F	補助パッケージ一覧 	150
付録 G	補助パッケージ : bxjscompat 	151
G.1	準備	151
G.2	X _Y TeX 部分	151
G.3	LuaTeX 部分	152
G.4	完了	154
付録 H	補助パッケージ : bxjscjkat 	154
H.1	準備	154
H.2	和文カテゴリコードの設定	155
H.3	ギリシャ・キリル文字の扱い	156
H.4	初期設定	163
H.5	完了	163
付録 I	補助パッケージ : bxjspandoc 	163
I.1	準備	163
I.2	パッケージ読込の阻止	164
I.3	fixltx2e パッケージ	164
I.4	cmap パッケージ	164
I.5	microtype パッケージ	165
I.6	Unicode 文字変換対策	165
I.7	PandoLa モジュール	166
I.8	完了	166

1 はじめに

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

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

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

これは L^AT_EX3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいてもともと奥村晴彦により作成されたものです。現在は日本語 T_EX 開発コミュニティにより GitHub で管理されています。

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

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

[2016-07-13] 日本語 T_EX 開発コミュニティによる管理に移行しました。

[2009-02-22] 田中琢爾氏による upL^AT_EX 対応パッチを取り込みました。

ここでは次のドキュメントクラス（スタイルファイル）を作ります。

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

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

以下では実際のコードに即して説明します。

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

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

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

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

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

■BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため `calc` を読み込む。

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

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

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

クラスの本体ではこの他に geometry パッケージが読み込まれる。

互換性のための補助パッケージを読み込む。

```
11 \IfFileExists{bxjscompat.sty}{%
```

```
12   \let\jsAtEndOfClass@gobble
```

```
13   \RequirePackage{bxjscompat}%
```

```
14 }{}
```

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

```
15 \let\jsArticle=a
```

```
16 \let\jsBook=b
```

```
17 \let\jsReport=r
```

```
18 \let\jsSlide=s
```

```
19 %<article>\let\jsDocClass\jsArticle
```

```
20 %<book>\let\jsDocClass\jsBook
```

```
21 %<report>\let\jsDocClass\jsReport
```

```
22 %<slide>\let\jsDocClass\jsSlide
```

`\jsEngine` [暗黙文字トークン] エンジン ($\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の種類) の種別：`j` = $\mathrm{p}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ 系、`x` = $\mathrm{X}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ 、`p` = $\mathrm{pdf}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ (含 DVI モード)、`l` = $\mathrm{Lua}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ 、`J` = $\mathrm{NTT}_{\mathrm{T}_{\mathrm{E}}\mathrm{X}}$ 、`0` = Omega 系、`n` = 以上の何れでもない。

```
23 \let\jsEngine=n
```

```
24 \def\bxjs@test@engine#1#2{%
```

```
25   \edef\bxjs@tmpa{\string#1}%
```

```
26   \edef\bxjs@tmpb{\meaning#1}%
```

```
27   \ifx\bxjs@tmpa\bxjs@tmpb #2\fi
```

```
28 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
```

```
29 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
```

```
30 \bxjs@test@engine\Omegaversion{\let\jsEngine=0}
```

```
31 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
```

```
32 \bxjs@test@engine\pdftexversion{\let\jsEngine=p}
```

```
33 \bxjs@test@engine\luatexversion{\let\jsEngine=l}
```

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

```
34 \newif\ifjsWithupTeX
```

```
35 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
```

```
36   \jsWithupTeXtrue
```

```
37 \fi\fi
```

```
38 \let\if@jsc@uplatex\ifjsWithupTeX
```

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

```
39 \newif\ifjsWithpTeXng
```

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

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

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

非サポートのエンジンの場合は強制終了させる。

※ NTT jTeX と Omega 系。

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

LuaTeX の場合、本クラス用の Lua モジュールを用意する。

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

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

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

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

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

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

※ LuaTeX 0.8x 版でのプリミティブ名変更に対応。

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

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

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

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

```

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

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

\bxjs@csletcs \bxjs@csletcs{<名前 1>}{<名前 2>} :
78 \def\bxjs@csletcs#1#2{%
79 \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}

\bxjs@catopt \bxjs@catopt{<文字列 1>}{<文字列 2>} : 2つの文字列を , で繋いだ文字列。ただし片方
が空の場合は , を入れない。完全展開可能。
80 \def\bxjs@catopt#1#2{%
81 #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

\bxjs@ifplus \@ifstar の + 版。
82 \def\bxjs@ifplus#1{\@ifnextchar+{\@firstoftwo{#1}}}

\bxjs@gset@tempcnta calc の整数式を用いて \@tempcnta の値を設定する。
83 \let\c@bxjs@tempcnta\@tempcnta
84 \def\bxjs@gset@tempcnta{\setcounter{bxjs@tempcnta}}

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

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

```

```

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

```

今の段階では Q/H だけが使用可能。

```

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

```

`\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。)

```

121 \def\jsAtEndOfClass{%
122 \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_{TEX} 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。

```

123 \@onlypreamble\bxjs@restore@jltrcc
124 \let\bxjs@restore@jltrcc\@empty
125 \if 1\jsEngine
126 \def\bxjs@change@jltrcc#1{%
127 \xdef\bxjs@restore@jltrcc{%
128 \bxjs@restore@jltrcc
129 \catcode`#1=\the\catcode`#1\relax}%
130 \catcode`#1=11\relax}
131 \@tfor\bxjs@x:=西暦\do
132 {\expandafter\bxjs@change@jltrcc\bxjs@x}
133 \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。


```

134 \bxjs@robust@def\jsInhibitGlue{%
135   \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。

136 \if@compatibility
137   \ClassError\bxjs@clsname
138   {Something went chaotic!\MessageBreak
139    (How come '\string\documentstyle' is there?)\MessageBreak
140    I cannot go a single step further...}
141   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
142    then there'll still be hope....}
143   \expandafter\@firstofone
144 \else \expandafter\@gobble
145 \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。
146 \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。
147 \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。
148 %<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。
149 %<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。

150 %<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。

151 \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。

```
152 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
153 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
154 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
155 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
156 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
157 \DeclareOption{b4paper}{\bxjs@setpaper{{257truemm}{364truemm}}}
158 \DeclareOption{b5paper}{\bxjs@setpaper{{182truemm}{257truemm}}}
159 \DeclareOption{b6paper}{\bxjs@setpaper{{128truemm}{182truemm}}}
160 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
161 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
162 \DeclareOption{b4j}{\bxjs@setpaper{{257truemm}{364truemm}}}
163 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
164 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
165 \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。

```
166 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
167 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
168 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
169 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
170 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}
```

■横置き 用紙の縦と横の長さを入れ換えます。

```
171 \newif\if@landscape
172 \@landscapefalse
173 \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。

```
174 \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。

```
175 %<!slide>\@slidefalse
```

```
176 %<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。

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

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

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

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

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

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

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

```
181 \jsSetQHLength\@tempdima{#1}%
```

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

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

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

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

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

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

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

JS クラス互換の `magstyle` 設定オプション。

```
209 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
210 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
211 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}
```

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

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

```
212 \if j\jsEngine
213 \hour\time \divide\hour by 60\relax
214 \@tempcnta\hour \multiply\@tempcnta 60\relax
215 \minute\time \advance\minute-\@tempcnta
216 \DeclareOption{tombow}{%
217   \tombowtrue \tombowdatetrue
218   \setlength{\@tombowwidth}{.1\p}%
```

```

219 \@bannertoken{%
220     \jobname\space(\number\year-\two@digits\month-\two@digits\day
221     \space\two@digits\hour:\two@digits\minute))}%
222 \maketombowbox}
223 \DeclareOption{tombo}{%
224     \tombowtrue \tombowdatefalse
225     \setlength{\@tombowwidth}{.1\p@}%
226     \maketombowbox}
227 \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。

```

228 \if j\jsEngine
229 \DeclareOption{mentuke}{%
230     \tombowtrue \tombowdatefalse
231     \setlength{\@tombowwidth}{\z@}%
232     \maketombowbox}
233 \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。

```

234 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
235 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
236 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 `twocolumn` で二段組になります。

```

237 \DeclareOption{onecolumn}{\@twocolumnfalse}
238 \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。

```

239 \DeclareOption{titlepage}{\@titlepagetrue}
240 \DeclareOption{notitlepage}{\@titlepagefalse}

```

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

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

```

241 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
242 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
243 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}

```

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

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

```

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

```

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

```

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

```

285 }}

■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。

```
286 % \DeclareOption{openbib}{%
287 %   \AtEndOfPackage{%
288 %     \renewcommand\@openbib@code{%
289 %       \advance\leftmargin\bibindent
290 %       \itemindent -\bibindent
291 %       \listparindent \itemindent
292 %       \parsep \z@}%
293 %     \renewcommand\newblock{\par}}}
```

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

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

```
294 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。
295 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
296 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
297 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
298 \define@key{bxjs}{enablejfam}[true]{%
299   \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。

```
300 \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。

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

■和文フォントメトリックの選択 このクラスファイルでは、和文 TFM として東京書籍印刷の小林肇さんの作られた JIS フォントメトリック (`jis`, `jisg`) を標準で使うことにしますが、従来の `min10`, `goth10` などを使いたいときは `mingoth` というオプションを指定します。また、`winjis` オプションで `winjis` メトリック (OTF パッケージと同じ `psitau` さん作; ソースに書かれた Windows の機種依存文字が `dvips`, `dvipdfmx` などで出力出来るようになる) が使えます。

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

BXJS クラスではここは和文ドライバの管轄。

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

BXJS クラスでは `geometry` パッケージがこの処理を行う。

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

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

```
312 \newif\ifbxjs@papersize
313 \bxjs@papersizetrue
314 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
315 \DeclareOption{papersize}{\bxjs@papersizetrue}
```

■英語化 オプション `english` を新設しました。

※`\if@english` は非ユニークで衝突耐性がない。

```
316 \newif\if@english
317 \@englishfalse
318 \DeclareOption{english}{\@englishttrue}
```

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

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

BXJS では当初から `bxjsreport` クラスが用意されている。

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

BXJS クラスでは、`nojslogo` を既定とする。

```
319 \newif\if@jslogo \@jslogofalse
320 \DeclareOption{jslogo}{\@jslogottrue}
321 \DeclareOption{nojslogo}{\@jslogofalse}
```

■複合設定オプション 密

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

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

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

※元の長さが 128 pt 以上の場合でも動作するように修正した。

```
322 \mathchardef\bxjs@isc@ll=128
323 \mathchardef\bxjs@isc@sl=259
324 \def\bxjs@isc@sl@h{65539 }
325 \def\bxjs@invscale#1#2{%
326   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
327   \ifdim\@tempdima<\bxjs@isc@ll\p@
328     \@tempcnta\@tempdima \multiply\@tempcnta\@cc@lvi
329     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cc@lvi
330   \else
331     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
332     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
333   \fi
334   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
335   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
```

```

336 \@tempdimb\@tempcnta\@ne
337 \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
338 \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
339 \@whiledim\@tempdimb<\@tempdimc\do{%
340 \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
341 \advance\@tempcntb\@ne \divide\@tempcntb\tw@
342 \ifdim #2\@tempcntb>\@tempdima
343 \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
344 \else \@tempdimb=\@tempcntb\@ne \fi}%
345 \xdef\bxjs@gtmpa{\the\@tempdimb}%
346 \endgroup #1=\bxjs@gtmpa\relax}

```

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

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

```

347 \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。

```

348 \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。

```

349 \def\bxjs@composite@proc{%
350 \bxjs@oldfontcommandstrue
351 \setkeys{bxjs}{ja=pandoc}%
352 \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…。

```

353 \def\bxjs@driver@opt{dvipdfmx}%
354 \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。

```

355 %\let\bxjs@engine@given\undefined

```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```

356 %\let\bxjs@engine@opt\undefined

```

エンジン明示指定のオプションの処理。

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

```

357 \DeclareOption{autodetect-engine}{%
358   \let\bxjs@engine@given=*}
359 \DeclareOption{latex}{%
360   \def\bxjs@engine@opt{latex}%
361   \let\bxjs@engine@given=n}
362 \DeclareOption{platex}{%
363   \def\bxjs@engine@opt{platex}%
364   \let\bxjs@engine@given=j}
365 \DeclareOption{uplatex}{%
366   \def\bxjs@engine@opt{uplatex}%
367   \let\bxjs@engine@given=u}
368 \DeclareOption{xelatex}{%
369   \def\bxjs@engine@opt{xelatex}%
370   \let\bxjs@engine@given=x}
371 \DeclareOption{pdflatex}{%
372   \def\bxjs@engine@opt{pdflatex}%
373   \let\bxjs@engine@given=p}
374 \DeclareOption{lualatex}{%
375   \def\bxjs@engine@opt{lualatex}%
376   \let\bxjs@engine@given=l}
377 \DeclareOption{platex-ng}{%
378   \def\bxjs@engine@opt{platex-ng}%
379   \let\bxjs@engine@given=g}
380 \DeclareOption{platex-ng*}{%
381   \def\bxjs@engine@opt{platex-ng*}%
382   \let\bxjs@platexng@nodrv=t%
383   \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。

```

384 %\let\bxjs@driver@given\@undefined
385 \let\bxjs@driver@@dvimode=0
386 \let\bxjs@driver@@dviplatform=1
387 \let\bxjs@driver@@pdfmode=2
388 \let\bxjs@driver@@xetex=3
389 \let\bxjs@driver@@dvips=4
390 \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。

```

391 %\let\bxjs@driver@opt\@undefined
392 \DeclareOption{dvips}{%
393   \def\bxjs@driver@opt{dvips}%
394   \let\bxjs@driver@given\bxjs@driver@@dvips}
395 \DeclareOption{dviout}{%
396   \def\bxjs@driver@opt{dviout}%
397   \let\bxjs@driver@given\bxjs@driver@@dvimode}
398 \DeclareOption{xdvi}{%

```

```

399 \def\bxjs@driver@opt{xdvi}%
400 \let\bxjs@driver@given\bxjs@driver@@dvimode}
401 \DeclareOption{dvipdfmx}{%
402 \def\bxjs@driver@opt{dvipdfmx}%
403 \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
404 \DeclareOption{nodvidriver}{%
405 \def\bxjs@driver@opt{nodvidriver}%
406 \let\bxjs@driver@given\bxjs@driver@@none}
407 \DeclareOption{pdftex}{%
408 \def\bxjs@driver@opt{pdftex}%
409 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
410 \DeclareOption{luatex}{%
411 \def\bxjs@driver@opt{luatex}%
412 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
413 \DeclareOption{xetex}{%
414 \def\bxjs@driver@opt{xetex}%
415 \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。

```

416 \DeclareOption{dvipdfmx-if-dvi}{%
417 \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。

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

`zw`、`noz` オプションの定義。

```

419 \DeclareOption{noz}{%
420 \bxjs@usezwfalse}
421 \DeclareOption{zw}{%
422 \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。

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

`nojs`、`js` オプションの定義。

```

424 \DeclareOption{nojs}{%
425 \bxjs@disguise@jsfalse}
426 \DeclareOption{js}{%
427 \bxjs@disguise@jstrue}

```

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

```
428 \newif\ifbxjs@precisetext
```

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

```

429 \DeclareOption{noprecisetext}{%
430   \bxjs@precisetextfalse}
431 \DeclareOption{precisetext}{%
432   \bxjs@precisetexttrue}

```

`\ifbxjs@simplejasetup` Xe_{La}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）。

```

433 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

```

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

```

434 \DeclareOption{nosimplejasetup}{%
435   \bxjs@simplejasetupfalse}
436 \DeclareOption{simplejasetup}{%
437   \bxjs@simplejasetuptrue}

```

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

```

438 \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。

```

439 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
440 \def\bxjs@tmpb#1 #2#3\@nil{%
441   \ifx1#2\bxjs@bigcodetrue \fi}
442 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil

```

`nobigcode` / `bigcode` オプションの定義。

```

443 \DeclareOption{nobigcode}{%
444   \bxjs@bigcodefalse}
445 \DeclareOption{bigcode}{%
446   \bxjs@bigcodetrue}

```

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

```

447 \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。

```

448 \DeclareOption{nooldfontcommands}{%
449   \bxjs@oldfontcommandsfalse}
450 \DeclareOption{oldfontcommands}{%
451   \bxjs@oldfontcommandstrue}

```

■keyval 型のオプション

```

452 \def\bxjs@setkey{%
453   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
454 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
455 \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`。

```

456 \def\bxjs@safe@setkeys#1#2{%
457   \let\bxjs@KV@errx\KV@errx
458   \let\KV@errx\bxjs@safe@setkeys@a
459   \setkeys{#1}{#2}%
460   \let\KV@errx\bxjs@KV@errx}
461 \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。

```

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

```

`\jsScale` 〔実数値マクロ〕和文スケール値。

```

473 \def\jsScale{0.924715}

```

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

```

474 %\let\bxjs@base@opt\@undefined

```

`base` オプションの処理。

```

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

```

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

```

479 %\let\bxjs@jbase@opt\@undefined

```

`jbase` オプションの処理。

```

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

```

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

```

482 %\let\bxjs@scale@opt\@undefined

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

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

```

`\bxjs@param@mag` `mag` オプションの値。

```

488 \let\bxjs@param@mag\relax

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

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

```

`\bxjs@jadriver` 和文ドライバの名前。

```

491 \let\bxjs@jadriver\relax
492 %\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）。
493 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
494 \define@key{bxjs}{ja}{\relax}{%
495   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}

```

`\jsJaFont` 和文フォント設定の名前。

```

496 \let\jsJaFont\@empty

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

```

`\jsJaParam` 和文ドライバパラメタの文字列。

```

498 \let\jsJaParam\@empty

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

```

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

```

500 \let\bxjs@magstyle@mag=m
501 \let\bxjs@magstyle@real=r
502 \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。

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

`\bxjs@magstyle@default` は既定の値を表す。

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

`magstyle` オプションの処理。

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

`\bxjs@geometry` `geometry` オプションの値。

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

`geometry` オプションの処理。

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

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

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

`fancyhdr` オプションの処理。

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



```

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

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

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

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

\bxjs@driver@given を未定義にしていることに注意。
551 \def\bxjs@driver@opt{#1}%
552 \let\bxjs@driver@given\undefined
553 \bxjs@dvi@opttrue
554 \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。
555 \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。
556 \newif\ifbxjs@force@chapterabstract
557 %<book>\bxjs@force@chapterabstracttrue

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

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

```

```

568 %\let\bxjs@textwidth@limit\@undefined
569 \define@key{bxjs}{textwidth-limit}{%
570   \edef\bxjs@textwidth@limit{#1}}

```

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

```

571 %\let\bxjs@textwidth@opt\@undefined
572 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
573 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

```

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

```

574 %\let\bxjs@number@of@lines@opt\@undefined
575 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
576 \define@key{bxjs}{number_of_lines}{\setkeys{bxjs}{number-of-lines=#1}}

```

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

```

577 %\let\bxjs@paragraph@mark\@undefined
578 \define@key{bxjs}{paragraph-mark}{%
579   \edef\bxjs@paragraph@mark{#1}}

```

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

```

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

```

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

```

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

```

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

```

589 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
590 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
591 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
592 \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。

```

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

```

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

```

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

```

```

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

```

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

```

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

```

617 \let\bxjs@ltx@removeelement\@removeelement
618 \def\@removeelement#1#2#3{%
619   \def\reserved@a{#2}%
620   \ifx\reserved@a\@empty \let#3\@empty
621   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
622   \fi}

```

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

```

623 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
624 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
625 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
626 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
627 \ProcessOptions\relax
628 \bxjs@composite@proc

```

後処理

```

629 \if@slide
630   \def\maybblue{\@ifundefined{ver@color.sty}{\color{blue}}}

```

```

631 \fi
632 \if@landscape
633   \setlength\@tempdima {\paperheight}
634   \setlength\paperheight{\paperwidth}
635   \setlength\paperwidth {\@tempdima}
636 \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。

```

637 \@onlypreamble\bxjs@purge@brace@elts
638 \def\bxjs@purge@brace@elts{%
639   \def\bxjs@tmpa{\gobble}%
640   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
641   \let\@classoptionslist\bxjs@tmpa}
642 \@onlypreamble\bxjs@purge@be@a
643 \def\bxjs@purge@be@a#1,{%
644   \ifx\@nil#1\relax\else
645     \bxjs@purge@be@b#1{\@nil
646     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
647     \expandafter\bxjs@purge@be@a
648   \fi}
649 \@onlypreamble\bxjs@purge@be@b
650 \def\bxjs@purge@be@b#1#2{\bxjs@purge@be@c}
651 \@onlypreamble\bxjs@purge@be@c
652 \def\bxjs@purge@be@c#1\@nil{%
653   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
654 \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。

```

655 \@expandtwoargs\@removeelement
656 {papersize}\@classoptionslist\@classoptionslist
657 \@expandtwoargs\@removeelement
658 {10pt}\@classoptionslist\@classoptionslist
659 \@expandtwoargs\@removeelement
660 {noscale}\@classoptionslist\@classoptionslist

```

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

[2016-11-09] pL^AT_EX/ upL^AT_EX を自動判別するオプション autodetect-engine を新設しました。upL^AT_EX の場合は、グローバルオプションに uplatex を追加することで、自動

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

このコードを削除。

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

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。

```
661 \ifx\bxjs@jadriver\relax\else
662   \let\bxjs@jadriver@given\bxjs@jadriver
663 \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。

```
664 \let\bxjs@tmpb\jsEngine
665 \ifx j\bxjs@tmpb\ifjsWithpTeXng
666   \let\bxjs@tmpb=g
667 \fi\fi
668 \ifx j\bxjs@tmpb\ifjsWithupTeX
669   \let\bxjs@tmpb=u
670 \fi\fi
671 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
672   \let\bxjs@tmpb=n
673 \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。）

```
674 \ifx *\bxjs@engine@given
675   \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。


```
676 \ifx j\bxjs@engine@given
677   \g@addto@macro\@classoptionslist{,latex}
678 \else\ifx u\bxjs@engine@given
679   \g@addto@macro\@classoptionslist{,uplatex}
680 \fi\fi
681 \fi
682 \ifx\bxjs@engine@given\@undefined\else
683   \ifx\bxjs@engine@given\bxjs@tmpb\else
684     \ClassError\bxjs{clsname
685       {Option '\bxjs@engine@opt' used on wrong engine}}\@ehc
686   \fi
687 \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。

```

688 \ifjsWithpTeXng
689   \g@addto@macro\@classoptionslist{,uplatex}
690 \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。

```

691 \@tempswatrue
692 \ifx \bxjs@driver@given\undefined\else
693   \ifjsInPdfMode
694     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
695       \@tempswafalse
696     \fi
697   \else\ifx x\jsEngine
698     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
699       \@tempswafalse
700     \fi
701   \else
702     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
703       \@tempswafalse
704     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
705       \@tempswafalse
706     \fi\fi
707   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
708     \@tempswafalse
709   \fi\fi
710 \fi\fi
711 \fi
712 \if@tempswa\else
713   \ClassError\bxjs@clsname
714     {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
715 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

716 \ifjsInPdfMode \@tempswafalse
717 \else\ifx x\jsEngine \@tempswafalse
718 \else\ifjsWithpTeXng \@tempswafalse
719 \else \@tempswatrue
720 \fi\fi\fi
721 \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)。

```

722   \ifx\bxjs@driver@opt\undefined \ifx\bxjs@jadriver@given\undefined\else
723     \ClassWarningNoLine\bxjs@clsname
724       {No driver option is given}
725   \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。)

```

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

```

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

```

735 \ifjsWithpTeXng
736   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
737     \let\bxjs@platexng@nodrv\@undefined
738   \else\ifx t\bxjs@platexng@nodrv\else
739     \g@addto@macro\@classoptionslist{,dvipdfmx}
740   \fi\fi
741 \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。

```

742 \ifx\bxjs@driver@given\bxjs@driver@@none
743   \bxjs@papersizefalse
744 \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。

```

745 \def\bxjs@@minimal{minimal}
746 \ifx\bxjs@jadriver\relax
747   \ifx j\jsEngine
748     \def\bxjs@jadriver{standard}
749   \else
750     \let\bxjs@jadriver\bxjs@@minimal
751   \fi
752 \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。

```

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

```

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

置き換えられる。)

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

base、jbase、scale の値を用いて和文スケール値を解決する。

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

```
769 \ifx\bxjs@jbase@opt\@undefined\else
770   \ifx\bxjs@base@opt\@undefined
```

jbase 指定済で base 未指定の場合は、\jsScale の値を採用して和文基底サイズを決定する。

```
771   \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
772   \bxjs@invscale\@tempdima\jsScale
773   \bxjs@setbasefontsize{\@tempdima}%
774 \else
```

jbase と base がともに指定済の場合は、それらの値から和文スケール値を決定する。

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

\Cjascale 和文クラス共通仕様（※ただし ZR 氏提唱）における、和文スケール値の変数。

```
786 \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。)

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



```
791 \repeat
792 \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。

```
793 \ifbxjs@disguise@js
794 %<book|report>\def\bxjs@js@clsname{jsbook}
795 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
796 \@namedef{ver@bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
797 \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。

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

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

```
799 \ifbxjs@oldfontcommands
800 \AtEndOfClass{\allowoldfontcommands}
801 \fi
```

■papersize スペシャルの出力 dvi ファイルの先頭に dvips の papersize special を書き込むことで、出力用紙サイズを設定します。これは dvipdfmx や最近の dviout にも有効です。どうやら papersize special には true 付の単位は許されず、かつ単位は常に true なものと扱われるようです。そこで、後で出てくる（☆）の部分、「\mag にあわせてスケール」よりも手前で実行しておくことになります。

トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが 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。

```
802 \newdimen\stockwidth \newdimen\stockheight
803 \begingroup\expandafter\expandafter\expandafter\endgroup
```

```

804 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
805 % \newdimen\stockwidth \newdimen\stockheight
806 \setlength{\stockwidth}{\paperwidth}
807 \setlength{\stockheight}{\paperheight}
808 \advance \stockwidth 2in
809 \advance \stockheight 2in
810 \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。

```

811 %<slide>\def\n@baseline{13}%
812 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
813 %<!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。

```

814 \ifx\bxjs@magstyle\bxjs@magstyle@mag
815 \jsc@magtrue
816 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
817 \jsc@mag@xrealtrue
818 \fi\fi

```

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

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

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

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

```

819 \ifx\bxjs@param@mag\relax
820 \@tempdima=\bxjs@param@basefontsize
821 \advance\@tempdima.001pt \multiply\@tempdima25
822 \divide\@tempdima16384\relax \@tempcmta=\@tempdima\relax
823 \edef\bxjs@param@mag{\the\@tempcmta}
824 \else
825 % mag 値が直接指定された場合
826 \bxjs@gset@tempcmta{\bxjs@param@mag}
827 \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
828 % 有効な mag 値の範囲は 1--32768

```

```

829 \edef\bxjs@param@mag{\the\@tempcnta}
830 \advance\@tempcnta100000
831 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
832 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
833 \edef\bxjs@param@basefontsize{\the\@tempdima}
834 \fi
835 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
836 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
837 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
838 \edef\jsc@magscale{\strip@pt\@tempdima}
839 \let\jsBaseFontSize\bxjs@param@basefontsize
840 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
841 %\typeout{\string\jsEngine: \meaning\jsEngine}
842 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
843 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
844 %\typeout{\string\jsc@magscale: \jsc@magscale}
845 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
846 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}

```

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

\mag する場合（現状はこれが既定）にコードの変更を低減するために、以下では必要に応じて、\jsc@mpt を \p@? と書く。その上で、\mag する場合は ? を無視して \p@ と解釈させ、\mag しない場合は ? を英字扱いにして \p@? という制御綴を \jsc@mpt と同値にする。※（多分 2.0 版あたりで）JS クラスに合わせるため \p@? 表記を止める予定。

```

847 \newdimen\jsc@mpt
848 \newdimen\jsc@mmm
849 \ifjsc@mag
850 \jsc@mpt=1\p@
851 \jsc@mmm=1mm
852 \catcode\?=9 % \p@? read as \p@
853 \else
854 \jsc@mpt=\jsc@magscale\p@
855 \jsc@mmm=\jsc@magscale mm
856 \catcode\?=11 \let\p@?\jsc@mpt
857 \fi
858 \chardef\bxjs@qmcc=\catcode\?\relax

```

ここで p_TE_X 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_w 非指定時は \zw を \jsZw と同義にする。

```

859 \newdimen\jsZw
860 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
861 \ifbxjs@usezw
862 \providecommand*\zw{\jsZw}

```

863 \fi

\zwspace 全角幅の水平空き。

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

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

865 \ifjsc@mag@xreal

866 \RequirePackage{type1cm}

867 \let\jsc@invscale\bxjs@invscale

ムニャムニャムニャ……。

868 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax

869 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

870 \let\jsc@get@external@font\get@external@font

871 \def\get@external@font{%

872 \jsc@preadjust@extract@font

873 \jsc@get@external@font}

874 \def\jsc@fstrunc#1{%

875 \edef\jsc@tmpa{\strip@pt#1}%

876 \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}

877 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%

878 \if#5*\else

879 \edef\jsc@tmpa{#1%

880 \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%

881 \fi}

882 \def\jsc@preadjust@extract@font{%

883 \let\jsc@req@size\f@size

884 \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale

885 \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@

886 \let\jsc@ref@size\jsc@tmpa

887 \let\f@size\jsc@ref@size}

888 \def\execute@size@function#1{%

889 \let\jsc@cref@size\f@size

890 \let\f@size\jsc@req@size

891 \csname s@fct@#1\endcsname}

892 \let\jsc@DeclareErrorFont\DeclareErrorFont

893 \def\DeclareErrorFont#1#2#3#4#5{%

894 \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc

895 \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}

896 \expandafter\jsc@DeclareErrorFont\jsc@tmpa}

897 \def\gen@sfcnt{%

898 \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%

899 \empty@sfcnt}

900 \def\genb@sfcnt{%

901 \edef\mandatory@arg{%

902 \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@{}}%

903 \empty@sfcnt}

904 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}

905 \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 906 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
```

```
\jsc@bigskip 907 %\def\jsc@medskip{\vspace\jsc@medskipamount}
```

```
908 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}
```

```
\jsc@smallskipamount
```

```
\jsc@medskipamount 909 \newskip\jsc@smallskipamount
```

```
\jsc@bigskipamount 910 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
```

```
911 %\newskip\jsc@medskipamount
```

```
912 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
```

```
913 %\newskip\jsc@bigskipamount
```

```
914 %\jsc@bigskipamoun =12\jsc@mpt plus 4\jsc@mpt minus 4\jsc@mpt
```

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

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

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

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

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

```
915 % \ifpapersize
```

```
916 % \setlength{\@tempdima}{\paperwidth}
```

```
917 % \setlength{\@tempdimb}{\paperheight}
```

```
918 % \iftombow
```

```
919 % \advance \@tempdima 2truein
```

```
920 % \advance \@tempdimb 2truein
```

```
921 % \fi
```

```
922 % \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
```

```
923 % \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 L^AT_EX で、auxiliary files に書き出されたときにスペースが食われないようにする修正が入りました。これに合わせて `{}` を補いました。

BXJS クラスでの変更点：

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

```
924 \chardef\bxjs@periodchar=`\.  
925 \bxjs@protected\def\bxjs@SE{%  
926   \ifnum\spacefactor<\@m \spacefactor\@m  
927   \else \spacefactor\sfcode\bxjs@periodchar  
928   \fi}  
929 \ifbxjs@fix@at@cmd  
930   \def\@{\bxjs@SE{}}  
931 \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>\@vipt</code>	7
<code>\@viipt</code>	8	<code>\@ixpt</code>	9	<code>\@xpt</code>	10
<code>\@xipt</code>	10.95	<code>\@xipt</code>	12	<code>\@xivpt</code>	14.4

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

`\kanjiskip` は pL^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。

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

```
932 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
933 \expandafter\bxjs@tmpa\expandafter{%
934   \set@fontsize{#1}{#2}{#3}%
935 % 末尾にコードを追加
936   \expandafter\def\expandafter\size@update\expandafter{%
937     \size@update
938     \jsFontSizeChanged}%
939 }
```

`\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。

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

`\jsResetDimen` ユーザ定義用のフック。

```
949 \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` の変わりに用いることにします。

```
950 \ifjsc@mag
951   \let\jsc@setfontsize\@setfontsize
952 \else
953   \def\jsc@setfontsize#1#2#3{%
954     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
955 % microtype 対策
956 \ifjsWithTeX\if j\jsEngine\else
957   \def\jsc@setfontsize#1#2#3{%
```

```

958 \edef\bxjs@sfs@next{%
959 \unexpanded{\@setfontsize#1}%
960 {\the\dimexpr#2\jsc@mpt\relax}{\the\dimexpr#3\jsc@mpt\relax}%
961 }\bxjs@sfs@next}
962 \fi\fi
963 \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。

```

964 \emergencystretch 3\jsZw

```

```

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

```

別行立て数式に入るときに `\narrowbaselines` が呼ばれるが、このコードでは「数式中で `\normalsize` などのサイズ命令 (`\@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。

```

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

```



```

982 \belowdisplayskip=\skip4
983 \belowdisplayshortskip=\skip6\relax}
984 \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。

```

985 \def\bxjs@if@narrowbaselines{%
986   \ifnarrowbaselines\expandafter\@firstoftwo
987   \else \expandafter\@secondoftwo
988   \fi
989 }

```

\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。

```

990 \renewcommand{\normalsize}{%
991   \bxjs@if@narrowbaselines{%
992     \jsc@setfontsize\normalsize\@xpt\@xipt
993   }{%else
994     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
995   }%

```

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。

```

996 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
997 \abovedisplayshortskip \z@ \@plus3\p@?
998 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
999 \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。

```

1000 \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。

```
1001 \normalsize
```

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

```
\Chs
```

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

```
1002 \ifx\Cht\undefined \newdimen\Cht \fi
1003 \ifx\Cdp\undefined \newdimen\Cdp \fi
1004 \ifx\Cwd\undefined \newdimen\Cwd \fi
1005 \ifx\Cvs\undefined \newdimen\Cvs \fi
1006 \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。

```
1007 \setlength\Cht{0.88\jsZw}
1008 \setlength\Cdp{0.12\jsZw}
1009 \setlength\Cwd{1\jsZw}
1010 \setlength\Cvs{\baselineskip}
1011 \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。

```
1012 \newcommand{\small}{%
1013   \bxjs@if@narrowbaselines{%
1014     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1015     %<kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1016   }{%else
1017     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1018     %<kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1019   }%
1020   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1021   \abovedisplayshortskip \z@ \@plus3\p@?
1022   \belowdisplayskip \abovedisplayskip
1023   \belowdisplayshortskip \belowdisplayskip
1024   \def\@listi{\leftmargin\leftmargini
1025               \topsep \z@
1026               \parsep \z@
```

1027 \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。

```

1028 \newcommand{\footnotesize}{%
1029   \bxjs@if@narrowbaselines{%
1030     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1031     %<kiyou>    \jsc@setfontsize\footnotesize{8.888}{11}%
1032   }{%else
1033     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1034     %<kiyou>    \jsc@setfontsize\footnotesize{8.888}{13.2418}%
1035   }%
1036   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1037   \abovedisplayshortskip \z@ \@plus2\p@?
1038   \belowdisplayskip \abovedisplayskip
1039   \belowdisplayshortskip \belowdisplayskip
1040   \def\@listi{\leftmargin\leftmargini
1041             \topsep \z@
1042             \parsep \z@
1043             \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 1044 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1045 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 1046 \if@twocolumn
1047 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1048 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1049 \else
1050 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
1051 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1052 \fi
1053 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1054 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1055 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1056 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1057 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1058 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}
```

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

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

```
1059 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}
```

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

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

```
1060 % \newcommand{\headfont}{\bfseries}
```

```
1061 \newcommand{\headfont}{\sffamily}
```

```
1062 % \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。

```
1063 %!kiyou>\setlength\columnsep{2\Cwd}
```

```
1064 %!kiyou>\setlength\columnsep{28truebp}
```

```
1065 \setlength\columnseprule{\z@}
```

■段落

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

```
\lineskiplimit 1066 \setlength\lineskip{1\jsc@mp}
```

```
1067 \setlength\normallineskip{1\jsc@mp}
```

```
\normallineskiplimit 1068 \setlength\lineskiplimit{1\jsc@mp}
```

```
1069 \setlength\normallineskiplimit{1\jsc@mp}
```

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

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

とすると、行送りが通常の 2 倍になります。ただし、これを設定すると、たとえ `\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。

```
1070 \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。

```
1071 \setlength\parskip{\z@}
```

```
1072 \if@slide
```

```

1073 \setlength\parindent{0\p@}
1074 \else
1075 \setlength\parindent{1\Cwd}
1076 \fi

```

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

```

\@highpenalty 1077 \@lowpenalty 51
1078 \@medpenalty 151
1079 \@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。

```
1080 % \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。

```
1081 % \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。

```

1082 \ifjsc@mag
1083 \mag=\bxjs@param@mag
1084 \fi
1085 \setlength{\topskip}{10\p@?}

```

`\jsSetQHLength` のための和文単位の定義。

```

1086 \def\bxjs@unit@trueQ{0.25truemm}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1087 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw

```

`\bxjs@param@paper` が長さ指定 (`{W}{H}`) の場合、`geometry` の形式 (`papersize={W,H}`) に変換する。

```

1088 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
1089 \def\bxjs@read@b{%
1090   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
1091   \else \expandafter\bxjs@read@d \fi}
1092 \def\bxjs@read@c#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1093 \def\bxjs@read@d#1\@nil{\bxjs@read@e#1,,\@nil}
1094 \def\bxjs@read@e#1,#2,#3\@nil{%
1095   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1096 \expandafter\bxjs@read@a\bxjs@param@paper\@empty\@empty\@nil

```

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

```
1097 \edef\bxjs@layout@paper{%
1098   \ifjsc@mag truedimen,\fi
1099   \if@landscape landscape,\fi
1100   \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。

```
1101 %<*article|report>
1102 \def\bxjs@layout@base{%
1103   headheight=\topskip,footskip=0.03367\paperheight,%
1104   headsep=\footskip-\topskip,includeheadfoot,%
1105 }
1106 \edef\bxjs@layout{\bxjs@layout@base
1107   hscale=0.76,hmarginratio=1:1,%
1108   vscale=0.83,vmarginratio=1:1,%
1109 }
1110 %</article|report>
1111 %<*book>
1112 \def\bxjs@layout@base{%
1113   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1114 }
1115 \ifbxjs@layout@buggyhmargin    %---
1116 % アレ
1117 \edef\bxjs@layout{\bxjs@layout@base
1118   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1119   vscale=0.83,vmarginratio=1:1,%
1120 }
1121 \else    %---
1122 % 非アレ
1123 \edef\bxjs@layout{\bxjs@layout@base
1124   hmargin=18\jsc@mmm,%
1125   vscale=0.83,vmarginratio=1:1,%
1126 }
1127 \fi    %---
1128 %</book>
1129 %<*slide>
1130 \def\bxjs@layout@base{%
1131   noheadfoot,%
1132 }
1133 \edef\bxjs@layout{\bxjs@layout@base
1134   hscale=0.9,hmarginratio=1:1,%
1135   vscale=0.944,vmarginratio=1:1,%
1136 }
1137 %</slide>
```

textwidth オプションの設定を反映する。

```
1138 %<!*book>
1139 \ifx\bxjs@textwidth@opt\undefined\else
1140   \jsSetQHLlength\@tempdima{\bxjs@textwidth@opt}
```

```

1141 \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1142 \fi
1143 %<*/book>
1144 \ifx\bxjs@number@of@lines@opt\@undefined\else
1145 \bxjs@set@tempcnta{\bxjs@number@of@lines@opt}
1146 \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1147 \fi

```

\fullwidth 〔寸法レジスタ〕ヘッダ・フッタ領域の横幅。

```

1148 \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。

```

1149 %<*/book>
1150 \newcommand\jsTextWidthLimit{40}
1151 \ifx\bxjs@textwidth@limit\@undefined\else
1152 \bxjs@set@tempcnta{\bxjs@textwidth@limit}
1153 \long\edef\jsTextWidthLimit{\the\@tempcnta}
1154 \fi
1155 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

1156 \def\bxjs@postproc@layout{%
    geometry のドライバを再設定する。
1157 \ifx\bxjs@geometry@driver\relax\else
1158 \let\Gm@driver\bxjs@geometry@driver
1159 \fi
    \textwidth の値を補正する。
1160 \ifbxjs@whole@zw@lines
1161 \@tempdimb=\textwidth
1162 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1163 \advance\textwidth.005pt\relax
1164 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1165 \advance\@tempdimb-\textwidth
1166 \advance\oddsidemargin 0.5\@tempdimb
1167 \advance\evensidemargin 0.5\@tempdimb
1168 \fi
1169 \fullwidth=\textwidth

```

bxjsbook の場合は、geometry が設定した \textwidth は \fullwidth として扱い、その値から実際の \textwidth を導出する。

```

1170 %<*/book>
1171 \@tempdima=\jsTextWidthLimit\Cwd
1172 \ifx\bxjs@textwidth@opt\@undefined\else
1173 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1174 \ifbxjs@whole@zw@lines
1175 \advance\@tempdima.005pt\relax
1176 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1177 \fi

```

```

1178 \fi
1179 \ifdim\textwidth>\@tempdima
1180 \textwidth=\@tempdima
1181 \addtolength\evensidemargin{\fullwidth-\textwidth}
1182 \fi
1183 %</book>

\textheight 関連の調整。
1184 \@tempdimb=\textheight
1185 \advance\textheight-\topskip
1186 \advance\textheight.005pt\relax
1187 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1188 \advance\textheight\topskip
1189 \advance\@tempdimb-\textheight
1190 \advance\topmargin0.5\@tempdimb

\headheight 関連の調整。
1191 \@tempdima=\topskip
1192 \advance\headheight\@tempdima
1193 \advance\topmargin-\@tempdima

marginpar 関連の調整。
1194 \setlength\marginparsep{\columnsep}
1195 \setlength\marginparpush{\baselineskip}
1196 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1197 -\textwidth-10\jsc@mmm-\marginparsep}
1198 \ifbxjs@whole@zw@lines
1199 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1200 \fi

連動する変数。
1201 \maxdepth=.5\topskip
1202 \stockwidth=\paperwidth
1203 \stockheight=\paperheight
1204 }

```

`\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。

```

1205 \edef\jsGeometryOptions{%
1206 \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージ読込

geoemtry オプションの値に応じて分岐する。

まずは `geometry=class` の場合。

```

1207 \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。

```
1208 \def\bxjs@geometry@guard@on{%
1209   \edef\bxjs@gg@tmp{\the\ht\strutbox}\ht\strutbox=10\jsc@empt}
1210 \def\bxjs@geometry@guard@off{%
1211   \ht\strutbox=\bxjs@gg@tmp\relax}
```

geometry のドライバオプション指定。nopapersize 指定時は、special 命令出力を抑止するためにドライバを none にする。そうでない場合は、クラスで指定したドライバオプションが引き継がれるので何もしなくてよいが、例外として、ドライバが dvipdfmx の時は、現状の geometry は dvipdfm を指定する必要がある。

```
1212 \ifbxjs@papersize
1213   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1214     \PassOptionsToPackage{dvipdfm}{geometry}
1215   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1216     \PassOptionsToPackage{dvipdfm}{geometry}
1217   \fi\fi
1218   \let\bxPapersizeSpecialDone=t
1219 \else
1220   \PassOptionsToPackage{driver=none}{geometry}
1221 \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。

```
1222 \AtBeginDocument{\bxjs@pre@geometry@hook}
1223 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1224 \bxjs@geometry@guard@on
1225 \edef\bxjs@next{%
1226   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1227 }\bxjs@next
1228 \bxjs@geometry@guard@off
1229 \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。

```
1230 \let\bxjs@geometry@driver\Gm@driver
1231 \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。

```
1232 \@onlypreamble\bxjs@pre@geometry@hook
1233 \def\bxjs@pre@geometry@hook{%
1234   \@ifpackageloaded{geometry}{%
1235     \@ifpackagelater{geometry}{2010/02/12}{-}{%else
1236       \PackageWarningNoLine\bxjs@clsname
```

```

1237     {The 'geometry' package installed\MessageBreak
1238       is too old (< v5.0)}}%
1239     \if x\jsEngine \ifnum\mag=\@m\else
1240       \def\bxjs@Gm@driver{pdftex}
1241       \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1242       \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1243     \fi\fi
1244   }%
1245   \ifjsWithpTeXng
1246     \ifx\Gm@driver\@empty
1247       \def\Gm@driver{pdftex}%
1248     \fi
1249   \fi
1250 }{}

```

`\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。

```

1251 \def\setpagelayout{%
1252   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1253     \@ifstar{\bxjs@setpagelayout@a@one}{\bxjs@setpagelayout@a@z@}}
1254 \def\bxjs@setpagelayout@a#1#2{%
1255   \ifcase#1% modify
1256     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1257   \or% reset(*)
1258     \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1259   \or% semireset(+)
1260     \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1261   \fi
1262   \bxjs@geometry@guard@on
1263   \edef\bxjs@next{%
1264     \noexpand\geometry{\bxjs@next}%
1265   }\bxjs@next
1266   \bxjs@geometry@guard@off
1267   \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。

```

1268 \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。

```

1269 \AtBeginDocument{\bxjs@check@page@layout}

```

```

1270 \@onlypreamble\bxjs@check@page@layout
1271 \def\bxjs@check@page@layout{%
1272   \ifdim\textwidth=.5\maxdimen
1273     \ClassError\bxjs@clsname
1274       {Page layout is not properly set}%
1275       {\@ehd}
1276   \fi}
1277 \def\jsUseMinimalPageLayout{%
1278   \setlength{\textwidth}{6.5in}%
1279   \setlength{\textheight}{8in}}

\setpagelayout はとりあえず無効にしておく。
1280 \let\bxjs@geometry@driver\relax
1281 \def\setpagelayout{%
1282   \bxjs@ifplus{\bxjs@pagelayout@a}{\%else
1283     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1284 \def\bxjs@pagelayout@a#1{%
1285   \ClassError\bxjs@clsname
1286     {Command '\string\setpagelayout' is not supported,\MessageBreak
1287     because 'geometry' value is not 'class'}\@eha}

geometry=user の場合の処理はここで終わり。
1288 \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。

```
1289 %<*\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。

```
1290 \setlength\topskip{1.38zw}%% from 10\jsc@mp (2016-08-17)
```

```

1291 \if@slide
1292   \setlength\headheight{0\jsc@empt}
1293 \else
1294   \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1295 \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³が、ここでは A4 判のときちょうど 1cm となるように、`\paperheight` の 0.03367 倍 (最小 `\baselineskip`) としました。書籍については、フッタは使わないことにして、ゼロにしました。

```

1296 %<*article|kiyou>
1297 \if@slide
1298   \setlength\footskip{0pt}
1299 \else
1300   \setlength\footskip{0.03367\paperheight}
1301   \ifdim\footskip<\baselineskip
1302     \setlength\footskip{\baselineskip}
1303   \fi
1304 \fi
1305 %</article|kiyou>
1306 %<jspf>\setlength\footskip{9\jsc@mmm}
1307 %<*book>
1308 \if@report
1309   \setlength\footskip{0.03367\paperheight}
1310   \ifdim\footskip<\baselineskip
1311     \setlength\footskip{\baselineskip}
1312   \fi
1313 \else
1314   \setlength\footskip{0pt}
1315 \fi
1316 %</book>
1317 %<*report>
1318 \setlength\footskip{0.03367\paperheight}
1319 \ifdim\footskip<\baselineskip
1320   \setlength\footskip{\baselineskip}
1321 \fi
1322 %</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)。

```

1323 %<*article>
1324 \if@slide
1325   \setlength\headsep{0\jsc@empt}

```

```

1326 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1327 \addtolength\headsep{10\jsc@mp}%% added (2016-10-08)
1328 \else
1329 \setlength\headsep{\footskip}
1330 \addtolength\headsep{-\topskip}
1331 \fi
1332 %</article>
1333 %<*book>
1334 \if@report
1335 \setlength\headsep{\footskip}
1336 \addtolength\headsep{-\topskip}
1337 \else
1338 \setlength\headsep{6\jsc@mm}
1339 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1340 \addtolength\headsep{10\jsc@mp}%% added (2016-10-08)
1341 \fi
1342 %</book>
1343 %<*report>
1344 \setlength\headsep{\footskip}
1345 \addtolength\headsep{-\topskip}
1346 %</report>
1347 %<*jspf>
1348 \setlength\headsep{9\jsc@mm}
1349 \addtolength\headsep{-\topskip}
1350 %</jspf>
1351 %<*kiyou>
1352 \setlength\headheight{0\jsc@mp}
1353 \setlength\headsep{0\jsc@mp}
1354 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1355 \addtolength\headsep{10\jsc@mp}%% added (2016-10-08)
1356 %</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。

```

1357 \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。

```

1358 \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。

```
1359 %<*article>
1360 \if@slide
1361   \setlength\fullwidth{0.9\paperwidth}
1362 \else
1363   \setlength\fullwidth{0.76\paperwidth}
1364 \fi
1365 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1366 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1367 \setlength\textwidth{\fullwidth}
1368 %</article>
1369 %<*book>
1370 \if@report
1371   \setlength\fullwidth{0.76\paperwidth}
1372 \else
1373   \setlength\fullwidth{\paperwidth}
1374   \addtolength\fullwidth{-36\jsc@mmm}
1375 \fi
1376 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1377 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1378 \setlength\textwidth{\fullwidth}
1379 \if@report \else
1380   \if@twocolumn \else
1381     \ifdim \fullwidth>40zw
1382       \setlength\textwidth{40zw}
1383     \fi
1384   \fi
1385 \fi
1386 %</book>
1387 %<*report>
1388 \setlength\fullwidth{0.76\paperwidth}
1389 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1390 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1391 \setlength\textwidth{\fullwidth}
1392 %</report>
1393 %<*jspf>
1394 \setlength\fullwidth{50zw}
1395 \addtolength\fullwidth{8\jsc@mmm}
1396 \setlength\textwidth{\fullwidth}
1397 %</jspf>
1398 %<*kiyou>
1399 \setlength\fullwidth{48zw}
1400 \addtolength\fullwidth{\columnsep}
```

```

1401 \setlength\textwidth{\fullwidth}
1402 %</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)。

```

1403 %<*article|book|report>
1404 \if@slide
1405   \setlength{\textheight}{0.95\paperheight}
1406 \else
1407   \setlength{\textheight}{0.83\paperheight}
1408 \fi
1409 \addtolength{\textheight}{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1410 \addtolength{\textheight}{-\headsep}
1411 \addtolength{\textheight}{-\footskip}
1412 \addtolength{\textheight}{-\topskip}
1413 \divide\textheight\baselineskip
1414 \multiply\textheight\baselineskip
1415 %</article|book|report>
1416 %<jspf>\setlength{\textheight}{51\baselineskip}
1417 %<kiyou>\setlength{\textheight}{47\baselineskip}
1418 \addtolength{\textheight}{\topskip}
1419 \addtolength{\textheight}{0.1\jsc@empt}
1420 %<jspf>\setlength{\mathindent}{10\jsc@mmm}

```

`\flushbottom` [2016-07-18] `\textheight` に念のため 0.1 ポイント余裕を持たせているのと同様に、`\flushbottom` にも余裕を持たせます。元の L^AT_EX 2_ε での完全な `\flushbottom` の定義は

```

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

```

ですが、次のようにします。

```

1421 \def\flushbottom{%

```

```

1422 \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1423 \let\@texttop\relax}

```

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

```

1424 \setlength\marginparsep{\columnsep}
1425 \setlength\marginparpush{\baselineskip}

```

`\oddsidemargin` それぞれ奇数ページ、偶数ページの左マージンから 1 インチ引いた値です。片面印刷では `\evensidemargin` `\oddsidemargin` が使われます。T_EX は上・左マージンに `1truein` を挿入しますが、トンボ関係のオプションが指定されると pL_AT_EX 2_ε (`plcore.ltx`) はトンボの内側に `1in` のスペース (`1truein` ではなく) を挿入するので、場合分けしています。

```

1426 \setlength{\oddsidemargin}{\paperwidth}
1427 \addtolength{\oddsidemargin}{-\fullwidth}
1428 \setlength{\oddsidemargin}{.5\oddsidemargin}
1429 \iftombow
1430 \addtolength{\oddsidemargin}{-1in}
1431 \else
1432 \addtolength{\oddsidemargin}{-\inv@mag in}
1433 \fi
1434 \setlength{\evensidemargin}{\oddsidemargin}
1435 \if@mparswitch
1436 \addtolength{\evensidemargin}{\fullwidth}
1437 \addtolength{\evensidemargin}{-\textwidth}
1438 \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。

```

1439 \setlength\marginparwidth{\paperwidth}
1440 \addtolength\marginparwidth{-\oddsidemargin}
1441 \addtolength\marginparwidth{-\inv@mag in}
1442 \addtolength\marginparwidth{-\textwidth}
1443 \addtolength\marginparwidth{-10\jsc@mmm}
1444 \addtolength\marginparwidth{-\marginparsep}
1445 \@tempdima=1zw
1446 \divide\marginparwidth\@tempdima
1447 \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)。

```

1448 \setlength\topmargin{\paperheight}

```



```

1449 \addtolength\topmargin{-\textheight}
1450 \if@slide
1451   \addtolength\topmargin{-\headheight}
1452 \else
1453   \addtolength\topmargin{-10\jsc@empt}}%% from -\topskip (2016-10-08); from -
      \headheight (2003-06-26)
1454 \fi
1455 \addtolength\topmargin{-\headsep}
1456 \addtolength\topmargin{-\footskip}
1457 \setlength\topmargin{0.5\topmargin}
1458 %<kiyou>\setlength\topmargin{81truebp}
1459 \iftombow
1460   \addtolength\topmargin{-1in}
1461 \else
1462   \addtolength\topmargin{-\inv@mag in}
1463 \fi
1464 %</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。

```
1465 \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。

```
1466 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

■フロート関連 フロート (図, 表) 関連のパラメータは $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_2\epsilon$ 本体で定義されていますが、ここで設定変更します。本文ページ (本文とフロートが共存するページ) とフロートだけのページで設定が異なります。ちなみに、カウンタは内部では `\c@` を名前に冠したマクロになっています。

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

[2003-08-23] ちょっと増やしました。

```
1467 \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。

1468 `\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。

1469 `\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。

1470 `\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。

1471 `\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。

1472 `\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。

1473 `\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。

1474 `\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。

1475 `\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。

1476 `\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。

1477 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

1478 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`

1479 `\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` 1480 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

1481 `\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`

```

1482 \setlength\@fptop{0\p@? \@plus 1fil}
1483 \setlength\@fpsep{8\p@? \@plus 2fil}
1484 \setlength\@fpbot{0\p@? \@plus 1fil}

```

`\@dblftop` 段抜きフロートについての値です。

```

\@dblfpsep 1485 \setlength\@dblftop{0\p@? \@plus 1fil}
\@dblfpbot 1486 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1487 \setlength\@dblfpbot{0\p@? \@plus 1fil}

```

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

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

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

```

1488 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1489 % \ifodd\c@page
1490 % \iftdir
1491 % \hbox{}\thispagestyle{empty}\newpage
1492 % \if@twocolumn\hbox{}\newpage\fi
1493 % \fi
1494 % \else
1495 % \ifydir
1496 % \hbox{}\thispagestyle{empty}\newpage
1497 % \if@twocolumn\hbox{}\newpage\fi
1498 % \fi
1499 % \fi\fi}
1500 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1501 % \ifodd\c@page
1502 % \ifydir
1503 % \hbox{}\thispagestyle{empty}\newpage
1504 % \if@twocolumn\hbox{}\newpage\fi
1505 % \fi
1506 % \else
1507 % \iftdir
1508 % \hbox{}\thispagestyle{empty}\newpage
1509 % \if@twocolumn\hbox{}\newpage\fi
1510 % \fi
1511 % \fi\fi}
1512 \def\pltx@cleartooddpage{\clearpage\if@twoside
1513 \ifodd\c@page\else
1514 \hbox{}\thispagestyle{empty}\newpage

```

```

1515 \if@twocolumn\hbox{}\newpage\fi
1516 \fi\fi}
1517 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1518 \ifodd\c@page
1519 \hbox{}\thispagestyle{empty}\newpage
1520 \if@twocolumn\hbox{}\newpage\fi
1521 \fi\fi}

```

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

```

1522 \let\pltx@cleartorightpage\pltx@cleartooddpage
1523 \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。

```

1524 %<*book|report>
1525 \if@openleft
1526 \let\cleardoublepage\pltx@cleartoleftpage
1527 \else\if@openright
1528 \let\cleardoublepage\pltx@cleartorightpage
1529 \fi\fi
1530 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

```

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

```

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

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

`\ps@empty` `empty` ページスタイルの定義です。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。

```
1531 % \def\ps@empty{%
1532 %   \let\mkboth\gobbletwo
1533 %   \let\@oddhead\empty
1534 %   \let\@oddfoot\empty
1535 %   \let\@evenhead\empty
1536 %   \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。

```
1537 \def\ps@plainfoot{%
1538   \let\mkboth\gobbletwo
1539   \let\@oddhead\empty
1540   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1541   \let\@evenhead\empty
1542   \let\@evenfoot\@oddfoot}
1543 \def\ps@plainhead{%
1544   \let\mkboth\gobbletwo
1545   \let\@oddfoot\empty
1546   \let\@evenfoot\empty
1547   \def\@evenhead{%
1548     \ifmparswitch \hss \fi
1549     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1550     \ifmparswitch\else \hss \fi}%
1551   \def\@oddhead{%
1552     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1553 %<book>\let\ps@plain\ps@plainhead
1554 %<!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。

```
1555 %<*article|slide>
1556 \if@twoside
1557   \def\ps@headings{%
1558     \let\@oddfoot\empty
1559     \let\@evenfoot\empty
1560     \def\@evenhead{\ifmparswitch \hss \fi
1561       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1562     }
```

```

1562     \if@mparswitch\else \hss \fi}%
1563 \def\@oddhead{%
1564     \underline{%
1565         \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1566 \let\@mkboth\markboth
1567 \def\sectionmark##1{\markboth{%
1568     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1569     ##1}}}%
1570 \def\subsectionmark##1{\markright{%
1571     \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1572     ##1}}}%
1573 }
1574 \else % if not twoside
1575 \def\ps@headings{%
1576     \let\@oddfoot\@empty
1577     \def\@oddhead{%
1578         \underline{%
1579             \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1580     \let\@mkboth\markboth
1581     \def\sectionmark##1{\markright{%
1582         \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1583         ##1}}}%
1584 \fi
1585 %</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ため、「\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。

```

1586 %<*book|report>
1587 \def\bxjs@maybe@autoxspacing{%
1588     \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1589 \newif\if@omit@number
1590 \def\ps@headings{%
1591     \let\@oddfoot\@empty
1592     \let\@evenfoot\@empty
1593     \def\@evenhead{%
1594         \if@mparswitch \hss \fi
1595         \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1596             \textbf{\thepage}\hfil\leftmark}}}%
1597         \if@mparswitch\else \hss \fi}%
1598     \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1599         {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1600     \let\@mkboth\markboth
1601     \def\chaptermark##1{\markboth{%
1602         \ifnum \c@secnumdepth >\@m@ne

```

```

1603     \if@mainmatter
1604     \if@omit@number\else
1605         \@chapapp\thechapter\@chappos\hskip1\jsZw
1606     \fi
1607     \fi
1608     \fi
1609     ##1}{}}%
1610 \def\sectionmark##1{\markright{%
1611     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1612     ##1}}}%
1613 %</book|report>

```

最後は学会誌の場合です。

```

1614 %<*jspf>
1615 \def\ps@headings{%
1616     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1617     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1618     \def\@oddhead{\normalfont\hfil \@title \hfil}
1619     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1620 %</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。

```

1621 \def\ps@myheadings{%
1622     \let\@oddfoot\@empty\let\@evenfoot\@empty
1623     \def\@evenhead{%
1624         \if@mparswitch \hss \fi%
1625         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1626         \if@mparswitch\else \hss \fi}%
1627     \def\@oddhead{%
1628         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1629     \let\@mkboth\@gobbletwo
1630 %<book|report> \let\chaptermark\@gobble
1631 \let\sectionmark\@gobble
1632 %<!book&!report> \let\subsectionmark\@gobble
1633 }

```

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

```

1637 % \date{\today}

\subtitle 副題を設定する。

\jsSubtitle ※プレアンブルにおいて \newcommand*{\subtitle}{...} が行われることへの対策として、\subtitle の定義を \title の実行まで遅延させることにする。もしどうしても主題より前に副題を設定したい場合は、\jsSubtitle 命令を直接用いればよい。

本体を \jsSubtitle として定義する。

1638 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}

1639 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1640 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}

1641 \AtBeginDocument{\bxjs@decl@subtitle}

1642 \def\bxjs@decl@subtitle{%

1643 \global\let\bxjs@decl@subtitle\relax

1644 \ifx\subtitle\@undefined

1645 \global\let\subtitle\jsSubtitle

1646 \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。

1647 \def\bxjs@annihilate@subtitle{%

1648 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi

1649 \global\let\jsSubtitle\relax}

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

\eauthor 1650 %<*\jspf>

\keywords 1651 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}

1652 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}

1653 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}

1654 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}

1655 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}

1656 %</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。

1657 \def\plainifnotempty{%

1658 \ifx \@oddhead \@empty

1659 \ifx \@oddfoot \@empty

1660 \else

1661 \thispagestyle{plainfoot}%


```

1662     \fi
1663     \else
1664         \thispagestyle{plainhead}%
1665     \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。

```

1666 %<*article|book|report|slide>
1667 \if@titlepage
1668     \newcommand{\maketitle}{%
1669         \begin{titlepage}%
1670             \let\footnotesize\small
1671             \let\footnoterule\relax
1672             \let\footnote\thanks
1673             \null\vfil
1674             \if@slide
1675                 {\footnotesize \@date}%
1676                 \begin{center}
1677                     \mbox{} \[\[1\jsZw]
1678                     \large
1679                     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1680                     \jsc@smallskip
1681                     \@title
1682                     \ifx\bxjs@subtitle\@undefined\else
1683                         \par\vskip\z@
1684                         {\small \bxjs@subtitle\par}
1685                     \fi
1686                     \jsc@smallskip
1687                     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1688                     \vfill
1689                     {\small \@author}%
1690                 \end{center}
1691             \else
1692                 \vskip 60\p@?
1693                 \begin{center}%
1694                     {\LARGE \@title \par}%
1695                     \ifx\bxjs@subtitle\@undefined\else
1696                         \vskip5\p@?
1697                         {\normalsize \bxjs@subtitle\par}
1698                     \fi
1699                     \vskip 3em%
1700                     {\large
1701                         \lineskip .75em
1702                         \begin{tabular}[t]{c}%
1703                             \@author

```

```

1704         \end{tabular}\par}%
1705         \vskip 1.5em
1706         {\large \@date \par}%
1707     \end{center}%
1708     \fi
1709     \par
1710     \@thanks\vfil\null
1711 \end{titlepage}%
1712 \setcounter{footnote}{0}%
1713 \global\let\thanks\relax
1714 \global\let\maketitle\relax
1715 \global\let\@thanks\@empty
1716 \global\let\@author\@empty
1717 \global\let\@date\@empty
1718 \global\let\@title\@empty
1719 \global\let\title\relax
1720 \global\let\author\relax
1721 \global\let\date\relax
1722 \global\let\and\relax
1723 \bxjs@annihilate@subtitle
1724 }%
1725 \else
1726 \newcommand{\maketitle}{\par
1727     \begingroup
1728         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1729         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1730         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1731             \parindent 1\jsZw\noindent
1732             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1733         \if@twocolumn
1734             \ifnum \col@number=\@ne
1735                 \@maketitle
1736             \else
1737                 \twocolumn[\@maketitle]%
1738             \fi
1739         \else
1740             \newpage
1741             \global\@topnum\z@ % Prevents figures from going at top of page.
1742             \@maketitle
1743         \fi
1744         \plainifnotempty
1745         \@thanks
1746     \endgroup
1747     \setcounter{footnote}{0}%
1748     \global\let\thanks\relax
1749     \global\let\maketitle\relax
1750     \global\let\@thanks\@empty
1751     \global\let\@author\@empty
1752     \global\let\@date\@empty

```

```

1753 \global\let\@title\@empty
1754 \global\let\title\relax
1755 \global\let\author\relax
1756 \global\let\date\relax
1757 \global\let\and\relax
1758 \bxjs@annihilate@subtitle
1759 }

```

\@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。

```

1760 \def\@maketitle{%
1761 \newpage\null
1762 \vskip 2em
1763 \begin{center}%
1764 \let\footnote\thanks
1765 {\LARGE \@title \par}%
1766 \ifx\bxjs@subtitle\@undefined\else
1767 \vskip3\p@?
1768 {\normalsize \bxjs@subtitle\par}
1769 \fi
1770 \vskip 1.5em
1771 {\large
1772 \lineskip .5em
1773 \begin{tabular}[t]{c}%
1774 \@author
1775 \end{tabular}\par}%
1776 \vskip 1em
1777 {\large \@date}%
1778 \end{center}%
1779 \par\vskip 1.5em
1780 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1781 }
1782 \fi
1783 %</article|book|report|slide>
1784 %<*jspf>
1785 \newcommand{\maketitle}{\par
1786 \begingroup
1787 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1788 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1789 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1790 \parindent 1\jsZw\noindent
1791 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1792 \twocolumn[\@maketitle]%
1793 \plainifnotempty
1794 \@thanks
1795 \endgroup
1796 \setcounter{footnote}{0}%
1797 \global\let\thanks\relax
1798 \global\let\maketitle\relax
1799 \global\let\@thanks\@empty

```

```

1800 \global\let\@author\@empty
1801 \global\let\@date\@empty
1802 % \global\let\@title\@empty % \@title は柱に使う
1803 \global\let\title\relax
1804 \global\let\author\relax
1805 \global\let\date\relax
1806 \global\let\and\relax
1807 \ifx\authors@mail\@undefined\else{%
1808     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1809     \footnotetext[0]{\itshape\authors@mail}%
1810 } \fi
1811 \global\let\authors@mail\@undefined}
1812 \def\@maketitle{%
1813     \newpage\null
1814     \vskip 6em % used to be 2em
1815     \begin{center}
1816         \let\footnote\thanks
1817         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1818         \lineskip .5em
1819         \ifx\@author\@undefined\else
1820             \vskip 1em
1821             \begin{tabular}[t]{c}%
1822                 \@author
1823             \end{tabular}\par
1824         \fi
1825         \ifx\@etitle\@undefined\else
1826             \vskip 1em
1827             {\large \@etitle \par}%
1828         \fi
1829         \ifx\@eauthor\@undefined\else
1830             \vskip 1em
1831             \begin{tabular}[t]{c}%
1832                 \@eauthor
1833             \end{tabular}\par
1834         \fi
1835         \vskip 1em
1836         \@date
1837     \end{center}
1838     \vskip 1.5em
1839     \centerline{\box\@abstractbox}
1840     \ifx\@keywords\@undefined\else
1841         \vskip 1.5em
1842         \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1843     \fi
1844     \vskip 1.5em}
1845 %</jspf>

```

8.2 章・節

ムニャムニャ……。

`\bxjs@label@sect` 節付 #1 の番号を出力する。節付 XXX に対して、`\labelXXX` が定義済ならそれが出力書式を表す。未定義ならばカウンタの出力書式 `\theXXX` が使われる。

```
1846 \def\bxjs@label@sect#1{%
1847   \expandafter\ifx\csname label#1\endcsname\relax
1848     \csname the#1\endcsname
1849   \else \csname label#1\endcsname
1850   \fi}
1851 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}
```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```
1852 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1853 \def\@secapp{\presectionname}
1854 \def\@secpos{\postsectionname}
1855 \fi
```

`\labelsection` 節番号の出力書式。

```
1856 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1857 \def\labelsection{\@secapp\thesection\@secpos}
1858 \fi
```

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

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

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

* この * 印がないと、見出し番号を付け、見出し番号のカウンタに 1 を加算します。
 別見出し 目次や柱に出力する見出しです。
 見出し 見出しです。

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

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

```

1859 \def\@startsection#1#2#3#4#5#6{%
1860   \if@noskipsec \leavevmode \fi
1861   \par
1862 % 見出し上の空きを \@tempskipa にセットする
1863   \@tempskipa #4\relax
1864 % \@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
1865   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1866 % 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
1867   \ifdim \@tempskipa <\z@
1868     \@tempskipa -\@tempskipa \@afterindentfalse
1869   \fi
1870   \if@nobreak
1871 %   \everypar{\everyparhook}% これは間違い
1872     \everypar{}%
1873   \else
1874     \addpenalty\@secpenalty
1875 % 次の行は削除
1876 %   \addvspace\@tempskipa
1877 % 次の \noindent まで追加
1878     \ifdim \@tempskipa >\z@
1879       \if@slide\else
1880         \null
1881         \vspace*{-\baselineskip}%
1882       \fi
1883       \vskip\@tempskipa
1884     \fi
1885   \fi
1886   \noindent
1887 % 追加終わり
1888   \@ifstar
1889     {\@ssect{#3}{#4}{#5}{#6}}%
1890     {\@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。

```

1891 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1892   \let\bxjs@if@ceph\@firstofone
1893 \else \let\bxjs@if@ceph\@gobble
1894 \fi

1895 \def\@sect#1#2#3#4#5#6[#7]#8{%
1896   \ifnum #2>\c@secnumdepth
1897     \let\@svsec\@empty
1898   \else
1899     \refstepcounter{#1}%
1900     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1901   \fi
1902 % 見出し後の空きを \@tempskipa にセット
1903   \@tempskipa #5\relax
1904 % 条件判断の順序を入れ替えました
1905   \ifdim \@tempskipa<\z@
1906     \def\@svsechd{%
1907       #6{\hskip #3\relax
1908         \@svsec #8}%
1909       \csname #1mark\endcsname{#7}%
1910       \addcontentsline{toc}{#1}{%
1911         \ifnum #2>\c@secnumdepth \else
1912           \protect\numberline{\bxjs@label@sect{#1}}%
1913         \fi
1914       #7}}% 目次にフルネームを載せるなら #8
1915   \else
1916     \begingroup
1917     \interlinepenalty \@M % 下から移動
1918     #6{%
1919       \@hangfrom{\hskip #3\relax\@svsec}%
1920 %     \interlinepenalty \@M % 上に移動
1921     #8\@@par}%
1922   \endgroup
1923   \csname #1mark\endcsname{#7}%
1924   \addcontentsline{toc}{#1}{%
1925     \ifnum #2>\c@secnumdepth \else
1926       \protect\numberline{\bxjs@label@sect{#1}}%
1927     \fi
1928     #7}% 目次にフルネームを載せるならここは #8
1929   \fi
1930   \@xsect{#5}}

```

二つ挿入した `\everyparhook` のうち後者が `\paragraph` 類の後で 2 回実行され、それ以降は前者が実行されます。

[2016-07-28] `slide` オプションと `twocolumn` オプションを同時に指定した場合の罫線の位置を微調整しました。

```

1931 \def\@xsect#1{%
1932 % 見出しの後ろの空きを \@tempskipa にセット

```

```

1933 \@tempskipa #1\relax
1934 % 条件判断の順序を変えました
1935 \ifdim \@tempskipa<\z@
1936 \nobreakfalse
1937 \global\@noskipsectrue
1938 \everypar{%
1939 \if@noskipsec
1940 \global\@noskipsecfalse
1941 {\setbox\z@\lastbox}%
1942 \clubpenalty\@M
1943 \begingroup \svsechd \endgroup
1944 \unskip
1945 \@tempskipa #1\relax
1946 \hskip -\@tempskipa
1947 \else
1948 \clubpenalty \@clubpenalty
1949 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1950 \fi\bxjs@if@ceph\everyparhook}%
1951 \else
1952 \par \nobreak
1953 \vskip \@tempskipa
1954 \@afterheading
1955 \fi
1956 \if@slide
1957 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1958 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1959 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1960 \fi
1961 \par % 2000-12-18
1962 \ignorespaces}
1963 \def\@ssect#1#2#3#4#5{%
1964 \@tempskipa #3\relax
1965 \ifdim \@tempskipa<\z@
1966 \def\@svsechd{#4{\hskip #1\relax #5}}%
1967 \else
1968 \begingroup
1969 #4{%
1970 \@hangfrom{\hskip #1}%
1971 \interlinepenalty \@M #5\@@par}%
1972 \endgroup
1973 \fi
1974 \@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 1975 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1976 % \newcommand*\sectionmark[1]{}
\paragraphmark
\subparagraphmark

```



```

1977 % \newcommand*{\subsectionmark}[1]{}
1978 % \newcommand*{\subsubsectionmark}[1]{}
1979 % \newcommand*{\paragraphmark}[1]{}
1980 % \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。

```

1981 %<!book&!report>\setcounter{secnumdepth}{3}
1982 %<book|report>\setcounter{secnumdepth}{2}

```

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

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

```

\c@subsection 1983 \newcounter{part}
1984 %<book|report>\newcounter{chapter}
\c@subsubsection 1985 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1986 %<!book&!report>\newcounter{section}
\c@subparagraph 1987 \newcounter{subsection}[section]
1988 \newcounter{subsubsection}[subsection]
1989 \newcounter{paragraph}[subsubsection]
1990 \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。

```

1991 \renewcommand{\thepart}{\@Roman\c@part}
1992 %<*&!book&!report>
1993 \ifnum\bxjs@label@section=\bxjs@label@section@compat
1994 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1995 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1996 \else
1997 \renewcommand{\thesection}{\@arabic\c@section}
1998 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1999 \fi
2000 %</&!book&!report>
2001 %<*&book|report>
2002 \renewcommand{\thechapter}{\@arabic\c@chapter}
2003 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2004 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}

```

```

2005 %</book|report>
2006 \renewcommand{\thesubsubsection}{%
2007   \thesubsection.\@arabic\c@subsubsection}
2008 \renewcommand{\theparagraph}{%
2009   \thesubsubsection.\@arabic\c@paragraph}
2010 \renewcommand{\thesubparagraph}{%
2011   \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。

```

2012 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2013 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

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

```

2014 %<*book|report>
2015 \newcommand\frontmatter{%
2016   \pltx@cleartooddpage
2017   \@mainmatterfalse
2018   \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。

```

2019 \newcommand\mainmatter{%
2020   \pltx@cleartooddpage
2021   \@mainmattertrue
2022   \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。

```

2023 \newcommand\backmatter{%
2024   \if@openleft
2025     \cleardoublepage
2026   \else\if@openright
2027     \cleardoublepage
2028   \else
2029     \clearpage
2030   \fi\fi
2031   \@mainmatterfalse}
2032 %</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。

```
2033 %<!*book&!report>
2034 \newcommand\part{%
2035   \if@noskipsec \leavevmode \fi
2036   \par
2037   \addvspace{4ex}%
2038   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2039   \secdef\@part\@spart}
2040 %</!*book&!report>
```

`book` および `report` クラスの場合は、少し複雑です。

```
2041 %<*book|report>
2042 \newcommand\part{%
2043   \if@openleft
2044     \cleardoublepage
2045   \else\if@openright
2046     \cleardoublepage
2047   \else
2048     \clearpage
2049   \fi\fi
2050   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2051   \if@twocolumn
2052     \onecolumn
2053     \@restonecoltrue
2054   \else
2055     \@restonecolfalse
2056   \fi
2057   \null\vfil
2058   \secdef\@part\@spart}
2059 %</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。

```
2060 %<*!book&!report>
2061 \def\@part[#1]#2{%
2062   \ifnum \c@secnumdepth >\m@ne
2063     \refstepcounter{part}%
2064     \addcontentsline{toc}{part}{%
2065       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2066   \else
2067     \addcontentsline{toc}{part}{#1}%
2068   \fi
2069   \markboth{}{}%
2070   {\parindent\z@
2071     \raggedright
2072     \interlinepenalty \@M
2073     \normalfont
2074     \ifnum \c@secnumdepth >\m@ne
2075       \Large\headfont\prepartname\thepart\postpartname
2076       \par\nobreak
2077     \fi
2078     \huge \headfont #2%
2079     \markboth{}{}\par}%
2080   \nobreak
2081   \vskip 3ex
2082   \@afterheading}
2083 %</!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。

```
2084 %<*book|report>
2085 \def\@part[#1]#2{%
2086   \ifnum \c@secnumdepth >-2\relax
2087     \refstepcounter{part}%
2088     \addcontentsline{toc}{part}{%
2089       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2090   \else
2091     \addcontentsline{toc}{part}{#1}%
2092   \fi
2093   \markboth{}{}%
2094   {\centering
2095     \interlinepenalty \@M
2096     \normalfont
2097     \ifnum \c@secnumdepth >-2\relax
2098       \huge\headfont \prepartname\thepart\postpartname
2099       \par\vskip20\p@?
2100     \fi
2101     \Huge \headfont #2\par}%
2102   \@endpart}
2103 %</book|report>
```

\@spart 番号を付けない部です。

```

2104 %<*!book&!report>
2105 \def\@spart#1{%
2106     \parindent \z@ \raggedright
2107     \interlinepenalty \@M
2108     \normalfont
2109     \huge \headfont #1\par}%
2110 \nobreak
2111 \vskip 3ex
2112 \@afterheading}
2113 %</!book&!report>
2114 %<*book|report>
2115 \def\@spart#1{%
2116     \centering
2117     \interlinepenalty \@M
2118     \normalfont
2119     \Huge \headfont #1\par}%
2120 \@endpart}
2121 %</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。

```

2122 %<*book|report>
2123 \def\@endpart{\vfil\newpage
2124     \if@twoside
2125     \if@openleft %% added (2017/02/24)
2126     \null\thispagestyle{empty}\newpage
2127     \else\if@openright %% added (2016/12/13)
2128     \null\thispagestyle{empty}\newpage
2129     \fi\fi %% added (2016/12/13, 2017/02/24)
2130 \fi
2131 \if@restonecol
2132     \twocolumn
2133 \fi}
2134 %</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。

```

2135 %<*book|report>
2136 \newcommand{\chapter}{%
2137     \if@openleft\cleardoublepage\else
2138     \if@openright\cleardoublepage\else\clearpage\fi\fi
2139     \plainifnotempty % 元: \thispagestyle{plain}

```

```

2140 \global\@topnum\z@
2141 \if@english \afterindentfalse \else \afterindenttrue \fi
2142 \secdef
2143     {\@omit@numberfalse\@chapter}%
2144     {\@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。

```

2145 \def\@chapter[#1]#2{%
2146     \ifnum \c@secnumdepth >\m@ne
2147         \if@mainmatter
2148             \refstepcounter{chapter}%
2149             \typeout{\@chapapp\thechapter\@chappos}%
2150             \addcontentsline{toc}{chapter}%
2151                 {\protect\numberline
2152 %             %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2153                 {\@chapapp\thechapter\@chappos}%
2154                 #1}%
2155         \else\addcontentsline{toc}{chapter}{#1}\fi
2156     \else
2157         \addcontentsline{toc}{chapter}{#1}%
2158     \fi
2159     \chaptermark{#1}%
2160     \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
2161     \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
2162     \if@twocolumn
2163         \@topnewpage[\@makechapterhead{#2}]%
2164     \else
2165         \@makechapterhead{#2}%
2166         \@afterheading
2167     \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。

```

2168 \def\@makechapterhead#1{%
2169     \vspace*{2\Cvs}% 欧文は 50pt
2170     {\parindent \z@ \raggedright \normalfont
2171         \ifnum \c@secnumdepth >\m@ne
2172             \if@mainmatter
2173                 \huge\headfont \@chapapp\thechapter\@chappos
2174                 \par\nobreak
2175                 \vskip \Cvs % 欧文は 20pt
2176             \fi
2177         \fi
2178         \interlinepenalty\@M
2179         \Huge \headfont #1\par\nobreak
2180         \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。

```

2181 \def\@schapter#1{%

```

```

2182 \chaptermark{#1}%
2183 \if@twocolumn
2184 \topnewpage[\@makeschapterhead{#1}]%
2185 \else
2186 \@makeschapterhead{#1}\@afterheading
2187 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2188 \def\@makeschapterhead#1{%
2189 \vspace*{2\Cvs}% 欧文は 50pt
2190 {\parindent \z@ \raggedright
2191 \normalfont
2192 \interlinepenalty\@M
2193 \Huge \headfont #1\par\nobreak
2194 \vskip 3\Cvs}} % 欧文は 40pt
2195 %</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。

```

2196 \if@twocolumn
2197 \newcommand{\section}{%
2198 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2199 \@startsection{section}{1}{\z@}%
2200 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2201 %<kiyou> {\Cvs}{0.5\Cvs}%
2202 % {\normalfont\large\headfont\@secapp}}
2203 {\normalfont\large\headfont\raggedright}}
2204 \else
2205 \newcommand{\section}{%
2206 \if@slide\clearpage\fi
2207 \@startsection{section}{1}{\z@}%
2208 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2209 {.5\Cvs \@plus.3\Cdp}% 後アキ
2210 % {\normalfont\Large\headfont\@secapp}}
2211 {\normalfont\Large\headfont\raggedright}}
2212 \fi

```

`\subsection` 同上です。

```

2213 \if@twocolumn
2214 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2215 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2216 {\normalfont\normalsize\headfont}}
2217 \else
2218 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2219 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ

```

```

2220    {.5\Cvs \@plus.3\Cdp}% 後アキ
2221    {\normalfont\large\headfont}}
2222 \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)。

```

2223 \if@twocolumn
2224   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2225     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2226     {\normalfont\normalsize\headfont}}
2227 \else
2228   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2229     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2230     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2231     {\normalfont\normalsize\headfont}}
2232 \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。

```

2233 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2234 \let\bxjs@org@paragraph@mark\jsParagraphMark
2235 \ifx\bxjs@paragraph@mark\@empty
2236   \let\jsParagraphMark\@empty
2237 \else\ifx\bxjs@paragraph@mark\@undefined\else
2238   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2239 \fi\fi
2240 \let\jsJaChar\@empty
2241 \if@twocolumn
2242   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2243     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2244     %<jspf> {\normalfont\normalsize\headfont}}
2245     %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2246 \else
2247   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%

```



```

2248      {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2249      {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2250 %<jspf>      {\normalfont\normalsize\headfont}}
2251 %<!jspf>      {\normalfont\normalsize\headfont\jsParagraphMark}}
2252 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2253 \if@twocolumn
2254   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2255     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2256     {\normalfont\normalsize\headfont}}
2257 \else
2258   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2259     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2260     {\normalfont\normalsize\headfont}}
2261 \fi

```

8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2262 \if@slide
2263   \setlength\leftmargini{1\jsZw}
2264 \else
2265   \if@twocolumn
2266     \setlength\leftmargini{2\jsZw}
2267   \else
2268     \setlength\leftmargini{3\jsZw}
2269   \fi
2270 \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 2271 \if@slide
\leftmarginv 2272   \setlength\leftmarginii {1\jsZw}
2273   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2274   \setlength\leftmarginiv {1\jsZw}
2275   \setlength\leftmarginv  {1\jsZw}
2276   \setlength\leftmarginvi {1\jsZw}
2277 \else
2278   \setlength\leftmarginii {2\jsZw}
2279   \setlength\leftmarginiii{2\jsZw}

```

```

2280 \setlength\leftmarginiv {2\jsZw}
2281 \setlength\leftmarginv {1\jsZw}
2282 \setlength\leftmarginvi {1\jsZw}
2283 \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。

```

2284 \setlength \labelsep {0.5\jsZw} % .5em
2285 \setlength \labelwidth{\leftmargini}
2286 \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。

```

2287 \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 2288 \@beginparpenalty -\@lowpenalty
\@itempenalty 2289 \@endparpenalty -\@lowpenalty
2290 \@itempenalty -\@lowpenalty

```

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

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

```

2291 \def\@listi{\leftmargin\leftmargini
2292 \parsep \z@
2293 \topsep 0.5\baselineskip
2294 \itemsep \z@ \relax}
2295 \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）。

```

2296 \@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 2297 \def\@listii{\leftmargin\leftmarginii
\@listiv 2298 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2299 \topsep \z@
\@listv 2300 \parsep \z@
\@listvi 2301 \itemsep\parsep}
2302 \def\@listiii{\leftmargin\leftmarginiii
2303 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2304 \topsep \z@
2305 \parsep \z@

```

```

2306 \itemsep\parsep}
2307 \def\@listiv {\leftmargin\leftmarginiv
2308             \labelwidth\leftmarginiv
2309             \advance\labelwidth-\labelsep}
2310 \def\@listv {\leftmargin\leftmarginv
2311             \labelwidth\leftmarginv
2312             \advance\labelwidth-\labelsep}
2313 \def\@listvi {\leftmargin\leftmarginvi
2314             \labelwidth\leftmarginvi
2315             \advance\labelwidth-\labelsep}

```

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

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

```

2316 \renewcommand{\theenumi}{\@arabic\c@enumi}
2317 \renewcommand{\theenumii}{\@alph\c@enumii}
2318 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2319 \renewcommand{\theenumiv}{\@Alph\c@enumiv}

```

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

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

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

```

2320 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2321 \newcommand*{\jsInJaParen}[1]{%
2322   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2323 \newcommand{\labelenumi}{\theenumi.}
2324 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
2325 \newcommand{\labelenumiii}{\theenumiii.}
2326 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2327 \renewcommand{\p@enumii}{\theenumi}
2328 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
2329 \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 2330 \newcommand\labelitemi{\textbullet}
\labelitemiii 2331 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiv 2332 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemv 2333 \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。

```
2334 \newenvironment{description}{%
2335   \list{}{%
2336     \labelwidth=\leftmargin
2337     \labelsep=1\jsZw
2338     \advance \labelwidth by -\labelsep
2339     \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。

```
2340 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}
```

■概要

`abstract` 概要（要旨、梗概）を出力する環境です。book クラスでは各章の初めにちょっとしたことを書くのに使います。titlepage オプション付きの article クラスでは、独立したページに出力されます。abstract 環境は元は quotation 環境で作られていましたが、quotation 環境の右マージンをゼロにしたので、list 環境で作り直しました。

JSPF スタイルでは実際の出力は `\maketitle` で行われます。

`bxjsreport` クラスの `abstract` 環境は：

- layout=v1 の場合は `jsbook + report` の動作を継承する。つまり `jsbook` と同じになる。
- layout=v2 の場合は新設の `jsreport` の動作を継承する。つまり `jsarticle (+ titlapage)` と同じになる。

`chapterabstract` `jsbook` の `abstract` 環境（「各章の初めにちょっとしたことを書く」ためのもの）を `chapterabstract` と呼ぶことにする。

```
2341 %<*book|report>
2342 \newenvironment{chapterabstract}{%
2343   \begin{list}{}{%
2344     \listparindent=1\jsZw
2345     \itemindent=\listparindent
```

```

2346 \rightmargin=0pt
2347 \leftmargin=5\jsZw\item[]\end{list}\vspace{\baselineskip}}
2348 %</book|report>

```

“普通の” abstract 環境の定義。

```

2349 %<*article|report|slide>
2350 \newbox\@abstractbox
2351 \if@titlepage
2352 \newenvironment{abstract}{%
2353 \titlepage
2354 \null\vfil
2355 \@beginparpenalty\@lowpenalty
2356 \begin{center}%
2357 \headfont \abstractname
2358 \@endparpenalty\@M
2359 \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。

```

2360 \par}%
2361 {\par\vfil\null\endtitlepage}
2362 \else
2363 \newenvironment{abstract}{%
2364 \if@twocolumn
2365 \ifx\maketitle\relax
2366 \section*{\abstractname}%
2367 \else
2368 \global\setbox\@abstractbox\hbox\bgroup
2369 \begin{minipage}[b]{\textwidth}
2370 \small\parindent1\jsZw
2371 \begin{center}%
2372 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2373 \end{center}%
2374 \list{}{%
2375 \listparindent\parindent
2376 \itemindent \listparindent
2377 \rightmargin \leftmargin}%
2378 \item\relax
2379 \fi
2380 \else
2381 \small
2382 \begin{center}%
2383 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2384 \end{center}%
2385 \list{}{%
2386 \listparindent\parindent
2387 \itemindent \listparindent
2388 \rightmargin \leftmargin}%
2389 \item\relax
2390 \fi}{\if@twocolumn

```

```

2391 \ifx\maketitle\relax
2392 \else
2393 \endlist\end{minipage}\egroup
2394 \fi
2395 \else
2396 \endlist
2397 \fi}
2398 \fi
2399 %</article|report|slide>
2400 %<*jspf>
2401 \newbox\@abstractbox
2402 \newenvironment{abstract}{%
2403 \global\setbox\@abstractbox\hbox\bgroup
2404 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2405 \small
2406 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2407 {\end{minipage}\egroup}
2408 %</jspf>

```

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

```

2409 %<*book|report>
2410 \ifbxjs@force@chapterabstract
2411 \let\abstract\chapterabstract
2412 \let\endabstract\endchapterabstract
2413 \fi
2414 %</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。

```

2415 %<*jspf>
2416 %\newbox\@keywordsbox
2417 %\newenvironment{keywords}{%
2418 % \global\setbox\@keywordsbox\hbox\bgroup
2419 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2420 % \small\parindent0\jsZw}%
2421 % {\end{minipage}\egroup}
2422 %</jspf>

```

■verse 環境

verse 詩のための verse 環境です。

```

2423 \newenvironment{verse}{%
2424 \let \=\@centercr
2425 \list{}{%
2426 \itemsep \z@

```

```

2427 \itemindent -2\jsZw % 元: -1.5em
2428 \listparindent\itemindent
2429 \rightmargin \z@
2430 \advance\leftmargin 2\jsZw}% 元: 1.5em
2431 \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。

```

2432 \newenvironment{quotation}{%
2433 \list{}{%
2434 \listparindent\parindent
2435 \itemindent\listparindent
2436 \rightmargin \z@}%
2437 \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。

```

2438 \newenvironment{quote}%
2439 {\list{}{\rightmargin\z@}\item\relax}{\endlist}

```

■定理など `ltthm.dtx` 参照。たとえば次のように定義します。

```

\newtheorem{definition}{定義}
\newtheorem{axiom}{公理}
\newtheorem{theorem}{定理}

```

[2001-04-26] 定理の中はイタリック体になりましたが、これでは和文がゴシック体になってしまうので、`\itshape` を削除しました。

[2009-08-23] `\bfseries` を `\headfont` に直し、`\labelsep` を 1zw にし、括弧を全角にしました。

```

2440 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2441 \item[\hskip \labelsep{\headfont #1\ #2}]}
2442 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2443 \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。

```

2444 \newenvironment{titlepage}{%

```

```

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

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```

2458 %<!*book&!report>
2459 \newcommand{\appendix}{\par
2460     \setcounter{section}{0}%
2461     \setcounter{subsection}{0}%
2462     \ifnum\bxjs@label@section=\bxjs@label@section@compat
2463         \gdef\presectionname{\appendixname}%
2464         \gdef\postsectionname{}%
2465     % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2466     \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2467     \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2468     \else
2469         \gdef\@secapp{\appendixname}%
2470         \gdef\@secpos{}%
2471         \gdef\thesection{\@Alph\c@section}%
2472     \fi}
2473 %</!*book&!report>
2474 %<*book|report>
2475 \newcommand{\appendix}{\par
2476     \setcounter{chapter}{0}%
2477     \setcounter{section}{0}%
2478     \gdef\@chapapp{\appendixname}%
2479     \gdef\@chappos{}%
2480     \gdef\thechapter{\@Alph\c@chapter}}
2481 %</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。

2482 `\setlength\arraycolsep{5\p@?}`

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

2483 `\setlength\tabcolsep{6\p@?}`

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

2484 `\setlength\arrayrulewidth{.4\p@}`

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

2485 `\setlength\doublerulesep{2\p@}`

■tabbing 環境

`\tabbingsep` `\'` コマンドで入るアキです。

2486 `\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。

2487 `\skip\@mpfootins = \skip\footins`

■framebox 環境

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

`\fbxrule` `\fbbox`, `\framebox` の罫線の幅です。

2488 `\setlength\fbxsep{3\p@?}`

2489 `\setlength\fbxrule{.4\p@}`

■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

2490 `%<!book&!report>\renewcommand \theequation {\@arabic\c@equation}`

2491 `%<*book|report>`

2492 `\@addtoreset{equation}{chapter}`

2493 `\renewcommand\theequation`

2494 `{\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}`

2495 `%</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。

2496 `% \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。

2497 `% \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。

```
2498 % \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。

```
2499 %<*/book&!report>
2500 \newcounter{figure}
2501 \renewcommand \thefigure {\@arabic\c@figure}
2502 %</!book&!report>
2503 %<*book|report>
2504 \newcounter{figure}[chapter]
2505 \renewcommand \thefigure
2506     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2507 %</book|report>
```

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

```
\ext@figure 2508 \def\fps@figure{tbp}
2509 \def\ftype@figure{1}
\fnum@figure 2510 \def\ext@figure{lof}
2511 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2512 \newenvironment{figure}%
2513     {\@float{figure}}%
2514     {\endfloat}
2515 \newenvironment{figure*}%
2516     {\@dblfloat{figure}}%
2517     {\enddblfloat}
```

■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。

```
2518 %<*&book&!report>
2519 \newcounter{table}
2520 \renewcommand\thetable{\@arabic\c@table}
2521 %</!book&!report>
2522 %<*&book|report>
2523 \newcounter{table}[chapter]
2524 \renewcommand \thetable
2525     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2526 %</book|report>
```

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

```
\ext@table 2527 \def\fps@table{tbp}
2528 \def\ftype@table{2}
\fnun@table 2529 \def\ext@table{lot}
2530 \def\fnun@table{\tablename\nobreak\thetable}
```

`table *` は段抜きのフロートです。

```
table* 2531 \newenvironment{table}%
2532     {\@float{table}}%
2533     {\end@float}
2534 \newenvironment{table*}%
2535     {\@dblfloat{table}}%
2536     {\end@dblfloat}
```

8.6 キャプション

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

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

```
2537 \newlength\abovecaptionskip
2538 \newlength\belowcaptionskip
2539 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2540 \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。

```
2541 %<*&!jspf>
2542 % \long\def\@makecaption#1#2{\small
2543 %     \advance\leftskip10\jsc@mmm
```

```

2544 % \advance\rightskip10\jsc@mmm
2545 % \vskip\abovecaptionskip
2546 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2547 % \ifdim \wd\@tempboxa >\hsize
2548 % #1\hskip1\jsZw\relax #2\par
2549 % \else
2550 % \global \@minipagefalse
2551 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2552 % \fi
2553 % \vskip\belowcaptionskip}}
2554 \long\def\@makecaption#1#2{{\small
2555 \advance\leftskip .0628\linewidth
2556 \advance\rightskip .0628\linewidth
2557 \vskip\abovecaptionskip
2558 \sbox\@tempboxa{#1\zwspace#2}%
2559 \ifdim \wd\@tempboxa <\hsize \centering \fi
2560 #1\zwspace#2\par
2561 \vskip\belowcaptionskip}}
2562 %</!jspf>
2563 %<*jspf>
2564 \long\def\@makecaption#1#2{%
2565 \vskip\abovecaptionskip
2566 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2567 \ifdim \wd\@tempboxa >\hsize
2568 {\small\sffamily
2569 \list{#1}{%
2570 \renewcommand{\makelabel}[1]{##1\hfil}
2571 \itemsep \z@
2572 \itemindent \z@
2573 \labelsep \z@
2574 \labelwidth 11\jsc@mmm
2575 \listparindent\z@
2576 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2577 \else
2578 \global \@minipagefalse
2579 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2580 \fi
2581 \vskip\belowcaptionskip}
2582 %</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 `\scrDeclareOldFontCommand` に倣い、これらの命令を使うときには警告を発することにしました。

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

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

```
2583 \newif\if@jsc@warnoldfontcmd
2584 \@jsc@warnoldfontcmdtrue
2585 \newif\if@jsc@warnoldfontcmdexception
2586 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand

2587 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2588   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2589   \DeclareOldFontCommand{#1}{%
2590     \bxjs@oldfontcmd{#1}#2%
2591   }{%
2592     \bxjs@oldfontcmd{#1}#3%
2593   }%
2594 }
2595 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2596   \ClassInfo\bxjs@clsname
2597   {Old font command '\string#1' is used!!\MessageBreak
2598     The first occurrence is}%
2599 }
```

```
\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。

2600 \newcommand*{\allowoldfontcommands}{%
2601   \@jsc@warnoldfontcmdexceptiontrue}
2602 \newcommand*{\disallowoldfontcommands}{%
2603   \@jsc@warnoldfontcmdexceptionfalse}

2604 \let\bxjs@oldfontcmd@list\@empty
2605 \def\bxjs@oldfontcmd#1{%
2606   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2607 \def\bxjs@oldfontcmd@a#1#2{%
2608   \if@jsc@warnoldfontcmdexception\else
2609     \global\@jsc@warnoldfontcmdfalse
2610     \ifx#1\relax
2611       \global\let#1=t%
2612       \jsc@warnoldfontcmd{#2}%
2613     \fi
2614   \fi}
```

```

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

```

2633 \def\bxjs@warnoldfontcmd@kick@final{%
2634   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2635 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2636 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2637 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2638 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2639 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2640 \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。

```

2641 \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。

```

2642 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2643 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2644 \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`

```
2645 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2646 \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。

```
2647 \newcommand\@pnumwidth{1.55em}
2648 \newcommand\@tocrmarg{2.55em}
2649 \newcommand\@dotsep{4.5}
2650 %<!book&!report>\setcounter{tocdepth}{2}
2651 %<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)

```
2652 \newdimen\jsc@tocl@width
2653 \newcommand{\tableofcontents}{%
2654 %<*book|report>
2655 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2656 \settowidth\@tempdima{\headfont\appendixname}%
2657 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2658 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2659 \if@twocolumn
2660 \@restonecoltrue\onecolumn
2661 \else
2662 \@restonecolfalse
2663 \fi
2664 \chapter*{\contentsname}%
2665 \@mkboth{\contentsname}{}%
2666 %</book|report>
2667 %<*&!book&!report>
2668 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2669 \settowidth\@tempdima{\headfont\appendixname}%
2670 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2671 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2672 \section*{\contentsname}%
2673 \@mkboth{\contentsname}{\contentsname}%
2674 %</&!book&!report>
2675 \@starttoc{toc}%
2676 %<book|report> \if@restonecol\twocolumn\fi
2677 }
```

`\l@part` 部の目次です。

```
2678 \newcommand*{\l@part}[2]{%
2679 \ifnum \c@tocdepth >-2\relax
2680 %<!book&!report> \addpenalty\@secpenalty
2681 %<book|report> \addpenalty{-\@highpenalty}%
2682 \addvspace{2.25em \@plus\p@}%
2683 \begingroup
2684 \parindent \z@
2685 % \@pnumwidth should be \@tocrmarg
2686 % \rightskip \@pnumwidth
2687 \rightskip \@tocrmarg
2688 \parfillskip -\rightskip
2689 {\leavevmode
2690 \large \headfont
2691 \setlength\@lnumwidth{4\jsZw}%
2692 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2693 \nobreak
```



```

2694 %<book|report>      \global\@nobeaktrue
2695 %<book|report>      \everypar{\global\@nobeakfalse\everypar{}}%
2696      \endgroup
2697      \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)

```

2698 %<*book|report>
2699 \newcommand*{\l@chapter}[2]{%
2700   \ifnum \c@tocdepth >\m@ne
2701     \addpenalty{-\@highpenalty}%
2702     \addvspace{1.0em \@plus\p@?}
2703 %   \vskip 1.0em \@plus\p@    % book.cls では↑がこうなっている
2704     \begingroup
2705       \parindent\z@
2706 %     \rightskip\@pnumwidth
2707     \rightskip\@tocrmarg
2708     \parfillskip-\rightskip
2709     \leavevmode\headfont
2710 %     % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2711     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2712     \advance\leftskip\@lnumwidth \hskip-\leftskip
2713     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2714     \penalty\@highpenalty
2715   \endgroup
2716   \fi}
2717 %</book|report>

```

\l@section 節の目次です。

```

2718 %<*!book&!report>
2719 \newcommand*{\l@section}[2]{%
2720   \ifnum \c@tocdepth >\z@
2721     \addpenalty{\@secpenalty}%
2722     \addvspace{1.0em \@plus\p@?}%
2723     \begingroup
2724       \parindent\z@
2725 %     \rightskip\@pnumwidth
2726     \rightskip\@tocrmarg
2727     \parfillskip-\rightskip
2728     \leavevmode\headfont
2729 %     % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2730     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2731     \advance\leftskip\@lnumwidth \hskip-\leftskip
2732     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2733   \endgroup
2734   \fi}
2735 %</!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。

```
2736 %<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] ここの \jsc@tocl@width から決めるようにしてみました。(by ts)
```

```
\l@subparagraph 2737 %<!*book&!report>
2738 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2739 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2740 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2741 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2742 %
2743 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2744 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2745 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2746 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2747 %
2748 \newcommand*{\l@subsection}{%
2749     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2750     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2751 \newcommand*{\l@subsubsection}{%
2752     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2753     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2754 \newcommand*{\l@paragraph}{%
2755     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2756     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2757 \newcommand*{\l@subparagraph}{%
2758     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2759     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2760 %</*!book&!report>
2761 %<*book|report>
2762 % \newcommand*{\l@subsection} {\@dottedtocline{2}{3.8em}{3.2em}}
2763 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2764 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2765 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2766 \newcommand*{\l@section}{%
2767     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2768     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2769 \newcommand*{\l@subsection}{%
2770     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2771     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2772 \newcommand*{\l@subsubsection}{%
2773     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2774     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2775 \newcommand*{\l@paragraph}{%
2776     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2777     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
```

```

2778 \newcommand*{\l@subparagraph}{%
2779     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2780     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2781 %</book|report>

```

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

```

2782 \newdimen\@lnumwidth
2783 \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。

```

2784 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2785     \vskip \z@ \@plus.2\p@?
2786     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2787     \parindent #2\relax\@afterindenttrue
2788     \interlinepenalty\@M
2789     \leavevmode
2790     \@lnumwidth #3\relax
2791     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2792     {#4}\nobreak
2793     \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2794     mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2795     \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2796 \newcommand{\listoffigures}{%
2797 %<*book|report>
2798 \if@twocolumn\@restonecoltrue\onecolumn
2799 \else\@restonecolfalse\fi
2800 \chapter*{\listfigurename}%
2801 \@mkboth{\listfigurename}{}%
2802 %</book|report>
2803 %<!*book&!report>
2804 \section*{\listfigurename}%
2805 \@mkboth{\listfigurename}{\listfigurename}%
2806 %</*!book&!report>
2807 \@starttoc{lof}%
2808 %<book|report> \if@restonecol\twocolumn\fi
2809 }

```

`\l@figure` 図目次の項目を出力します。

```

2810 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```
2811 \newcommand{\listoftables}{%
2812 %<*book|report>
2813 \if@twocolumn\@restonecoltrue\onecolumn
2814 \else\@restonecolfalse\fi
2815 \chapter*{\listtablename}%
2816 \@mkboth{\listtablename}{}%
2817 %</book|report>
2818 %<*&book&!report>
2819 \section*{\listtablename}%
2820 \@mkboth{\listtablename}{\listtablename}%
2821 %</!book&!report>
2822 \@starttoc{lot}%
2823 %<book|report> \if@restonecol\twocolumn\fi
2824 }
```

`\l@table` 表目次は図目次と同じです。

```
2825 \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。

```
2826 \newdimen\bibindent
2827 \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。

```
2828 \newenvironment{thebibliography}[1]{%
2829 \@jsc@warnoldfontcmdexceptiontrue
2830 \global\let\presectionname\relax
2831 \global\let\postsectionname\relax
2832 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2833 %<*&kiyou>
2834 \vspace{1.5\baselineskip}
2835 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2836 \vspace{0.5\baselineskip}
2837 %</kiyou>
2838 %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2839 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2840 \list{\@biblabel{\@arabic{c@enumiv}}}%
2841 {\settowidth\labelwidth{\@biblabel{#1}}%
2842 \leftmargin\labelwidth
2843 \advance\leftmargin\labelsep
2844 \@openbib@code
2845 \usecounter{enumiv}}%
```

```

2846      \let\p@enumiv\@empty
2847      \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2848 %<kiyou>   \small
2849   \sloppy
2850   \clubpenalty4000
2851   \@clubpenalty\clubpenalty
2852   \widowpenalty4000%
2853   \sfcode`\.\@m}
2854   {\def\@noitemerr
2855     {\@latex@warning{Empty `thebibliography' environment}}}%
2856   \endlist}

```

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

```

2857 \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。

```

2858 \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。

```

2859 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

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

```

2860 % \def\@citex[#1]#2{\leavevmode
2861 %   \let\@citea\@empty
2862 %   \@cite{\@for\@citeb:=#2\do
2863 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2864 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2865 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2866 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2867 %     \G@refundefinedtrue
2868 %     \@latex@warning
2869 %       {Citation `'\@citeb' on page \thepage \space undefined}}%
2870 %     {\@cite@ofmt{\csname b@\@citeb\endcsname}}}\@citeb}}{#1}}
2871 % \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。

```

2872 % \DeclareRobustCommand\cite{\unskip
2873 %   \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}}
2874 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2875 %   , \jsInhibitGlue\ #2\fi}}}\$}

```

10.3 索引

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

```
2876 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2877     \if@twocolumn
2878         \onecolumn\@restonecolfalse
2879     \else
2880         \clearpage\@restonecoltrue
2881     \fi
2882     \columnseprule.4pt \columnsep 2\jsZw
2883     \ifx\multicols\undefined
2884 %<book|report>         \twocolumn[\@makeschapterhead{\indexname}%
2885 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}}%
2886 %<!book&!report>      \def\presectionname{}\def\postsectionname{}%
2887 %<!book&!report>      \twocolumn[\section*{\indexname}}%
2888     \else
2889         \ifdim\textwidth<\fullwidth
2890             \setlength{\evensidemargin}{\oddsidemargin}
2891             \setlength{\textwidth}{\fullwidth}
2892             \setlength{\linewidth}{\fullwidth}
2893 %<book|report>         \begin{multicols}{3}[\chapter*{\indexname}%
2894 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}}%
2895 %<!book&!report>      \def\presectionname{}\def\postsectionname{}%
2896 %<!book&!report>      \begin{multicols}{3}[\section*{\indexname}}%
2897     \else
2898 %<book|report>         \begin{multicols}{2}[\chapter*{\indexname}%
2899 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}}%
2900 %<!book&!report>      \def\presectionname{}\def\postsectionname{}%
2901 %<!book&!report>      \begin{multicols}{2}[\section*{\indexname}}%
2902     \fi
2903     \fi
2904 %<book|report>         \@mkboth{\indexname}{}%
2905 %<!book&!report>      \@mkboth{\indexname}{\indexname}%
2906     \plainifnotempty % \thispagestyle{plain}
2907     \parindent\z@
2908     \parskip\z@ \@plus .3\p@?\relax
2909     \let\item\@idxitem
2910     \raggedright
2911     \footnotesize\narrowbaselines
2912 }{
2913     \ifx\multicols\undefined
2914         \if@restonecol\onecolumn\fi
2915     \else
2916         \end{multicols}
2917     \fi
2918     \clearpage
```

2919 }

`\@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` 2920 `\newcommand{\@idxitem}{\par\hangindent 4\jsZw}` % 元 40pt

`\subsubitem` 2921 `\newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}}` % 元 20pt

2922 `\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。

2923 `\newcommand{\indexspace}{\par \vskip 10\p@? \@plus5\p@? \@minus3\p@?\relax}`

`\seename` 索引の `\see`, `\seesalso` 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。

2924 `\newcommand\seename{\if@english see\else \rightarrow\fi}`

2925 `\newcommand\alsoname{\if@english see also\else \rightarrow\fi}`

10.4 脚注

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

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

2926 `\ifx\pltx@foot@penalty\@undefined`

2927 `\let\footnotes@ve=\footnote`

2928 `\def\footnote{\jsInhibitGlue\footnotes@ve}`

2929 `\let\footnotemarks@ve=\footnotemark`

2930 `\def\footnotemark{\jsInhibitGlue\footnotemarks@ve}`

2931 `\fi`

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

pT_EX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
2932 \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。

```
2933 % \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。

```
2934 \renewcommand{\footnoterule}{%
2935   \kern-2.6\p@? \kern-.4\p@
2936   \hrule width .4\columnwidth
2937   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2938 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

[2018-03-11] `\next` などいくつかの内部命令を `\jsc@...` 付きのユニークな名前にしました。

```
2939 \long\def\@footnotetext{%
2940   \insert\footins\bgroup
2941     \normalfont\footnotesize
2942     \interlinepenalty\interfootnotelinepenalty
2943     \splittopskip\footnotesep
2944     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2945     \hsize\columnwidth \@parboxrestore
2946     \protected@edef\@currentlabel{%
2947       \csname p@footnote\endcsname\@thefnmark
2948     }%
2949     \color@begingroup
2950     \@makefntext{%
2951       \rule\z@\footnotesep\ignorespaces}%
2952     \futurelet\jsc@next\jsc@fo@t}
2953 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@fo@t
2954               \else \let\jsc@next\jsc@fo@t\fi \jsc@next}
```



```

2955 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
2956 \def\jsc@f@t#1{#1\jsc@@foot}
2957 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
2958   \ifx\plt@foot@penalty\@undefined\else
2959     \ifhmode\null\fi
2960     \ifnum\plt@foot@penalty=z\else
2961       \penalty\plt@foot@penalty
2962       \plt@foot@penalty\z@
2963     \fi
2964   \fi}

```

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

```

2965 \newcommand\@makefntext[1]{%
2966   \advance\leftskip 3\jsZw
2967   \parindent 1\jsZw
2968   \noindent
2969   \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。

```

2970 % \def\@xfootnotenext[#1]{%
2971 %   \begingroup
2972 %     \ifnum#1>z@
2973 %       \csname c@\@mpfn\endcsname #1\relax
2974 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2975 %     \else
2976 %       \unrestored@protected@xdef\@thefnmark{}%
2977 %     \fi
2978 %   \endgroup
2979 %   \@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。

```
2980 \let\jsInhibitGlueAtParTop\@empty
```

`\everyparhook` 全ての段落の冒頭で実行されるフック。これの初期値を先述の `\jsInhibitGlueAtParTop` とする。

```
2981 \def\everyparhook{\jsInhibitGlueAtParTop}
2982 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2983 \AtBeginDocument{\everypar{\everyparhook}}
2984 \fi
```

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

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

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

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

```
2985 \ifx j\jsEngine
2986 \def\@inhibitglue{%
2987   \futurelet\@let@token\@@inhibitglue}
2988 \begingroup
2989 \let\GDEF=\gdef
2990 \let\CATCODE=\catcode
2991 \let\ENDGROUP=\endgroup
2992 \CATCODE`k=12
2993 \CATCODE`a=12
2994 \CATCODE`n=12
2995 \CATCODE`j=12
2996 \CATCODE`i=12
2997 \CATCODE`c=12
2998 \CATCODE`h=12
2999 \CATCODE`r=12
3000 \CATCODE`t=12
3001 \CATCODE`e=12
3002 \GDEF\KANJI@CHARACTER{kanji character }
3003 \ENDGROUP
3004 \def\@@inhibitglue{%
3005   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3006   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
3007   \def\jsc@ig@temp{#1}%
3008   \ifx\jsc@ig@temp\@empty
3009     \ifnum\the\inhibitxspcode`#2=2\relax
```

```

3010     \inhibitglue
3011     \fi
3012   \fi}
3013 \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。

```

3014 \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。

```

3015 \def\@doendpe{%
3016   \@endpetrue
3017   \def\par{%
3018     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3019   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3020 \def\@setminipage{%
3021   \@minipagetrue
3022   \everypar{\@minipagefalse\everypar{\everyparhook}}}%
3023 }

```

`\item` 命令の直後です。

```

3024 \def\@item[#1]{%
3025   \if@noperitem
3026     \@donoperitem
3027   \else
3028     \if@inlabel
3029       \indent \par
3030     \fi
3031     \ifhmode
3032       \unskip\unskip \par
3033     \fi
3034     \if@newlist
3035       \if@nobreak
3036         \@nbitem
3037       \else
3038         \addpenalty\@beginparpenalty
3039         \addvspace\@topsep
3040         \addvspace{-\parskip}%
3041       \fi
3042     \else
3043       \addpenalty\@itempenalty

```

```

3044     \addvspace\itemsep
3045     \fi
3046     \global\@inlabeltrue
3047     \fi
3048     \everypar{%
3049         \@minipagefalse
3050         \global\@newlistfalse
3051         \if@inlabel
3052             \global\@inlabelfalse
3053             {\setbox\z@\lastbox
3054              \ifvoid\z@
3055                \kern-\itemindent
3056              \fi}%
3057         \box\@labels
3058         \penalty\z@
3059     \fi
3060     \if@nobreak
3061         \@nobreakfalse
3062         \clubpenalty \@M
3063     \else
3064         \clubpenalty \@clubpenalty
3065         \everypar{\everyparhook}%
3066     \fi
3067     \everyparhook}%
3068     \if@noitemarg
3069         \@noitemargfalse
3070     \if@nmbrlist
3071         \refstepcounter\@listctr
3072     \fi
3073     \fi
3074     \sbox\@tempboxa{\makelabel{#1}}%
3075     \global\setbox\@labels\hbox{%
3076         \unhbox\@labels
3077         \hskip \itemindent
3078         \hskip -\labelwidth
3079         \hskip -\labelsep
3080         \ifdim \wd\@tempboxa >\labelwidth
3081             \box\@tempboxa
3082         \else
3083             \hbox to\labelwidth {\unhbox\@tempboxa}%
3084         \fi
3085         \hskip \labelsep}%
3086     \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。

```

3087 \def\@afterheading{%
3088     \@nobreaktrue
3089     \everypar{%

```

```

3090 \if@nobreak
3091 \clubpenalty \@clubpenalty \@M
3092 \if@afterindent \else
3093 {\setbox\z@\lastbox}%
3094 \fi
3095 \else
3096 \clubpenalty \@clubpenalty
3097 \everypar{\everyparhook}%
3098 \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

```

3100 \fi

```

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

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

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

```

3101 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3102 \AtEndOfPackage{%
3103 \def\@gnewline #1{%
3104 \ifvmode
3105 \nolnerr
3106 \else
3107 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3108 \jsInhibitGlue \ignorespaces
3109 \fi}
3110 }
3111 \fi

```

12 いろいろなロゴ

$\text{I}\text{A}\text{T}\text{E}\text{X}$ 関連のロゴを作り直します。

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

`BXJS` クラスでも `jslogo` オプション指定の場合に `jslogo` パッケージを読み込むように

した。ただし JS クラスと異なり、既定では読み込まない。

※`\小`、`\上小` の制御綴は定義しない。

```
3112 \if@jslogo
3113   \IfFileExists{jslogo.sty}{%
3114     \RequirePackage{jslogo}%
3115   }{%
3116     \ClassWarningNoLine{bxjs@clsname}
3117       {The package 'jslogo' is not installed.\MessageBreak
3118        It is included in the recent release of\MessageBreak
3119        the 'jsclasses' bundle}
3120   }
3121 \fi
```

13 amsmath との衝突の回避

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

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

[2016-11-19] 本家の `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。

```
3122 %\let\ltx@ifnextchar\@ifnextchar
3123 %\def\ProvidesFile#1{%
3124 %  \begingroup
3125 %    \catcode`\ 10 %
3126 %    \ifnum \endlinechar<256 %
3127 %      \ifnum \endlinechar>\m@ne
3128 %        \catcode\endlinechar 10 %
3129 %      \fi
3130 %    \fi
3131 %    \@makeother\/%
3132 %    \@makeother\&%
3133 %    \ltx@ifnextchar [{\@providesfile{#1}}{\@providesfile{#1} []}]}
```

14 初期設定

■いろいろな語

```
\prepartname
\postpartname 3134 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3135 \newcommand{\postpartname}{\if@english\else 部\fi}
\postchaptername
\presectionname
\postsectionname
```

```

3136 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
3137 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
3138 \newcommand{\presectionname}{}% 第
3139 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3140 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3141 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3142 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname

\bibname 3143 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3144 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3145 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 3146 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3147 %<jspf>\newcommand{\figurename}{Fig.~}
3148 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3149 %<jspf>\newcommand{\tablename}{Table~}

\appendixname

\abstractname 3150 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3151 \newcommand{\appendixname}{\if@english \else 付録\fi}
3152 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

環境変数 `SOURCE_DATE_EPOCH` / `FORCE_SOURCE_DATE` が設定されている場合は“今日”が過去・未来の日付になる可能性がある。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。

```

3153 \@tempwafalse
3154 \if p\jsEngine \@tempwattrue \fi
3155 \if n\jsEngine \@tempwattrue \fi
3156 \if@tempswa \expandafter\@firstoftwo
3157 \else \expandafter\@secondoftwo
3158 \fi
3159 {%
3160 % 欧文 8bitTeX の場合
3161 \newif\ifjsSeireki \jsSeirekitrue
3162 \def\bxjs@decl@Seireki@cmds{%
3163   \def\西暦{\jsSeirekitrue}%
3164   \def\和暦{\jsSeirekifalse}}
3165 \def\Seireki{\jsSeirekitrue}

```

```

3166 \def\Wareki{\jsSeirekifalse}
3167 \def\bxjs@if@use@seireki{%
3168   \ifjsSeireki \expandafter\@firstoftwo
3169   \else \expandafter\@secondoftwo \fi}
3170 }{%
3171 \newif\if 西暦 \西暦 true
3172 \def\bxjs@decl@Seireki@cmds{%
3173   \def\西暦{\西暦 true}%
3174   \def\和暦{\西暦 false}}
3175 \def\Seireki{\西暦 true}
3176 \def\Wareki{\西暦 false}
3177 \def\bxjs@if@use@seireki{%
3178   \if 西暦 \expandafter\@firstoftwo
3179   \else \expandafter\@secondoftwo \fi}
3180 }
3181 \bxjs@decl@Seireki@cmds
3182 \let\bxjs@unxp\@firstofone \let\bxjs@onxp\@firstofone
3183 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded
3184   \def\bxjs@onxp{\unexpanded\expandafter}}
3185 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3186   \def\bxjs@iai{\noexpand~}
3187 \else \def\bxjs@iai{}
3188 \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。

```

3189 \newcount\bxjs@jyear
3190 \let\bxjs@gengo\@empty \bxjs@jyear\year
3191 \@tempcnta=\the\year\two@digits\month\two@digits\day\relax
3192 \ifnum\@tempcnta<19700101
3193 \else\ifnum\@tempcnta<19890108
3194   \def\bxjs@gengo{\昭和}\advance\bxjs@jyear-1925\relax
3195   \let\heisei\bxjs@jyear
3196 \else\ifnum\@tempcnta<20190501
3197   \def\bxjs@gengo{\平成}\advance\bxjs@jyear-1988\relax
3198   \let\heisei\bxjs@jyear
3199 \else

```

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

```

3200 \IfFileExists{bxwareki.sty}{\IfFileExists{bxwareki2019.def}{%
3201   \RequirePackage{bxwareki}[]%
3202   \let\bxjs@gengo\WarekiGengo
3203 }{}{}%
3204 \if 1\jsEngine \ifx\bxjs@gengo\@empty
3205   \toks@{}
3206   \directlua{pcall(function()
3207     tex.toks[0] = bxjs.get_new_gengo()

```



```

3208     end)}}
3209     \edef\bxjs@gengo{\the\toks@}
3210     \fi\fi

    新元号が既知の場合にのみ、和暦の年を求める。

3211     \ifx\bxjs@gengo\@empty\else \advance\bxjs@jyear-2018\relax \fi
3212 \fi\fi\fi
3213 \edef\jyear{%
3214     \ifx\bxjs@gengo\@empty \the\bxjs@jyear\bxjs@iai
3215     \else \bxjs@onxp{\bxjs@gengo}%
3216         \ifnum\bxjs@jyear=\@ne \bxjs@unxp{元}%
3217         \else \bxjs@iai\the\bxjs@jyear\bxjs@iai
3218         \fi
3219     \fi}

```

`\today` 英語、西暦、和暦で場合分けをする。

```

3220 \edef\bxjs@today{%
3221     \if@english
3222         \ifcase\month\or
3223             January\or February\or March\or April\or May\or June\or
3224             July\or August\or September\or October\or November\or December\fi
3225         \space\number\day, \number\year
3226     \else
3227         \noexpand\bxjs@if@use@seireki{%
3228             \number\year\bxjs@iai\bxjs@unxp{年}%
3229             \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3230             \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3231         }{%
3232             \bxjs@onxp{\jyear}\bxjs@unxp{年}%
3233             \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3234             \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3235         }%
3236     \fi}
3237 \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。

```

3238 \AtBeginDocument{%
3239     \ifx\bb1@jpn@Seirekitrue\@undefined\else
3240         \bxjs@decl@Seireki@cmds
3241         \g@addto@macro\datejapanese{%
3242             \let\today\bxjs@today}%
3243     \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)

```
3244 \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。

```
3245 %<slide>\pagestyle{empty}%
3246 %<article|report>\pagestyle{plain}%
3247 %<book>\pagestyle{headings}%
3248 \pagenumbering{arabic}
3249 \if@twocolumn
3250   \twocolumn
3251   \sloppy
3252   \flushbottom
3253 \else
3254   \onecolumn
3255   \raggedbottom
3256 \fi
3257 %<*slide>
3258 \renewcommand\familydefault{\sfdefault}
3259 \raggedright
3260 %</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。

```
3261 \catcode`\?=12
3262 \ifx\bxjs@jadriver\relax\else
3263 \input{bxjsja-\bxjs@jadriver.def}
3264 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
3265 \bxjs@restore@jltrcc
3266 %</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。

```
3267 %<*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

```
3268 %<*minimal>
```

```
3269 %% 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.
```

```
3270 \def\DeclareJaTextFontCommand#1#2{%
```

```
3271   \DeclareRobustCommand#1[1]{%
```

```
3272     \relax
```

```
3273     \ifmmode \expandafter\nfss@text \fi
```

```
3274     {#2##1}}%
```

```
3275 }
```

```
\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。
```

```
3276 \def\DeclareJaMathFontCommand#1#2{%
```

```
3277   \DeclareRobustCommand#1[1]{%
```

```
3278     \relax
```

```
3279     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3280     \nfss@text{\fontfamily\familydefault
```

```
3281       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3282       #2##1}}%
```

```

3283 }%
3284 }

```

`\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。

```

3285 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3286 \@onlypreamble\bxjs@if@sf@default
3287 \def\bxjs@if@sf@default#1{%
3288   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3289   \AtBeginDocument{%
3290     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3291 }

```

`\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。

```

3292 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3293 \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。

```

3294 \def\jsLetHeadChar#1#2{%
3295   \begingroup
3296   \escapechar=`\ %
3297   \let\bxjs@tmpa={% brace-match-hack
3298   \bxjs@let@hchar@exp#2}%
3299   \endgroup
3300   \let#1\bxjs@g@tmpa}
3301 \def\bxjs@let@hchar@exp{%
3302   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3303 \def\bxjs@let@hchar@exp@a{%
3304   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3305   \bxjs@let@hchar@out\let\relax
3306   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3307   \bxjs@let@hchar@out\let\space%
3308   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3309   \bxjs@let@hchar@out\let\@backslashchar
3310   }{\bxjs@let@hchar@exp@b}}}%
3311 \def\bxjs@let@hchar@exp@b#1{%
3312   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3313 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3314   %\message{<#1#2>}%
3315   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3316 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3317 \bxjs@let@hchar@out\let\relax
3318 }{%else
3319 \expandafter\bxjs@let@hchar@exp
3320 }%
3321 }{%else
3322 \bxjs@let@hchar@chr#1%
3323 }}
3324 \def\bxjs@let@hchar@chr#1{%
3325 \bxjs@let@hchar@out\def{{#1}}}%
3326 \def\bxjs@let@hchar@out#1#2{%
3327 \global#1\bxjs@g@tmpa#2\relax
3328 \toks@{\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3329 \chardef\bxjs@let@hchar@csta=128
3330 \chardef\bxjs@let@hchar@cstb=192
3331 \chardef\bxjs@let@hchar@cstc=224
3332 \chardef\bxjs@let@hchar@cstd=240
3333 \chardef\bxjs@let@hchar@cste=248
3334 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3335 \def\bxjs@let@hchar@chr@ue#1{%
3336 \@tempcnta=#1\relax
3337 %\message{\the\@tempcnta}%
3338 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3339 \bxjs@let@hchar@chr@ue@a#1%
3340 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3341 \bxjs@let@hchar@out\let\relax
3342 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3343 \bxjs@let@hchar@chr@ue@b
3344 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3345 \bxjs@let@hchar@chr@ue@c
3346 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3347 \bxjs@let@hchar@chr@ue@d
3348 }{%else
3349 \bxjs@let@hchar@out\let\relax
3350 }}}}
3351 \def\bxjs@let@hchar@chr@ue@a#1{%
3352 \bxjs@let@hchar@out\def{{#1}}}%
3353 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3354 \bxjs@let@hchar@out\def{{#1#2}}}%
3355 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3356 \bxjs@let@hchar@out\def{{#1#2#3}}}%
3357 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3358 \bxjs@let@hchar@out\def{{#1#2#3#4}}}%

```

B.2 (u)pTeX 用の設定

```
3359 \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。

```
3360 \def\bxjs@let@hchar@chr@pp#1#2{%
3361   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3362 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3363   %\message{(\meaning#3:\meaning#4)}%
3364   \bxjs@cond\if#1k\fi{%
3365     \bxjs@let@hchar@out\def{#{#4}}%
3366   }{%else
3367     \bxjs@let@hchar@chr@ue#3#4%
3368   }}
3369 \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。

```
3370 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3371 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3372 \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。

```
3373 \@onlypreamble\bxjs@declarefontshape
3374 \ifjsWithupTeX
3375 \def\bxjs@declarefontshape{%
3376   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3377   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3378   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3379   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3380 }
3381 \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。

```
3382 \else
3383 \def\bxjs@declarefontshape{%
3384   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3385   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3386   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3387 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3388 }
3389 \def\bxjs@sizereference{jis}
3390 \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。

```

3391 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
3392   \def\bxjs@y{#5}}
3393 \ifjsWithpTeXng \def\bxjs@y{10}%
3394 \else
3395 \expandafter\expandafter\expandafter\bxjs@tmpa
3396 \expandafter\string\the\jfont\relax
3397 \fi
3398 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3399               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3400   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
3401    \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。

```

3402 \begingroup
3403 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3404 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3405 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3406 % 幅が丁度 10pt なら補正は不要
3407 \ifdim\wd\z@=10pt
3408   \global\let\bxjs@scale\jsScale
3409 \else
3410 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3411   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3412   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3413   \bxjs@invscale\@tempdima\bxjs@tmpa
3414   \xdef\bxjs@scale{\strip@pt\@tempdima}
3415 \fi
3416 \endgroup
3417 %\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。

```

3418 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3419 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3420 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3421 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```



```

3422 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3423 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3424 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3425 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3426 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3427 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3428 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3429 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3430 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3431 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3432 \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。

```

3433 \DeclareRobustCommand\rmfamily
3434   {\not@math@alphabet\rmfamily\mathrm
3435    \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3436 \DeclareRobustCommand\sffamily
3437   {\not@math@alphabet\sffamily\mathsf
3438    \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3439 \DeclareRobustCommand\ttfamily
3440   {\not@math@alphabet\ttfamily\mathtt
3441    \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3442 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3443 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3444 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3445 \fi
3446 \bxjs@if@sf@default{%
3447   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3448 \selectfont

```

`\bxjs@parse@qh` の処理は不要になるので無効化する。

```

3449 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3450 \let\bxjs@parse@qh@a\@undefined
3451 \let\bxjs@parse@qh@b\@undefined

```

■パラメタの設定

```

3452 \prebreakpenalty\jis"2147=10000
3453 \postbreakpenalty\jis"2148=10000
3454 \prebreakpenalty\jis"2149=10000
3455 \inhibitxspcode`!=1
3456 \inhibitxspcode`〒=2
3457 \xspcode`+=3
3458 \xspcode`\%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3459 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3460   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```
3461 \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。

```
3462 \begingroup
```

```
3463 \catcode`\!=0
```

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

```
3464 \gdef\bxjs@ptex@dir{%
```

```
3465   !iftdir t%
```

```
3466   !else!ifydir y%
```

```
3467   !else ?%
```

```
3468   !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。

```
3469 % 古い \@makefnmark の定義
```

```
3470 \long\def\bxjs@tmpa{\hbox{%
```

```
3471   !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
```

```
3472   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
```

```
3473 \ifx\@makefnmark\bxjs@tmpa
```

```
3474 \long\gdef\@makefnmark{%
```

```
3475   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
```

```
3476   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
```

```
3477 \fi
```

```
3478 \endgroup
```

B.3 pdfTeX 用の処理

```
3479 \else\ifx p\jsEngine
```

```
3480 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

```
3481 \@onlypreamble\bxjs@cjk@loaded
```

```
3482 \def\bxjs@cjk@loaded{%
```

```
3483   \def\@footnotemark{%
```

```
3484     \leavevmode
```

```
3485     \ifhmode
```

```
3486       \edef\@xsf{\the\spacefactor}%
```

```
3487       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
```

```
3488         \unkern\unkern
```

```
3489         \ifdim\lastskip>\z@ \unskip \fi
```

```
3490       \fi\fi
```

```
3491       \nobreak
```

```

3492 \fi
3493 \@makefnmark
3494 \ifhmode \spacefactor\@x@sf \fi
3495 \relax}%
3496 \let\bxjs@cjk@loaded\relax
3497 }
3498 \AtBeginDocument{%
3499 \ifpackageloaded{CJK}{%
3500 \bxjs@cjk@loaded
3501 }{}%
3502 }

```

B.4 X_YTeX 用の処理

```

3503 \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。

```

3504 \def\bxjs@let@hchar@chr#1{%
3505 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3506 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3507 \bxjs@let@hchar@chr@xe
3508 }{\bxjs@let@hchar@out\def{{#1}}}}
3509 \def\bxjs@let@hchar@chr@xe#1{%
3510 \lccode`0=`#1\relax
3511 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3512 \ifx\XeTeXgenerateactualtext\@undefined\else
3513 \def\bxjs@do@precisetext{%
3514 \XeTeXgenerateactualtext=\@ne}
3515 \fi

```

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

```

3516 \@onlypreamble\bxjs@do@simplejasetup
3517 \def\bxjs@do@simplejasetup{%
3518 \ifnum\XeTeXinterchartokenstate>\z@
3519 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3520 \jsSimpleJaSetup
3521 \ClassInfo\bxjs@clsname
3522 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3523 \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

3524 \newcommand*{\jsSimpleJaSetup}{%
3525 \XeTeXlinebreaklocale "ja"\relax
3526 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3527 \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理（エンジン共通）

3528 \fi\fi\fi

simplejasetup オプションの処理。

3529 \ifx\bxjs@do@simplejasetup\@undefined\else

3530 \AtBeginDocument{%

3531 \ifbxjs@simplejasetup

3532 \bxjs@do@simplejasetup

3533 \fi}

3534 \fi

precisetext オプションの処理。

3535 \ifbxjs@precisetext

3536 \ifx\bxjs@do@precisetext\@undefined

3537 \ClassWarning\bxjs@clsname

3538 {The current engine does not support the\MessageBreak

3539 'precisetext' option\@gobble}

3540 \else

3541 \bxjs@do@precisetext

3542 \fi

3543 \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。

3544 \@onlypreamble\bxjs@check@everyparhook

3545 \def\bxjs@check@everyparhook{%

3546 \ifx\jsInhibitGlueAtParTop\@empty

3547 \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%

3548 \ifx\everyparhook\bxjs@tmpa

3549 \let\everyparhook\@empty

3550 \fi

3551 \fi}

3552 \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。

3553 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

まず \everypar を“乗っ取る” 処理を行う。

3554 \let\bxjs@everypar\everypar

3555 \newtoks\everypar

3556 \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。

3557 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%

3558 \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。

3559 \ifbxjs@fancyhdr

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

3560 \@onlypreamble\bxjs@adjust@fancyhdr

3561 \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。

3562 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%

3563 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%

3564 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

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

3566 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3567 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3568 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%

3569 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%

3570 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3571 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3572 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3573 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3574 \def\bxjs@tmpa{\rm\thepage\strut}%

3575 \def\bxjs@tmpb{\thepage\strut}%

3576 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi

3577 \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。

3578 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth

3579 \setlength{\@tempdima}{\fullwidth-\textwidth}%

3580 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%

3581 }\bxjs@tmpa

3582 \fi\fi

3583 \PackageInfo{bxjs}{clsname

3584 {Patch to fancyhdr is applied\@gobble}}

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

3585 \def\bxjs@pagestyle@hook{%

3586 \@ifpackageloaded{fancyhdr}{%

3587 \bxjs@adjust@fancyhdr

3588 \global\let\bxjs@adjust@fancyhdr\relax

```
3589 }{}}
```

`\pagestyle` にフックを入れ込む。

```
3590 \let\bxjs@org@pagestyle\pagestyle
```

```
3591 \def\pagestyle{%
```

```
3592   \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。

```
3593 \AtBeginDocument{%
```

```
3594   \bxjs@pagestyle@hook
```

```
3595   \global\let\bxjs@pagestyle@hook\relax}
```

■和文空白命令

```
3596 \ifbxjs@jaspace@cmd
```

`\jaenspace` 半角幅の水平空き。

```
3597   \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。

```
3598   \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。

```
3599   \def\_{\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3600   \DeclareRobustCommand*{\jaspace}[1]{%
```

```
3601     \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
```

```
3602     \ClassError\bxjs@clsname
```

```
3603     {Unknown jaspac: #1}{\@eha}%
```

```
3604   \else
```

```
3605     \csname bxjs@jaspace@@#1\endcsname
```

```
3606   \fi}
```

```
3607   \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
```

```
3608   \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
```

```
3609   \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
```

```
3610 \fi
```

終わり。

```
3611 \fi
```

以上で終わり。

```
3612 %</minimal>
```

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
3613 %<*standard>
3614 %% このファイルは日本語文字を含みます
3615 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3616 \bxjs@simplejasetupfalse
```

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

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

```
3617 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。

3618 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3619 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3620 \define@key{bxjsStd}{jis2004}[true]{%
3621   \bxjs@set@keyval{jis2004}{#1}{}}
```

`\ifbxjs@jp@units` 和文用単位 (`zw`、`zh`、`(true)Q`、`(true)H`) を使えるようにするか。

```
3622 \newif\ifbxjs@jp@units

units オプションの処理。

3623 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3624 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3625 \define@key{bxjsStd}{units}[true]{%
3626   \bxjs@set@keyval{units}{#1}{}}
```

`\bxjs@jp@font` フォントパッケージの追加オプション。

```
3627 \let\bxjs@jp@font\@empty
```

font オプションの処理。

```
3628 \define@key{bxjsStd}{font}{%
```

```
3629 \edef\bxjs@jp@font{#1}}
```

実際の `japaram` の値を適用する。

```
3630 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
```

```
3631 \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。

```
3632 \ifbxjs@jp@jismmiv
```

```
3633 \g@addto@macro\@classoptionslist{,jis2004}
```

```
3634 % \ifpackagewith 判定への対策
```

```
3635 \PassOptionsToPackage{jis2004}{otf}
```

```
3636 \fi
```

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

```
3637 \if j\jsEngine
```

```
3638 \bxjs@jp@unitsfalse
```

```
3639 \fi
```

`units` パラメタが有効の場合は、`bxcalc` パッケージの `\usepTeXunits` 命令を実行して和文用単位を有効化する。

```
3640 \ifbxjs@jp@units
```

```
3641 \IfFileExists{bxcalc.sty}{%
```

```
3642 \RequirePackage{bxcalc}[2018/01/28]%v1.0a
```

```
3643 \ifx\usepTeXunits\@undefined
```

```
3644 \PackageWarningNoLine\bxjs@clsname
```

```
3645 {Cannot support pTeX units (zw etc.), since\MessageBreak
```

```
3646 the package 'bxcalc' is too old}%
```

```
3647 \bxjs@jp@unitsfalse
```

```
3648 \else \usepTeXunits
```

```
3649 \fi
```

```
3650 }{%else
```

```
3651 \PackageWarningNoLine\bxjs@clsname
```

```
3652 {Cannot support pTeX units (zw etc.), since\MessageBreak
```

```
3653 the package 'bxcalc' is unavailable}%
```

```
3654 \bxjs@jp@unitsfalse
```

```
3655 }
```

```
3656 \fi
```

`bxcalc` で和文用単位をサポートした場合は、`\bxjs@parse@qh` の処理は不要になるので無効化する。

```
3657 \ifbxjs@jp@units
```



```

3658 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3659 \let\bxjs@parse@qh@a\@undefined
3660 \let\bxjs@parse@qh@b\@undefined
3661 \fi

```

`\bxjs@let@lenexpr \bxjs@let@lenexpr\CS{(長さ式)}` : 長さ式に `bxcalc` の展開を適用した結果のトークン列を `\CS` に代入する。

```

3662 \ifbxjs@jp@units
3663   \def\bxjs@let@lenexpr#1#2{%
3664     \edef#1{#2}%
3665     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3666 \else
3667   \def\bxjs@let@lenexpr{\edef}
3668 \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。

```

3669 \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 3670 \@tempdima=0.25mm
3671 \protected\edef\jq{\dimexpr\the\@tempdima\relax}
3672 \let\jH\jq

```

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

```

\trueH 3673 \ifjsc@mag
3674   \@tempdimb=\jsBaseFontSize\relax
3675   \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3676   \@tempdima=2.5mm
3677   \bxjs@invscale\@tempdima\bxjs@tmpa
3678   \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3679   \@tempdima=10pt
3680   \bxjs@invscale\@tempdima\bxjs@tmpa
3681   \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3682 \else \let\trueQ\jq \let\bxjs@truept\p@
3683 \fi
3684 \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。

```

3685 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3686 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3687 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3688 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}

```

3689 \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。

```
3690 \def\jafontsize#1#2{%
3691   \begingroup
3692     \bxjs@jafontsize@a{#1}%
3693     \@tempdimb\jsInverseScale\@tempdima
3694     \bxjs@jafontsize@a{#2}%
3695     \xdef\bxjs@g@tmpa{%
3696       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3697   \endgroup\bxjs@g@tmpa}
3698 \def\bxjs@jafontsize@a#1{%
3699   \bxjs@parse@qh{#1}%
3700   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3701   \@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。

```
3702 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3703 \newcommand*\setkanjiskip[1]{%
3704   \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3705   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3706 \newcommand*\getkanjiskip{%
3707   \bxjs@kanjiskip}
```

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

```
3708 \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。(pT_EX 以外)

```
\bxjs@disable@kanjiskip 3709 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3710   \bxjs@kanjiskip@enabledtrue
3711   \bxjs@reset@kanjiskip}
3712 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3713   \bxjs@kanjiskip@enabledfalse
3714   \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。

```
3715 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3716   \ifbxjs@kanjiskip@enabled
3717     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3718   \else \@tempskipa\z@
```

```

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

```

3740 \g@addto@macro\jsResetDimen{%
3741 \bxjs@reset@kanjiskip
3742 \bxjs@reset@xkanjiskip}
3743 \let\bxjs@apply@kanjiskip\relax
3744 \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。

```

3745 \@onlypreamble\bxjs@adjust@jafont
3746 \def\bxjs@adjust@jafont#1{%
3747 \ifx\jsJaFont\bxjs@auto
3748 \bxjs@get@kanjiEmbed
3749 \ifx\bxjs@kanjiEmbed\relax
3750 \let\bxjs@tmpa\@empty
3751 \else
3752 \let\bxjs@tmpa\bxjs@kanjiEmbed
3753 \fi
3754 \else
3755 \let\bxjs@tmpa\jsJaFont

```

```

3756 \fi
3757 \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3758 \ClassWarningNoLine\bxjs@clsname
3759 {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3760 not available on the current situation}%
3761 \let\bxjs@tmpa\@empty
3762 \fi\fi
3763 }
3764 \def\bxjs@@auto{auto}
3765 \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。

```

3766 \let\bxjs@kanjiEmbed\relax

```

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

```

3767 \@onlypreamble\bxjs@get@kanjiEmbed
3768 \def\bxjs@get@kanjiEmbed{%
3769 \begingroup\setbox\z@=\hbox{%
3770 \global\let\bxjs@g@tmpa\relax
3771 \endlinechar\m@ne
3772 \let\do\@makeother\dospecials
3773 \catcode32=10 \catcode12=10 %form-feed
3774 \let\bxjs@tmpa\@empty
3775 \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3776 \ifeof\@inputcheck\else
3777 \read\@inputcheck to\bxjs@tmpa
3778 \closein\@inputcheck
3779 \fi
3780 \ifx\bxjs@tmpa\@empty\else
3781 \openin\@inputcheck="\bxjs@tmpa"\relax
3782 \@tempswatrue
3783 \loop\if@tempswa
3784 \read\@inputcheck to\bxjs@tmpa
3785 \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3786 \ifx\bxjs@tmpb\relax\else
3787 \global\let\bxjs@g@tmpa\bxjs@tmpb
3788 \@tempswafalse
3789 \fi
3790 \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3791 \ifx\bxjs@tmpb\relax\else
3792 \global\let\bxjs@g@tmpa\bxjs@tmpb
3793 \@tempswafalse
3794 \fi
3795 \ifeof\@inputcheck \@tempswafalse \fi
3796 \repeat
3797 \fi
3798 }\endgroup
3799 \let\bxjs@kanjiEmbed\bxjs@g@tmpa

```

```

3800 }
3801 \@onlypreamble\bxjs@get@ke@a
3802 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3803   \ifx$#1$\def\bxjs@tmpb{#2}%
3804   \else \let\bxjs@tmpb\relax
3805   \fi}
3806 \@onlypreamble\bxjs@get@ke@b
3807 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3808   \ifx$#1$\def\bxjs@tmpb{#2}%
3809   \else \let\bxjs@tmpb\relax
3810   \fi}

```

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

```

3811 \newcommand*\jachar[1]{%
3812   \begingroup
3813     \jsLetHeadChar\bxjs@tmpa{#1}%
3814     \ifx\bxjs@tmpa\relax
3815       \ClassWarningNoLine\bxjs@clsname
3816       {Illegal argument given to \string\jachar}%
3817     \else
3818       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3819     \fi
3820   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3821 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3822 \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。

```

3823 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3824 \@onlypreamble\bxjs@fix@hyperref@unicode
3825 \def\bxjs@fix@hyperref@unicode#1{%
3826   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3827   \@namedef{KV@Hyp@bxjs/hook}##1{%
3828     \KV@Hyp@unicode{##1}%
3829     \def\KV@Hyp@unicode####1{%
3830       \expandafter\ifx\csname if##1\expandafter\endcsname
3831       \csname if####1\endcsname\else
3832       \ClassWarningNoLine\bxjs@clsname
3833       {Blcoked hyperref option 'unicode=####1'}%
3834     \fi
3835   }%

```

```

3836 }%
3837 }

```

`\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。

```

3838 \@onlypreamble\jsCheckHyperrefUnicode
3839 \let\jsCheckHyperrefUnicode\@empty
3840 \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。

```

3841 \@onlypreamble\bxjs@check@hyperref@unicode
3842 \def\bxjs@check@hyperref@unicode#1{%
3843   \g@addto@macro\jsCheckHyperrefUnicode{%
3844     \@tempwafalse
3845     \begingroup
3846       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3847         \aftergroup\@tempwattrue \fi
3848       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3849         \csname if#1\endcsname
3850       \aftergroup\@tempwattrue \fi
3851     \endgroup
3852     \if@tempwa\else
3853       \ClassError\bxjs@clsname
3854         {The value of hyperref 'unicode' key is not suitable\MessageBreak
3855          for the present engine (must be #1)}%
3856       {\@ehc}%
3857     \fi}}

```

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

```

3858 \@onlypreamble\bxjs@urgent@special
3859 \def\bxjs@urgent@special#1{%
3860   \AtBeginDvi{\special{#1}}%
3861   \AtBeginDocument{%
3862     \@ifpackageloaded{atbegshi}{%
3863       \begingroup
3864         \toks\z@{\special{#1}}%
3865         \toks\tw@{\expandafter{\AtBegShi@HookFirst}}%
3866         \xdef\AtBegShi@HookFirst{\the\toks@the\toks\tw@}%
3867       \endgroup
3868     }{}%
3869   }%
3870 }

```

`\bxjs@resolve@jafont@paren` jafont パラメタ値内の () を解決する。

```

3871 \def\bxjs@resolve@jafont@paren#1{%
3872   \def\bxjs@tmpb{\let#1}%
3873   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3874 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3875   \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3876   \else

```

```

3877 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3878 \bxjs@tmpb\bxjs@tmpa
3879 \fi}

```

C.2 pTeX 用設定

```
3880 \if j\jsEngine
```

■ 共通命令の実装

```

3881 \def\bxjs@apply@kanjiskip{%
3882 \kanjiskip\@tempskipa}
3883 \def\bxjs@apply@xkanjiskip{%
3884 \xkanjiskip\@tempskipa}

```

\jaJaChar のサブマクロ。

```

3885 \def\bxjs@jachar#1{%
3886 \bxjs@jachar@a#1...\@nil}
3887 \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。

```
3888 \ifx.#2#1%
```

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

```

3889 \else\ifx.#3%
3890 \@tempcnta`#1 \multiply\@tempcnta64
3891 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3892 \bxjs@jachar@b
3893 \else\ifx.#4%
3894 \@tempcnta`#1 \multiply\@tempcnta64
3895 \advance\@tempcnta`#2 \multiply\@tempcnta64
3896 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3897 \bxjs@jachar@b
3898 \else
3899 \@tempcnta`#1 \multiply\@tempcnta64
3900 \advance\@tempcnta`#2 \multiply\@tempcnta64
3901 \advance\@tempcnta`#3 \multiply\@tempcnta64
3902 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3903 \bxjs@jachar@b
3904 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3905 \ifjsWithupTeX
3906 \def\bxjs@jachar@b{\kchar\@tempcnta}
3907 \else
3908 \def\bxjs@jachar@b{%
3909 \ifx\bxUInt\@undefined\else
3910 \bxUInt{\@tempcnta}%
3911 \fi}
3912 \fi

```

和欧文間空白の命令 `\jathinspace` の実装。

```
3913 \ifbxjs@jaspace@cmd
3914   \def\jathinspace{\hskip\xkanjiskip}
3915 \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。

```
3916 \ifbxjs@jp@jismmiv
3917   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3918   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3919 \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。

```
3920 \let\bxjs@jafont@paren\@firstofone
3921 \let\bxjs@tmpa\jsJaFont
3922 \ifx\bxjs@tmpa\bxjs@@auto
3923   \let\bxjs@tmpa\@empty
3924 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3925   \def\bxjs@tmpa{noembed}
3926 \fi\fi
3927 \bxjs@resolve@jafont@paren\bxjs@tmpa
3928 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3929 \ifx\bxjs@tmpa\@empty\else
3930   \edef\bxjs@next{%
3931     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3932   }\bxjs@next
3933 \fi
```

■`otf` パッケージ対策 インストールされている `otf` パッケージが `scale` オプションに対応している場合は `scale=(\jsScale の値)` を事前に `otf` に渡す。

※ `scale` 対応は 1.7b6 版 [2013/11/17] から。

※ `otf.sty` の中に「`\RequirePackage{keyval}`」の行が存在するかにより判定している。
(もっといい方法はないのか……。)

```
3934 \begingroup
3935   \global\let\@gtempa\relax
3936   \catcode`\|=0 \catcode`\=12
3937   |def|bxjs@check#1|@nil{%
3938     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3939   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3940     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3941   |catcode`\|=0 \catcode`\=12
3942   \def\bxjs@check@b#1keyval#2\@nnil{%
3943     \ifx$#2$\else
```



```

3944 \xdef\@gtempa{%
3945 \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3946 \fi}
3947 \@firstofone{%
3948 \catcode10=12 \endlinechar\m@ne
3949 \let\do\@makeother \dospecials \catcode32=10
3950 \openin\@inputcheck=otf.sty\relax
3951 \@tempswatrue
3952 \loop\if@tempswa
3953 \ifeof\@inputcheck \@tempswafalse \fi
3954 \if@tempswa
3955 \read\@inputcheck to\bxjs@line
3956 \expandafter\bxjs@check\bxjs@line\@nil
3957 \fi
3958 \repeat
3959 \closein\@inputcheck
3960 \endgroup}
3961 \@gtempa

```

■hyperref 対策 unicode にはいけない。

```

3962 \ifbxjs@hyperref@enc
3963 \bxjs@check@hyperref@unicode{false}
3964 \fi

```

tounicode special 命令を出力する。

```

3965 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3966 \else\ifjsWithpTeXng T\else F\fi\fi T%
3967 \ifnum\jis"2121="A1A1 %euc
3968 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3969 \else\ifnum\jis"2121="8140 %sjis
3970 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3971 \else\ifnum\jis"2121="3000 %uptex
3972 \ifbxjs@bigcode
3973 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3974 \PassOptionsToPackage{bigcode}{pxjahyper}
3975 \else
3976 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3977 \fi
3978 \fi\fi\fi
3979 \let\bxToUnicodeSpecialDone=t
3980 \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。

```

3981 \ifx f\bxjs@enablejfam\else
3982 \@enablejfamtrue
3983 \fi

```

実際に和文用の数式ファミリを設定を行う。

```

3984 \if@enablejfam
3985   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3986   \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3987   \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3988   \jfam\symmincho
3989   \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3990   \AtBeginDocument{%
3991     \ifx\reDeclareMathAlphabet\undefined\else
3992       \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3993       \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3994       \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3995     \fi}
3996 \fi

```

C.3 pdfTeX 用設定 : CJK + bxcjkatype

```

3997 \else\if p\jsEngine

```

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

※ Pandoc モードでは autotilde を指定しない。

```

3998 \bxjs@adjust@jafont{f}
3999 \let\bxjs@jafont@paren\@firstofone
4000 \bxjs@resolve@jafont@paren\bxjs@tmpa
4001 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4002 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4003 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4004   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4005 \fi
4006 \edef\bxjs@next{%
4007   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4008 }\bxjs@next
4009 \bxjs@cjk@loaded

```

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

※取りあえず固定はしない。

```

4010 \ifbxjs@hyperref@enc
4011   \PassOptionsToPackage{unicode}{hyperref}
4012 \fi

```

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

※ bxcjkatype を whole 付きで使っていることが前提。

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

```

4013 \ifx\bxCJKtypeHyperrefPatchDone\@undefined
4014 \begingroup
4015   \CJK@input{UTF8.bdg}
4016 \endgroup
4017 \g@addto@macro\pdfstringdefPreHook{%
4018   \@nameuse{CJK@UTF8Binding}%
4019 }
4020 \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。

```

4021 \ifx\bxCJKtypeHyperrefPatchDone\@undefined
4022 \g@addto@macro\pdfstringdefPreHook{%
4023   \ifx~\bxjs@@CJKtilde
4024     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4025     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4026     \let~\@empty
4027   \fi
4028 }
4029 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4030 \def\bxjs@@tildecmd{~}
4031 \def\bxjs@LetUnexpandableSpace#1{%
4032   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4033     \bxjs@org@LetUnexpandableSpace#1%
4034   \fi}
4035 \fi

```

■ 共通命令の実装

```

4036 \newskip\jsKanjiSkip
4037 \newskip\jsXKanjiSkip
4038 \ifx\CJKecglue\@undefined
4039   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4040 \fi
4041 \let\autospacing\bxjs@enable@kanjiskip
4042 \let\noautospacing\bxjs@disable@kanjiskip
4043 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4044 \def\bxjs@apply@kanjiskip{%
4045   \jsKanjiSkip\@tempskipa
4046   \let\CJKglue\bxjs@CJKglue}
4047 \let\autoxspacing\bxjs@enable@xkanjiskip
4048 \let\noautoxspacing\bxjs@disable@xkanjiskip
4049 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4050 \def\bxjs@apply@xkanjiskip{%
4051   \jsXKanjiSkip\@tempskipa
4052   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4053 \def\bxjs@jachar#1{%
4054   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4055 \ifbxjs@jaspace@cmd
4056   \protected\def\jathinspace{\CJKecglue}
4057 \fi

```

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

```

4058 \ifx t\bxjs@enablejfam
4059   \ClassWarningNoLine\bxjs@clsname
4060     {You cannot use 'enablejfam=true', since the\MessageBreak
4061       CJK package does not support Japanese math}
4062 \fi

```

C.4 X_YTeX 用設定 : xeCJK + zxjatype

```

4063 \else\if x\jsEngine

```

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

```

4064 \RequirePackage{zxjatype}
4065 \PassOptionsToPackage{no-math}{fontspec}%!
4066 \PassOptionsToPackage{xetex}{graphicx}%!
4067 \PassOptionsToPackage{xetex}{graphics}%!
4068 \ifx\zxJaFamilyName\@undefined
4069   \ClassError\bxjs@clsname
4070     {xeCJK or zxjatype is too old}\@ehc
4071 \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。

```

4072 \bxjs@adjust@jafont{f}
4073 \let\bxjs@jafont@paren\@gobble
4074 \bxjs@resolve@jafont@paren\bxjs@tmpa
4075 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4076 \ifx\bxjs@tmpa\@empty
4077   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
4078   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
4079 \else
4080   \edef\bxjs@next{%
4081     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4082   }\bxjs@next
4083 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X_YTeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、X_YTeX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過

去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「Xe_{La}TeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```
4084 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4085   \ifbxjs@hyperref@enc
4086     \PassOptionsToPackage{unicode}{hyperref}
4087   \fi
4088 \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。

```
4089 \AtEndOfPackage{%
4090 \def\@gnewline #1{%
4091   \ifvmode \@nolnerr
4092   \else
4093     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4094     \nobreak \hskip-1sp\hskip1sp\relax
4095     \ignorespaces
4096   \fi}
4097 }
```

■共通命令の実装

```
4098 \newskip\jsKanjiSkip
4099 \newskip\jsXKanjiSkip
4100 \ifx\CJKecglue\@undefined
4101   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4102 \fi
4103 \let\autospacing\bxjs@enable@kanjiskip
4104 \let\noautospacing\bxjs@disable@kanjiskip
4105 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4106 \def\bxjs@apply@kanjiskip{%
4107   \jsKanjiSkip\@tempskipa
4108   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4109 \let\autoxspacing\bxjs@enable@xkanjiskip
4110 \let\noautoxspacing\bxjs@disable@xkanjiskip
4111 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4112 \def\bxjs@apply@xkanjiskip{%
4113   \jsXKanjiSkip\@tempskipa
4114   \xeCJKsetup{CJKecglue={\bxjs@CJKecglue}}}
```

\mcfamily、\gtfamily は本来は zxjatype の方で定義すべきであろうが、現状は暫定的

にここで定義する。

```
4115 \ifx\mcfamily\@undefined
4116   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4117   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4118 \fi
```

`\jachar` のサブマクロの実装。

```
4119 \def\bxjs@jachar#1{%
4120   \xeCJKDeclareCharClass{CJK}{`#1}\relax
4121   #1}
```

`\jathinspace` の実装。

```
4122 \ifbxjs@jaspace@cmd
4123   \protected\def\jathinspace{\CJKecglue}
4124 \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。

```
4125 \ifx t\bxjs@enablejfam
4126   \@enablejfamtrue
4127 \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。

```
4128 \if@enablejfam
4129   \xeCJKsetup{CJKmath=true}
4130 \fi
```

C.5 Lua_T_EX 用設定 : Lua_T_EX-ja

```
4131 \else\if l\jsEngine
```

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

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

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

```
4132 \let\zw\@undefined
4133 \RequirePackage{luatexja}
4134 \edef\bxjs@next{%
4135   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4136 }\bxjs@next
```

■和文フォント定義 `\jsJaFont` が指定された場合は、その値をオプションとして `luatexja-preset` を読み込む。非指定の場合は、`luatexja-preset` パッケージの `ipaex`

オプション (IPAex フォント使用) と等価な設定を用いる (luatexja-preset は読み込まない)。

```
4137 \bxjs@adjust@jafont{t}
4138 \ifx\bxjs@tmpa\bxjs@noEmbed
4139   \def\bxjs@tmpa{noembed}
4140 \fi
4141 \let\bxjs@jafont@paren\@gobble
4142 \bxjs@resolve@jafont@paren\bxjs@tmpa
4143 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4144 \ifx\bxjs@tmpa\@empty
4145   \defaultfontfeatures{ Kerning=Off }
4146   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
4147   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
4148 \else
4149   \edef\bxjs@next{%
4150     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4151   }\bxjs@next
4152 \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。

```
4153 \DeclareRobustCommand\rmfamily
4154   {\not@math@alphabet\rmfamily\mathrm
4155     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4156 \DeclareRobustCommand\sffamily
4157   {\not@math@alphabet\sffamily\mathsf
4158     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4159 \DeclareRobustCommand\ttfamily
4160   {\not@math@alphabet\ttfamily\mathtt
4161     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4162 \AtBeginDocument{%
4163   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4164   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4165   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4166 \bxjs@if@sf@default{%
4167   \renewcommand\kanjifamilydefault{\gtdefault}}
```

■和文パラメタの設定

```
4168 % 次の3つは既定値の通り
4169 %\ltjsetparameter{prebreakpenalty}={`',10000}
4170 %\ltjsetparameter{postbreakpenalty}={`",10000}
4171 %\ltjsetparameter{prebreakpenalty}={`",10000}
4172 \ltjsetparameter{jaxspmde}={`!',1}
4173 \ltjsetparameter{jaxspmde}={`〒,2}
4174 \ltjsetparameter{alxspmde}={`+,3}
4175 \ltjsetparameter{alxspmde}={`%,3}
```

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

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の `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。

```
4176 \ifx\ltjfakeparbegin\@undefined
4177   \protected\def\ltjfakeparbegin{%
4178     \ifhmode
4179       \relax\directlua{%
4180         luatexja.jfmglue.create_beginpar_node()}}
4181   \fi}
4182 \fi
```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```
4183 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4184 \begingroup
4185   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4186   \@gobble\if\def\bxjs@tmpa{\@{\everypar}\fi}
4187   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4188     \@gobble\if\def\bxjs@tmpa{\@{\everypar\everyparhook}\fi}\fi
4189     \directlua{
4190       local function patchcmd(cs, code, from, to)
4191         tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4192           :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4193       end
4194       patchcmd(\@xsect, [[\meaning\xsect]],
4195         \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4196       patchcmd(\@item, [[\meaning\@item]],
4197         \bxjs@tmpa, \@{\ltjfakeparbegin})}
4198 \endgroup
4199 \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。

```
4200 \ifbxjs@hyperref@enc
4201   \PassOptionsToPackage{unicode}{hyperref}
4202   \bxjs@check@hyperref@unicode{true}
4203 \fi
```

■共通命令の実装

```
4204 \protected\def\autospacing{%
4205   \ltjsetparameter{autospacing=true}}
4206 \protected\def\noautospacing{%
4207   \ltjsetparameter{autospacing=false}}
4208 \protected\def\autoxspacing{%
4209   \ltjsetparameter{autoxspacing=true}}
4210 \protected\def\noautoxspacing{%
4211   \ltjsetparameter{autoxspacing=false}}
4212 \def\bxjs@apply@kanjiskip{%
```



```

4213 \ltjsetparameter{kanjiskip={\@tempskipa}}
4214 \def\bxjs@apply@xkanjiskip{%
4215 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

\jachar のサブマクロの実装。

4216 \def\bxjs@jachar#1{%
4217 \ltjjachar`#1\relax}

\jathinspace の実装。

4218 \ifbxjs@jaspace@cmd
4219 \protected\def\jathinspace{%
4220 \hskip\ltjgetparameter{xkanjiskip}\relax}
4221 \fi

```

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

```

4222 \ifx f\bxjs@enablejfam
4223 \ClassWarningNoLine\bxjs@clsname
4224 {You cannot use 'enablejfam=false', since the\MessageBreak
4225 LuaTEX-ja always provides Japanese math families}
4226 \fi

```

C.6 共通処理 (2)

```

4227 \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 4228 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4229 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4230 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4231 \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。

```

4232 \ifx\mathmc\@undefined
4233 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4234 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4235 \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)。

```

4236 \ifbxjs@jaspace@cmd
4237 \bxjs@protected\def\bxjs@choice@jathinspace{%
4238 \relax\ifmmode \mskip\medmuskip

```

```

4239     \else \jathinspace\ignorespaces
4240     \fi}
4241     \jsAtEndOfClass{%
4242     \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4243     \else \def\>{\protect\bxjs@choice@jathinspace}%
4244     \fi}
4245 \fi

```

■和文・和欧文間空白の初期値

```

4246 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4247 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4248 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4249 \fi

```

以上で終わり。

```

4250 %</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。

```

4251 %<*modern>
4252 \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。

```

4253 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4254 \def\encodingdefault{T1}%
4255 \input{t1enc.def}%
4256 \fontencoding\encodingdefault\selectfont
4257 \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。

```

4258 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4259 \renewcommand{\rmdefault}{lmr}
4260 \renewcommand{\sfdefault}{lms}
4261 \renewcommand{\ttdefault}{lmtt}
4262 \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。

```

4263 \DeclareFontShape{OMX}{cmex}{m}{n}{%

```

```

4264 <-7.5>cmex7<7.5-8.5>cmex8%
4265 <8.5-9.5>cmex9<9.5->cmex10}{}%
4266 \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。
4267 \def\cmex@opt{10}

```

D.2 fixltx2e 読込

※ fixltx2e 廃止前の L^AT_EX カーネルの場合。

```

4268 \ifx\@IncludeInRelease\@undefined
4269 \RequirePackage{fixltx2e}
4270 \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。

```

4271 \RequirePackage{bxjscjkat}

```

D.4 完了

おしまい。

```

4272 %</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。

```

4273 %<*pandoc>
4274 \input{bxjsja-standard.def}
4275 \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。

```

4276 \@onlypreamble\bxjs@set@dupload@proc
4277 \def\bxjs@set@dupload@proc#1{%
4278   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}

```

```

4279 \@onlypreamble\bxjs@set@dupload@proc@a
4280 \def\bxjs@set@dupload@proc@a#1{%
4281   \@onlypreamble#1\def#1##1}

```

\@if@ptions \@if@ptions の再定義。

```

4282 \@onlypreamble\bxjs@org@if@ptions
4283 \let\bxjs@org@if@ptions\@if@ptions
4284 \newif\ifbxjs@dlp
4285 \def\@if@ptions#1#2#3{%
4286   \bxjs@dlpfalse
4287   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4288   \ifx\bxjs@tmpa\bxjs@tmpb
4289     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4290       \bxjs@dlptrue \fi
4291   \fi
4292   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
4293   \else \expandafter\bxjs@org@if@ptions
4294   \fi {#1}{#2}{#3}}
4295 \AtBeginDocument{%
4296   \let\@if@ptions\bxjs@org@if@ptions}
4297 \@onlypreamble\bxjs@do@dupload@proc
4298 \def\bxjs@do@dupload@proc#1#2#3{%
4299   \csname bxjs@dlp/#2.#1\endcsname{#3}%
4300   \@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。

```

4301 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0

```

Polyglossia について。

```

4302 \pandocSkipLoadPackage{polyglossia}
4303 \bxjs@set@dupload@proc{polyglossia.sty}{%
4304   \ClassWarning\bxjs@clsname
4305   {Loading of polyglossia is blocked}}
4306 \ifx\setmainlanguage\@undefined
4307 \newcommand*\setmainlanguage[2][]{%
4308 \newcommand*\setotherlanguage[2][]{%
4309   \ifcat_#2_\else
4310     \expandafter\let\csname #2\endcsname\@empty
4311     \expandafter\let\csname end#2\endcsname\@empty
4312     \expandafter\let\csname text#2\endcsname\@firstofone
4313   \fi}
4314 \newcommand*\setotherlanguages[2][]{%
4315   \@for\bxjs@tmpa:={#2}\do{%
4316     \setotherlanguage{\bxjs@tmpa}}}

```

```
4317 \fi
4318 \else
```

Babel について。

```
4319 \pandocSkipLoadPackage{babel}
4320 \bxjs@set@dupload@proc{babel.sty}{%
4321   \ClassWarning\bxjs@clsname
4322     {Loading of babel is blocked}}
4323 \let\foreignlanguage\@secondoftwo
4324 \let\otherlanguage\@gobble
4325 \let\endotherlanguage\@empty
4326 \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。

```
4327 \bxjs@set@dupload@proc{geometry.sty}{%
4328   \setpagelayout*{#1}}
```

E.4 CJKmainfont 変数

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

```
4329 \if 1\jsEngine
4330   \pandocSkipLoadPackage{xCJK}
4331   \providecommand*\setCJKmainfont{\setmainfont}
4332 \fi
```

E.5 paragraph のマーク

BXJS クラスでは `\paragraph` の見出しの前に `\jsParagraphMark` で指定したマークが付加され、既定ではこれは “■” である。しかし、この規定は `\paragraph` が本来のレイアウトを保っている、すなわち「行内見出しである」「節番号が付かない」ことが前提になっていると考えられる。Pandoc はこの規定を変更することがある（特に既定で `\paragraph` を別行見出しに再定義する）ため、変更された場合は `\jsParagraphMark` の既定値を空にする。

Pandoc がプレアンブルで行う再定義の結果を調べるため、`begin-document` フックを利用する。

```
4333 \AtBeginDocument{%
4334   \@tempwafalse
```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```
4335   \ifx\oldparagraph\undefined\else
```

```
4336 \tempwattrue
4337 \fi
```

\paragraph が番号付きの場合は、マーク変更が必要である。

```
4338 \ifnum\c@secnumdepth>3
4339 \tempwattrue
4340 \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。

```
4341 \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4342 \let\jsParagraphMark\empty
4343 \fi\fi}
```

E.6 全角空白文字

```
4344 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4345 \catcode"3000=\active
4346 \begingroup \catcode`\!=7
4347 \protected\gdef!!!!3000{\zwspace}
4348 \endgroup
4349 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4350 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4351 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4352 \fi\fi
```

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

```
4353 \if j\jsEngine
4354 \bxjs@fix@hyperref@unicode{false}
4355 \else
4356 \bxjs@fix@hyperref@unicode{true}
4357 \fi
```

E.7 完了

おしまい。

```
4358 %</pandoc>
```

和文ドライバ実装はここまで。

```
4359 %</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。

4360 %<*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

4361 %<*compat>

4362 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4363 \let\bxac@engine=n

4364 \def\bxac@do#1#2{%

4365 \edef\bxac@tmpa{\string#1}%

4366 \edef\bxac@tmpb{\meaning#1}%

4367 \ifx\bxac@tmpa\bxac@tmpb #2\fi}

4368 \bxac@do\XeTeXversion{\let\bxac@engine=x}

4369 \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。

4370 \ifx\jsAtEndOfClass\@undefined

4371 \let\bxac@delayed@if@bxjs\@firstofone

4372 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass

4373 \fi

\ImposeOldLuaTeXBehavior ムニャムニャ。

\RevokeOldLuaTeXBehavior 4374 \newif\ifbxac@in@old@behavior

4375 \let\ImposeOldLuaTeXBehavior\relax

4376 \let\RevokeOldLuaTeXBehavior\relax

G.2 X_YTEX 部分

4377 \ifx x\bxac@engine

X_YTEX 文字クラスのムニャムニャ。

4378 \@onlypreamble\bxac@adjust@charclass

4379 \bxac@delayed@if@bxjs{%

4380 \@ifpackageloaded{xeCJK}{-}{%else

4381 \ifx\xe@alloc@intercharclass\@undefined\else

4382 \ifnum\xe@alloc@intercharclass=\z@

4383 \PackageInfo\bxac@pkgname

4384 {Setting up interchar class for CJK...\@gobble}%

4385 \InputIfFileExists{load-unicode-xetex-classes.tex}{%

4386 \xe@alloc@intercharclass=3

```

4387     }{%else
4388         \PackageWarning\bxac@pkgname
4389         {Cannot find file 'load-unicode-xetex-classes.tex'%
4390         \@gobble}%
4391     }%
4392 \fi\fi
4393 \ifnum\XeTeXcharclass"3041=\z@
4394     \PackageInfo\bxac@pkgname
4395     {Adjusting interchar class for CJK...\@gobble}%
4396     \@for\bxac@x={%
4397         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4398         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4399         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4400         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4401         31FF%
4402     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4403 \fi
4404 }%
4405 }

```

以上。

```
4406 \fi
```

G.3 LuaTeX 部分

```

4407 \ifx l\bxac@engine
      ムニヤムニヤ。
4408 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4409     \chardef\pdfTeXversion=200
4410     \def\pdfTeXrevision{0}
4411     \let\pdfTeXbanner\luatexbanner
4412 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior
4413 \begingroup\expandafter\expandafter\expandafter\endgroup
4414 \expandafter\ifx\csgname outputmode\endcsgname\relax\else
4415 \def\bxac@ob@list{%
4416     \do{\let}\pdfoutput{\outputmode}%
4417     \do{\let}\pdfpagewidth{\pagewidth}%
4418     \do{\let}\pdfpageheight{\pageheight}%
4419     \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4420     \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}}
4421 \def\bxac@ob@do#1#2{\begingroup
4422     \expandafter\bxac@ob@do@a\csgname bxac@\string#2\endcsgname{#1}#2}
4423 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4424     \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4425     \else \let#3#1\relax \let#1\@undefined
4426     \fi}
4427 \protected\def\ImposeOldLuaTeXBehavior{%
4428     \unless\ifbxac@in@old@behavior

```



```

4429 \bxac@in@old@behaviortrue
4430 \let\do\bxac@ob@do \bxac@ob@list
4431 \fi}
4432 \protected\def\RevokeOldLuaTeXBehavior{%
4433 \ifbxac@in@old@behavior
4434 \bxac@in@old@behaviorfalse
4435 \let\do\bxac@ob@do \bxac@ob@list
4436 \fi}
4437 \fi

```

漢字および完成形ハングルのカテゴリコードのムニャムニャ。

```

4438 \ifnum\luatexversion>64 \directlua{
4439   local function range(cs, ce, cc, ff)
4440     if ff or not tex.getcatcode(cs) == cc then
4441       local setcc = tex.setcatcode
4442       for c = cs, ce do setcc(c, cc) end
4443     end
4444   end
4445   range(0x3400, 0x4DB5, 11, false)
4446   range(0x4DB5, 0x4DBF, 11, true)
4447   range(0x4E00, 0x9FCC, 11, false)
4448   range(0x9FCD, 0x9FFF, 11, true)
4449   range(0xAC00, 0xD7A3, 11, false)
4450   range(0x20000, 0x2A6D6, 11, false)
4451   range(0x2A6D7, 0x2A6FF, 11, true)
4452   range(0x2A700, 0x2B734, 11, false)
4453   range(0x2B735, 0x2B73F, 11, true)
4454   range(0x2B740, 0x2B81D, 11, false)
4455   range(0x2B81E, 0x2B81F, 11, true)
4456   range(0x2B820, 0x2CEA1, 11, false)
4457   range(0x2CEA2, 0x2FFFD, 11, true)
4458 } \fi

```

新元号のムニャムニャ。

```

4459 \begingroup
4460 \catcode`\~ = 12 \let\0\relax \let\1\relax \let\2\relax
4461 \directlua{
4462   if not bxjs then bxjs = {} end
4463   function bxjs.get_new_gengo()
4464     local lc_time = nil
4465     local ok, ret = pcall(function()
4466       assert(os.type == "unix")
4467       lc_time = os.setlocale(nil, "time")
4468       assert(os.setlocale("ja_JP.utf8", "time") or
4469         os.setlocale("ja_JP.UTF-8", "time"))
4470       local heisei = "\229\185\179\230\136\144"
4471       local kanji = "[\228-\233][\128-\191][\128-\191]"
4472       local gh = os.date("\037EC", 1500000000)
4473       local gn = os.date("\037EC", 1600000000)

```

```

4474         assert(gh == heisei and gn ~= heisei and
4475             gn:match("^"..kanji..kanji.."$$"))
4476         return gn
4477     end)
4478     os.setlocale(lc_time, "time")
4479     return ok and ret or ""
4480 end}
4481 \endgroup

```

以上。

```
4482 \fi
```

G.4 完了

おしまい。

```
4483 %</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

```

4484 %<*cjkat>
4485 \def\bxjx@pkgname{bxjscjkat}
4486 \newcount\bxjx@canta
4487 \@onlypreamble\bxjx@tmpdo
4488 \@onlypreamble\bxjx@tmpdo@a
4489 \@onlypreamble\bxjx@tmpdo@b

```

\bxjx@engine エンジンの種別。

```

4490 \let\bxjx@engine=n
4491 \def\bxjx@tmpdo#1#2{%
4492     \edef\bxjx@tmpa{\string#1}%
4493     \edef\bxjx@tmpb{\meaning#1}%
4494     \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4495 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4496 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4497 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4498 \bxjx@tmpdo\pdfTeXversion{\let\bxjx@engine=p}
4499 \bxjx@tmpdo\luaTeXversion{\let\bxjx@engine=l}

```

それぞれのエンジンで、前提となる日本語処理パッケージが実際に読み込まれているかを
 検査する。

```

4500 \def\bxjx@tmpdo#1#2{%
4501     \if#1\bxjx@engine
4502         \@ifpackageloaded{#2}{-}{%else

```

```

4503 \PackageError\bxjx@pkgname
4504 {Package '#2' must be loaded}%
4505 {Package loading is aborted.\MessageBreak\@ehc}%
4506 \endinput}
4507 \fi}
4508 \bxjx@tmpdo{p}{bxCJKatype}
4509 \bxjx@tmpdo{x}{xeCJK}
4510 \bxjx@tmpdo{l}{luatexja}

```

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

```

4511 \ifx\TextOrMath\@undefined
4512 \RequirePackage{fixltx2e}
4513 \fi

```

H.2 和文カテゴリコードの設定

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

※ LuaT_EX-ja との相違点：A830、A960、1B000。

```

4514 \if u\bxjx@engine
4515 \@for\bxjx@tmpa:={%
4516 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4517 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4518 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4519 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4520 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4521 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4522 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4523 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4524 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4525 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4526 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4527 FE70,FFF0,%
4528 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4529 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4530 10480,104B0,10500,10530,10600,10800,10840,10860,%
4531 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4532 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4533 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4534 11200,11280,112B0,11300,11400,11480,11580,11600,%
4535 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4536 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4537 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4538 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4539 1EE00,1F000,1F030,1F0A0,1F300,1F600,1F650,1F680,%
4540 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4541 00C0%
4542 }\do{%

```

```

4543 \@tempcnta="\bxjx@tmpa\relax
4544 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4545 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4546 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4547 \fi

```

H.3 ギリシャ・キリル文字の扱い

「特定 CJK 曖昧文字」について、和文・欧文扱いを制御できるようにする。ここで「特定 CJK 曖昧文字」とは以下に該当する文字の集合を指す：

- Unicode と JIS X 0213 に共通して含まれるギリシャ文字・キリル文字。
- Latin-1 の上位部分と JIS X 0208 に共通して含まれる文字（LuaT_EX-j_a の定める“範囲 8”）。

`\bxjx@grkcyr@list` 「特定 CJK 曖昧文字」に関する情報をもつ `\do`-リスト。各項目の形式は以下の通り：

`\do{〈Unicode 符号値〉}{〈対象 fontenc〉}{〈テキスト LICR〉}{〈数式 LICR〉}`

※数式で使わない文字は〈数式 LICR〉を空にする。

```

4548 \@onlypreamble\bxjx@grkcyr@list
4549 \def\bxjx@grkcyr@list{%
4550 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4551 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4552 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4553 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4554 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4555 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4556 \do{0397}{LGR}{\textEta}{H}%           % GR. C. L. ETA
4557 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4558 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4559 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4560 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4561 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4562 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
4563 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
4564 \do{039F}{LGR}{\textOmicron}{O}%      % GR. C. L. OMICRON
4565 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
4566 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
4567 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4568 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
4569 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4570 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
4571 \do{03A7}{LGR}{\textChi}{X}%           % GR. C. L. CHI
4572 \do{03A8}{LGR}{\textPsi}{\Psi}%         % GR. C. L. PSI
4573 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA
4574 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
4575 \do{03B2}{LGR}{\textbeta}{\beta}%       % GR. S. L. BETA
4576 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA

```

4577	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4578	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4579	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4580	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4581	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4582	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4583	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4584	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4585	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4586	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4587	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4588	\do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
4589	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4590	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4591	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4592	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4593	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4594	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4595	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4596	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4597	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4598	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4599	\do{0401}{T2A}{\CYRYO}{}	% CY. C. L. IO
4600	\do{0410}{T2A}{\CYRA}{}	% CY. C. L. A
4601	\do{0411}{T2A}{\CYRB}{}	% CY. C. L. BE
4602	\do{0412}{T2A}{\CYRV}{}	% CY. C. L. VE
4603	\do{0413}{T2A}{\CYRG}{}	% CY. C. L. GHE
4604	\do{0414}{T2A}{\CYRD}{}	% CY. C. L. DE
4605	\do{0415}{T2A}{\CYRE}{}	% CY. C. L. IE
4606	\do{0416}{T2A}{\CYRZH}{}	% CY. C. L. ZHE
4607	\do{0417}{T2A}{\CYRZ}{}	% CY. C. L. ZE
4608	\do{0418}{T2A}{\CYRI}{}	% CY. C. L. I
4609	\do{0419}{T2A}{\CYRISHRT}{}	% CY. C. L. SHORT I
4610	\do{041A}{T2A}{\CYRK}{}	% CY. C. L. KA
4611	\do{041B}{T2A}{\CYRL}{}	% CY. C. L. EL
4612	\do{041C}{T2A}{\CYRM}{}	% CY. C. L. EM
4613	\do{041D}{T2A}{\CYRN}{}	% CY. C. L. EN
4614	\do{041E}{T2A}{\CYRO}{}	% CY. C. L. O
4615	\do{041F}{T2A}{\CYRP}{}	% CY. C. L. PE
4616	\do{0420}{T2A}{\CYRR}{}	% CY. C. L. ER
4617	\do{0421}{T2A}{\CYRS}{}	% CY. C. L. ES
4618	\do{0422}{T2A}{\CYRT}{}	% CY. C. L. TE
4619	\do{0423}{T2A}{\CYRU}{}	% CY. C. L. U
4620	\do{0424}{T2A}{\CYRF}{}	% CY. C. L. EF
4621	\do{0425}{T2A}{\CYRH}{}	% CY. C. L. HA
4622	\do{0426}{T2A}{\CYRC}{}	% CY. C. L. TSE
4623	\do{0427}{T2A}{\CYRCH}{}	% CY. C. L. CHE
4624	\do{0428}{T2A}{\CYRSH}{}	% CY. C. L. SHA
4625	\do{0429}{T2A}{\CYRSHCH}{}	% CY. C. L. SHCHA

4626	\do{042A}{T2A}{\CYRHRDSN}{}	% CY. C. L. HARD SIGN
4627	\do{042B}{T2A}{\CYRERY}{}	% CY. C. L. YERU
4628	\do{042C}{T2A}{\CYRSFTSN}{}	% CY. C. L. SOFT SIGN
4629	\do{042D}{T2A}{\CYREREV}{}	% CY. C. L. E
4630	\do{042E}{T2A}{\CYRYU}{}	% CY. C. L. YU
4631	\do{042F}{T2A}{\CYRYA}{}	% CY. C. L. YA
4632	\do{0430}{T2A}{\cyra}{}	% CY. S. L. A
4633	\do{0431}{T2A}{\cyrb}{}	% CY. S. L. BE
4634	\do{0432}{T2A}{\cyrv}{}	% CY. S. L. VE
4635	\do{0433}{T2A}{\cyrg}{}	% CY. S. L. GHE
4636	\do{0434}{T2A}{\cyrd}{}	% CY. S. L. DE
4637	\do{0435}{T2A}{\cyre}{}	% CY. S. L. IE
4638	\do{0436}{T2A}{\cyrrzh}{}	% CY. S. L. ZHE
4639	\do{0437}{T2A}{\cyrrz}{}	% CY. S. L. ZE
4640	\do{0438}{T2A}{\cyri}{}	% CY. S. L. I
4641	\do{0439}{T2A}{\cyrishrt}{}	% CY. S. L. SHORT I
4642	\do{043A}{T2A}{\cyrk}{}	% CY. S. L. KA
4643	\do{043B}{T2A}{\cyrll}{}	% CY. S. L. EL
4644	\do{043C}{T2A}{\cyrm}{}	% CY. S. L. EM
4645	\do{043D}{T2A}{\cyrn}{}	% CY. S. L. EN
4646	\do{043E}{T2A}{\cyro}{}	% CY. S. L. O
4647	\do{043F}{T2A}{\cyrp}{}	% CY. S. L. PE
4648	\do{0440}{T2A}{\cyrr}{}	% CY. S. L. ER
4649	\do{0441}{T2A}{\cyrs}{}	% CY. S. L. ES
4650	\do{0442}{T2A}{\cyrt}{}	% CY. S. L. TE
4651	\do{0443}{T2A}{\cyru}{}	% CY. S. L. U
4652	\do{0444}{T2A}{\cyrf}{}	% CY. S. L. EF
4653	\do{0445}{T2A}{\cyrrh}{}	% CY. S. L. HA
4654	\do{0446}{T2A}{\cyrc}{}	% CY. S. L. TSE
4655	\do{0447}{T2A}{\cyrrch}{}	% CY. S. L. CHE
4656	\do{0448}{T2A}{\cyrrsh}{}	% CY. S. L. SHA
4657	\do{0449}{T2A}{\cyrrshch}{}	% CY. S. L. SHCHA
4658	\do{044A}{T2A}{\cyrhrdsn}{}	% CY. S. L. HARD SIGN
4659	\do{044B}{T2A}{\cyrery}{}	% CY. S. L. YERU
4660	\do{044C}{T2A}{\cyrsftsn}{}	% CY. S. L. SOFT SIGN
4661	\do{044D}{T2A}{\cyrerev}{}	% CY. S. L. E
4662	\do{044E}{T2A}{\cyryu}{}	% CY. S. L. YU
4663	\do{044F}{T2A}{\cyrya}{}	% CY. S. L. YA
4664	\do{0451}{T2A}{\cyryo}{}	% CY. S. L. IO
4665	\do{00A7}{TS1}{\textsection}{\mathsection}	% SECTION SYMBOL
4666	\do{00A8}{TS1}{\textasciidieresis}{}	% DIAERESIS
4667	\do{00B0}{TS1}{\textdegree}{\mathdegree}	% DEGREE SIGN
4668	\do{00B1}{TS1}{\textpm}{\pm}	% PLUS-MINUS SIGN
4669	\do{00B4}{TS1}{\textasciicircum}{}	% ACUTE ACCENT
4670	\do{00B6}{TS1}{\textparagraph}{\mathparagraph}	% PILCROW SIGN
4671	\do{00D7}{TS1}{\texttimes}{\times}	% MULTIPLICATION SIGN
4672	\do{00F7}{TS1}{\textdiv}{\div}	% DIVISION SIGN
4673	}	

`\mathdegree` 面倒なので補っておく。

```
4674 \providecommand*\mathdegree{{}^{\circ}}
```

`\ifbxjx@gcc@cjk` [スイッチ]「特定 CJK 曖昧文字」を和文扱いにするか。

```
4675 \newif\ifbxjx@gcc@cjk
```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```
4676 \newcommand*\greekasCJK{%
```

```
4677   \bxjx@gcc@cjktrue}
```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```
4678 \newcommand*\nogreekasCJK{%
```

```
4679   \bxjx@gcc@cjkfalse}
```

`\bxjx@fake@grk` `\bxjx@fake@grk{〈出力文字〉}{〈基準文字〉}` : ラテン文字で代用される数式ギリシャ文字の出力を行う。〈基準文字〉(`\mathchardef` の制御綴) の数式クラスと数式ファミリを引き継いで、〈出力文字〉(ASCII 文字トークン) の文字コードの数式文字を出力する。例えば、`\Pi` の意味が `\mathchar"7005` である場合、`\bxjx@fake@grk{B}{\Pi}` は `\mathchar"7042` を実行する。

※フォントパッケージ使用時の再定義を考慮して、〈基準文字〉が `\mathchardef` であるかを検査し、そうでない場合はフォールバックとして単に 〈出力文字〉 を実行する。

```
4680 \def\bxjx@tmpdo#1\relax{%
4681   \def\bxjx@fake@grk##1##2{%
4682     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4683   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4684     \ifx\##1\%
4685       \bxjx@canta##4\divide\bxjx@canta\@cclvi
4686       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
4687       \mathchar\bxjx@canta
4688     \else ##3\fi}
4689 }\expandafter\bxjx@tmpdo\string\mathchar\relax
```

■pdfTeX・upTeX の場合

```
4690 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0
```

- `\[bxjx@KC/〈符号値〉]` : その文字が「特定曖昧 CJK 文字」に該当する場合に定義済になる。

まず `inputenc` を読み込んで入力エンコーディングを `utf8` に変更する。

※「既定 UTF-8 化」後の L^AT_EX においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```
4691 \@ifpackageloaded{inputenc}{}{%else
4692   \RequirePackage[utf8]{inputenc}}
4693 \def\bxjx@tmpa{utf8}
4694 \ifx\bxjx@tmpa\inputencdoingname
4695   \PackageWarningNoLine\bxjx@pkgname
```

```

4696   {Input encoding changed to utf8}%
4697   \inputencoding{utf8}%
4698 \fi

```

upTeX の場合に、「特定曖昧 CJK 文字」を含むブロックの和文カテゴリコードを変更する。

```

4699 \if u\bxjx@engine
4700 \kcatcode"0370=15
4701 \kcatcode"0400=15
4702 \kcatcode"0500=15
4703 \fi

```

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

```

4704 \def\bxjx@tmpdo#1{%
4705   \@tempcnta="#1\relax
4706   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4707 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

引数 = \[bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式 LICR>}

“数式中の動作”を決定する。<数式 LICR> が空（数式非対応）なら警告を出す。

```

4708   \ifx\\#5\\%
4709     \def\bxjx@tmpa{\@inmathwarn#4}%

```

<数式 LICR> が英字である場合は \bxjx@fake@grk で出力する。大文字なら \Pi、小文字なら \pi を基準文字にする。

```

4710   \else\ifcat A\noexpand#5%
4711     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4712       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%

```

それ以外は <数式 LICR> をそのまま実行する。

```

4713   \else \def\bxjx@tmpa{#5}%
4714   \fi\fi
4715   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4716   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

4717 \if u\bxjx@engine
4718 \def\bxjx@tmpdo@b#1#2#3#4#5{%

```

引数 = \[bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式中の動作>}

当該の Unicode 文字の動作は「テキストでは <LICR>、数式では <数式中の動作>」となる。LICR は現在エンコーディングで有効な定義がある場合はそれが実行されるはずである。（つまり、現在が LGR である場合はギリシャ文字は常に欧文扱いになる。）それ以外の場合は LICR を \bxjx@ja@or@not に帰着させる。この際に、和文用の定義として当該の kchardef を使用し、その制御綴として \[bxjx@KC/...] を流用している。

```

4719   \kchardef#1=\@tempcnta
4720   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4721   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```


pdf_T_EX の場合も処理はほとんど同じ。ただし、和文用の定義として `\UTF{〈符号値〉}` を使う (`\UTF` は `bxCJKatype` の命令)。`\[bxjx@KC/...]` は使わないが定義済にする必要がある。

```
4722 \else\if p\bxjx@engine
4723 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4724   \mathchardef#1=\@tempcnta
4725   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4726   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4727 \fi\fi
```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```
4728 \let\do\bxjx@tmpdo \bxjx@grkcyr@list
```

`\bxjx@DeclareUnicodeCharacter` `\bxjx@DeclareUnicodeCharacter` を改変して、「特定 CJK 曖昧文字」の場合に再定義を抑制したもの。

```
4729 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4730 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4731 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4732 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4733   \count@=#1\relax
4734   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4735     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4736   \else
4737     \wlog{ \space\space skipped defining Unicode char U+#1}%
4738   \fi}
```

`\bxjx@ja@or@not` `\bxjx@ja@or@not{〈和文用定義〉}{〈対象 fontenc〉}{〈LICR〉}` : `\[no]greekasCJK` の状態に応じて和文または欧文で文字を出力する。

```
4739 \def\bxjx@ja@or@not#1#2#3{%
```

`\greekasCJK` の場合は、無条件に `〈和文用定義〉` を実行する。

```
4740   \ifbxjx@gcc@CJK #1%
```

`\nogreekasCJK` の場合は、対象のエンコーディングに変更して `LICR` を実行するが、そのエンコーディングが未定義の場合は（フォールバックとして）`和文用定義` を使う。

```
4741   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4742   \else \UseTextSymbol{#2}{#3}%
4743   \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。

```
4744 \begingroup
4745 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4746 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4747   \noexpand\bxjx@swap@DUC@cmd
4748   \the\toks@
4749   \noexpand\bxjx@swap@DUC@cmd}}
4750 \endgroup\next
4751 \def\bxjx@swap@DUC@cmd{%
4752   \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```

4753 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4754 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
4755 \let\bxjx@tmpa\relax}

```

以上。

■Xe_{La}TeX・Lua_{TeX} の場合

```

4756 \else\ifnum0\if x\bxjx@engine1\fi\if 1\bxjx@engine1\fi>0

```

各文字について、数式中の動作を定義する。

```

4757 \def\bxjx@tmpdo#1{%
4758   \bxjx@cmta="#1\relax
4759   \begingroup
4760     \lccode`~=\bxjx@cmta
4761     \lowercase{\endgroup
4762       \bxjx@tmpdo@a{~}}{#1}}
4763 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

〈数式 LICR〉が空なら何もしない。空でない場合、up_{La}TeX の場合と同じ方法で“数式中の動作”を決定し、当該の文字を math active にしてその動作を設定する。

```

4764   \ifx\\#5\\\let\bxjx@tmpa\relax
4765   \else\ifcat A\noexpand#5%
4766     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4767       {\ifnum\uccode`#5=#5\noexpand\pi\else\noexpand\pi\fi}}%
4768   \else \def\bxjx@tmpa{#5}%
4769   \fi\fi
4770   \ifx\bxjx@tmpa\relax\else
4771     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4772   \fi}

```

「Unicode な数式」の設定が行われているかを（簡易的に）検査して、そうでない場合にのみ、以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

4773 \mathchardef\bxjx@tmpa="119
4774 \ifx\bxjx@tmpa\pi \let\do\bxjx@tmpdo \bxjx@grkcyr@list \fi

```

次に、テキストにおいて「特定 CJK 曖昧文字」の扱いが `\[no]greekasCJK` で切り替わるようにする。

Lua_{TeX} の場合は、Lua_{TeX}-ja の `jacharrange` の設定を変更する。

※ “範囲 2” がギリシャ・キリル文字、“範囲 8” が Latin-1 の記号。

```

4775 \if 1\bxjx@engine
4776   \protected\def\greekasCJK{%
4777     \bxjx@gcc@cjktrue
4778     \ltjsetparameter{jacharrange={+2, +8}}
4779   \protected\def\nogreekasCJK{%
4780     \bxjx@gcc@cjkfalse
4781     \ltjsetparameter{jacharrange={-2, -8}}
4782 \fi

```

Xe_{La}TeX の場合、xeCJK は Xe_{La}TeX の文字クラス定義を参照しているので、対象文字の文字クラスを変更する。

```

4783 \if x\bxjx@engine
4784   \let\bxjx@gcc@cjkl@list\@empty
4785   \def\do#1#2#3#4{%
4786     \edef\bxjx@gcc@cjkl@list{\bxjx@gcc@cjkl@list
4787       \noexpand\XeTeXcharclass"#1\bxjx@canta}}
4788   \bxjx@grkcyrl@list
4789   \protected\def\greekasCJK{%
4790     \bxjx@gcc@cjkltrue
4791     \bxjx@canta=\@ne \bxjx@gcc@cjkl@list}
4792   \protected\def\nogreekasCJK{%
4793     \bxjx@gcc@cjklfalse
4794     \bxjx@canta=\z@ \bxjx@gcc@cjkl@list}
4795 \fi

    以上。
4796 \fi\fi

```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```

4797 \nogreekasCJK

```

H.5 完了

おしまい。

```

4798 %</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

```

4799 %<*ancpandoc>
4800 %% このファイルは日本語文字を含みます。
4801 \def\bxjsp@pkgname{bxjscjkat}

\bxjsp@engine エンジンの種別。

4802 \let\bxjsp@engine=n
4803 \@onlypreamble\bxjsp@do
4804 \def\bxjsp@do#1#2{%
4805   \edef\bxjsp@tmpa{\string#1}%

```

```

4806 \edef\bxjsp@tmpb{\meaning#1}%
4807 \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
4808 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
4809 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
4810 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
4811 \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。

```

4812 \newcommand*\pandocSkipLoadFile[1]{%
4813 \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4814 \def\bxjsp@skip@load@file@a#1#2{%
4815 \ifx#1\relax
4816 \def#1{2001/01/01}%
4817 \PackageInfo\bxjsp@pkgname
4818 {File '#2' marked as loaded\@gobble}%
4819 \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。

```

4820 \newcommand*\pandocSkipLoadPackage[1]{%
4821 \pandocSkipLoadFile{#1.sty}}

```

I.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。

```

4822 \ifx\@IncludeInRelease\@undefined\else
4823 \pandocSkipLoadPackage{fixltx2e}
4824 \fi

```

I.4 cmap パッケージ

エンジンが (u)p`LATEX` のときに `cmap` パッケージが読み込まれるのを阻止する。(実際は警告が出るだけで無害であるが。)

```

4825 \if j\bxjsp@engine
4826 \pandocSkipLoadPackage{cmap}
4827 \fi

```

I.5 microtype パッケージ

警告が多すぎなので消す。

```
4828 \if j\bxjsp@engine \else
4829   \PassOptionsToPackage{verbose=silent}{microtype}
4830 \fi
```

エンジンが (u)pL^AT_EX のときに microtype パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は standard ドライバでこの処理を行っていたが、元来は Pandoc 用の処理なので、1.5 版で pandoc に移動。

```
4831 \if j\bxjsp@engine
4832   \pandocSkipLoadPackage{microtype}
4833   \newcommand*\UseMicrotypeSet[2][]{\fi}
4834 \fi
```

I.6 Unicode 文字変換対策

Pandoc で L^AT_EX 形式に書き出す場合は、元データ中の一部の Unicode 文字を「L^AT_EX の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

…→\ldots{} ‘→` ’→' “→`` ”→’’

日本語 L^AT_EX では「L^AT_EX の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「\ldots{} を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の `\ldots` の実装。非数式でありかつ後続が `{}` の場合は代わりに … を実行する。

```
4835 \DeclareRobustCommand{\pandocLdots}{\fi}
4836   \relax\ifmmode \expandafter\bxjsp@org@ldots
4837   \else \expandafter\bxjsp@ldots@a
4838   \fi}
4839 \def\bxjsp@ja@ellipsis{…}
4840 \let\bxjsp@org@ldots\ldots
4841 \def\bxjsp@ldots@a{%
4842   \futurelet\bxjsp@tok\bxjsp@ldots@b}
4843 \def\bxjsp@ldots@b{%
4844   \ifx\bxjsp@tok\bgroup \expandafter\bxjsp@ldots@c
4845   \else \expandafter\bxjsp@org@ldots
4846   \fi}
4847 \def\bxjsp@ldots@c{%
4848   \afterassignment\bxjsp@ldots@d \let\bxjsp@tok=}
4849 \def\bxjsp@ldots@d{%
4850   \futurelet\bxjsp@tok\bxjsp@ldots@e}
```

```

4851 \def\bxjsp@ldots@e{%
4852   \ifx\bxjsp@tok\egroup \expandafter\bxjsp@ldots@f
4853   \else \expandafter\bxjsp@ldots@g
4854   \fi}
4855 \def\bxjsp@ldots@f{%
4856   \bxjsp@ja@ellipsis \let\bxjsp@tok=}
4857 \def\bxjsp@ldots@g{%
4858   \expandafter\bxjsp@org@ldots\expandafter{\romannumeral-`} }

```

\ldots の実装を置き換える。

```

4859 \AtBeginDocument{%
4860   \let\bxjsp@org@ldots\ldots
4861   \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。

```

4862 \IfFileExists{bxpandola.sty}{%
4863   \RequirePackage{bxpandola}\relax
4864   \PackageInfo\bxjsp@pkgname
4865   {PandoLa module is loaded\@gobble}
4866 }{}

```

I.8 完了

おしまい。

```

4867 %</ancpandoc>

```

補助パッケージ実装はここまで。

```

4868 %</anc>

```