

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”)





v1.9f [2019/03/10]

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

目次

1	はじめに	3
2	オプション	10
3	和文フォントの変更	38
4	フォントサイズ	39
5	レイアウト	44
5.1	ページレイアウト	45
6	改ページ (日本語 TeX 開発コミュニティ版のみ)	59
7	ページスタイル	61
8	文書のマークアップ	64
8.1	表題	64
8.2	章・節	69
8.3	リスト環境	82
8.4	パラメータの設定	89
8.5	フロート	90
8.6	キャプション	92
9	フォントコマンド	93

10	相互参照	95
10.1	目次の類	95
10.2	参考文献	101
10.3	索引	103
10.4	脚注	104
11	段落の頭へのグルー挿入禁止	106
12	いろいろなロゴ	110
13	amsmath との衝突の回避	111
14	初期設定	111
付録 A	和文ドライバの仕様 罫	116
付録 B	和文ドライバ : minimal 罫	117
B.1	補助マクロ	117
B.2	(u)pTeX 用の設定	119
B.3	pdfTeX 用の処理	124
B.4	X _Y TeX 用の処理	124
B.5	後処理 (エンジン共通)	125
付録 C	和文ドライバ : standard 罫	128
C.1	共通処理 (1)	128
C.2	pTeX 用設定	136
C.3	pdfTeX 用設定 : CJK + bxcjkjatype	139
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	141
C.5	LuaTeX 用設定 : LuaTeX-ja	143
C.6	共通処理 (2)	147
付録 D	和文ドライバ : modern 罫	148
D.1	フォント設定	148
D.2	fixltx2e 読込	149
D.3	和文カテゴリコード	149
D.4	完了	149
付録 E	和文ドライバ : pandoc 罫	149
E.1	dupload システム	150
E.2	lang 変数	151
E.3	geometry 変数	153
E.4	CJKmainfont 変数	153
E.5	paragraph のマーク	154

E.6	全角空白文字	154
E.7	完了	155
付録 F	補助パッケージ一覧 	155
付録 G	補助パッケージ : bxjscompat 	155
G.1	準備	155
G.2	X _q TeX 部分	156
G.3	LuaTeX 部分	156
G.4	完了	158
付録 H	補助パッケージ : bxjscjkat 	158
H.1	準備	158
H.2	和文カテゴリコードの設定	159
H.3	ギリシャ・キリル文字の扱い	160
H.4	初期設定	167
H.5	完了	167
付録 I	補助パッケージ : bxjspandoc 	167
I.1	準備	167
I.2	パッケージオプション	168
I.3	パッケージ読込の阻止	168
I.4	fixltx2e パッケージ	168
I.5	cmap パッケージ	169
I.6	microtype パッケージ	169
I.7	Unicode 文字変換対策	169
I.8	PandoLa モジュール	170
I.9	完了	170

1 はじめに

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

```

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

```

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

する解説である。

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

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

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

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

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

ここでは次のドキュメントクラス（スタイルファイル）を作ります。

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

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

以下では実際のコードに即して説明します。

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

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

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

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

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

■ BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため calc を読み込む。

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

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

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

クラスの本体ではこの他に geometry パッケージが読み込まれる。

TODO: 依存パッケージの情報。

互換性のための補助パッケージを読み込む。

```
11 \IfFileExists{bxjscompat.sty}{%
12   \let\jsAtEndOfClass@gobble
13   \RequirePackage{bxjscompat}%
14 }{}
```

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

```
15 \let\jsArticle=a
16 \let\jsBook=b
17 \let\jsReport=r
18 \let\jsSlide=s
19 %<article>\let\jsDocClass\jsArticle
20 %<book>\let\jsDocClass\jsBook
21 %<report>\let\jsDocClass\jsReport
22 %<slide>\let\jsDocClass\jsSlide
```

`\jsEngine` [暗黙文字トークン] エンジン (T_EX の種類) の種別: `j = pTEX` 系、`x = XYTEX`、`p = pdfTEX` (含 DVI モード)、`l = LuaTEX`、`J = NTT jTEX`、`0 = Omega` 系、`n =`以上の何れでもない。

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

`\ifjsWithupTeX` [スイッチ] エンジンが (内部漢字コードが Unicode の) upT_EX 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 pT_EX-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

```

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

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

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

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

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

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

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

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

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

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

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

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

`\bxjs@ifplus` `\@ifstar` の + 版。

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

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

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

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

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

```

97   \fi
98   \endgroup
99   #1=\bxjs@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)pL^AT_EX の場合はこれらの和文単位はエンジンでサポートされる。しかし和文フォントの設定が完了するまでは zw/zh の値は正しくない。

```

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

```

今の段階では Q/H だけが使用可能。

```

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

```

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

```

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

```

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

```

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

```

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


```

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

```

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

```

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

```

一時的な手続き用の制御綴。

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

2 オプション

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

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

`\if@restonecol` 段組のときに真になる論理変数です。

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

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

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

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

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

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

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

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

BXJS では report 系でも定義されることに注意。

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

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

JS クラスと異なり、初期値は偽とする。

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

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

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

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

```
167 \@onlypreamble\bxjs@setpaper
168 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
169 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
170 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
171 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
172 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
173 \DeclareOption{b4paper}{\bxjs@setpaper{{257trueemm}{364trueemm}}}
174 \DeclareOption{b5paper}{\bxjs@setpaper{{182trueemm}{257trueemm}}}
175 \DeclareOption{b6paper}{\bxjs@setpaper{{128trueemm}{182trueemm}}}
176 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
177 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
178 \DeclareOption{b4j}{\bxjs@setpaper{{257trueemm}{364trueemm}}}
179 \DeclareOption{b5j}{\bxjs@setpaper{{182trueemm}{257trueemm}}}
180 \DeclareOption{a4var}{\bxjs@setpaper{{210trueemm}{283trueemm}}}
181 \DeclareOption{b5var}{\bxjs@setpaper{{182trueemm}{230trueemm}}}
```

※...var を Pandoc で使えるように後ろに `paper` をつけた形を用意する。

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

■横置き 用紙の縦と横の長さを入れ換えます。

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

■slide オプション `slide` を新設しました。

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

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

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

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

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

[2016-07-24] オプティカルサイズを調整するために NFSS 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。

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

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

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

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

※クラスオプションのトークン列の中に展開可能なトークンがある場合, \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。

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

`\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。

```
203 \newif\ifjsc@mag
204 \newif\ifjsc@mag@xreal
205 %\let\jsc@magscale\undefined
206 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
207 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
208 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
209 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
210 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
211 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
212 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
```

```

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

```

JS クラス互換の magstyle 設定オプション。

```

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

取りあえず、pT_EX 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。

```

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

```

244 \if j\jsEngine

```

```

245 \DeclareOption{mentuke}{%
246   \tombowtrue \tombowdatefalse
247   \setlength{\@tombowwidth}{\z@}%
248   \maketombowbox}
249 \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。

```

250 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
251 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
252 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 `twocolumn` で二段組になります。

```

253 \DeclareOption{onecolumn}{\@twocolumnfalse}
254 \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。

```

255 \DeclareOption{titlepage}{\@titlepagetrue}
256 \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 L^AT_EX 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。

```

257 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
258 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
259 %<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。

```

260 \def\eqnarray{%
261   \stepcounter{equation}%
262   \def\@currentlabel{\p@equation\theequation}%
263   \global\@eqnswtrue
264   \m@th
265   \global\@eqcnt\z@
266   \tabskip\@centering
267   \let\\\@eqnocr
268   $$\everycr{}\halign to\displaywidth\bgroup
269     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1

```

```

270 &\global\@eqcnt\@ne \hfil$\displaystyle{##{}}$\hfil
271 &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
272 &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
273 \tabskip\z@skip
274 \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。

```

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

```

■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。

```

302 % \DeclareOption{openbib}{%
303 % \AtEndOfPackage{%
304 % \renewcommand\@openbib@code{%
305 % \advance\leftmargin\bibindent
306 % \itemindent -\bibindent
307 % \listparindent \itemindent
308 % \parsep \z@}%
309 % \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：

```
310 %\let\bxjs@enablejfam\undefined
```

enablejfam オプションの処理。

```
311 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
```

```
312 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
```

```
313 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
```

```
314 \define@key{bxjs}{enablejfam}[true]{%
```

```
315 \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。

```
316 \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。

```
317 \let\ifjsDraft\iffalse
```

```
318 \@onlypreamble\bxjs@draft
```

```
319 \def\bxjs@draft#1{%
```

```
320 \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
```

```
321 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
```

```
322 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
```

```
323 \g@addto@macro\bxjs@begin@document@hook{%
```

```
324 \expandafter\ifx\csname ifdraft\endcsname\relax
```

```
325 \expandafter\let\csname ifdraft\endcsname
```

```
326 \csname ifjsDraft\endcsname
```

■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, dviptdms など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。

```
328 \newif\ifbxjs@papersize
329 \bxjs@papersizetrue
330 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
331 \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。

```
332 \newif\if@english
333 \@englishfalse
334 \DeclareOption{english}{\@englishtrue}
```

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


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

BXJS では当初から bxjsreport クラスが用意されている。

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

BXJS クラスでは、nojslogo を既定とする。

```
335 \newif\if@jslogo \@jslogofalse
336 \DeclareOption{jslogo}{\@jslogotrue}
337 \DeclareOption{nojslogo}{\@jslogofalse}
```

■複合設定オプション 

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

`\bxjs@invscale` `\bxjs@invscale` は T_EX 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。

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

複合設定オプションとは、「エンジンやドライバや和文ドライバの設定を含む、複数の設定

を一度に行うオプション」のことである。ある特定の設定を短く書く必要が高いと判断される場合に用意される。

pandoc オプションは、Pandoc で L^AT_EX 用の既定テンプレートを用いて他形式から L^AT_EX (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。

```
363 \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。

```
364 \g@addto@macro\bxjs@post@option@hook{%
```

```
365 \bxjs@oldfontcommandstrue
```

```
366 \setkeys{bxjs}{ja=pandoc}%
```

```
367 \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…。

```
368 \def\bxjs@driver@opt{dvipdfmx}%
```

```
369 \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。

```
370 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
371 %\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。

```
372 \DeclareOption{autodetect-engine}{%
```

```
373 \let\bxjs@engine@given=*}
```

```
374 \DeclareOption{latex}{%
```

```
375 \def\bxjs@engine@opt{latex}%
```

```
376 \let\bxjs@engine@given=n}
```

```
377 \DeclareOption{platex}{%
```

```
378 \def\bxjs@engine@opt{platex}%
```

```
379 \let\bxjs@engine@given=j}
```

```
380 \DeclareOption{uplatex}{%
```

```
381 \def\bxjs@engine@opt{uplatex}%
```

```
382 \let\bxjs@engine@given=u}
```

```
383 \DeclareOption{xelatex}{%
```

```
384 \def\bxjs@engine@opt{xelatex}%
```

```
385 \let\bxjs@engine@given=x}
```

```
386 \DeclareOption{pdflatex}{%
```

```

387 \def\bxjs@engine@opt{pdflatex}%
388 \let\bxjs@engine@given=p}
389 \DeclareOption{lualatex}{%
390 \def\bxjs@engine@opt{lualatex}%
391 \let\bxjs@engine@given=l}
392 \DeclareOption{platex-ng}{%
393 \def\bxjs@engine@opt{platex-ng}%
394 \let\bxjs@engine@given=g}
395 \DeclareOption{platex-ng*}{%
396 \def\bxjs@engine@opt{platex-ng*}%
397 \let\bxjs@platexng@nodrv=t%
398 \let\bxjs@engine@given=g}

```

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

```

399 %\let\bxjs@driver@given\@undefined
400 \let\bxjs@driver@@dvimode=0
401 \let\bxjs@driver@@dvipdfmx=1
402 \let\bxjs@driver@@pdfmode=2
403 \let\bxjs@driver@@xetex=3
404 \let\bxjs@driver@@dvips=4
405 \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。

```

406 %\let\bxjs@driver@opt\@undefined
407 \DeclareOption{dvips}{%
408 \def\bxjs@driver@opt{dvips}%
409 \let\bxjs@driver@given\bxjs@driver@@dvips}
410 \DeclareOption{dviout}{%
411 \def\bxjs@driver@opt{dviout}%
412 \let\bxjs@driver@given\bxjs@driver@@dvimode}
413 \DeclareOption{xdvi}{%
414 \def\bxjs@driver@opt{xdvi}%
415 \let\bxjs@driver@given\bxjs@driver@@dvimode}
416 \DeclareOption{dvipdfmx}{%
417 \def\bxjs@driver@opt{dvipdfmx}%
418 \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
419 \DeclareOption{nodvidriver}{%
420 \def\bxjs@driver@opt{nodvidriver}%
421 \let\bxjs@driver@given\bxjs@driver@@none}
422 \DeclareOption{pdftex}{%
423 \def\bxjs@driver@opt{pdftex}%
424 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
425 \DeclareOption{luatex}{%
426 \def\bxjs@driver@opt{luatex}%
427 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
428 \DeclareOption{xetex}{%
429 \def\bxjs@driver@opt{xetex}%
430 \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。

```
431 \DeclareOption{dvipdfmx-if-dvi}{%
432   \setkeys{bxjs}{dvi=dvipdfmx}}
```

■その他の BXJS 独自オプション

TODO: 互換用オプションを分離する (2.0 版で?)。

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

```
433 \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。

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

`nobigcode` / `bigcode` オプションの定義。

```
438 \DeclareOption{nobigcode}{%
439   \bxjs@bigcodefalse}
440 \DeclareOption{bigcode}{%
441   \bxjs@bigcodetrue}
```

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

```
442 \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 `enabledeprecatdfontcommands` であるがこれはチヨットアレなので避けた。

```
443 \DeclareOption{nooldfontcommands}{%
444   \bxjs@oldfontcommandsfalse}
445 \DeclareOption{oldfontcommands}{%
446   \bxjs@oldfontcommandstrue}
```

■`keyval` 型のオプション

その他のオプションは `keyval` の機構を用いて処理する。

```
447 \DeclareOption*{%
448   \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
```

```
449 \expandafter\bxjs@next\expandafter{\CurrentOption}}
```

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

```
450 \def\bxjs@safe@setkeys#1#2{%
451   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
452   \setkeys{#1}{#2}%
453   \let\KV@errx\bxjs@save@KV@errx}
```

`\bxjs@declare@enum@option` `\bxjs@declare@enum@option{<オプション名>}{<enum名>}`
“<オプション名>=<値>”のオプション指定に対して、`\[bxjs@<enum名>]`を`\[bxjs@<enum名>@<値>]`に等値する（後者の制御綴が未定義の場合はエラー）、という動作を規定する。

```
454 \@onlypreamble\bxjs@declare@enum@option
455 \def\bxjs@declare@enum@option#1#2{%
456   \define@key{bxjs}{#1}{%
457     \expandafter\ifx\csname bxjs@#2@##1\endcsname\relax
458     \bxjs@error@keyval{#1}{##1}%
459     \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@##1}%
460     \fi}}
```

`\bxjs@declare@bool@option` `\bxjs@declare@bool@option{<オプション名>}{<スイッチ名>}`
“<オプション名>=<真偽値>”のオプション指定に対して、`\if[bxjs@<スイッチ名>]`を設定する、という動作を規定する。

```
461 \@onlypreamble\bxjs@declare@bool@option
462 \def\bxjs@declare@bool@option#1#2{%
463   \define@key{bxjs}{#1}[true]{%
464     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
465     \bxjs@error@keyval{#1}{##1}%
466     \else \@nameuse{bxjs@#2##1}%
467     \fi}}
```

`\bxjs@set@keyval` `\bxjs@set@keyval{<key>}{<value>}{<error>}`
`\bxjs@kv@<key>@<value>`が定義済ならそれを実行し、未定義ならエラーを出す。

```
468 \def\bxjs@set@keyval#1#2#3{%
469   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
470   \ifx\bxjs@next\relax
471     \bxjs@error@keyval{#1}{#2}%
472     #3%
473   \else \bxjs@next
474   \fi}
475 \@onlypreamble\bxjs@error@keyval
476 \def\bxjs@error@keyval#1#2{%
477   \ClassError\bxjs@clsname
478   {Invalid value '#2' for option #1}\@ehc}
```

`\jsScale` [実数値マクロ] 和文スケール値。

```
479 \def\jsScale{0.924715}
```

```

\bxjs@base@opt 明示された base オプションの値。
480 %\let\bxjs@base@opt\@undefined

    base オプションの処理。
481 \define@key{bxjs}{base}{%
482   \edef\bxjs@base@opt{#1}%
483   \bxjs@setbasefontsize{#1}}
484 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

\bxjs@jbase@opt 明示された jbase オプションの値。
485 %\let\bxjs@jbase@opt\@undefined

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

\bxjs@scale@opt 明示された scale オプションの値。
488 %\let\bxjs@scale@opt\@undefined

    scale オプションの処理。
489 \define@key{bxjs}{scale}{%
490   \edef\bxjs@scale@opt{#1}%
491   \let\jsScale\bxjs@scale@opt}
492 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

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

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

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

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

\bxjs@jadriver 和文ドライバの名前。
497 \let\bxjs@jadriver\relax
498 %\let\bxjs@jadriver@opt\@undefined

    ja オプションの処理。
    ※jadriver は 0.9 版で用いられた旧称。
    ※単なる ja という指定は無視される (Pandoc 対策)。
499 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver@opt{#1}}
500 \define@key{bxjs}{ja}{\relax}{%
501   \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}

\jsJaFont 和文フォント設定の名前。
502 \let\jsJaFont\@empty

```

jafont オプションの処理。

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

\jsJaParam 和文ドライバパラメタの文字列。

```
504 \let\jsJaParam@empty
```

japaram オプションの処理。

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

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

```
506 \let\bxjs@magstyle@mag=m
507 \let\bxjs@magstyle@real=r
508 \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。

```
509 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
510 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
511 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

\bxjs@magstyle@default は既定の値を表す。

```
512 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
513 \ifx l\jsEngine \ifnum\luatexversion>86
514   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
515 \fi\fi
516 \ifjsWithpTeXng
517   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
518 \fi
519 \let\bxjs@magstyle\bxjs@magstyle@default
```

magstyle オプションの処理。

```
520 \define@key{bxjs}{magstyle}{%
521   \expandafter\let\expandafter\bxjs@magstyle\csname
522     bxjs@magstyle@#1\endcsname
523   \ifx\bxjs@magstyle\relax
524     \ClassError\bxjs@clsname
525       {Invalid value '#1' for option magstyle}\@ehc
526   \let\bxjs@magstyle\bxjs@magstyle@default
527   \fi}
```

\bxjs@geometry geometry オプションの値。

```
528 \let\bxjs@geometry@class=c
529 \let\bxjs@geometry@user=u
530 \let\bxjs@geometry\bxjs@geometry@class
```

geometry オプションの処理。

```
531 \define@key{bxjs}{geometry}{%
532   \expandafter\let\expandafter\bxjs@geometry\csname
```



```

533   bxjs@geometry@#1\endcsname
534   \ifx\bxjs@geometry\relax
535     \ClassError\bxjs@clsname
536     {Invalid value '#1' for option geometry}\@ehc
537     \let\bxjs@geometry\bxjs@geometry@class
538   \fi}

```

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

```

539 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

    fancyhdr オプションの処理。
540 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
541 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
542 \define@key{bxjs}{fancyhdr}[true]{%
543   \bxjs@set@keyval{fancyhdr}{#1}{}}

```

`\ifbxjs@dvi@opt` dvi オプションが指定されたか。

```

544 \newif\ifbxjs@dvi@opt

    DVI モードのドライバとドライバ種別との対応。
545 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
546 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
547 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
548 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
549 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

    dvi オプションの処理。
550 \define@key{bxjs}{dvi}{%
551   \expandafter\let\expandafter\bxjs@tmpa\csname
552     bxjs@dvidriver@@#1\endcsname
553   \ifx\bxjs@tmpa\relax
554     \ClassError\bxjs@clsname
555     {Invalid value '#1' for option dvi}\@ehc
556   \else

    \bxjs@driver@given を未定義にしていることに注意。
557     \def\bxjs@driver@opt{#1}%
558     \let\bxjs@driver@given\@undefined
559     \bxjs@dvi@opttrue
560   \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。

```

561 \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。

```

562 \newif\ifbxjs@force@chapterabstract
563 %<book>\bxjs@force@chapterabstracttrue

```

layout オプションの処理。

```
564 \@namedef{bxjs@kv@layout@v1}{%
565 %<book>\bxjs@layout@buggyhmargintrue
566 %<report>\bxjs@force@chapterabstracttrue
567 }
568 \@namedef{bxjs@kv@layout@v2}{%
569 %<book>\bxjs@layout@buggyhmarginfalse
570 %<report>\bxjs@force@chapterabstractfalse
571 }
572 \define@key{bxjs}{layout}{%
573   \bxjs@set@keyval{layout}{#1}{}}
```

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

```
574 %\let\bxjs@textwidth@limit@opt\undefined
575 \define@key{bxjs}{textwidth-limit}{%
576   \edef\bxjs@textwidth@limit@opt{#1}}
```

\bxjs@textwidth@opt textwidth の指定値。

```
577 %\let\bxjs@textwidth@opt\undefined
578 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
579 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}
```

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

```
580 %\let\bxjs@number@of@lines@opt\undefined
581 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
582 \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。

```
583 %\let\bxjs@paragraph@mark\undefined
584 \define@key{bxjs}{paragraph-mark}{%
585   \edef\bxjs@paragraph@mark{#1}}
```

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

```
586 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
587 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
588 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
589 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}
```

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

```
590 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
591 \let\bxjs@kv@jaspaccmd@true\bxjs@jaspace@cmdtrue
592 \let\bxjs@kv@jaspaccmd@false\bxjs@jaspace@cmdfalse
593 \define@key{bxjs}{jaspac-cmd}[true]{\bxjs@set@keyval{jaspaccmd}{#1}{}}
594 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{jaspaccmd}{#1}{}}
```

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

```
595 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
596 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
597 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
598 \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。

```
599 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
600 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
601 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
602 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}
```

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

```
603 \chardef\bxjs@everyparhook@none=0
604 \chardef\bxjs@everyparhook@compat=1
605 \chardef\bxjs@everyparhook@modern=2
606 \if j\jsEngine
607 \let\bxjs@everyparhook\bxjs@everyparhook@compat
608 \else
609 \let\bxjs@everyparhook\bxjs@everyparhook@modern
610 \fi
611 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
612 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
613 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
614 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}
```

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

```
615 \chardef\bxjs@label@section@none=0
616 \chardef\bxjs@label@section@compat=1
617 \chardef\bxjs@label@section@modern=2
618 \let\bxjs@label@section\bxjs@label@section@compat
619 \def\bxjs@kv@labelsection@none{\let\bxjs@label@section\bxjs@label@section@none}
620 \def\bxjs@kv@labelsection@compat{\let\bxjs@label@section\bxjs@label@section@compat}
621 \def\bxjs@kv@labelsection@modern{\let\bxjs@label@section\bxjs@label@section@modern}
622 \define@key{bxjs}{label-section}{\bxjs@set@keyval{labelsection}{#1}{}}
```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```
623 \newif\ifbxjs@usezw \bxjs@usezwtrue
624 \bxjs@declare@bool@option{use-zw}{usezw}
625 \DeclareOption{noz}{\setkeys{bxjs}{use-zw=false}}
626 \DeclareOption{zw}{\setkeys{bxjs}{use-zw=true}}
```

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

```
627 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
628 \bxjs@declare@bool@option{disguise-js}{disguise@js}
629 \DeclareOption{nojs}{\setkeys{bxjs}{disguise-js=false}}
630 \DeclareOption{js}{\setkeys{bxjs}{disguise-js=true}}
```

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

```
631 \newif\ifbxjs@precisetext
632 \bxjs@declare@bool@option{precise-text}{precisetext}
633 \DeclareOption{noprecisetext}{\setkeys{bxjs}{precise-text=false}}
634 \DeclareOption{precisetext}{\setkeys{bxjs}{precise-text=true}}
```

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

```

635 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
636 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}
637 \DeclareOption{nosimplejasetup}{\setkeys{bxjs}{simple-ja-setup=false}}
638 \DeclareOption{simplejasetup}{\setkeys{bxjs}{simple-ja-setup=true}}

```

■ 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。

```

639 \let\bxjs@org@removeelement\@removeelement
640 \def\@removeelement#1#2#3{%
641   \def\reserved@a{#2}%
642   \ifx\reserved@a\@empty \let#3\@empty
643   \else \bxjs@org@removeelement{#1}{#2}{#3}%
644   \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。

```

645 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
646 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
647 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
648 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
649 \ProcessOptions\relax
650 \bxjs@post@option@hook

```

後処理

```

651 \if@slide
652   \def\maybeblue{\@ifundefined{ver@color.sty}{}{\color{blue}}}
653 \fi
654 \if@landscape
655   \setlength\@tempdima {\paperheight}
656   \setlength\paperheight{\paperwidth}
657   \setlength\paperwidth {\@tempdima}
658 \fi

```

■ グローバルオプションの整理

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

```

659 \def\bxjs@tmpdo{%

```

```

660 \def\bxjs@tmpa{\@gobble}%
661 \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
662 \let\@classoptionslist\bxjs@tmpa}
663 \def\bxjs@tmpdo@a#1,{%
664 \ifx\@nil#1\relax\else
665 \bxjs@tmpdo@b#1}\@nil
666 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
667 \expandafter\bxjs@tmpdo@a
668 \fi}
669 \def\bxjs@tmpdo@b#1#\{bxjs@tmpdo@c}
670 \def\bxjs@tmpdo@c#1\@nil{%
671 \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
672 \bxjs@tmpdo

```

papersize、10pt、noscale の各オプションは他のパッケージと衝突を起こす可能性があるため、グローバルオプションから外す。

```

673 \@expandtwoargs\@removeelement
674 {papersize}\@classoptionslist\@classoptionslist
675 \@expandtwoargs\@removeelement
676 {10pt}\@classoptionslist\@classoptionslist
677 \@expandtwoargs\@removeelement
678 {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。

ここを削除。

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

正規化前の和文ドライバの値を `\bxjs@jadriver` に設定する。

```

679 \ifx\bxjs@jadriver@opt\@undefined\else
680 \let\bxjs@jadriver\bxjs@jadriver@opt
681 \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。

```

682 \let\bxjs@tmpb\jsEngine
683 \ifx j\bxjs@tmpb\ifjsWithpTeXng

```

```

684 \let\bxjs@tmpb=g
685 \fi\fi
686 \ifx j\bxjs@tmpb\ifjsWithupTeX
687 \let\bxjs@tmpb=u
688 \fi\fi
689 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
690 \let\bxjs@tmpb=n
691 \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。)

```

692 \ifx *\bxjs@engine@given
693 \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。

```

694 \ifx j\bxjs@engine@given
695 \g@addto@macro\@classoptionslist{,latex}
696 \else\ifx u\bxjs@engine@given
697 \g@addto@macro\@classoptionslist{,uplatex}
698 \fi\fi
699 \fi
700 \ifx\bxjs@engine@given\@undefined\else
701 \ifx\bxjs@engine@given\bxjs@tmpb\else
702 \ClassError\bxjs@clsname
703 {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
704 \fi
705 \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。

```

706 \ifjsWithpTeXng
707 \g@addto@macro\@classoptionslist{,uplatex}
708 \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。

```

709 \@tempswatruue
710 \ifx \bxjs@driver@given\@undefined\else
711 \ifjsInPdfMode
712 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
713 \@tempswafalse
714 \fi
715 \else\ifx x\jsEngine
716 \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
717 \@tempswafalse
718 \fi
719 \else
720 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
721 \@tempswafalse

```

```

722 \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
723 \@tempswafalse
724 \fi\fi
725 \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
726 \@tempswafalse
727 \fi\fi
728 \fi\fi
729 \fi
730 \if@tempswa\else
731 \ClassError\bxjs@clsname
732 {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
733 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

734 \ifjsInPdfMode \@tempswafalse
735 \else\ifx x\jsEngine \@tempswafalse
736 \else\ifjsWithpTeXng \@tempswafalse
737 \else \@tempswatru
738 \fi\fi\fi
739 \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)。

```

740 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@opt\@undefined\else
741 \ClassWarningNoLine\bxjs@clsname
742 {No driver option is given}
743 \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。)

```

744 \ifbxjs@dvi@opt
745 \edef\bxjs@next{%
746 \let\noexpand\bxjs@driver@given
747 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
748 \noexpand\g@addto@macro\noexpand\@classoptionslist
749 {,\bxjs@driver@opt}%
750 }\bxjs@next
751 \fi
752 \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。

```

753 \ifjsWithpTeXng
754 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
755 \let\bxjs@platexng@nodrv\@undefined
756 \else\ifx t\bxjs@platexng@nodrv\else
757 \g@addto@macro\@classoptionslist{,dvipdfmx}
758 \fi\fi

```

```
759 \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。

```
760 \ifx\bxjs@driver@given\bxjs@driver@@none
```

```
761 \bxjs@papersizefalse
```

```
762 \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。

```
763 \def\bxjs@@minimal{minimal}
```

```
764 \ifx\bxjs@jadriver\relax
```

```
765 \ifx j\jsEngine
```

```
766 \def\bxjs@jadriver{standard}
```

```
767 \else
```

```
768 \let\bxjs@jadriver\bxjs@@minimal
```

```
769 \fi
```

```
770 \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。

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

```
772 \ifx\bxjs@engine@given\@undefined
```

```
773 \ClassError\bxjs@clsname
```

```
774 {An engine option must be explicitly given}%
```

```
775 {When you use a Japanese-driver you must specify a correct\MessageBreak
```

```
776 engine option.\MessageBreak\@ehc}
```

```
777 \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。)

```
778 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
```

```
779 \ifx\bxjs@magstyle\bxjs@magstyle@mag
```

```
780 \let\bxjs@magstyle\bxjs@magstyle@default
```

```
781 \ClassError\bxjs@clsname
```

```
782 {The engine does not support 'magstyle=usemag'}%
```

```
783 {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
```

```
784 The default value 'nomag*' is used instead.\MessageBreak \@ehc}
```

```
785 \fi
```

```
786 \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。

```
787 \ifx\bxjs@jbase@opt\@undefined\else
```

```
788 \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。


```

789 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
790 \bxjs@invscale\@tempdima\jsScale
791 \bxjs@setbasefontsize{\@tempdima}%
792 \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。
793 \ifx\bxjs@scale@opt\undefined\else
794 \ClassWarningNoLine\bxjs@clsname
795 {Redundant 'scale' option is ignored}%
796 \fi
797 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
798 \@tempdimb=\bxjs@param@basefontsize\relax
799 \edef\jsScale{\strip@pt\@tempdimb}%
800 \bxjs@invscale\@tempdima\jsScale
801 \edef\jsScale{\strip@pt\@tempdima}%
802 \fi
803 \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。

```
804 \let\Cjascale\jsScale
```

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

```

805 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
806 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
807 \catcode\@tempcnta\active
808 \advance\@tempcnta\@ne
809 \repeat
810 \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。

```

811 \ifbxjs@disguise@js
812 %<book|report>\def\bxjs@js@clsname{jsbook}
813 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
814 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
815 \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。

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

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

```

817 \ifbxjs@oldfontcommands
818 \AtEndOfClass{\allowoldfontcommands}
819 \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。

```
820 \newdimen\stockwidth \newdimen\stockheight
821 \begingroup\expandafter\expandafter\expandafter\endgroup
822 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
823 % \newdimen\stockwidth \newdimen\stockheight
824 \setlength{\stockwidth}{\paperwidth}
825 \setlength{\stockheight}{\paperheight}
826 \advance \stockwidth 2in
827 \advance \stockheight 2in
828 \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。

```
829 %<slide>\def\n@baseline{13}%
830 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
831 %<!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。

```
832 \ifx\bxjs@magstyle\bxjs@magstyle@mag
```

```

833 \jsc@magtrue
834 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
835 \jsc@mag@xrealtrue
836 \fi\fi

```

サイズの変更は T_EX 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。

```

837 \ifx\bxjs@param@mag\relax
838 \@tempdima=\bxjs@param@basefontsize
839 \advance\@tempdima.001pt \multiply\@tempdima25
840 \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
841 \edef\bxjs@param@mag{\the\@tempcnta}
842 \else
843 % mag 値が直接指定された場合
844 \bxjs@gset@tempcnta{\bxjs@param@mag}
845 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
846 % 有効な mag 値の範囲は 1--32768
847 \edef\bxjs@param@mag{\the\@tempcnta}
848 \advance\@tempcnta100000
849 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
850 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
851 \edef\bxjs@param@basefontsize{\the\@tempdima}
852 \fi
853 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
854 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
855 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
856 \edef\jsc@magscale{\strip@pt\@tempdima}
857 \let\jsBaseFontSize\bxjs@param@basefontsize

```

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

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

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

```
858 \newdimen\jsc@mpt
859 \newdimen\jsc@mmm
860 \ifjsc@mag
861   \jsc@mpt=1\p@
862   \jsc@mmm=1mm
863   \catcode`\?=9 % \p@? read as \p@
864 \else
865   \jsc@mpt=\jsc@magscale\p@
866   \jsc@mmm=\jsc@magscale mm
867   \catcode`\?=11 \let\p@?\jsc@mpt
868 \fi
869 \chardef\bxjs@qmc=\catcode`\?\relax
870 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode`\?=12\relax}
```

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

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

```
871 \newdimen\jsZw
872 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
873 \ifbxjs@usezw
874   \providecommand*\zw{\jsZw}
875 \fi
```

`\zwspace` 全角幅の水平空き。

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

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

```
877 \ifjsc@mag@xreal
878   \RequirePackage{type1cm}
879   \let\jsc@invscale\bxjs@invscale
```

ムニヤムニヤムニヤ……。

```
880 \ifbxjs@TUenc
881   \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
882 \else
883   \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
884 \fi
885 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
886 \let\jsc@get@external@font\get@external@font
887 \def\get@external@font{%
888   \jsc@preadjust@extract@font
889   \jsc@get@external@font}
890 \def\jsc@fstrunc#1{%
891   \edef\jsc@tmpa{\strip@pt#1}%
892   \expandafter\jsc@fstrunc@a\jsc@tmpa.***\@nil}
893 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
894   \if#5*\else
```

```

895     \edef\jsc@tmpa{#1%
896     \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
897     \fi}
898     \def\jsc@preadjust@extract@font{%
899     \let\jsc@req@size\fontsize
900     \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
901     \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
902     \let\jsc@ref@size\jsc@tmpa
903     \let\fontsize\jsc@ref@size}
904     \def\execute@size@function#1{%
905     \let\jsc@c@ref@size\fontsize
906     \let\fontsize\jsc@req@size
907     \csname s@fct@#1\endcsname}
908     \let\jsc@DeclareErrorFont\DeclareErrorFont
909     \def\DeclareErrorFont#1#2#3#4#5{%
910     \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
911     \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
912     \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
913     \def\gen@sfcnt{%
914     \edef\mandatory@arg{\mandatory@arg\jsc@c@ref@size}%
915     \empty@sfcnt}
916     \def\genb@sfcnt{%
917     \edef\mandatory@arg{%
918     \mandatory@arg\expandafter\genb@x\jsc@c@ref@size..\@}%
919     \empty@sfcnt}
920     \ifbxjs@TUenc\else
921     \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
922     \fi
923     \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 924 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
925 %\def\jsc@medskip{\vspace\jsc@medskipamount}
\jsc@bigskip 926 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```
\jsc@smallskipamount
```

```

\jsc@medskipamount 927 \newskip\jsc@smallskipamount
928 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
\jsc@bigskipamount 929 %\newskip\jsc@medskipamount
930 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
931 %\newskip\jsc@bigskipamount
932 %\jsc@bigskipamount =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。

```
933 % \ifpapersize
934 %   \setlength{\@tempdima}{\paperwidth}
935 %   \setlength{\@tempdimb}{\paperheight}
936 %   \iftombow
937 %     \advance \@tempdima 2truein
938 %     \advance \@tempdimb 2truein
939 %   \fi
940 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
941 % \fi
```

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

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

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

BXJS クラスでの変更点：

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

```
942 \chardef\bxjs@periodchar=`\ .
943 \bxjs@protected\def\bxjs@SE{%
944   \ifnum\spacefactor<\@m \spacefactor\@m
945   \else \spacefactor\sfcode\bxjs@periodchar
946   \fi}
947 \ifbxjs@fix@at@cmd
948   \def\@{\bxjs@SE{}}
949 \fi
```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

```
950 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
951 \expandafter\bxjs@tmpa\expandafter{%
952   \set@fontsize{#1}{#2}{#3}%
953 % 末尾にコードを追加
954   \expandafter\def\expandafter\size@update\expandafter{%
955     \size@update
956     \jsFontSizeChanged}%
957 }
```

`\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。

```
958 \newcommand*\jsFontSizeChanged{%
959   \jsZw=\f@size\p@
960   \jsZw=\jsScale \jsZw
961   \ifdim\parindent>\z@
962     \if@english \parindent=1em
963     \else       \parindent=1\jsZw
964   \fi
965   \fi\relax
966   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
967 \newcommand*\jsResetDimen{}
```

`\jsc@setfontsize` クラスファイルの内部では、拡大率も考慮した `\jsc@setfontsize` を `\@setfontsize` の代わりに用いることにします。

```
968 \ifjscssmag
969   \let\jsc@setfontsize\@setfontsize
970 \else
971   \def\jsc@setfontsize#1#2#3{%
972     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
973 % microtype 対策
974 \ifjswitheTeX\if j\jsEngine\else
975   \def\jsc@setfontsize#1#2#3{%
976     \edef\bxjs@sfs@next{%
977       \unexpanded{\@setfontsize#1}%
978       {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
979     }\bxjs@sfs@next}
980 \fi\fi
981 \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。

```
982 \emergencystretch 3\jsZw
```

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

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

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

```
983 \newif\ifnarrowbaselines
984 \if@english
985   \narrowbaselinestrue
986 \fi
987 \def\narrowbaselines{%
988   \narrowbaselinestrue
989   \skip0=\abovedisplayskip
990   \skip2=\abovedisplayshortskip
991   \skip4=\belowdisplayskip
992   \skip6=\belowdisplayshortskip
993 % 一時的に警告を無効化する
994   \let\bxjs@save@nomath\@nomath
995   \let\@nomath\@gobble
996   \@currsize\selectfont
997   \let\@nomath\bxjs@save@nomath
998   \abovedisplayskip=\skip0
999   \abovedisplayshortskip=\skip2
1000  \belowdisplayskip=\skip4
1001  \belowdisplayshortskip=\skip6\relax}
1002 \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。

```
1003 \def\bxjs@if@narrowbaselines{%
1004   \ifnarrowbaselines\expandafter\@firstoftwo
1005   \else \expandafter\@secondoftwo
1006   \fi
1007 }
```

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

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

公称 10 ポイントの和文フォントが約 9.25 ポイント (アスキーのもの 0.961 倍) である

こともあり、行送りがかなりゆったりとしたと思います。実際、 $16/9.25 \approx 1.73$ であり、和文の推奨値の一つ「二分四分」(1.75) に近づきました。

microtype 対策のため if 文を避ける。

```
1008 \renewcommand{\normalsize}{%
1009   \bxjs@if@narrowbaselines{%
1010     \jsc@setfontsize\normalsize\@xpt\@xipt
1011   }{%else
1012     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1013   }%
```

数式の上のアキ(\abovedisplayskip), 短い数式の上のアキ(\abovedisplayskip), 数式の下のアキ(\belowdisplayskip) 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。

```
1014 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1015 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1016 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1017 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
```

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。

```
1018 \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。

```
1019 \normalsize
```

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

\Chs

\Cwd 等の変数は pT_EX 系以外では未定義なのでここで定義する。

```
1020 \ifx\Cht\@undefined \newdimen\Cht \fi
1021 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1022 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1023 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1024 \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。

```
1025 \setlength\Cht{0.88\jsZw}
1026 \setlength\Cdp{0.12\jsZw}
```

```

1027 \setlength\Cwd{1\jsZw}
1028 \setlength\Cvs{\baselineskip}
1029 \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。

```

1030 \newcommand{\small}{%
1031   \bxjs@if@narrowbaselines{%
1032     <!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1033     <kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1034   }{%else
1035     <!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1036     <kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1037   }%
1038   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1039   \abovedisplayshortskip \z@ \@plus3\p@?
1040   \belowdisplayskip \abovedisplayskip
1041   \belowdisplayshortskip \belowdisplayskip
1042   \def\@listi{\leftmargin\leftmargini
1043             \topsep \z@
1044             \parsep \z@
1045             \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。

```

1046 \newcommand{\footnotesize}{%
1047   \bxjs@if@narrowbaselines{%
1048     <!kiyou>   \jsc@setfontsize\footnotesize\@viiipt{9.5}%
1049     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1050   }{%else
1051     <!kiyou>   \jsc@setfontsize\footnotesize\@viiipt{11}%
1052     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1053   }%
1054   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1055   \abovedisplayshortskip \z@ \@plus2\p@?
1056   \belowdisplayskip \abovedisplayskip
1057   \belowdisplayshortskip \belowdisplayskip
1058   \def\@listi{\leftmargin\leftmargini
1059             \topsep \z@
1060             \parsep \z@
1061             \itemsep \parsep}}

```

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

```

\tiny
\large
\Large
\LARGE
\huge

```

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

[2004-11-03] `\HUGE` を追加。

```

1062 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1063 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
1064 \if@twocolumn
1065 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
1066 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1067 \else
1068 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
1069 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1070 \fi
1071 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1072 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1073 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1074 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1075 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1076 \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。

```

1077 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}

```

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

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

```

1078 % \newcommand{\headfont}{\bfseries}
1079 \newcommand{\headfont}{\sffamily}
1080 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

5 レイアウト

■二段組

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

```

1081 %<kiyou>\setlength\columnsep{2\Cwd}
1082 %<kiyou>\setlength\columnsep{28truebp}
1083 \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 1084 \setlength\lineskip{1\jsc@empt}
\normallineskip 1085 \setlength\normallineskip{1\jsc@empt}
\normallineskiplimit 1086 \setlength\lineskiplimit{1\jsc@empt}
1087 \setlength\normallineskiplimit{1\jsc@empt}

```

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

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

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

```
1088 \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。

```

1089 \setlength\parskip{\z@}
1090 \if@slide
1091 \setlength\parindent{0\p@}
1092 \else
1093 \setlength\parindent{1\Cwd}
1094 \fi

```

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

```

\@highpenalty 1095 \@lowpenalty 51
1096 \@medpenalty 151
1097 \@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。

```
1098 % \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。

```
1099 % \brokenpenalty 100
```

5.1 ページレイアウト

BXJS ではページレイアウトの処理は `geometry` パッケージが担当している。

■準備

`\bxjs@bd@pre@geometry@hook` begin-document フックのコード内で、`geometry` パッケージが挿入するコードの直前で実行されるフック。

```
1100 \@onlypreamble\bxjs@bd@pre@geometry@hook
1101 \let\bxjs@bd@pre@geometry@hook\@empty
```

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
1102 \ifjsc@mag
1103 \mag=\bxjs@param@mag
1104 \fi
1105 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLlength` のための和文単位の定義。

```
1106 \def\bxjs@unit@trueQ{0.25trueem}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1107 \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。

```
1108 \def\bxjs@tmpdo{\futurelet\bxjs@tmpa\bxjs@tmpdo@a}
1109 \def\bxjs@tmpdo@a{%
1110   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@tmpdo@b
1111   \else \expandafter\bxjs@tmpdo@c \fi}
1112 \def\bxjs@tmpdo@b#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1113 \def\bxjs@tmpdo@c#1\@nil{\bxjs@tmpdo@d#1,,\@nil}
1114 \def\bxjs@tmpdo@d#1,#2,#3\@nil{%
1115   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1116 \expandafter\bxjs@tmpdo\bxjs@param@paper\@empty\@empty\@nil
```

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

```
1117 \edef\bxjs@layout@paper{%
1118   \ifjsc@mag truedimen,\fi
1119   \if@landscape landscape,\fi
1120   \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。

```
1121 %<article|report>
1122 \def\bxjs@layout@base{%
1123   headheight=\topskip,footskip=0.03367\paperheight,%
1124   headsep=\footskip-\topskip,includeheadfoot,%
1125 }
1126 \edef\bxjs@layout{\bxjs@layout@base
1127   hscale=0.76,hmarginratio=1:1,%
1128   vscale=0.83,vmarginratio=1:1,%
1129 }
1130 %</article|report>
```

```

1131 %<*book>
1132 \def\bxjs@layout@base{%
1133   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1134 }
1135 \ifbxjs@layout@buggyhmargin    %---
1136 % アレ
1137 \edef\bxjs@layout{\bxjs@layout@base
1138   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1139   vscale=0.83,vmarginratio=1:1,%
1140 }
1141 \else                            %---
1142 % 非アレ
1143 \edef\bxjs@layout{\bxjs@layout@base
1144   hmargin=18\jsc@mmm,%
1145   vscale=0.83,vmarginratio=1:1,%
1146 }
1147 \fi                                %---
1148 %</book>
1149 %<*slide>
1150 \def\bxjs@layout@base{%
1151   noheadfoot,%
1152 }
1153 \edef\bxjs@layout{\bxjs@layout@base
1154   hscale=0.9,hmarginratio=1:1,%
1155   vscale=0.944,vmarginratio=1:1,%
1156 }
1157 %</slide>

```

`textwidth` オプションの設定を反映する。

```

1158 %<*!book>
1159 \ifx\bxjs@textwidth@opt\@undefined\else
1160   \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1161   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1162 \fi
1163 %</!book>
1164 \ifx\bxjs@number@of@lines@opt\@undefined\else
1165   \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1166   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1167 \fi

```

`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1168 \newdimen\fullwidth
```

`\bxjs@textwidth@limit` [寸法値マクロ] `bxjsbook` における、`\textwidth` 上限の値。

`\jsTextWidthLimit` [実数値マクロ] `\bxjs@textwidth@limit` の全角 (`\Cwd`) 単位での値。

```

1169 %<*book>
1170 \newcommand\jsTextWidthLimit{40}
1171 \@tempdima=\jsTextWidthLimit\Cwd

```

```

1172 \ifx\bxjs@textwidth@limit@opt\undefined\else
1173   \bxjs@gset@tempcnta{\bxjs@textwidth@limit@opt}
1174   \@tempdima=\@tempcnta\Cwd
1175 \fi
1176 \ifx\bxjs@textwidth@opt\undefined\else
1177   \jsSetQHLLength\@tempdima{\bxjs@textwidth@opt}
1178 \fi
1179 \edef\bxjs@textwidth@limit{\the\@tempdima}
1180 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1181   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1182   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1183 \fi
1184 %</book>

```

`\bxjs@preproc@layout` geometry の前処理。

geometry は `\topskip` が標準の行高 (`\ht\strutbox`) より小さくならないようにする自動調整を行うが、これをどうするかは未検討。今のところ、単純に回避 (無効化) している。

```

1185 \def\bxjs@preproc@layout{%
1186   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mpt}

```

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

```

1187 \def\bxjs@postproc@layout{%
    geometry のドライバを再設定する。
1188   \ifx\bxjs@geometry@driver\relax\else
1189     \let\Gm@driver\bxjs@geometry@driver
1190   \fi
    \ht\strutbox の値を元に戻す。
1191   \ht\strutbox=\bxjs@save@ht@strutbox\relax
    \textwidth の値を補正する。
1192   \ifbxjs@whole@zw@lines
1193     \@tempdimb=\textwidth
1194     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1195     \advance\textwidth.005pt\relax
1196     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1197     \advance\@tempdimb-\textwidth
1198     \advance\oddsidemargin 0.5\@tempdimb
1199     \advance\evensidemargin 0.5\@tempdimb
1200   \fi
1201   \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。

```

1202 %<*book>
1203   \@tempdima=\bxjs@textwidth@limit\relax
1204   \ifbxjs@whole@zw@lines

```



```

1205 \advance\@tempdima.005pt\relax
1206 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1207 \fi
1208 \ifdim\textwidth>\@tempdima
1209 \textwidth=\@tempdima
1210 \addtolength\evensidemargin{\fullwidth-\textwidth}
1211 \fi
1212 %</book>

```

\textheight 関連の調整。

```

1213 \@tempdimb=\textheight
1214 \advance\textheight-\topskip
1215 \advance\textheight.005pt\relax
1216 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1217 \advance\textheight\topskip
1218 \advance\@tempdimb-\textheight
1219 \advance\topmargin0.5\@tempdimb

```

\headheight 関連の調整。

```

1220 \@tempdima=\topskip
1221 \advance\headheight\@tempdima
1222 \advance\topmargin-\@tempdima

```

marginpar 関連の調整。

```

1223 \setlength\marginparsep{\columnsep}
1224 \setlength\marginparpush{\baselineskip}
1225 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1226 -\textwidth-10\jsc@mmm-\marginparsep}
1227 \ifbxjs@whole@zw@lines
1228 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1229 \fi

```

連動する変数。

```

1230 \maxdepth=.5\topskip
1231 \stockwidth=\paperwidth
1232 \stockheight=\paperheight
1233 }

```


\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。

```

1234 \edef\jsGeometryOptions{%
1235 \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージを読み込む 

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

```

1236 \ifx\bxjs@geometry\bxjs@geometry@class

```

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

```

1237 \ifbxjs@papersize
1238   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1239     \PassOptionsToPackage{dvipdfm}{geometry}
1240   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1241     \PassOptionsToPackage{dvipdfm}{geometry}
1242   \fi\fi
1243   \let\bxPapersizeSpecialDone=t
1244 \else
1245   \PassOptionsToPackage{driver=none}{geometry}
1246 \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。

```

1247 \AtBeginDocument{\bxjs@bd@pre@geometry@hook}
1248 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1249 \bxjs@preproc@layout
1250 \edef\bxjs@next{%
1251   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1252 }\bxjs@next
1253 \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。

```

1254 \let\bxjs@geometry@driver\Gm@driver
1255 \bxjs@postproc@layout

```

`geometry` のドライバ自動判別に対する前処理。

```

1256 \g@addto@macro\bxjs@bd@pre@geometry@hook{%

```

`geometry` の 4.x 版に関する対策。

※`geometry` の 4.x 版の使用は BXJS 1.2 版より非推奨とし、2.0 版でサポートを廃止する予定。

```

1257   \@ifpackagelater{geometry}{2010/02/12}{-}{%else
1258     \PackageWarningNoLine\bxjs@clsname
1259     {The 'geometry' package installed\MessageBreak
1260      is too old (< v5.0)}%
1261     \if x\jsEngine \ifnum\mag=\@m\else
1262       \def\bxjs@Gm@driver{pdftex}
1263       \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1264       \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1265       \fi\fi}%

```

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

```

1266 \ifjsWithpTeXng
1267 \ifx\Gm@driver\@empty
1268 \def\Gm@driver{pdftex}%
1269 \fi
1270 \fi}

```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```

1271 \def\setpagelayout{%
1272 \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1273 \@ifstar{\bxjs@setpagelayout@a@ne}{\bxjs@setpagelayout@a@z@}}
1274 \def\bxjs@setpagelayout@a#1#2{%
1275 \ifcase#1% modify
1276 \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1277 \or% reset(*)
1278 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1279 \or% semireset(+)
1280 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1281 \fi
1282 \bxjs@preproc@layout
1283 \edef\bxjs@next{%
1284 \noexpand\geometry{\bxjs@next}%
1285 }\bxjs@next
1286 \bxjs@postproc@layout}

```

■ geometry パッケージを読み込まない

`geometry=user` の場合の処理。

```

1287 \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。

```

1288 \g@addto@macro\bxjs@begin@document@hook{%
1289 \ifdim\textwidth=.5\maxdimen
1290 \ClassError\bxjs@clsname
1291 {Page layout is not properly set}%
1292 {\@ehd}%
1293 \fi}
1294 \def\jsUseMinimalPageLayout{%
1295 \setlength{\textwidth}{6.5in}%
1296 \setlength{\textheight}{8in}}

```

`\setpagelayout` はとりあえず無効にしておく。

```

1297 \let\bxjs@geometry@driver\relax
1298 \def\setpagelayout{%
1299 \bxjs@ifplus{\bxjs@pagelayout@a}{%else

```

```

1300 \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1301 \def\bxjs@pagelayout@a#1{%
1302 \ClassError\bxjs@clsname
1303 {Command '\string\setpagelayout' is not supported,\MessageBreak
1304 because 'geometry' value is not 'class'}\@eha}
1305 %
1306 \fi\fi

```

■ JS クラスと共通処理の開始

ここからのコードは以下の点を除いて JS クラスのものを踏襲する。

- zw の代わりに `\jsZw` を用いる。
- `article/report/book/slide` の切り分けの処理が異なる。

※ diff が崩壊するのを避けるためオリジナルのコードを無効化した状態で挿入しておく。

```
1307 %<*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 f 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。

```

1308 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1309 \if@slide
1310 \setlength\headheight{0\jsc@mpt}
1311 \else
1312 \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
    06-26)
1313 \fi

```

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

```

1314 %<*article|kiyou>
1315 \if@slide

```

```

1316 \setlength\footskip{0pt}
1317 \else
1318 \setlength\footskip{0.03367\paperheight}
1319 \ifdim\footskip<\baselineskip
1320 \setlength\footskip{\baselineskip}
1321 \fi
1322 \fi
1323 %</article|kiyou>
1324 %<jspf>\setlength\footskip{9\jsc@mmm}
1325 %<*book>
1326 \if@report
1327 \setlength\footskip{0.03367\paperheight}
1328 \ifdim\footskip<\baselineskip
1329 \setlength\footskip{\baselineskip}
1330 \fi
1331 \else
1332 \setlength\footskip{0pt}
1333 \fi
1334 %</book>
1335 %<*report>
1336 \setlength\footskip{0.03367\paperheight}
1337 \ifdim\footskip<\baselineskip
1338 \setlength\footskip{\baselineskip}
1339 \fi
1340 %</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)。

```

1341 %<*article>
1342 \if@slide
1343 \setlength\headsep{0\jsc@empt}
1344 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1345 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1346 \else
1347 \setlength\headsep{\footskip}
1348 \addtolength\headsep{-\topskip}
1349 \fi
1350 %</article>
1351 %<*book>
1352 \if@report
1353 \setlength\headsep{\footskip}
1354 \addtolength\headsep{-\topskip}
1355 \else
1356 \setlength\headsep{6\jsc@mmm}
1357 \addtolength\headsep{-\topskip}%% added (2016-10-08)

```

```

1358 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1359 \fi
1360 %</book>
1361 %<*report>
1362 \setlength\headsep{\footskip}
1363 \addtolength\headsep{-\topskip}
1364 %</report>
1365 %<*jspf>
1366 \setlength\headsep{9\jsc@mmm}
1367 \addtolength\headsep{-\topskip}
1368 %</jspf>
1369 %<*kiyou>
1370 \setlength\headheight{0\jsc@mpt}
1371 \setlength\headsep{0\jsc@mpt}
1372 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1373 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1374 %</kiyou>

```

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

```
1375 \setlength\maxdepth{.5\topskip}
```

■本文の幅と高さ

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

```
1376 \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。

```

1377 %<*article>
1378 \if@slide
1379 \setlength\fullwidth{0.9\paperwidth}
1380 \else
1381 \setlength\fullwidth{0.76\paperwidth}
1382 \fi
1383 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi

```

```

1384 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1385 \setlength\textwidth{\fullwidth}
1386 %</article>
1387 %<*book>
1388 \if@report
1389 \setlength\fullwidth{0.76\paperwidth}
1390 \else
1391 \setlength\fullwidth{\paperwidth}
1392 \addtolength\fullwidth{-36\jsc@mmm}
1393 \fi
1394 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1395 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1396 \setlength\textwidth{\fullwidth}
1397 \if@report \else
1398 \if@twocolumn \else
1399 \ifdim \fullwidth>40zw
1400 \setlength\textwidth{40zw}
1401 \fi
1402 \fi
1403 \fi
1404 %</book>
1405 %<*report>
1406 \setlength\fullwidth{0.76\paperwidth}
1407 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1408 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1409 \setlength\textwidth{\fullwidth}
1410 %</report>
1411 %<*jspf>
1412 \setlength\fullwidth{50zw}
1413 \addtolength\fullwidth{8\jsc@mmm}
1414 \setlength\textwidth{\fullwidth}
1415 %</jspf>
1416 %<*kiyou>
1417 \setlength\fullwidth{48zw}
1418 \addtolength\fullwidth{\columnsep}
1419 \setlength\textwidth{\fullwidth}
1420 %</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)。

```
1421 %<*article|book|report>
1422 \if@slide
1423   \setlength{\textheight}{0.95\paperheight}
1424 \else
1425   \setlength{\textheight}{0.83\paperheight}
1426 \fi
1427 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
      \headheight (2003-06-26)
1428 \addtolength{\textheight}{-\headsep}
1429 \addtolength{\textheight}{-\footskip}
1430 \addtolength{\textheight}{-\topskip}
1431 \divide\textheight\baselineskip
1432 \multiply\textheight\baselineskip
1433 %</article|book|report>
1434 %<jspf>\setlength{\textheight}{51\baselineskip}
1435 %<kiyou>\setlength{\textheight}{47\baselineskip}
1436 \addtolength{\textheight}{\topskip}
1437 \addtolength{\textheight}{0.1\jsc@mpt}
1438 %<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。

```
1439 \def\flushbottom{%
1440   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1441   \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。

```
1442 \setlength\marginparsep{\columnsep}
1443 \setlength\marginparpush{\baselineskip}
```

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

```
1444 \setlength{\oddsidemargin}{\paperwidth}
```



```

1445 \addtolength{\oddsidemargin}{-\fullwidth}
1446 \setlength{\oddsidemargin}{.5\oddsidemargin}
1447 \iftombow
1448   \addtolength{\oddsidemargin}{-1in}
1449 \else
1450   \addtolength{\oddsidemargin}{-\inv@mag in}
1451 \fi
1452 \setlength{\evensidemargin}{\oddsidemargin}
1453 \if@mparswitch
1454   \addtolength{\evensidemargin}{\fullwidth}
1455   \addtolength{\evensidemargin}{-\textwidth}
1456 \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。

```

1457 \setlength\marginparwidth{\paperwidth}
1458 \addtolength\marginparwidth{-\oddsidemargin}
1459 \addtolength\marginparwidth{-\inv@mag in}
1460 \addtolength\marginparwidth{-\textwidth}
1461 \addtolength\marginparwidth{-10\jsc@mmm}
1462 \addtolength\marginparwidth{-\marginparsep}
1463 \@tempdima=1zw
1464 \divide\marginparwidth\@tempdima
1465 \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)。

```

1466 \setlength\topmargin{\paperheight}
1467 \addtolength\topmargin{-\textheight}
1468 \if@slide
1469   \addtolength\topmargin{-\headheight}
1470 \else
1471   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
     \headheight (2003-06-26)
1472 \fi
1473 \addtolength\topmargin{-\headsep}
1474 \addtolength\topmargin{-\footskip}
1475 \setlength\topmargin{0.5\topmargin}
1476 %<kiyou>\setlength\topmargin{81truebp}
1477 \iftombow
1478   \addtolength\topmargin{-1in}
1479 \else

```

```
1480 \addtolength\topmargin{-\inv@mag in}
1481 \fi
1482 %</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。

```
1483 \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。

```
1484 \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。

```
1485 \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。

```
1486 \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。

```
1487 \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。

```
1488 \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。

```
1489 \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。

```
1490 \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。

```
1491 \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。

```
1492 \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。

```
1493 \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。

```
1494 \renewcommand{\dblfloatpagefraction}{.8}
```

`\floatsep` `\floatsep` はページ上部・下部のフロート間の距離です。`\textfloatsep` はページ上部・

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

`\intertextsep` 文との距離です。

```
1495 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1496 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1497 \setlength\intertextsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

`\dblfloatsep` 二段組のときの段抜きのフロートについての値です。

```
\dbltextfloatsep 1498 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1499 \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 1500 \setlength\@fptop{0\p@? \@plus 1fil}
```

```
1501 \setlength\@fpsep{8\p@? \@plus 2fil}
```

```
1502 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
\@dblfpsep 1503 \setlength\@dblfpsep{0\p@? \@plus 1fil}
```

```
1504 \setlength\@dblfpsep{8\p@? \@plus 2fil}
```

```
\@dblfpbot 1505 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ (日本語 T_EX 開発コミュニティ版のみ)

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

```
\pltx@cleartoorightpage
```

```
\pltx@cleartoevenpage
```

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

となっています。

```

1506 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1507 % \ifodd\c@page
1508 %   \iftdir
1509 %     \hbox{}\thispagestyle{empty}\newpage
1510 %     \if@twocolumn\hbox{}\newpage\fi
1511 %   \fi
1512 % \else
1513 %   \ifydir
1514 %     \hbox{}\thispagestyle{empty}\newpage
1515 %     \if@twocolumn\hbox{}\newpage\fi
1516 %   \fi
1517 % \fi\fi}
1518 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1519 % \ifodd\c@page
1520 %   \ifydir
1521 %     \hbox{}\thispagestyle{empty}\newpage
1522 %     \if@twocolumn\hbox{}\newpage\fi
1523 %   \fi
1524 % \else
1525 %   \iftdir
1526 %     \hbox{}\thispagestyle{empty}\newpage
1527 %     \if@twocolumn\hbox{}\newpage\fi
1528 %   \fi
1529 % \fi\fi}
1530 \def\pltx@cleartooddpage{\clearpage\if@twoside
1531 \ifodd\c@page\else
1532 \hbox{}\thispagestyle{empty}\newpage
1533 \if@twocolumn\hbox{}\newpage\fi
1534 \fi\fi}
1535 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1536 \ifodd\c@page
1537 \hbox{}\thispagestyle{empty}\newpage
1538 \if@twocolumn\hbox{}\newpage\fi
1539 \fi\fi}

```

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

```

1540 \let\pltx@cleartorightpage\pltx@cleartooddpage
1541 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

クラスの場合に`\cleardoublepage`を再定義します。

```
1542 %<*book|report>
1543 \if@openleft
1544   \let\cleardoublepage\ptx@cleartoleftpage
1545 \else\if@openright
1546   \let\cleardoublepage\ptx@cleartorightpage
1547 \fi\fi
1548 %</book|report>
```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

柱を設定する命令は、右の柱が左の柱の下位にある場合は十分まともに動作します。たとえば左マークを `\chapter`, 右マークを `\section` で変更する場合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、おかしな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。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 本体で定義されているものをコメントアウトした形で載せておきます。

```
1549 % \def\ps@empty{%
1550 %   \let\mkboth@gobbletwo
1551 %   \let\@oddhead@empty
1552 %   \let\@oddfoot@empty
1553 %   \let\@evenhead@empty
1554 %   \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。

```
1555 \def\ps@plainfoot{%
1556   \let\@mkboth\@gobbletwo
1557   \let\@oddhead\@empty
1558   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1559   \let\@evenhead\@empty
1560   \let\@evenfoot\@oddfoot}
1561 \def\ps@plainhead{%
1562   \let\@mkboth\@gobbletwo
1563   \let\@oddfoot\@empty
1564   \let\@evenfoot\@empty
1565   \def\@evenhead{%
1566     \if@mparswitch \hss \fi
1567     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1568     \if@mparswitch\else \hss \fi}%
1569   \def\@oddhead{%
1570     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1571 %<book>\let\ps@plain\ps@plainhead
1572 %<!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。

```
1573 %<*article|slide>
1574 \if@twoside
1575   \def\ps@headings{%
1576     \let\@oddfoot\@empty
1577     \let\@evenfoot\@empty
1578     \def\@evenhead{\if@mparswitch \hss \fi
1579       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1580     \if@mparswitch\else \hss \fi}%
1581     \def\@oddhead{%
1582       \underline{%
1583         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1584     \let\@mkboth\markboth
1585     \def\sectionmark##1{\markboth{%
1586       \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1587       ##1}{}}%
1588     \def\subsectionmark##1{\markright{%
1589       \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1590       ##1}}%
1591   }
1592 \else % if not twoside
1593   \def\ps@headings{%
1594     \let\@oddfoot\@empty
1595     \def\@oddhead{%
```

```

1596     \underline{%
1597     \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}\hss}%
1598     \let\@mkboth\markboth
1599     \def\sectionmark##1{\markright{%
1600     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1601     ##1}}
1602 \fi
1603 %</article|slide>

```

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

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

```

1604 %<*book|report>
1605 \def\bxjs@maybe@autoxspacing{%
1606   \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1607 \newif\if@omit@number
1608 \def\ps@headings{%
1609   \let\@oddfont\@empty
1610   \let\@evenfont\@empty
1611   \def\@evenhead{%
1612     \if@mparswitch \hss \fi
1613     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1614     \textbf{\thepage}\hfil\leftmark}}}%
1615     \if@mparswitch\else \hss \fi}%
1616   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1617     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1618   \let\@mkboth\markboth
1619   \def\chaptermark##1{\markboth{%
1620     \ifnum \c@secnumdepth >\m@ne
1621       \if@mainmatter
1622         \if@omit@number\else
1623           \@chapapp\thechapter\@chappos\hskip1\jsZw
1624         \fi
1625       \fi
1626     \fi
1627     ##1}{}}%
1628   \def\sectionmark##1{\markright{%
1629     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1630     ##1}}}%
1631 %</book|report>

```

最後は学会誌の場合です。

```

1632 %<*jspf>
1633 \def\ps@headings{%
1634   \def\@oddfont{\normalfont\hfil\thepage\hfil}

```

```

1635 \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1636 \def\@oddhead{\normalfont\hfil \@title \hfil}
1637 \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1638 %</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。

```

1639 \def\ps@myheadings{%
1640 \let\@oddfoot\empty\let\@evenfoot\empty
1641 \def\@evenhead{%
1642 \ifmparswitch \hss \fi%
1643 \hbox to \fullwidth{\thepage\hfil\leftmark}%
1644 \ifmparswitch\else \hss \fi}%
1645 \def\@oddhead{%
1646 \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1647 \let\@mkboth\@gobbletwo
1648 %<book|report> \let\chaptermark\@gobble
1649 \let\sectionmark\@gobble
1650 %<!book&!report> \let\subsectionmark\@gobble
1651 }

```

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 1652 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1653 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1654 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1655 % \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。

```

1656 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1657 %\let\bxjs@subtitle\undefined

\title にフックを入れる。

1658 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1659 \AtBeginDocument{\bxjs@decl@subtitle}
1660 \def\bxjs@decl@subtitle{%

```



```

1661 \global\let\bxjs@decl@subtitle\relax
1662 \ifx\subtitle\@undefined
1663 \global\let\subtitle\jsSubtitle
1664 \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。

```

1665 \def\bxjs@annihilate@subtitle{%
1666 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1667 \global\let\jsSubtitle\relax}

```

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

```

\author 1668 %<*jspf>
\keywords 1669 \newcommand*\etitle}[1]{\gdef\@etitle{#1}}
1670 \newcommand*\eauthor}[1]{\gdef\@eauthor{#1}}
1671 \newcommand*\keywords}[1]{\gdef\@keywords{#1}}
1672 \newcommand*\email}[1]{\gdef\authors@mail{#1}}
1673 \newcommand*\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1674 %</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。

```

1675 \def\plainifnotempty{%
1676 \ifx \@oddhead \@empty
1677 \ifx \@oddfoot \@empty
1678 \else
1679 \thispagestyle{plainfoot}%
1680 \fi
1681 \else
1682 \thispagestyle{plainhead}%
1683 \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。

```

1684 %<*article|book|report|slide>
1685 \if@titlepage
1686 \newcommand{\maketitle}{%
1687 \begin{titlepage}%
1688 \let\footnotesize\small

```

```

1689 \let\footnoterule\relax
1690 \let\footnote\thanks
1691 \null\vfil
1692 \if@slide
1693   {\footnotesize \@date}%
1694   \begin{center}
1695     \mbox{} \\\[1\jssZw]
1696     \large
1697     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1698     \jsc@smallskip
1699     \@title
1700     \ifx\bxjs@subtitle\@undefined\else
1701       \par\vskip\z@
1702       {\small \bxjs@subtitle\par}
1703     \fi
1704     \jsc@smallskip
1705     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1706     \vfill
1707     {\small \@author}%
1708   \end{center}
1709 \else
1710 \vskip 60\p@?
1711 \begin{center}%
1712   {\LARGE \@title \par}%
1713   \ifx\bxjs@subtitle\@undefined\else
1714     \vskip5\p@?
1715     {\normalsize \bxjs@subtitle\par}
1716   \fi
1717   \vskip 3em%
1718   {\large
1719     \lineskip .75em
1720     \begin{tabular}[t]{c}%
1721       \@author
1722     \end{tabular}\par}%
1723   \vskip 1.5em
1724   {\large \@date \par}%
1725 \end{center}%
1726 \fi
1727 \par
1728 \@thanks\vfil\null
1729 \end{titlepage}%
1730 \setcounter{footnote}{0}%
1731 \global\let\thanks\relax
1732 \global\let\maketitle\relax
1733 \global\let\@thanks\@empty
1734 \global\let\@author\@empty
1735 \global\let\@date\@empty
1736 \global\let\@title\@empty
1737 \global\let\title\relax

```

```

1738 \global\let\author\relax
1739 \global\let\date\relax
1740 \global\let\and\relax
1741 \bxjs@annihilate@subtitle
1742 }%
1743 \else
1744 \newcommand{\maketitle}{\par
1745 \begin{group}
1746 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1747 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1748 \long\def\@makefnmark##1{\advance\leftskip 3\jsZw
1749 \parindent 1\jsZw\noindent
1750 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1751 \if@twocolumn
1752 \ifnum \col@number=\@ne
1753 \maketitle
1754 \else
1755 \twocolumn[\maketitle]%
1756 \fi
1757 \else
1758 \newpage
1759 \global\@topnum\z@ % Prevents figures from going at top of page.
1760 \maketitle
1761 \fi
1762 \plainifnotempty
1763 \@thanks
1764 \endgroup
1765 \setcounter{footnote}{0}%
1766 \global\let\thanks\relax
1767 \global\let\maketitle\relax
1768 \global\let\@thanks\@empty
1769 \global\let\@author\@empty
1770 \global\let\@date\@empty
1771 \global\let\@title\@empty
1772 \global\let\title\relax
1773 \global\let\author\relax
1774 \global\let\date\relax
1775 \global\let\and\relax
1776 \bxjs@annihilate@subtitle
1777 }

```

`\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。

```

1778 \def\maketitle{%
1779 \newpage\null
1780 \vskip 2em
1781 \begin{center}%
1782 \let\footnote\thanks
1783 {\LARGE \@title \par}%
1784 \ifx\bxjs@subtitle\undefined\else

```

```

1785     \vskip3\p@?
1786     {\normalsize \bxjs@subtitle\par}
1787     \fi
1788     \vskip 1.5em
1789     {\large
1790     \lineskip .5em
1791     \begin{tabular}[t]{c}%
1792     \@author
1793     \end{tabular}\par}%
1794     \vskip 1em
1795     {\large \@date}%
1796     \end{center}%
1797     \par\vskip 1.5em
1798 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1799 }
1800 \fi
1801 %</article|book|report|slide>
1802 %<*jspf>
1803 \newcommand{\maketitle}{\par
1804 \begingroup
1805 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1806 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1807 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1808 \parindent 1\jsZw\noindent
1809 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1810 \twocolumn[\@maketitle]%
1811 \plainifnotempty
1812 \@thanks
1813 \endgroup
1814 \setcounter{footnote}{0}%
1815 \global\let\thanks\relax
1816 \global\let\maketitle\relax
1817 \global\let\@thanks\@empty
1818 \global\let\@author\@empty
1819 \global\let\@date\@empty
1820 % \global\let\@title\@empty % \@title は柱に使う
1821 \global\let\title\relax
1822 \global\let\author\relax
1823 \global\let\date\relax
1824 \global\let\and\relax
1825 \ifx\authors@mail\@undefined\else{%
1826 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1827 \footnotetext[0]{\itshape\authors@mail}%
1828 }\fi
1829 \global\let\authors@mail\@undefined}
1830 \def\@maketitle{%
1831 \newpage\null
1832 \vskip 6em % used to be 2em
1833 \begin{center}

```

```

1834 \let\footnote\thanks
1835 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1836 \lineskip .5em
1837 \ifx\@author\@undefined\else
1838 \vskip 1em
1839 \begin{tabular}[t]{c}%
1840 \@author
1841 \end{tabular}\par
1842 \fi
1843 \ifx\@etitle\@undefined\else
1844 \vskip 1em
1845 {\large \@etitle \par}%
1846 \fi
1847 \ifx\@eauthor\@undefined\else
1848 \vskip 1em
1849 \begin{tabular}[t]{c}%
1850 \@eauthor
1851 \end{tabular}\par
1852 \fi
1853 \vskip 1em
1854 \@date
1855 \end{center}
1856 \vskip 1.5em
1857 \centerline{\box\@abstractbox}
1858 \ifx\@keywords\@undefined\else
1859 \vskip 1.5em
1860 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1861 \fi
1862 \vskip 1.5em}
1863 %</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。

```

1864 \def\bxjs@label@sect#1{%
1865 \expandafter\ifx\csname label#1\endcsname\relax
1866 \csname the#1\endcsname
1867 \else \csname label#1\endcsname
1868 \fi}
1869 \def\@secntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1870 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1871 \def\@secapp{\presectionname}
1872 \def\@secpos{\postsectionname}
1873 \fi

```

`\labelsection` 節番号の出力書式。

```

1874 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1875 \def\labelsection{\@secapp\thesection\@secpos}
1876 \fi

```

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

```

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

```

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```

1877 \def\@startsection#1#2#3#4#5#6{%
1878   \if@noskipsec \leavevmode \fi
1879   \par
1880 % 見出し上の空きを \@tempskipa にセットする
1881   \@tempskipa #4\relax
1882 % \@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
1883   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1884 % 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
1885   \ifdim \@tempskipa <\z@

```

```

1886 \@tempkipa -\@tempkipa \@afterindentfalse
1887 \fi
1888 \if@nobreak
1889 % \everypar{\everyparhook}% これは間違い
1890 \everypar{}%
1891 \else
1892 \addpenalty\@secpenalty
1893 % 次の行は削除
1894 % \addvspace\@tempkipa
1895 % 次の \noindent まで追加
1896 \ifdim \@tempkipa >\z@
1897 \if@slide\else
1898 \null
1899 \vspace*{-\baselineskip}%
1900 \fi
1901 \vskip\@tempkipa
1902 \fi
1903 \fi
1904 \noindent
1905 % 追加終わり
1906 \@ifstar
1907 {\@ssect{#3}{#4}{#5}{#6}}%
1908 {\@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。

```

1909 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1910 \let\bxjs@if@ceph\@firstofone
1911 \else \let\bxjs@if@ceph\@gobble
1912 \fi

```

```

1913 \def\@sect#1#2#3#4#5#6[#7]#8{%
1914 \ifnum #2>\c@secnumdepth
1915 \let\@svsec\@empty
1916 \else
1917 \refstepcounter{#1}%
1918 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1919 \fi
1920 % 見出し後の空きを \@tempkipa にセット
1921 \@tempkipa #5\relax
1922 % 条件判断の順序を入れ替えました
1923 \ifdim \@tempkipa<\z@
1924 \def\@svsechd{%
1925 #6{\hskip #3\relax

```

```

1926     \@svsec #8}%
1927     \csname #1mark\endcsname{#7}%
1928     \addcontentsline{toc}{#1}{%
1929         \ifnum #2>\c@secnumdepth \else
1930             \protect\numberline{\bxjs@label@sect{#1}}%
1931         \fi
1932         #7}}% 目次にフルネームを載せるなら #8
1933 \else
1934     \begingroup
1935     \interlinepenalty \@M % 下から移動
1936     #6{%
1937         \@hangfrom{\hskip #3\relax\@svsec}%
1938 %     \interlinepenalty \@M % 上に移動
1939     #8\@@par}%
1940 \endgroup
1941     \csname #1mark\endcsname{#7}%
1942     \addcontentsline{toc}{#1}{%
1943         \ifnum #2>\c@secnumdepth \else
1944             \protect\numberline{\bxjs@label@sect{#1}}%
1945         \fi
1946         #7}}% 目次にフルネームを載せるならここは #8
1947 \fi
1948 \@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。

```

1949 \def\@xsect#1{%
1950 % 見出しの後ろの空きを \@tempskipa にセット
1951     \@tempskipa #1\relax
1952 % 条件判断の順序を変えました
1953     \ifdim \@tempskipa<\z@
1954         \@nbreakfalse
1955         \global\@noskipsectrue
1956     \everypar{%
1957         \if@noskipsec
1958             \global\@noskipsecfalse
1959             {\setbox\z@\lastbox}%
1960             \clubpenalty\@M
1961             \begingroup \@svsechd \endgroup
1962             \unskip
1963             \@tempskipa #1\relax
1964             \hskip -\@tempskipa
1965         \else
1966             \clubpenalty \@clubpenalty
1967             \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1968             \fi\bxjs@if@ceph\everyparhook}%

```



```

1969 \else
1970 \par \nobreak
1971 \vskip \@tempskipa
1972 \@afterheading
1973 \fi
1974 \if@slide
1975 {\vskip\if@twocolumn-5\jsc@mpt\else-6\jsc@mpt\fi
1976 \maybeblue\hrule height0\jsc@mpt depth1\jsc@mpt
1977 \vskip\if@twocolumn 4\jsc@mpt\else 7\jsc@mpt\fi\relax}%
1978 \fi
1979 \par % 2000-12-18
1980 \ignorespaces}
1981 \def\@ssect#1#2#3#4#5{%
1982 \@tempskipa #3\relax
1983 \ifdim \@tempskipa<\z@
1984 \def\@svsechd{#4{\hskip #1\relax #5}}%
1985 \else
1986 \begingroup
1987 #4{%
1988 \@hangfrom{\hskip #1}%
1989 \interlinepenalty \@M #5\@@par}%
1990 \endgroup
1991 \fi
1992 \@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 1993 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1994 % \newcommand*\sectionmark[1]{}
\paragraphmark 1995 % \newcommand*\subsectionmark[1]{}
\subparagraphmark 1996 % \newcommand*\subsubsectionmark[1]{}
1997 % \newcommand*\paragraphmark[1]{}
1998 % \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。

```

1999 %<!book&!report>\setcounter{secnumdepth}{3}
2000 %<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 2001 \newcounter{part}
\c@subsubsection 2002 %<book|report>\newcounter{chapter}
\c@paragraph 2003 %<book|report>\newcounter{section}[chapter]
\c@subparagraph 2004 %<!book&!report>\newcounter{section}

```

```

2005 \newcounter{subsection}[section]
2006 \newcounter{subsubsection}[subsection]
2007 \newcounter{paragraph}[subsubsection]
2008 \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。

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

```

2009 \renewcommand{\thepart}{\@Roman\c@part}
2010 %<!*book&!report>
2011 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2012 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2013 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2014 \else
2015 \renewcommand{\thesection}{\@arabic\c@section}
2016 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2017 \fi
2018 %</!*book&!report>
2019 %<*book|report>
2020 \renewcommand{\thechapter}{\@arabic\c@chapter}
2021 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2022 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2023 %</book|report>
2024 \renewcommand{\thesubsubsection}{%
2025   \thesubsection.\@arabic\c@subsubsection}
2026 \renewcommand{\theparagraph}{%
2027   \thesubsubsection.\@arabic\c@paragraph}
2028 \renewcommand{\thesubparagraph}{%
2029   \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。

2030 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2031 %<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` がどうかにかかわらず奇数ページまで繰るように修正することで、問題を解消しました。実は、`LATEX` の標準クラスでは1998年に修正されていた問題です（コミュニティ版 `pLATEX` の標準クラス 2017/03/05 も参照）。

```
2032 %<*book|report>
2033 \newcommand\frontmatter{%
2034   \pltx@cleartooddpage
2035   \@mainmatterfalse
2036   \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。

```
2037 \newcommand\mainmatter{%
2038   \pltx@cleartooddpage
2039   \@mainmattertrue
2040   \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。

```
2041 \newcommand\backmatter{%
2042   \if@openleft
2043     \cleardoublepage
2044   \else\if@openright
2045     \cleardoublepage
2046   \else
2047     \clearpage
2048   \fi\fi
2049   \@mainmatterfalse}
2050 %</book|report>
```

■部

`\part` 新しい部を始めます。

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

```
\secdef{星なし}{星あり}
```

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

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

まず `book` と `report` のクラス以外です。

```

2051 %<!*book&!report>
2052 \newcommand\part{%
2053   \if@noskipsec \leavevmode \fi
2054   \par
2055   \addvspace{4ex}%
2056   \if@english \afterindentfalse \else \afterindenttrue \fi
2057   \secdef\@part\@spart}
2058 %</!*book&!report>

```

book および report クラスの場合は、少し複雑です。

```

2059 %<*book|report>
2060 \newcommand\part{%
2061   \if@openleft
2062     \cleardoublepage
2063   \else\if@openright
2064     \cleardoublepage
2065   \else
2066     \clearpage
2067   \fi\fi
2068   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2069   \if@twocolumn
2070     \onecolumn
2071     \@restonecoltrue
2072   \else
2073     \@restonecolfalse
2074   \fi
2075   \null\vfil
2076   \secdef\@part\@spart}
2077 %</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。

```

2078 %<!*book&!report>
2079 \def\@part[#1]#2{%
2080   \ifnum \c@secnumdepth >\m@ne
2081     \refstepcounter{part}%
2082     \addcontentsline{toc}{part}{%
2083       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2084   \else
2085     \addcontentsline{toc}{part}{#1}%
2086   \fi
2087   \markboth{}{}%
2088   {\parindent\z@
2089     \raggedright
2090     \interlinepenalty \@M
2091     \normalfont
2092     \ifnum \c@secnumdepth >\m@ne
2093       \Large\headfont\prepartname\thepart\postpartname

```

```

2094     \par\nobreak
2095     \fi
2096     \huge \headfont #2%
2097     \markboth{}{}\par}%
2098     \nobreak
2099     \vskip 3ex
2100     \@afterheading}
2101 %</!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。

```

2102 %<*book|report>
2103 \def\@part[#1]#2{%
2104   \ifnum \c@secnumdepth >-2\relax
2105     \refstepcounter{part}%
2106     \addcontentsline{toc}{part}{%
2107       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2108   \else
2109     \addcontentsline{toc}{part}{#1}%
2110   \fi
2111   \markboth{}{}%
2112   {\centering
2113     \interlinepenalty \@M
2114     \normalfont
2115     \ifnum \c@secnumdepth >-2\relax
2116       \huge\headfont \prepartname\thepart\postpartname
2117       \par\vskip20\p@?
2118     \fi
2119     \Huge \headfont #2\par}%
2120   \@endpart}
2121 %</book|report>

```

\@spart 番号を付けない部です。

```

2122 %<*!book&!report>
2123 \def\@spart#1{%
2124   \parindent \z@ \raggedright
2125   \interlinepenalty \@M
2126   \normalfont
2127   \huge \headfont #1\par}%
2128   \nobreak
2129   \vskip 3ex
2130   \@afterheading}
2131 %</!book&!report>
2132 %<*book|report>
2133 \def\@spart#1{%
2134   \centering
2135   \interlinepenalty \@M
2136   \normalfont
2137   \Huge \headfont #1\par}%
2138   \@endpart}

```

2139 %</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。

2140 %<*book|report>

2141 `\def\@endpart{\vfil\newpage`

2142 `\if@twoside`

2143 `\if@openleft %% added (2017/02/24)`

2144 `\null\thispagestyle{empty}\newpage`

2145 `\else\if@openright %% added (2016/12/13)`

2146 `\null\thispagestyle{empty}\newpage`

2147 `\fi\fi %% added (2016/12/13, 2017/02/24)`

2148 `\fi`

2149 `\if@restonecol`

2150 `\twocolumn`

2151 `\fi}`

2152 %</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。

2153 %<*book|report>

2154 `\newcommand{\chapter}{%`

2155 `\if@openleft\cleardoublepage\else`

2156 `\if@openright\cleardoublepage\else\clearpage\fi\fi`

2157 `\plainifnotempty % 元: \thispagestyle{plain}`

2158 `\global\@topnum\z@`

2159 `\if@english \@afterindentfalse \else \@afterindenttrue \fi`

2160 `\secdef`

2161 `{\@omit@numberfalse\@chapter}%`

2162 `{\@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。

2163 `\def\@chapter[#1]#2{%`

2164 `\ifnum \c@secnumdepth >\m@ne`

2165 `\if@mainmatter`

2166 `\refstepcounter{chapter}%`

2167 `\typeout{\@chapapp\thechapter\@chappos}%`

2168 `\addcontentsline{toc}{chapter}%`

2169 `{\protect\numberline`

2170 `%{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%`

2171 `{\@chapapp\thechapter\@chappos}%`

```

2172     #1}%
2173     \else\addcontentsline{toc}{chapter}{#1}\fi
2174 \else
2175     \addcontentsline{toc}{chapter}{#1}%
2176 \fi
2177 \chaptermark{#1}%
2178 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
2179 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
2180 \if@twocolumn
2181     \@topnewpage[\@makechapterhead{#2}]%
2182 \else
2183     \@makechapterhead{#2}%
2184     \@afterheading
2185 \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。

```

2186 \def\@makechapterhead#1{%
2187     \vspace*{2\Cvs}% 欧文は 50pt
2188     {\parindent \z@ \raggedright \normalfont
2189     \ifnum \c@secnumdepth >\m@ne
2190         \if@mainmatter
2191             \huge\headfont \@chapapp\thechapter\@chappos
2192             \par\nobreak
2193             \vskip \Cvs % 欧文は 20pt
2194         \fi
2195     \fi
2196     \interlinepenalty\M
2197     \Huge \headfont #1\par\nobreak
2198     \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。

```

2199 \def\@schapter#1{%
2200     \chaptermark{#1}%
2201     \if@twocolumn
2202         \@topnewpage[\@makeschapterhead{#1}]%
2203     \else
2204         \@makeschapterhead{#1}\@afterheading
2205     \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2206 \def\@makeschapterhead#1{%
2207     \vspace*{2\Cvs}% 欧文は 50pt
2208     {\parindent \z@ \raggedright
2209     \normalfont
2210     \interlinepenalty\M
2211     \Huge \headfont #1\par\nobreak
2212     \vskip 3\Cvs}} % 欧文は 40pt
2213 %</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。

```
2214 \if@twocolumn
2215   \newcommand{\section}{%
2216 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2217   \@startsection{section}{1}{\z@}%
2218 %<kiyou>   {0.6\Cvs}{0.4\Cvs}%
2219 %<kiyou>   {\Cvs}{0.5\Cvs}%
2220 %   {\normalfont\large\headfont\@secapp}}
2221     {\normalfont\large\headfont\raggedright}}
2222 \else
2223   \newcommand{\section}{%
2224     \if@slide\clearpage\fi
2225     \@startsection{section}{1}{\z@}%
2226     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2227     {.5\Cvs \@plus.3\Cdp}% 後アキ
2228 %   {\normalfont\Large\headfont\@secapp}}
2229     {\normalfont\Large\headfont\raggedright}}
2230 \fi
```

`\subsection` 同上です。

```
2231 \if@twocolumn
2232   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2233     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2234     {\normalfont\normalsize\headfont}}
2235 \else
2236   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2237     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2238     {.5\Cvs \@plus.3\Cdp}% 後アキ
2239     {\normalfont\large\headfont}}
2240 \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)。

```
2241 \if@twocolumn
2242   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2243     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2244     {\normalfont\normalsize\headfont}}
2245 \else
2246   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2247     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2248     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2249     {\normalfont\normalsize\headfont}}
2250 \fi
```


`\paragraph` 見出しの後ろで改行されません。

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

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

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

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

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

```
2251 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2252 \let\bxjs@org@paragraph@mark\jsParagraphMark
2253 \if\bxjs@paragraph@mark\@empty
2254   \let\jsParagraphMark\@empty
2255 \else\if\bxjs@paragraph@mark\undefined\else
2256   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2257 \fi\fi
2258 \let\jsJaChar\@empty
2259 \if@twocolumn
2260   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2261     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2262     {\normalfont\normalsize\headfont}}
2263 %<jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
2264 \else
2265   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2266     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2267     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2268     {\normalfont\normalsize\headfont}}
2269 %<jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
2270 \fi
```

`\subparagraph` 見出しの後ろで改行されません。

```
2271 \if@twocolumn
2272   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2273     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2274     {\normalfont\normalsize\headfont}}
2275 \else
2276   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2277     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2278     {\normalfont\normalsize\headfont}}
2279 \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。

```
2280 \if@slide
2281   \setlength\leftmargini{1\jsZw}
2282 \else
2283   \if@twocolumn
2284     \setlength\leftmargini{2\jsZw}
2285   \else
2286     \setlength\leftmargini{3\jsZw}
2287   \fi
2288 \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 2289 \if@slide
\leftmarginv 2290   \setlength\leftmarginii {1\jsZw}
2291   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2292   \setlength\leftmarginiv {1\jsZw}
2293   \setlength\leftmarginv  {1\jsZw}
2294   \setlength\leftmarginvi {1\jsZw}
2295 \else
2296   \setlength\leftmarginii {2\jsZw}
2297   \setlength\leftmarginiii{2\jsZw}
2298   \setlength\leftmarginiv {2\jsZw}
2299   \setlength\leftmarginv  {1\jsZw}
2300   \setlength\leftmarginvi {1\jsZw}
2301 \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。

```
2302 \setlength \labelsep {0.5\jsZw} % .5em
2303 \setlength \labelwidth{\leftmargini}
2304 \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。

```
2305 \setlength\partopsep{z@} % {2\p@ \@plus 1\p@ \@minus 1\p@}
```

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

`\@endparpenalty`

`\@itempenalty`

```

2306 \@beginparpenalty -\@lowpenalty
2307 \@endparpenalty -\@lowpenalty
2308 \@itempenalty -\@lowpenalty

```

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

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

```

2309 \def\@listi{\leftmargin\leftmarginI
2310   \parsep \z@
2311   \topsep 0.5\baselineskip
2312   \itemsep \z@ \relax}
2313 \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）。

```
2314 \@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 2315 \def\@listii{\leftmargin\leftmarginii
\@listiv 2316   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2317   \topsep \z@
\@listv 2318   \parsep \z@
\@listvi 2319   \itemsep\parsep}
2320 \def\@listiii{\leftmargin\leftmarginiii
2321   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2322   \topsep \z@
2323   \parsep \z@
2324   \itemsep\parsep}
2325 \def\@listiv {\leftmargin\leftmarginiv
2326   \labelwidth\leftmarginiv
2327   \advance\labelwidth-\labelsep}
2328 \def\@listv {\leftmargin\leftmarginv
2329   \labelwidth\leftmarginv
2330   \advance\labelwidth-\labelsep}
2331 \def\@listvi {\leftmargin\leftmarginvi
2332   \labelwidth\leftmarginvi
2333   \advance\labelwidth-\labelsep}

```

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

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

れ算用数字, 小文字アルファベット, 小文字ローマ数字, 大文字アルファベットで番号を出力する命令です。

```
2334 \renewcommand{\theenumi}{\@arabic\c@enumi}
2335 \renewcommand{\theenumii}{\@alph\c@enumii}
2336 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2337 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

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

`\labelenumiv`

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

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

```
2338 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2339 \newcommand*{\jsInJaParen}[1]{%
2340   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2341 \newcommand{\labelenumi}{\theenumi.}
2342 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
2343 \newcommand{\labelenumiii}{\theenumiii.}
2344 \newcommand{\labelenumiv}{\theenumiv.}
```

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

```
\p@enumiv 2345 \renewcommand{\p@enumii}{\theenumi}
2346 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
2347 \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 2348 \newcommand\labelitemi{\textbullet}
\labelitemiii 2349 \newcommand\labelitemii{\normalfont\bfseries \textendash}
2350 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2351 \newcommand\labelitemiv{\textperiodcentered}
```

■description 環境

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

```
2352 \newenvironment{description}{%
2353   \list{}{%
```

```

2354 \labelwidth=\leftmargin
2355 \labelsep=1\jsZw
2356 \advance \labelwidth by -\labelsep
2357 \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。

```

2358 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}

```

■概要

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

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

`bxjsreport` クラスの `abstract` 環境は :

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

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

```

2359 %<*book|report>
2360 \newenvironment{chapterabstract}{%
2361 \begin{list}{}{%
2362 \listparindent=1\jsZw
2363 \itemindent=\listparindent
2364 \rightmargin=0pt
2365 \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2366 %</book|report>

```

“普通の” `abstract` 環境の定義。

```

2367 %<*article|report|slide>
2368 \newbox\@abstractbox
2369 \if@titlepage
2370 \newenvironment{abstract}{%
2371 \titlepage
2372 \null\vfil
2373 \@beginparpenalty\@lowpenalty
2374 \begin{center}%
2375 \headfont \abstractname
2376 \@endparpenalty\@M
2377 \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。

```
2378 \par}%
2379 {\par\vfil\null\endtitlepage}
2380 \else
2381 \newenvironment{abstract}{%
2382 \if@twocolumn
2383 \ifx\maketitle\relax
2384 \section*{\abstractname}%
2385 \else
2386 \global\setbox\@abstractbox\hbox\bgroup
2387 \begin{minipage}[b]{\textwidth}
2388 \small\parindent1\jsZw
2389 \begin{center}%
2390 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2391 \end{center}%
2392 \list{}{%
2393 \listparindent\parindent
2394 \itemindent \listparindent
2395 \rightmargin \leftmargin}%
2396 \item\relax
2397 \fi
2398 \else
2399 \small
2400 \begin{center}%
2401 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2402 \end{center}%
2403 \list{}{%
2404 \listparindent\parindent
2405 \itemindent \listparindent
2406 \rightmargin \leftmargin}%
2407 \item\relax
2408 \fi}{\if@twocolumn
2409 \ifx\maketitle\relax
2410 \else
2411 \endlist\end{minipage}\egroup
2412 \fi
2413 \else
2414 \endlist
2415 \fi}
2416 \fi
2417 %</article|report|slide>
2418 %<*jspf>
2419 \newbox\@abstractbox
2420 \newenvironment{abstract}{%
2421 \global\setbox\@abstractbox\hbox\bgroup
2422 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2423 \small
2424 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%

```

```
2425 {\end{minipage}\egroup}
2426 %</jspf>
```

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

```
2427 %<*book|report>
2428 \ifbxjs@force@chapterabstract
2429 \let\abstract\chapterabstract
2430 \let\endabstract\endchapterabstract
2431 \fi
2432 %</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。

```
2433 %<*jspf>
2434 %\newbox\@keywordsbox
2435 %\newenvironment{keywords}{%
2436 % \global\setbox\@keywordsbox\hbox\bgroup
2437 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2438 % \small\parindent0\jsZw}%
2439 % {\end{minipage}\egroup}
2440 %</jspf>
```

■verse 環境

verse 詩のための verse 環境です。

```
2441 \newenvironment{verse}{%
2442 \let \\\=\@centercr
2443 \list{}{%
2444 \itemsep \z@
2445 \itemindent -2\jsZw % 元: -1.5em
2446 \listparindent\itemindent
2447 \rightmargin \z@
2448 \advance\leftmargin 2\jsZw}% 元: 1.5em
2449 \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。

```
2450 \newenvironment{quotation}{%
2451 \list{}{%
2452 \listparindent\parindent
2453 \itemindent\listparindent
2454 \rightmargin \z@}%
2455 \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。

```
2456 \newenvironment{quote}%
2457   {\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。

```
2458 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2459   \item[\hskip \labelsep{\headfont #1\ #2}]}
2460 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2461   \item[\hskip \labelsep{\headfont #1\ #2 (#3) ]}}
```

titlepage タイトルを独立のページに出力するのに使われます。

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

```
2462 \newenvironment{titlepage}{%
2463   %<book> \pltx@cleartooddpage %% 2017-02-24
2464   \if@twocolumn
2465     \@restonecoltrue\onecolumn
2466   \else
2467     \@restonecolfalse\newpage
2468   \fi
2469   \thispagestyle{empty}%
2470   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2471   24
2472   }%
2473   {\if@restonecol\twocolumn \else \newpage \fi
2474   \if@twoside\else
2475     \setcounter{page}\@ne
2476   \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2476 %<!*book&!report>
2477 \newcommand{\appendix}{\par
2478   \setcounter{section}{0}%
2479   \setcounter{subsection}{0}%
2480   \ifnum\bxjs@label@section=\bxjs@label@section@compat
2481     \gdef\presectionname{\appendixname}%
2482     \gdef\postsectionname{}}%
2483 % \gdef\thesection{\@Alph@c@section}% [2003-03-02]
2484 \gdef\thesection{\presectionname\@Alph@c@section\postsectionname}%
2485 \gdef\thesubsection{\@Alph@c@section.\@arabic@c@subsection}%
2486 \else
2487 \gdef\@secapp{\appendixname}%
2488 \gdef\@secpos{}}%
2489 \gdef\thesection{\@Alph@c@section}%
2490 \fi}
2491 %</!*book&!report>
2492 %<*book|report>
2493 \newcommand{\appendix}{\par
2494   \setcounter{chapter}{0}%
2495   \setcounter{section}{0}%
2496   \gdef\@chapapp{\appendixname}%
2497   \gdef\@chappos{}}%
2498 \gdef\thechapter{\@Alph@c@chapter}}
2499 %</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。

```
2500 \setlength\arraycolsep{5\p@?}
```

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

```
2501 \setlength\tabcolsep{6\p@?}
```

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

```
2502 \setlength\arrayrulewidth{.4\p@}
```

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

```
2503 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2504 \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。

```
2505 \skip\@mpfootins = \skip\footins
```

■framebox 環境

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

`\fboxrule` `\fbox`, `\framebox` の罫線の幅です。

```
2506 \setlength\fboxsep{3\p@?}
2507 \setlength\fboxrule{.4\p@}
```

■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

```
2508 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2509 %<*book|report>
2510 \@addtoreset{equation}{chapter}
2511 \renewcommand\theequation
2512   {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
2513 %</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。

```
2514 % \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。

```
2515 % \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。

```
2516 % \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。

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

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

```
\ext@figure 2526 \def\fps@figure{tbp}
\fnm@figure 2527 \def\ftype@figure{1}
2528 \def\ext@figure{lof}
2529 \def\fnm@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

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

■table 環境

`\c@table` 表番号カウンタと表番号を出力するコマンドです。アスキー版では `\thechapter.` が `\thetable \thechapter{}` になっていますが、ここではオリジナルのままにしています。

```
2536 %<!*book&!report>
2537 \newcounter{table}
2538 \renewcommand\thetable{\@arabic\c@table}
2539 %</!book&!report>
2540 %<*book|report>
2541 \newcounter{table}[chapter]
2542 \renewcommand \thetable
2543     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2544 %</book|report>
```

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

```
\ext@table 2545 \def\fps@table{tbp}
2546 \def\ftype@table{2}
\fnunm@table 2547 \def\ext@table{lot}
2548 \def\fnunm@table{\tablename\nobreak\thetable}
```

`table *` は段抜きのフロートです。

```
table* 2549 \newenvironment{table}%
2550         {\@float{table}}%
2551         {\end@float}
2552 \newenvironment{table*}%
2553         {\@dblfloat{table}}%
2554         {\end@dblfloat}
```

8.6 キャプション

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

`\abovecaptionskip` それぞれキャプションの前後に挿入されるスペースです。`\belowcaptionskip` が 0 になっ
`\belowcaptionskip` ていたもので、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。

```
2555 \newlength\abovecaptionskip
2556 \newlength\belowcaptionskip
2557 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2558 \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。

```
2559 %<*\jpsf>
2560 % \long\def\@makecaption#1#2{{\small
2561 %   \advance\leftskip10\jsc@mmm
2562 %   \advance\rightskip10\jsc@mmm
2563 %   \vskip\abovecaptionskip
2564 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2565 %   \ifdim \wd\@tempboxa >\hsize
2566 %     #1\hskip1\jsZw\relax #2\par
2567 %   \else
2568 %     \global \@minipagefalse
2569 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2570 %   \fi
2571 %   \vskip\belowcaptionskip}}
2572 \long\def\@makecaption#1#2{{\small
2573   \advance\leftskip .0628\linewidth
```

```

2574 \advance\rightskip .0628\linewidth
2575 \vskip\abovecaptionskip
2576 \sbox\@tempboxa{#1\zwspace#2}%
2577 \ifdim \wd\@tempboxa <\hsize \centering \fi
2578 #1\zwspace#2\par
2579 \vskip\belowcaptionskip}}
2580 %</!jspf>
2581 %<*jspf>
2582 \long\def\@makecaption#1#2{%
2583 \vskip\abovecaptionskip
2584 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2585 \ifdim \wd\@tempboxa >\hsize
2586 {\small\sffamily
2587 \list{#1}{%
2588 \renewcommand{\makelabel}[1]{##1\hfil}
2589 \itemsep \z@
2590 \itemindent \z@
2591 \labelsep \z@
2592 \labelwidth 11\jsc@mmm
2593 \listparindent\z@
2594 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2595 \else
2596 \global \@minipagefalse
2597 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2598 \fi
2599 \vskip\belowcaptionskip}
2600 %</jspf>

```

9 フォントコマンド

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

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

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

```

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

```

```

2601 \newif\if@jsc@warnoldfontcmd
2602 \@jsc@warnoldfontcmdtrue

```

```

2603 \newif\if@jsc@warnoldfontcmdexception
2604 \@jsc@warnoldfontcmdexceptionfalse

```

\jsc@DeclareOldFontCommand

```

2605 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2606   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2607   \DeclareOldFontCommand{#1}{%
2608     \bxjs@oldfontcmd{#1}#2%
2609   }{%
2610     \bxjs@oldfontcmd{#1}#3%
2611   }%
2612 }
2613 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2614   \ClassInfo\bxjs@clsname
2615   {Old font command '\string#1' is used!!\MessageBreak
2616     The first occurrence is}%
2617 }

```

\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。

```

2618 \newcommand*{\allowoldfontcommands}{%
2619   \@jsc@warnoldfontcmdexceptiontrue}
2620 \newcommand*{\disallowoldfontcommands}{%
2621   \@jsc@warnoldfontcmdexceptionfalse}

2622 \let\bxjs@oldfontcmd@list\@empty
2623 \def\bxjs@oldfontcmd#1{%
2624   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2625 \def\bxjs@oldfontcmd@a#1#2{%
2626   \if@jsc@warnoldfontcmdexception\else
2627     \global\@jsc@warnoldfontcmdfalse
2628     \ifx#1\relax
2629       \global\let#1=t%
2630       \jsc@warnoldfontcmd{#2}%
2631     \fi
2632   \fi}
2633 \def\bxjs@warnoldfontcmd@final{%
2634 % \par
2635   \global\let\bxjs@warnoldfontcmd@final\@empty
2636   \let\@tempa\@empty
2637   \def\do##1{%
2638     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2639       \edef\@tempa{\@tempa \space\string##1}\fi}
2640   \bxjs@oldfontcmd@list
2641   \ifx\@tempa\@empty\else
2642     \ClassWarningNoLine\bxjs@clsname
2643     {Some old font commands were used in text:\MessageBreak

```

```

2644     \space\@tempa\MessageBreak
2645     You should note, that since 1994 LaTeX2e provides a\MessageBreak
2646     new font selection scheme called NFSS2 with several\MessageBreak
2647     new, combinable font commands. The
2648     class provides\MessageBreak
2649     the old font commands only for compatibility}
2650 \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。

```

2651 \def\bxjs@warnoldfontcmd@kick@final{%
2652   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2653 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2654 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2655 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 2656 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\tt 2657 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2658 \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。

```

2659 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2660 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2661 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2662 \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 2663 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2664 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}

```

10 相互参照

10.1 目次の類

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

```

\contentsline{section}{タイトル}{ページ}

```

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

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

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

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

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

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

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

という書式です。

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

インデント 左側の字下げ量です。

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

`\@pnumwidth` ページ番号の入る箱の幅です。

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

`\@dotsep` 点の間隔です (単位 `mu`)。

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

```
2665 \newcommand\@pnumwidth{1.55em}
2666 \newcommand\@tocrmarg{2.55em}
2667 \newcommand\@dotsep{4.5}
2668 %<!book&!report>\setcounter{tocdepth}{2}
2669 %<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)

```
2670 \newdimen\jsc@tocl@width
2671 \newcommand{\tableofcontents}{%
2672 %<*book|report>
2673 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2674 \settowidth\@tempdima{\headfont\appendixname}%
```



```

2675 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2676 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2677 \iftwocolumn
2678 \@restonecoltrue\onecolumn
2679 \else
2680 \@restonecolfalse
2681 \fi
2682 \chapter*{\contentsname}%
2683 \@mkboth{\contentsname}{}%
2684 %</book|report>
2685 %<!*book&!report>
2686 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2687 \settowidth\@tempdima{\headfont\appendixname}%
2688 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2689 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2690 \section*{\contentsname}%
2691 \@mkboth{\contentsname}{\contentsname}%
2692 %</!book&!report>
2693 \@starttoc{toc}%
2694 %<book|report> \if@restonecol\twocolumn\fi
2695 }

```

\l@part 部の目次です。

```

2696 \newcommand*{\l@part}[2]{%
2697 \ifnum \c@tocdepth >-2\relax
2698 %<!book&!report> \addpenalty\@secpenalty
2699 %<book|report> \addpenalty{-\@highpenalty}%
2700 \addvspace{2.25em \@plus\p@?}%
2701 \begingroup
2702 \parindent \z@
2703 % \@pnumwidth should be \@tocrmarg
2704 % \rightskip \@pnumwidth
2705 \rightskip \@tocrmarg
2706 \parfillskip -\rightskip
2707 {\leavevmode
2708 \large \headfont
2709 \setlength\@lnumwidth{4\jsZw}%
2710 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2711 \nobreak
2712 %<book|report> \global\@nobreaktrue
2713 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2714 \endgroup
2715 \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)

```

2716 %<!*book|report>
2717 \newcommand*{\l@chapter}[2]{%

```

```

2718 \ifnum \c@tocdepth >\m@ne
2719 \addpenalty{-\@highpenalty}%
2720 \addvspace{1.0em \@plus\p@?}
2721 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2722 \begingroup
2723 \parindent\z@
2724 % \rightskip\@pnumwidth
2725 \rightskip\@tocrmarg
2726 \parfillskip-\rightskip
2727 \leavevmode\headfont
2728 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2729 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2730 \advance\leftskip\@lnumwidth \hskip-\leftskip
2731 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2732 \penalty\@highpenalty
2733 \endgroup
2734 \fi}
2735 %</book|report>

```

\l@section 節の目次です。

```

2736 %<!*book&!report>
2737 \newcommand*{\l@section}[2]{%
2738 \ifnum \c@tocdepth >\z@
2739 \addpenalty{\@secpenalty}%
2740 \addvspace{1.0em \@plus\p@?}%
2741 \begingroup
2742 \parindent\z@
2743 % \rightskip\@pnumwidth
2744 \rightskip\@tocrmarg
2745 \parfillskip-\rightskip
2746 \leavevmode\headfont
2747 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2748 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2749 \advance\leftskip\@lnumwidth \hskip-\leftskip
2750 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2751 \endgroup
2752 \fi}
2753 %</!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。

```

2754 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

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

\l@subsubsection しれません。

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

```

\l@subparagraph 2755 %<!*book&!report>
2756 % \newcommand*{\l@subsection}{\@dottedtocline{2}{1.5em}{2.3em}}

```

```

2757 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2758 % \newcommand*\l@paragraph}{\@dottedtocline{4}{7.0em}{4.1em}}
2759 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{10em}{5em}}
2760 %
2761 % \newcommand*\l@subsection}{\@dottedtocline{2}{1zw}{3zw}}
2762 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2763 % \newcommand*\l@paragraph}{\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2764 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2765 %
2766 \newcommand*\l@subsection}{%
2767     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2768     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2769 \newcommand*\l@subsubsection}{%
2770     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2771     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2772 \newcommand*\l@paragraph}{%
2773     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2774     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2775 \newcommand*\l@subparagraph}{%
2776     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2777     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2778 %</!book&!report>
2779 %<*book|report>
2780 % \newcommand*\l@subsection}{\@dottedtocline{2}{3.8em}{3.2em}}
2781 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2782 % \newcommand*\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
2783 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
2784 \newcommand*\l@section}{%
2785     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2786     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2787 \newcommand*\l@subsection}{%
2788     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2789     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2790 \newcommand*\l@subsubsection}{%
2791     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2792     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2793 \newcommand*\l@paragraph}{%
2794     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2795     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2796 \newcommand*\l@subparagraph}{%
2797     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2798     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2799 %</book|report>

```

`\numberline` 欧文版 L^AT_EX では `\numberline{...}` は幅 `\@tempdima` の箱に左詰めで出力する命令で
`\@lnumwidth` 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。

```
2800 \newdimen\@lnumwidth
2801 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}
```

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

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

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

とします。

```
2802 \def\jsTocLine{\leaders\hbox{%
2803   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2804 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2805   \vskip \z@ \@plus.2\p@?
2806   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2807     \parindent #2\relax\@afterindenttrue
2808     \interlinepenalty\@M
2809     \leavevmode
2810     \@lnumwidth #3\relax
2811     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2812     {#4}\nobreak
2813     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2814       \hfil\normalfont \normalcolor #5}\par}\fi}
```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```
2815 \newcommand{\listoffigures}{%
2816 %<*book|report>
2817   \if@twocolumn\@restonecoltrue\onecolumn
2818   \else\@restonecolfalse\fi
2819   \chapter*{\listfigurename}%
2820   \@mkboth{\listfigurename}{}%
2821 %</book|report>
2822 %<!*book&!report>
2823   \section*{\listfigurename}%
2824   \@mkboth{\listfigurename}{\listfigurename}%
2825 %</!*book&!report>
2826   \@starttoc{lof}%
2827 %<book|report> \if@restonecol\twocolumn\fi
2828 }
```

`\l@figure` 図目次の項目を出力します。

```
2829 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```

2830 \newcommand{\listoftables}{%
2831 %<*book|report>
2832 \if@twocolumn\@restonecoltrue\onecolumn
2833 \else\@restonecolfalse\fi
2834 \chapter*{\listtablename}%
2835 \@mkboth{\listtablename}{}%
2836 %</book|report>
2837 %<!*book&!report>
2838 \section*{\listtablename}%
2839 \@mkboth{\listtablename}{\listtablename}%
2840 %<!/book&!report>
2841 \@starttoc{lot}%
2842 %<book|report> \if@restonecol\twocolumn\fi
2843 }

```

`\l@table` 表目次は図目次と同じです。

```
2844 \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。

```

2845 \newdimen\bibindent
2846 \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。

```

2847 \newenvironment{thebibliography}[1]{%
2848 \@jsc@warnoldfontcmdexceptiontrue
2849 \global\let\presectionname\relax
2850 \global\let\postsectionname\relax
2851 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2852 %<*kiyou>
2853 \vspace{1.5\baselineskip}
2854 \subsection*{\refname}\@mkboth{\refname}{\refname}%
2855 \vspace{0.5\baselineskip}
2856 %</kiyou>
2857 %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2858 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2859 \list{\@biblabel{\@arabic{c@enumiv}}}%
2860 \settotalwidth\labelwidth{\@biblabel{#1}}%
2861 \leftmargin\labelwidth
2862 \advance\leftmargin\labelsep
2863 \@openbib@code
2864 \usecounter{enumiv}%
2865 \let\p@enumiv\@empty

```

```

2866      \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2867 %<kiyou> \small
2868 \sloppy
2869 \clubpenalty4000
2870 \@clubpenalty\clubpenalty
2871 \widowpenalty4000%
2872 \sfcode`.\@m}
2873 {\def\@noitemerr
2874   {\@latex@warning{Empty `thebibliography' environment}}}%
2875 \endlist}

```

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

```
2876 \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。

```
2877 \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。

```
2878 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

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

```

2879 % \def\@citex[#1]#2{\leavevmode
2880 %   \let\@citea\@empty
2881 %   \@cite{\@for\@citeb:=#2\do
2882 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2883 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2884 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2885 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2886 %     \G@refundefinedtrue
2887 %     \@latex@warning
2888 %       {Citation `@\@citeb' on page \thepage \space undefined}}%
2889 %     {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2890 % \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。

```

2891 % \DeclareRobustCommand\cite{\unskip
2892 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2893 %   \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2894 %     , \jsInhibitGlue\ #2\fi) }}$}

```

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)。

```
2895 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2896     \if@twocolumn
2897         \onecolumn\@restonecolfalse
2898     \else
2899         \clearpage\@restonecoltrue
2900     \fi
2901     \columnseprule.4pt \columnsep 2\jsw
2902     \ifx\multicols\@undefined
2903 %<book|report>         \twocolumn[\@makeschapterhead{\indexname}]%
2904 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}]%
2905 %<!book&!report>     \def\presectionname{}\def\postsectionname{}%
2906 %<!book&!report>     \twocolumn[\section*{\indexname}]%
2907     \else
2908         \ifdim\textwidth<\fullwidth
2909             \setlength{\evensidemargin}{\oddsidemargin}
2910             \setlength{\textwidth}{\fullwidth}
2911             \setlength{\linewidth}{\fullwidth}
2912 %<book|report>         \begin{multicols}{3}[\chapter*{\indexname}]%
2913 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}]%
2914 %<!book&!report>     \def\presectionname{}\def\postsectionname{}%
2915 %<!book&!report>     \begin{multicols}{3}[\section*{\indexname}]%
2916     \else
2917 %<book|report>         \begin{multicols}{2}[\chapter*{\indexname}]%
2918 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}]%
2919 %<!book&!report>     \def\presectionname{}\def\postsectionname{}%
2920 %<!book&!report>     \begin{multicols}{2}[\section*{\indexname}]%
2921     \fi
2922     \fi
2923 %<book|report>     \@mkboth{\indexname}{}%
2924 %<!book&!report>     \@mkboth{\indexname}{\indexname}%
2925     \plainifnotempty % \thispagestyle{plain}
2926     \parindent\z@
2927     \parskip\z@ \@plus .3\p@\relax
2928     \let\item\@idxitem
2929     \raggedright
2930     \footnotesize\narrowbaselines
2931 }{
2932     \ifx\multicols\@undefined
2933         \if@restonecol\onecolumn\fi
2934     \else
2935         \end{multicols}
2936     \fi
2937     \clearpage
```

2938 }

`\@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` 2939 `\newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt`

`\subsubitem` 2940 `\newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt`

2941 `\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。

2942 `\newcommand{\indexspace}{\par \vskip 10\p@? \@plus5\p@? \@minus3\p@?\relax}`

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

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

2943 `\newcommand\seename{\if@english see\else →\fi}`

2944 `\newcommand\alsoname{\if@english see also\else →\fi}`

10.4 脚注

`\footnote` 和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため、`\footnotemark` `\inhibitglue` を入れることにします。pL^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` の代わりに `\jsInhibitGlue` を使う。

2945 `\ifx\pltx@foot@penalty\@undefined`

2946 `\let\footnotes@ve=\footnote`

2947 `\def\footnote{\jsInhibitGlue\footnotes@ve}`

2948 `\let\footnotemarks@ve=\footnotemark`

2949 `\def\footnotemark{\jsInhibitGlue\footnotemarks@ve}`

2950 `\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 `newtxtext` や `newpctext` 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 `defaultsup` オプションを付けてください (qa:57284, qa:57287)。

```
2951 \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。

```
2952 % \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。

```
2953 \renewcommand{\footnoterule}{%
2954   \kern-2.6\p@? \kern-.4\p@
2955   \hrule width .4\columnwidth
2956   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2957 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

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

```
2958 \long\def\@footnotetext{%
2959   \insert\footins\bgroup
2960     \normalfont\footnotesize
2961     \interlinepenalty\interfootnotelinepenalty
2962     \splittopskip\footnotesep
2963     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2964     \hsize\columnwidth \@parboxrestore
2965     \protected@edef\@currentlabel{%
2966       \csname p@footnote\endcsname\@thefnmark
2967     }%
2968     \color@begingroup
2969     \@makefnmark%
2970     \rule\z@\footnotesep\ignorespaces}%
2971     \futurelet\jsc@next\jsc@fo@t}
2972 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@fo@t
2973               \else \let\jsc@next\jsc@fo@t\fi \jsc@next}
```

```

2974 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
2975 \def\jsc@f@t#1{#1\jsc@@foot}
2976 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
2977 \ifx\pltx@foot@penalty\@undefined\else
2978 \ifhmode\null\fi
2979 \ifnum\pltx@foot@penalty=\z@\else
2980 \penalty\pltx@foot@penalty
2981 \pltx@foot@penalty\z@
2982 \fi
2983 \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。

```

2984 \newcommand\@makefntext[1]{%
2985 \advance\leftskip 3\jsZw
2986 \parindent 1\jsZw
2987 \noindent
2988 \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。

```

2989 % \def\@xfootnotenext[#1]{%
2990 % \begingroup
2991 % \ifnum#1>\z@
2992 % \csname c@\@mpfn\endcsname #1\relax
2993 % \unrestored@protected@xdef\@thefnmark{\thempfn}%
2994 % \else
2995 % \unrestored@protected@xdef\@thefnmark{}%
2996 % \fi
2997 % \endgroup
2998 % \@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。

```
2999 \let\jsInhibitGlueAtParTop\@empty
```

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

```
3000 \def\everyparhook{\jsInhibitGlueAtParTop}
3001 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
3002 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3003 \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。間違っって別の箇所で使う危険性が高いので、専用の命令 `\jsc@ig@temp` に置き換えました (Issue #54)。

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

```
3004 \ifx j\jsEngine
3005 \def\@inhibitglue{%
3006   \futurelet\@let@token\@@inhibitglue}
3007 \begingroup
3008 \let\GDEF=\gdef
3009 \let\CATCODE=\catcode
3010 \let\ENDGROUP=\endgroup
3011 \CATCODE`k=12
3012 \CATCODE`a=12
3013 \CATCODE`n=12
3014 \CATCODE`j=12
3015 \CATCODE`i=12
3016 \CATCODE`c=12
3017 \CATCODE`h=12
3018 \CATCODE`r=12
3019 \CATCODE`t=12
3020 \CATCODE`e=12
3021 \GDEF\KANJI@CHARACTER{kanji character }
3022 \ENDGROUP
3023 \def\@@inhibitglue{%
3024   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3025   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
3026   \def\jsc@ig@temp{#1}%
3027   \ifx\jsc@ig@temp\@empty
3028     \ifnum\the\inhibitxspcode`#2=2\relax
```

```

3029     \inhibitglue
3030     \fi
3031 \fi}
3032 \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。

```

3033 \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。

```

3034 \def\@doendpef%
3035     \@endpetrue
3036     \def\par{%
3037         \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3038     \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3039 \def\@setminipagef%
3040     \@minipagetrue
3041     \everypar{\@minipagefalse\everypar{\everyparhook}}%
3042 }

```

`\item` 命令の直後です。

```

3043 \def\@item[#1]{%
3044     \if@noparitem
3045         \@donoparitem
3046     \else
3047         \if@inlabel
3048             \indent \par
3049         \fi
3050         \ifhmode
3051             \unskip\unskip \par
3052         \fi
3053         \if@newlist
3054             \if@nobreak
3055                 \@nbitem
3056             \else
3057                 \addpenalty\@beginparpenalty
3058                 \addvspace\@topsep
3059                 \addvspace{-\parskip}%
3060             \fi
3061         \else
3062             \addpenalty\@itempenalty

```

```

3063     \addvspace\itemsep
3064     \fi
3065     \global\@inlabeltrue
3066     \fi
3067     \everypar{%
3068         \@minipagefalse
3069         \global\@newlistfalse
3070         \if@inlabel
3071             \global\@inlabelfalse
3072             {\setbox\z@\lastbox
3073              \ifvoid\z@
3074               \kern-\itemindent
3075              \fi}%
3076         \box\@labels
3077         \penalty\z@
3078     \fi
3079     \if@nobreak
3080         \@nobreakfalse
3081         \clubpenalty \@M
3082     \else
3083         \clubpenalty \@clubpenalty
3084         \everypar{\everyparhook}%
3085     \fi
3086     \everyparhook}%
3087     \if@noitemarg
3088         \@noitemargfalse
3089     \if@nmbrlist
3090         \refstepcounter\@listctr
3091     \fi
3092     \fi
3093     \sbox\@tempboxa{\makelabel{#1}}%
3094     \global\setbox\@labels\hbox{%
3095         \unhbox\@labels
3096         \hskip \itemindent
3097         \hskip -\labelwidth
3098         \hskip -\labelsep
3099         \ifdim \wd\@tempboxa >\labelwidth
3100             \box\@tempboxa
3101         \else
3102             \hbox to\labelwidth {\unhbox\@tempboxa}%
3103         \fi
3104         \hskip \labelsep}%
3105     \ignorespaces}

3106 \def\@afterheading{%
3107     \@nobreaktrue
3108     \everypar{%

```

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。

```

3109 \if@nobreak
3110 \@nobreakfalse
3111 \clubpenalty \@M
3112 \if@afterindent \else
3113   {\setbox\z@\lastbox}%
3114 \fi
3115 \else
3116 \clubpenalty \@clubpenalty
3117 \everypar{\everyparhook}%
3118 \fi\everyparhook}}

```

「\everyparhook 用のパッチ処理」はここまで。

```
3119 \fi
```

\@gnewline についてはちょっと複雑な心境です。もともとの pL^AT_EX 2_ε 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。

```

3120 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3121 \AtEndOfPackage{%
3122 \def\@gnewline #1{%
3123   \ifvmode
3124     \@nolnerr
3125   \else
3126     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3127     \jsInhibitGlue \ignorespaces
3128   \fi}
3129 }
3130 \fi

```

12 いろいろなロゴ

L^AT_EX 関連のロゴを作り直します。

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

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

した。ただし JS クラスと異なり、既定では読み込まない。

※`\小`、`\上小` の制御綴は定義しない。

```
3131 \if@jslogo
3132   \IfFileExists{jslogo.sty}{%
3133     \RequirePackage{jslogo}%
3134   }{%
3135     \ClassWarningNoLine{bxjs@clsname
3136       {The package 'jslogo' is not installed.\MessageBreak
3137         It is included in the recent release of\MessageBreak
3138         the 'jsclasses' bundle}
3139   }
3140 \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` `\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: しっぽ愛好家さん)。

[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。

```
3141 %\let\ltx@ifnextchar\@ifnextchar
3142 %\def\ProvidesFile#1{%
3143 %   \begingroup
3144 %     \catcode\ 10 %
3145 %     \ifnum \endlinechar<256 %
3146 %       \ifnum \endlinechar>\m@ne
3147 %         \catcode\endlinechar 10 %
3148 %       \fi
3149 %     \fi
3150 %     \@makeoother\/%
3151 %     \@makeoother\&%
3152 %     \ltx@ifnextchar [{\@providesfile{#1}}{\@providesfile{#1} []}]}
```

14 初期設定

■いろいろな語

```
\prepartname
\postpartname 3153 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3154 \newcommand{\postpartname}{\if@english\else 部\fi}
\postchaptername
\presectionname
\postsectionname
```

```

3155 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
3156 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
3157 \newcommand{\presectionname}{}% 第
3158 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3159 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3160 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3161 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname

\bibName 3162 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3163 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3164 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 3165 %<!jspf>\newcommand{\figurename}{\if@english Fig.-\else 図\fi}
3166 %<jspf>\newcommand{\figurename}{Fig.-}
3167 %<!jspf>\newcommand{\tablename}{\if@english Table-\else 表\fi}
3168 %<jspf>\newcommand{\tablename}{Table~}

\appendixname

\abstractname 3169 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3170 \newcommand{\appendixname}{\if@english \else 付録\fi}
3171 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

環境変数 SOURCE_DATE_EPOCH / FORCE_SOURCE_DATE が設定されている場合は“今日”が過去・未来の日付になる可能性がある。BXJS クラスでは、和暦の扱いは bxwareki パッケージに任せることを原則とする。ただし、bxwareki が使えない場合は自前で処理する。

```

3172 \onlypreamble\bxjs@decl@Seireki@cmds
3173 \@tempwafalse
3174 \if p\jsEngine \@tempwattrue \fi
3175 \if n\jsEngine \@tempwattrue \fi
3176 \if@tempswa \expandafter\@firstoftwo
3177 \else \expandafter\@secondoftwo
3178 \fi
3179 {%
3180 % 欧文 8bitTeX の場合
3181 \newif\ifjsSeireki \jsSeirekitrue
3182 \def\bxjs@decl@Seireki@cmds{%
3183 \def\西暦{\jsSeirekitrue}%
3184 \def\和暦{\jsSeirekifalse}}
3185 \def\Seireki{\jsSeirekitrue}

```



```

3186 \def\Wareki{\jsSeirekifalse}
3187 \def\bxjs@if@use@seireki{%
3188   \ifjsSeireki \expandafter\@firstoftwo
3189   \else \expandafter\@secondoftwo \fi}
3190 }{%
3191 \newif\if 西暦 \西暦 true
3192 \def\bxjs@decl@Seireki@cmds{%
3193   \def\西暦{\西暦 true}%
3194   \def\和暦{\西暦 false}}
3195 \def\Seireki{\西暦 true}
3196 \def\Wareki{\西暦 false}
3197 \def\bxjs@if@use@seireki{%
3198   \if 西暦 \expandafter\@firstoftwo
3199   \else \expandafter\@secondoftwo \fi}
3200 }
3201 \bxjs@decl@Seireki@cmds
3202 \let\bxjs@unxp\@firstofone \let\bxjs@onxp\@firstofone
3203 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded
3204   \def\bxjs@onxp{\unexpanded\expandafter}}
3205 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3206   \def\bxjs@iai{\noexpand~}
3207 \else \def\bxjs@iai{}
3208 \fi

```

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

```

3209 \newif\ifbxjs@bxwareki@avail
3210 \IfFileExists{bxwareki.sty}{%
3211   \RequirePackage{bxwareki} []%
3212   \bxjs@bxwareki@availtrue}{%

```

\jayear 和暦における年の表記（元号 + 年数）。

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

```

3213 \ifbxjs@bxwareki@avail
3214   \let\jayear\warekiyear
3215   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3216     \newcount\heisei \heisei=\value{warekiyear}
3217   \fi

   bxwareki が使えない場合のフォールバック処理。平成のみに対応する。
3218 \else
3219   \@tempcnta=\the\year\two@digits\month\two@digits\day\relax
3220   \ifnum\@tempcnta<19890108 \else \ifnum\@tempcnta<20190501
3221     \newcount\heisei \heisei\year \advance\heisei-1988\relax
3222   \fi\fi
3223   \ifnum \ifx\heisei\undefined\z@\else\heisei\fi >\@ne
3224     \edef\jayear{\bxjs@unxp{平成}\bxjs@iai \the\heisei \bxjs@iai}
3225   \else \edef\jayear{\the\year \bxjs@iai}
3226   \fi
3227 \fi

```

`\today` 英語、西暦、和暦で場合分けをする。

```
3228 \let\bxjs@next\relax
3229 \ifbxjs@bxwareki@avail
3230   \ifx\warekigengo@empty\else
3231     \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3232   \fi
3233 \else\ifnum \ifx\heisei@undefined\z@\else\heisei\fi >\@ne
3234   \def\bxjs@next{%
3235     \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
3236     \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3237     \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}}
3238 \fi\fi
3239 \edef\bxjs@today{%
3240   \if@english
3241     \ifcase\month\or
3242       January\or February\or March\or April\or May\or June\or
3243       July\or August\or September\or October\or November\or December\fi
3244     \space\number\day, \number\year
3245   \else
3246     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3247     \else \noexpand\bxjs@if@use@seireki
3248     \fi {%
3249       \number\year\bxjs@iai\bxjs@unxp{年}%
3250       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3251       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3252     }\bxjs@next}%
3253 \fi}
3254 \let\today\bxjs@today
```

texjporg 版の日本語用 Babel 定義ファイル (`japanese.1df`) が読み込まれた場合に影響を受けないようにする。

```
3255 \g@addto@macro\bxjs@begin@document@hook{%
3256   \ifx\bbl@jpn@Seirekitrue@undefined\else
3257     \bxjs@decl@Seireki@cmds
3258     \g@addto@macro\datejapanese{%
3259       \let\today\bxjs@today}%
3260   \fi}
```

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

```
3261 \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。

```
3262 %<slide>\pagestyle{empty}%
```

```
3263 %<article|report>\pagestyle{plain}%
3264 %<book>\pagestyle{headings}%
3265 \pagenumbering{arabic}
3266 \if@twocolumn
3267   \twocolumn
3268   \sloppy
3269   \flushbottom
3270 \else
3271   \onecolumn
3272   \raggedbottom
3273 \fi
3274 %<*slide>
3275   \renewcommand\familydefault{\sfdefault}
3276   \raggedright
3277 %</slide>
```

■ BXJS 独自の追加処理 ㊦

フックを実行する。

```
3278 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3279 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

```
3280 %</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」はこの設定と辻褃が合うように和文フォントサイズを調整する必要がある。ほとんどの場合、和文フォントを NFSS で規定する際に `\jsScale` の値をスケール値として与えれば上手くいく。

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

以下で標準で用意されている和文ドライバの実装を示す。

```
3281 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

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

B.1 補助マクロ

```
3282 %<*minimal>
```

```
3283 %% 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.
```

```
3284 \def\DeclareJaTextFontCommand#1#2{%
```

```
3285   \DeclareRobustCommand#1[1]{%
```

```
3286     \relax
```

```
3287     \ifmmode \expandafter\nfss@text \fi
```

```
3288     {#2##1}}%
```

```
3289 }
```

```
\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。
```

```
3290 \def\DeclareJaMathFontCommand#1#2{%
```

```
3291   \DeclareRobustCommand#1[1]{%
```

```
3292     \relax
```

```
3293     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3294     \nfss@text{\fontfamily\familydefault
```

```
3295         \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3296         #2##1}}%
```

```

3297 }%
3298 }

```

`\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。

```

3299 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3300 \@onlypreamble\bxjs@if@sf@default
3301 \def\bxjs@if@sf@default#1{%
3302   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3303   \g@addto@macro\bxjs@begin@document@hook{%
3304     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3305 }

```

`\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。

```

3306 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3307 \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。

```

3308 \def\jsLetHeadChar#1#2{%
3309   \begingroup
3310     \escapechar=`\ %
3311     \let\bxjs@tmpa={% brace-match-hack
3312       \bxjs@let@hchar@exp#2}%
3313   \endgroup
3314   \let#1\bxjs@g@tmpa}
3315 \def\bxjs@let@hchar@exp{%
3316   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3317 \def\bxjs@let@hchar@exp@a{%
3318   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3319   \bxjs@let@hchar@out\let\relax
3320 }{\bxjs@cond@ifcat\noexpand\@let@token@sptoken\fi% 空白
3321   \bxjs@let@hchar@out\let\space%
3322 }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3323   \bxjs@let@hchar@out\let\@backslashchar
3324 }{\bxjs@let@hchar@exp@b}}}}
3325 \def\bxjs@let@hchar@exp@b#1{%
3326   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3327 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3328 %\message{<#1#2>}%
3329   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3330 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3331 \bxjs@let@hchar@out\let\relax
3332 }{%else
3333 \expandafter\bxjs@let@hchar@exp
3334 }%
3335 }{%else
3336 \bxjs@let@hchar@chr#1%
3337 }}
3338 \def\bxjs@let@hchar@chr#1{%
3339 \bxjs@let@hchar@out\def{#{1}}}}
3340 \def\bxjs@let@hchar@out#1#2{%
3341 \global#1\bxjs@g@tmpa#2\relax
3342 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3343 \chardef\bxjs@let@hchar@csta=128
3344 \chardef\bxjs@let@hchar@cstb=192
3345 \chardef\bxjs@let@hchar@cstc=224
3346 \chardef\bxjs@let@hchar@cstd=240
3347 \chardef\bxjs@let@hchar@cste=248
3348 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3349 \def\bxjs@let@hchar@chr@ue#1{%
3350 \@tempcnta=#1\relax
3351 }\message{\the\@tempcnta}%
3352 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3353 \bxjs@let@hchar@chr@ue@a#1%
3354 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3355 \bxjs@let@hchar@out\let\relax
3356 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3357 \bxjs@let@hchar@chr@ue@b
3358 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3359 \bxjs@let@hchar@chr@ue@c
3360 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3361 \bxjs@let@hchar@chr@ue@d
3362 }{%else
3363 \bxjs@let@hchar@out\let\relax
3364 }}}}}
3365 \def\bxjs@let@hchar@chr@ue@a#1{%
3366 \bxjs@let@hchar@out\def{#{1}}}}
3367 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3368 \bxjs@let@hchar@out\def{#{1#2}}}}
3369 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3370 \bxjs@let@hchar@out\def{#{1#2#3}}}}
3371 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3372 \bxjs@let@hchar@out\def{#{1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
3373 \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。

```
3374 \def\bxjs@let@hchar@chr@pp#1#2{%
3375   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3376 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3377   \message{(\meaning#3:\meaning#4)}%
3378   \bxjs@cond\if#1k\fi{%
3379     \bxjs@let@hchar@out\def{#4}}%
3380   }{%else
3381     \bxjs@let@hchar@chr@ue#3#4%
3382   }}
3383 \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。

```
3384 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3385 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3386 \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。

```
3387 \@onlypreamble\bxjs@declarefontshape
3388 \ifjsWithupTeX
3389 \def\bxjs@declarefontshape{%
3390   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3391   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3392   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3393   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3394 }
3395 \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。

```
3396 \else
3397 \def\bxjs@declarefontshape{%
3398   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3399   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3400   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{%
```



```

3401 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3402 }
3403 \def\bxjs@sizereference{jis}
3404 \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。

```

3405 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3406   \def\bxjs@tmpb{#5}}
3407 \ifjswithpteng \def\bxjs@tmpb{10}%
3408 \else
3409 \expandafter\expandafter\expandafter\bxjs@next
3410 \expandafter\string\the\jfont\relax
3411 \fi
3412 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3413                 \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3414   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3415     \expandafter\let\csname\bxjs@tmpa/\bxjs@tmpb\endcsname=\@undefined}

```

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

```

3416 \begingroup
3417 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3418 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3419 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3420 % 幅が丁度 10pt なら補正は不要
3421 \ifdim\wd\z@=10pt
3422   \global\let\bxjs@scale\jsScale
3423 \else
3424 %  $(10*s)/(10*f)$  として計算、\bxjs@invscale は BXJS で定義
3425   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3426   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3427   \bxjs@invscale\@tempdima\bxjs@tmpa
3428   \xdef\bxjs@scale{\strip@pt\@tempdima}
3429 \fi
3430 \endgroup
3431 %\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。

```

3432 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3433 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{}
3434 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{}
3435 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{}

```

```

3436 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3437 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3438 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3439 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3440 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3441 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3442 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3443 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3444 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3445 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3446 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3447 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{-}
3448 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3449 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
3450 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
3451 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3452 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3453 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3454 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3455 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3456 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3457 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3458 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3459 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3460 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3461 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3462 \DeclareFontShape{\jsc@JTn}{gt}{b}{sl}{<->ssub*gt/bx/n}{-}

```

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

```

3463 \DeclareRobustCommand\rmfamily
3464   {\not@math@alphabet\rmfamily\mathrm
3465     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3466 \DeclareRobustCommand\sffamily
3467   {\not@math@alphabet\sffamily\mathsf
3468     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3469 \DeclareRobustCommand\ttfamily
3470   {\not@math@alphabet\ttfamily\mathtt
3471     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3472 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3473 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3474 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3475 \fi
3476 \bxjs@if@sf@default{%
3477   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```
3478 \selectfont
```

`\bxjs@parse@qh` の処理は pTeX 系では不要になるので無効化する (つまり

`\jsSetQHLlength` は `\setlength` と等価になる)。

```
3479 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3480 \let\bxjs@parse@qh@a\undefined
3481 \let\bxjs@parse@qh@b\undefined
```

■パラメタの設定

```
3482 \prebreakpenalty\jis"2147=10000
3483 \postbreakpenalty\jis"2148=10000
3484 \prebreakpenalty\jis"2149=10000
3485 \inhibitxspcode`!=1
3486 \inhibitxspcode`¯=2
3487 \xspcode`+=3
3488 \xspcode`%=3
```

"80~"FF の範囲の `\spcode` を 3 に変更。

```
3489 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3490   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
```

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

```
3491 \let\jsInhibitGlueAtParTop\@inhibitglue
```

`\jsResetDimen` は空のままよい。

■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。

```
3492 \begingroup
3493 \catcode`\!=0
```

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

```
3494 \gdef\bxjs@ptex@dir{%
3495   !iftdir t%
3496   !else!ifydir y%
3497   !else ?%
3498   !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。

```
3499 % 古い \@makefnmark の定義
3500 \long\def\bxjs@tmpa{\hbox{%
3501   !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
3502   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3503 \ifx\@makefnmark\bxjs@tmpa
3504 \long\gdef\@makefnmark{%
3505   !ifydir \hbox{}\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}%
3506   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3507 \fi
3508 \endgroup
```

B.3 pdfTeX 用の処理

```
3509 \else\ifx p\jsEngine
3510 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3511 \@onlypreamble\bxjs@cjk@loaded
3512 \def\bxjs@cjk@loaded{%
3513   \def\@footnotemark{%
3514     \leavevmode
3515     \ifhmode
3516       \edef\@x@sf{\the\spacefactor}%
3517       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3518         \unkern\unkern
3519         \ifdim\lastskip>\z@ \unskip \fi
3520       \fi\fi
3521       \nobreak
3522     \fi
3523     \@makefnmark
3524     \ifhmode \spacefactor\@x@sf \fi
3525     \relax}%
3526 \let\bxjs@cjk@loaded\relax
3527 }
3528 \g@addto@macro\bxjs@begin@document@hook{%
3529   \@ifpackageloaded{CJK}{%
3530     \bxjs@cjk@loaded
3531   }{}%
3532 }
```

B.4 XeTeX 用の処理

```
3533 \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。

```
3534 \def\bxjs@let@hchar@chr#1{%
3535   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3536   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3537     \bxjs@let@hchar@chr@xe
3538   }{\bxjs@let@hchar@out\def{{#1}}}}
3539 \def\bxjs@let@hchar@chr@xe#1{%
3540   \lccode`0=`#1\relax
3541   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3542 \@onlypreamble\bxjs@do@precisetext
3543 \ifx\XeTeXgenerateactualtext\@undefined\else
3544   \def\bxjs@do@precisetext{%
3545     \XeTeXgenerateactualtext=\@one}
3546 \fi
```

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

```
3547 \@onlypreamble\bxjs@do@simplejasetup
3548 \def\bxjs@do@simplejasetup{%
3549   \ifnum\XeTeXinterchartokenstate>\z@
3550   \else\ifnum\stricmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3551     \jsSimpleJaSetup
3552     \ClassInfo\bxjs@clsname
3553     {\string\jsSimpleJaSetup' is applied\@gobble}%
3554   \fi\fi}
```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```
3555 \newcommand*{\jsSimpleJaSetup}{%
3556   \XeTeXlinebreaklocale "ja"\relax
3557   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3558   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理 (エンジン共通)

```
3559 \fi\fi\fi
```

`simplejasetup` オプションの処理。

```
3560 \ifx\bxjs@do@simplejasetup\undefined\else
3561   \g@addto@macro\bxjs@begin@document@hook{%
3562     \ifbxjs@simplejasetup
3563       \bxjs@do@simplejasetup
3564     \fi}
3565 \fi
```

`precisetext` オプションの処理。

```
3566 \ifbxjs@precisetext
3567   \ifx\bxjs@do@precisetext\undefined
3568     \ClassWarning\bxjs@clsname
3569     {The current engine does not support the\MessageBreak
3570     'precise-text' option\@gobble}
3571   \else
3572     \bxjs@do@precisetext
3573   \fi
3574 \fi
```

■段落頭でのグルー挿入禁止 本体開始時において `\everyparhook` を検査して、“結局何もしない” ことになっている場合は、副作用を完全に無くするために `\everyparhook` を空にする。

```
3575 \g@addto@macro\bxjs@begin@document@hook{%
3576   \ifx\jsInhibitGlueAtParTop\@empty
3577     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3578     \ifx\everyparhook\bxjs@tmpa
3579       \let\everyparhook\@empty
3580     \fi
3581   \fi}
```

`everyparhook=modern` の場合の、`\everyparhook` の有効化の実装。

※本体開始時ではなく最初から有効化していることに注意。

```
3582 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern
```

まず `\everypar` を“乗っ取る”処理を行う。

```
3583 \let\bxjs@everypar\everypar
```

```
3584 \newtoks\everypar
```

```
3585 \everypar\bxjs@everypar
```

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

```
3586 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
```

```
3587 \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。

```
3588 \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。

```
3589 \@onlypreamble\bxjs@adjust@fancyhdr
```

```
3590 \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。

```
3591 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
```

```
3592 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
```

```
3593 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
```

```
3594 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
```

```
3595 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
```

```
3596 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
```

```
3597 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
```

```
3598 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
```

```
3599 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
```

```
3600 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
```

```
3601 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
```

```
3602 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
```

```
3603 \def\bxjs@tmpa{\rm\thepage\strut}%
```

```
3604 \def\bxjs@tmpb{\thepage\strut}%
```

```
3605 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
```

```
3606 \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。

```
3607 \ifx\fullwidth@undefined\else \ifdim\textwidth<\fullwidth
```

```

3608 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3609 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3610 } \bxjs@tmpa
3611 \fi\fi
3612 \PackageInfo\bxjs@clsname
3613 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3614 \def\bxjs@pagestyle@hook{%
3615 \ifpackageloaded{fancyhdr}{%
3616 \bxjs@adjust@fancyhdr
3617 \global\let\bxjs@adjust@fancyhdr\relax
3618 }{}}

```

`\pagestyle` にフックを入れ込む。

```

3619 \let\bxjs@org@pagestyle\pagestyle
3620 \def\pagestyle{%
3621 \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。

```

3622 \g@addto@macro\bxjs@begin@document@hook{%
3623 \bxjs@pagestyle@hook
3624 \global\let\bxjs@pagestyle@hook\relax}

```

■和文空白命令

```

3625 \ifbxjs@jaspace@cmd

```

`\jaenspace` 半角幅の水平空き。

```

3626 \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。

```

3627 \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。

```

3628 \def\_{\zwspace}

```

`\jaspace` `jlreq` クラスと互換の命令。

```

3629 \DeclareRobustCommand*{\jaspace}[1]{%
3630 \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3631 \ClassError\bxjs@clsname
3632 {Unknown jaspacer: #1}{\@eha}%
3633 \else
3634 \csname bxjs@jaspace@@#1\endcsname
3635 \fi}
3636 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3637 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3638 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}

```

3639 \fi

終わり。

3640 \fi

以上で終わり。

3641 %</minimal>

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

3642 %<*standard>

3643 %% このファイルは日本語文字を含みます

3644 \input{bxjsja-minimal.def}

`simplejasetup` は `standard` では無効になる。

3645 \bxjs@simplejasetupfalse

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

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

3646 \newif\ifbxjs@jp@jismmiv

`jis2004` オプションの処理。


```

3647 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3648 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3649 \define@key{bxjsStd}{jis2004}[true]{%
3650   \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。

```

3651 \newif\ifbxjs@jp@units

      units オプションの処理。

3652 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3653 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3654 \define@key{bxjsStd}{units}[true]{%
3655   \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。

```

3656 \let\bxjs@jp@font\@empty

      font オプションの処理。

3657 \define@key{bxjsStd}{font}{%
3658   \edef\bxjs@jp@font{#1}}

      実際の japaram の値を適用する。

3659 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3660 \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。

```

3661 \ifbxjs@jp@jismmiv
3662   \g@addto@macro\@classoptionslist{,jis2004}
3663 % \ifpackagewith 判定への対策
3664   \PassOptionsToPackage{jis2004}{otf}
3665 \fi

```

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

```

3666 \if j\jsEngine
3667   \bxjs@jp@unitsfalse
3668 \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。

```

3669 \ifbxjs@jp@units
3670   \IfFileExists{bxcalc.sty}{%
3671     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3672     \ifx\usepTeXunits\undefined
3673       \PackageWarningNoLine{bxjs@clsname}
3674         {Cannot support pTeX units (zw etc.), since\MessageBreak
3675         the package 'bxcalc' is too old}%

```

```

3676     \bxjs@jp@unitsfalse
3677     \else \useTeXunits
3678     \fi
3679   }{%else
3680     \PackageWarningNoLine\bxjs@clsname
3681     {Cannot support pTeX units (zw etc.), since\MessageBreak
3682     the package 'bxcalc' is unavailable}%
3683     \bxjs@jp@unitsfalse
3684   }
3685 \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。

```

3686 \ifbxjs@jp@units
3687 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3688 \let\bxjs@parse@qh@a\@undefined
3689 \let\bxjs@parse@qh@b\@undefined
3690 \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。

```

3691 \ifbxjs@jp@units
3692 \def\bxjs@let@lenexpr#1#2{%
3693   \edef#1{#2}%
3694   \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3695 \else
3696 \def\bxjs@let@lenexpr{\edef}
3697 \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。

```
3698 \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 3699 \@tempdima=0.25mm
3700 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3701 \let\jH\jQ

```

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

```

\trueH 3702 \ifjsc@mag
3703   \@tempdimb=\jsBaseFontSize\relax
3704   \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3705   \@tempdima=2.5mm
3706   \bxjs@invscale\@tempdima\bxjs@tmpa
3707   \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}

```

```

3708 \tempdima=10pt
3709 \bxjs@invscale\tempdima\bxjs@tmpa
3710 \protected\edef\bxjs@truept{\dimexpr\the\tempdima\relax}
3711 \else \let\trueQ\jQ \let\bxjs@truept\p@
3712 \fi
3713 \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。

```

3714 \tempdima\trueQ \bxjs@invscale\tempdima\jsScale
3715 \protected\edef\ascQ{\dimexpr\the\tempdima\relax}
3716 \tempdima\bxjs@truept \bxjs@invscale\tempdima\jsScale
3717 \protected\edef\ascpt{\dimexpr\the\tempdima\relax}
3718 \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。

```

3719 \def\jafontsize#1#2{%
3720 \beginingroup
3721 \bxjs@jafontsize@a{#1}%
3722 \tempdimb\jsInverseScale\tempdima
3723 \bxjs@jafontsize@a{#2}%
3724 \xdef\bxjs@g@tmpa{%
3725 \noexpand\fontsize{\the\tempdimb}{\the\tempdima}}%
3726 \endgroup\bxjs@g@tmpa}
3727 \def\bxjs@jafontsize@a#1{%
3728 \bxjs@parse@qh{#1}%
3729 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3730 \@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。

```
3731 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```

3732 \newcommand*\setkanjiskip[1]{%
3733 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3734 \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3735 \newcommand*\getkanjiskip{%
3736 \bxjs@kanjiskip}

```

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

```
3737 \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 3738 \bxjs@robust@def\bxjs@enable@kanjiskip{%  
3739 \bxjs@kanjiskip@enabledtrue  
3740 \bxjs@reset@kanjiskip}  
3741 \bxjs@robust@def\bxjs@disable@kanjiskip{%  
3742 \bxjs@kanjiskip@enabledfalse  
3743 \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。

```
3744 \bxjs@robust@def\bxjs@reset@kanjiskip{%  
3745 \ifbxjs@kanjiskip@enabled  
3746 \setlength{\@tempskipa}{\bxjs@kanjiskip}%  
3747 \else \@tempskipa\z@  
3748 \fi  
3749 \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 3750 \def\bxjs@xkanjiskip{Opt}
```

```
\getxkanjiskip 3751 \newcommand*\setxkanjiskip[1]{%
```

```
\ifbxjs@xkanjiskip@enabled 3752 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
```

```
3753 \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3754 \newcommand*\getxkanjiskip{%
```

```
3755 \bxjs@xkanjiskip}
```

```
\bxjs@disable@xkanjiskip 3756 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3757 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3758 \bxjs@xkanjiskip@enabledtrue
```

```
3759 \bxjs@reset@xkanjiskip}
```

```
3760 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3761 \bxjs@xkanjiskip@enabledfalse
```

```
3762 \bxjs@reset@xkanjiskip}
```

```
3763 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```
3764 \ifbxjs@xkanjiskip@enabled
```

```
3765 \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
```

```
3766 \else \@tempskipa\z@
```

```
3767 \fi
```

```
3768 \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。

```
3769 \g@addto@macro\jsResetDimen{%
```

```
3770 \bxjs@reset@kanjiskip
```

```
3771 \bxjs@reset@xkanjiskip}
```

```
3772 \let\bxjs@apply@kanjiskip\relax
```

```
3773 \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。

```
3774 \@onlypreamble\bxjs@adjust@jafont
3775 \def\bxjs@adjust@jafont#1{%
3776   \ifx\jsJaFont\bxjs@@auto
3777     \bxjs@get@kanjiEmbed
3778     \ifx\bxjs@kanjiEmbed\relax
3779       \let\bxjs@tmpa\@empty
3780     \else
3781       \let\bxjs@tmpa\bxjs@kanjiEmbed
3782     \fi
3783   \else
3784     \let\bxjs@tmpa\jsJaFont
3785   \fi
3786   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3787     \ClassWarningNoLine\bxjs@clsname
3788     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3789     not available on the current situation}%
3790     \let\bxjs@tmpa\@empty
3791   \fi\fi
3792 }
3793 \def\bxjs@@auto{auto}
3794 \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。

```
3795 \let\bxjs@kanjiEmbed\relax
```

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

```
3796 \@onlypreamble\bxjs@get@kanjiEmbed
3797 \def\bxjs@get@kanjiEmbed{%
3798   \begingroup\setbox\z@=\hbox{%
3799     \global\let\bxjs@g@tmpa\relax
3800     \endlinechar\m@ne
3801     \let\do\@makeother\dospecials
3802     \catcode32=10 \catcode12=10 %form-feed
3803     \let\bxjs@tmpa\@empty
3804     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3805     \ifeof\@inputcheck\else
3806       \read\@inputcheck to\bxjs@tmpa
3807       \closein\@inputcheck
3808     \fi
3809     \ifx\bxjs@tmpa\@empty\else
3810       \openin\@inputcheck="\bxjs@tmpa"\relax
3811       \@tempwattrue
3812       \loop\if@tempwa
3813         \read\@inputcheck to\bxjs@tmpa
3814         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3815       \ifx\bxjs@tmpb\relax\else
3816         \global\let\bxjs@g@tmpa\bxjs@tmpb
```

```

3817         \@tempwafalse
3818         \fi
3819         \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3820         \ifx\bxjs@tmpb\relax\else
3821             \global\let\bxjs@g@tmpa\bxjs@tmpb
3822             \@tempwafalse
3823         \fi
3824         \ifeof\@inputcheck \@tempwafalse \fi
3825     \repeat
3826 \fi
3827 }\endgroup
3828 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3829 }
3830 \@onlypreamble\bxjs@get@ke@a
3831 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3832     \ifx$#1$\def\bxjs@tmpb{#2}%
3833     \else \let\bxjs@tmpb\relax
3834     \fi}
3835 \@onlypreamble\bxjs@get@ke@b
3836 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3837     \ifx$#1$\def\bxjs@tmpb{#2}%
3838     \else \let\bxjs@tmpb\relax
3839     \fi}

```

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

```

3840 \newcommand*\jachar[1]{%
3841     \begingroup
3842         \jsLetHeadChar\bxjs@tmpa{#1}%
3843         \ifx\bxjs@tmpa\relax
3844             \ClassWarningNoLine\bxjs@clsname
3845                 {Illegal argument given to \string\jachar}%
3846         \else
3847             \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3848         \fi
3849     \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3850 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3851 \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。

```

3852 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3853 \@onlypreamble\bxjs@fix@hyperref@unicode
3854 \def\bxjs@fix@hyperref@unicode#1{%
3855   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3856   \@namedef{KV@Hyp@bxjs/hook}##1{%
3857     \KV@Hyp@unicode{##1}%
3858     \def\KV@Hyp@unicode####1{%
3859       \expandafter\ifx\csname if##1\expandafter\endcsname
3860         \csname if####1\endcsname\else
3861         \ClassWarningNoLine\bxjs@clsname
3862         {Blcked hyperref option 'unicode=####1'}%
3863       \fi
3864     }%
3865   }%
3866 }

```

\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。

```

3867 \@onlypreamble\jsCheckHyperrefUnicode
3868 \let\jsCheckHyperrefUnicode\@empty
3869 \g@addto@macro\bxjs@begin@document@hook{\jsCheckHyperrefUnicode}

```

\bxjs@check@hyperref@unicode hyperref の unicode オプションの値を本体開始時に検証する。

```

3870 \@onlypreamble\bxjs@check@hyperref@unicode
3871 \def\bxjs@check@hyperref@unicode#1{%
3872   \g@addto@macro\jsCheckHyperrefUnicode{%
3873     \@tempswafalse
3874     \begingroup
3875     \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3876     \aftergroup\@tempwattrue \fi
3877     \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3878     \csname if#1\endcsname
3879     \aftergroup\@tempwattrue \fi
3880   \endgroup
3881   \if@tempswa\else
3882     \ClassError\bxjs@clsname
3883     {The value of hyperref 'unicode' key is not suitable\MessageBreak
3884     for the present engine (must be #1)}%
3885     {\@ehc}%
3886   \fi}}

```

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

```

3887 \@onlypreamble\bxjs@urgent@special
3888 \def\bxjs@urgent@special#1{%
3889   \AtBeginDvi{\special{#1}}%
3890   \g@addto@macro\bxjs@begin@document@hook{%
3891     \@ifpackageloaded{atbegshi}{%
3892       \begingroup
3893       \toks\z@{\special{#1}}%
3894       \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3895       \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%

```

```

3896     \endgroup
3897   }{}%
3898 }%
3899 }

```

`\bxjs@resolve@jafont@paren` jafont パラメタ値内の()を解決する。`\bxjs@resolve@jafont@paren\CS` で、`\CS` の内容中の (...) を `\bxjs@jafont@paren{...}` に置き換える。

```

3900 \onlypreamble\bxjs@resolve@jafont@paren
3901 \def\bxjs@resolve@jafont@paren#1{%
3902   \def\bxjs@tmpb{\let#1}%
3903   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3904 \onlypreamble\bxjs@resolve@jafont@paren@a
3905 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3906   \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3907   \else
3908     \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3909     \bxjs@tmpb\bxjs@tmpa
3910   \fi}

```

C.2 pTeX 用設定

```
3911 \if j\jsEngine
```

■ 共通命令の実装

```

3912 \def\bxjs@apply@kanjiskip{%
3913   \kanjiskip\@tempskipa}
3914 \def\bxjs@apply@xkanjiskip{%
3915   \xkanjiskip\@tempskipa}

```

`\jaJaChar` のサブマクロ。

```

3916 \def\bxjs@jachar#1{%
3917   \bxjs@jachar@a#1...\@nil}
3918 \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。

```
3919   \ifx.#2#1%
```

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

```

3920   \else\ifx.#3%
3921     \@tempcnta`#1 \multiply\@tempcnta64
3922     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3923     \bxjs@jachar@b
3924   \else\ifx.#4%
3925     \@tempcnta`#1 \multiply\@tempcnta64
3926     \advance\@tempcnta`#2 \multiply\@tempcnta64
3927     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3928     \bxjs@jachar@b
3929   \else

```



```

3930 \@tempcnta`#1 \multiply\@tempcnta64
3931 \advance\@tempcnta`#2 \multiply\@tempcnta64
3932 \advance\@tempcnta`#3 \multiply\@tempcnta64
3933 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3934 \bxjs@jachar@b
3935 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3936 \ifjsWithupTeX
3937 \def\bxjs@jachar@b{\kchar\@tempcnta}
3938 \else
3939 \def\bxjs@jachar@b{%
3940 \ifx\bxUInt\@undefined\else
3941 \bxUInt{\@tempcnta}%
3942 \fi}
3943 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

3944 \ifbxjs@jaspace@cmd
3945 \def\jathinspace{\hskip\xkanjiskip}
3946 \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。

```

3947 \ifbxjs@jp@jismmiv
3948 \PassOptionsToPackage{prefer2004jis}{pxchfon}
3949 \PassOptionsToPackage{prefer2004jis}{pxbabel}
3950 \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。

```

3951 \let\bxjs@jafont@paren\@firstofone
3952 \let\bxjs@tmpa\jsJaFont
3953 \ifx\bxjs@tmpa\bxjs@@auto
3954 \let\bxjs@tmpa\@empty
3955 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3956 \def\bxjs@tmpa{noembed}
3957 \fi\fi
3958 \bxjs@resolve@jafont@paren\bxjs@tmpa
3959 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3960 \ifx\bxjs@tmpa\@empty\else
3961 \edef\bxjs@next{%
3962 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3963 }\bxjs@next
3964 \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……。)

```
3965 \begingroup
3966   \global\let\bxjs@g@tmpa\relax
3967   \catcode`\|=0 \catcode`\|=12
3968   |def|bxjs@tmpdo#1|@nil{%
3969     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
3970   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
3971     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
3972   |catcode`\|=0 \catcode`\|=12
3973   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
3974     \ifx$#2$\else
3975       \xdef\bxjs@g@tmpa{%
3976         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3977     \fi}
3978 \@firstofone{%
3979   \catcode10=12 \endlinechar\m@ne
3980   \let\do\@makeother \dospecials \catcode32=10
3981   \openin\@inputcheck=otf.sty\relax
3982   \@tempwattrue
3983   \loop\if@tempwa
3984     \ifeof\@inputcheck \@tempwafalse \fi
3985     \if@tempwa
3986       \read\@inputcheck to\bxjs@next
3987       \expandafter\bxjs@tmpdo\bxjs@next\@nil
3988     \fi
3989     \repeat
3990   \closein\@inputcheck
3991 \endgroup}
3992 \bxjs@g@tmpa
```

■hyperref 対策 unicode にはいけない。

```
3993 \ifbxjs@hyperref@enc
3994   \bxjs@check@hyperref@unicode{false}
3995 \fi
```

tounicode special 命令を出力する。

```
3996 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3997   \else\ifjsWithpTeXng T\else F\fi\fi T%
3998   \ifnum\jis"2121="A1A1 %euc
3999     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4000   \else\ifnum\jis"2121="8140 %sjis
4001     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4002   \else\ifnum\jis"2121="3000 %uptex
```

```

4003 \ifbxjs@bigcode
4004 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4005 \PassOptionsToPackage{bigcode}{pxjahyper}
4006 \else
4007 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4008 \fi
4009 \fi\fi\fi
4010 \let\bxToUnicodeSpecialDone=t
4011 \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。

```

4012 \ifx f\bxjs@enablejfam\else
4013 \@enablejfamtrue
4014 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

4015 \if@enablejfam
4016 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4017 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4018 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4019 \jfam\symmincho
4020 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4021 \g@addto@macro\bxjs@begin@document@hook{%
4022 \ifx\reDeclareMathAlphabet\undefined\else
4023 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4024 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4025 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4026 \fi}
4027 \fi

```

C.3 pdfTeX 用設定 : CJK + bxcjkatype

```

4028 \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。

```

4029 \bxjs@adjust@jafont{f}
4030 \let\bxjs@jafont@paren\@firstofone
4031 \bxjs@resolve@jafont@paren\bxjs@tmpa
4032 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4033 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4034 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4035 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4036 \fi

```

```

4037 \edef\bxjs@next{%
4038   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkjatype}[2013/10/15]% v0.2c
4039 }\bxjs@next
4040 \bxjs@cjk@loaded

```

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

※取りあえず固定はしない。

```

4041 \ifbxjs@hyperref@enc
4042   \PassOptionsToPackage{unicode}{hyperref}
4043 \fi

```

\hypersetup 命令で (CJK* 環境に入れなくても) 日本語文字を含む文書情報を設定できるようにするための細工。

※ bxcjkjatype を whole 付きで使っていることが前提。

※パッケージオプションでの指定に対応するのは、「アクティブな高位バイトトークンがその場で展開されてしまう」ため困難である。

```

4044 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
4045 \begingroup
4046   \CJK@input{UTF8.bdg}
4047 \endgroup
4048 \g@addto@macro\pdfstringdefPreHook{%
4049   \@nameuse{CJK@UTF8Binding}%
4050 }
4051 \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。

```

4052 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
4053 \g@addto@macro\pdfstringdefPreHook{%
4054   \ifx~\bxjs@@CJKtilde
4055     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4056     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4057     \let~\@empty
4058   \fi
4059 }
4060 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4061 \def\bxjs@@tildecmd{~}
4062 \def\bxjs@LetUnexpandableSpace#1{%
4063   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4064     \bxjs@org@LetUnexpandableSpace#1%
4065   \fi}
4066 \fi

```

■共通命令の実装

```

4067 \newskip\jsKanjiSkip
4068 \newskip\jsXKanjiSkip
4069 \ifx\CJKecglue\@undefined
4070   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4071 \fi

```

```

4072 \let\autospacing\bxjs@enable@kanjiskip
4073 \let\noautospacing\bxjs@disable@kanjiskip
4074 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4075 \def\bxjs@apply@kanjiskip{%
4076   \jsKanjiSkip\@tempskipa
4077   \let\CJKglue\bxjs@CJKglue}
4078 \let\autoxspacing\bxjs@enable@xkanjiskip
4079 \let\noautoxspacing\bxjs@disable@xkanjiskip
4080 \protected\def\bxjs@CJKe glue{\hskip\jsXKanjiSkip}
4081 \def\bxjs@apply@xkanjiskip{%
4082   \jsXKanjiSkip\@tempskipa
4083   \let\CJKe glue\bxjs@CJKe glue}

```

\jachar のサブマクロの実装。

```

4084 \def\bxjs@jachar#1{%
4085   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4086 \ifbxjs@jaspace@cmd
4087   \protected\def\jathinspace{\CJKe glue}
4088 \fi

```

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

```

4089 \ifx t\bxjs@enablejfam
4090   \ClassWarningNoLine\bxjs@clsname
4091     {You cannot use 'enablejfam=true', since the\MessageBreak
4092     CJK package does not support Japanese math}
4093 \fi

```

C.4 Xe_{La}TeX 用設定：xeCJK + zxjatype

```

4094 \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。

```

4095 \RequirePackage{zxjatype}
4096 \PassOptionsToPackage{no-math}{fontspec}%!
4097 \PassOptionsToPackage{xetex}{graphicx}%!
4098 \PassOptionsToPackage{xetex}{graphics}%!
4099 \ifx\zxJaFamilyName\@undefined
4100   \ClassError\bxjs@clsname
4101     {xeCJK or zxjatype is too old}\@ehc
4102 \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。

```

4103 \bxjs@adjust@jafont{f}
4104 \let\bxjs@jafont@paren\@gobble

```

```

4105 \bxjs@resolve@jafont@paren\bxjs@tmpa
4106 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4107 \ifx\bxjs@tmpa\@empty
4108   \setCJKmainfont[BoldFont=ipaexg.ttf]{ipaexm.ttf}
4109   \setCJKsansfont[BoldFont=ipaexg.ttf]{ipaexg.ttf}
4110 \else
4111   \edef\bxjs@next{%
4112     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4113   }\bxjs@next
4114 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

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

これを踏まえて、ここでは、「X_gTeX 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。

```

4115 \ifnum\strcomp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4116   \ifbxjs@hyperref@enc
4117     \PassOptionsToPackage{unicode}{hyperref}
4118   \fi
4119 \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。

```

4120 \AtEndOfPackage{%
4121 \def\@gnewline #1{%
4122   \ifvmode \@nolnerr
4123   \else
4124     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4125     \nobreak \hskip-1sp\hskip1sp\relax
4126     \ignorespaces
4127   \fi}
4128 }

```

■共通命令の実装

```

4129 \newskip\jsKanjiSkip

```

```

4130 \newskip\jsXKanjiSkip
4131 \ifx\CJKecglue\@undefined
4132 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4133 \fi
4134 \let\autospacing\bxjs@enable@kanjiskip
4135 \let\noautospacing\bxjs@disable@kanjiskip
4136 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4137 \def\bxjs@apply@kanjiskip{%
4138 \jsKanjiSkip\@tempskipa
4139 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4140 \let\autoxspacing\bxjs@enable@xkanjiskip
4141 \let\noautoxspacing\bxjs@disable@xkanjiskip
4142 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4143 \def\bxjs@apply@xkanjiskip{%
4144 \jsXKanjiSkip\@tempskipa
4145 \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。

```

4146 \ifx\mcfamily\@undefined
4147 \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4148 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4149 \fi

```

`\jachar` のサブマクロの実装。

```

4150 \def\bxjs@jachar#1{%
4151 \xeCJKDeclareCharClass{CJK}{`#1}\relax
4152 #1}

```

`\jathinspace` の実装。

```

4153 \ifbxjs@jaspace@cmd
4154 \protected\def\jathinspace{\CJKecglue}
4155 \fi

```

■和文数式ファミリ 和文数式ファミリ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。

```

4156 \ifx t\bxjs@enablejfam
4157 \@enablejfamtrue
4158 \fi

```

実際に和文用の数式ファミリの設定を行う。

※ FIXME: 要検討。

```

4159 \if@enablejfam
4160 \xeCJKsetup{CJKmath=true}
4161 \fi

```

C.5 Lua_TE_X 用設定 : Lua_TE_X-ja

```

4162 \else\if l\jsEngine

```

■ LuaTeX-jā 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。

```
4163 \let\zw\@undefined
4164 \RequirePackage{luatexja}
4165 \edef\bxjs@next{%
4166   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4167 }\bxjs@next
```

フォント代替の明示的定義。

```
4168 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4169 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4170 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4171 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4172 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4173 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4174 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4175 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4176 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4177 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4178 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4179 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4180 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4181 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4182 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4183 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4184 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4185 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4186 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4187 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4188 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4189 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4190 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4191 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4192 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4193 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4194 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4195 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4196 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4197 \DeclareFontShape{JT3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
```

■ 和文フォント定義 \jsJaFont が指定された場合は、その値をオプションとして luatexja-preset を読み込む。非指定の場合は、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)。

```
4198 \bxjs@adjust@jafont{t}
4199 \ifx\bxjs@tmpa\bxjs@noEmbed
4200   \def\bxjs@tmpa{noembed}
4201 \fi
4202 \let\bxjs@jafont@paren@gobble
4203 \bxjs@resolve@jafont@paren\bxjs@tmpa
4204 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4205 \ifx\bxjs@tmpa@empty
4206   \defaultfontfeatures{ Kerning=Off }
4207   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
4208   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
4209 \else
4210   \edef\bxjs@next{%
4211     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4212   }\bxjs@next
4213 \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。

```
4214 \DeclareRobustCommand\rmfamily
4215   {\not@math@alphabet\rmfamily\mathrm
4216     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4217 \DeclareRobustCommand\sffamily
4218   {\not@math@alphabet\sffamily\mathsf
4219     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4220 \DeclareRobustCommand\ttfamily
4221   {\not@math@alphabet\ttfamily\mathtt
4222     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4223 \g@addto@macro\bxjs@begin@document@hook{%
4224   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4225   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4226   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4227 \bxjs@if@sf@default{%
4228   \renewcommand\kanjifamilydefault{\gtdefault}}
```

■和文パラメタの設定

```
4229 % 次の3つは既定値の通り
4230 %\ltjsetparameter{prebreakpenalty={`',10000}}
4231 %\ltjsetparameter{postbreakpenalty={`",10000}}
4232 %\ltjsetparameter{prebreakpenalty={`",10000}}
4233 \ltjsetparameter{jaxspmode={`!,1}}
4234 \ltjsetparameter{jaxspmode={`〒,2}}
4235 \ltjsetparameter{alxspmode={`+,3}}
4236 \ltjsetparameter{alxspmode={`\%,3}}
```

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

※`\jsInhibitGlueAtParTop` は使わない。

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

```
4237 \ifx\ltjfakeparbegin\@undefined
4238   \protected\def\ltjfakeparbegin{%
4239     \ifhmode
4240       \relax\directlua{%
4241         luatexja.jfmglue.create_beginpar_node()}}
4242   \fi}
4243 \fi
```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```
4244 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4245 \begingroup
4246   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4247   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}
4248   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4249     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}\fi
4250   \directlua{
4251     local function patchcmd(cs, code, from, to)
4252       tex.sprint(code:gsub(from:gsub("%W", "%%%0"), "%0"..to)
4253         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1)..")
4254     end
4255     patchcmd(\@xsect, [[\meaning\xsect]],
4256       \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4257     patchcmd(\@item, [[\meaning\@item]],
4258       \bxjs@tmpa, \@{\ltjfakeparbegin})}
4259 \endgroup
4260 \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。

```
4261 \ifbxjs@hyperref@enc
4262   \PassOptionsToPackage{unicode}{hyperref}
4263   \bxjs@check@hyperref@unicode{true}
4264 \fi
```

■共通命令の実装

```
4265 \protected\def\autospacing{%
4266   \ltjsetparameter{autospacing=true}}
4267 \protected\def\noautospacing{%
4268   \ltjsetparameter{autospacing=false}}
4269 \protected\def\autoxspacing{%
4270   \ltjsetparameter{autoxspacing=true}}
4271 \protected\def\noautoxspacing{%
4272   \ltjsetparameter{autoxspacing=false}}
4273 \def\bxjs@apply@kanjiskip{%
```

```

4274 \ltjsetparameter{kanjiskip={\@tempskipa}}
4275 \def\bxjs@apply@xkanjiskip{%
4276 \ltjsetparameter{xkanjiskip={\@tempskipa}}
      \jachar のサブマクロの実装。
4277 \def\bxjs@jachar#1{%
4278 \ltjjachar`#1\relax}
      \jathinspace の実装。
4279 \ifbxjs@jaspace@cmd
4280 \protected\def\jathinspace{%
4281 \hskip\ltjgetparameter{xkanjiskip}\relax}
4282 \fi

```

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

```

4283 \ifx f\bxjs@enablejfam
4284 \ClassWarningNoLine\bxjs@clsname
4285 {You cannot use 'enablejfam=false', since the\MessageBreak
4286 LuaTeX-ja always provides Japanese math families}
4287 \fi

```

C.6 共通処理 (2)

```
4288 \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 4289 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4290 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4291 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4292 \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。

```

4293 \ifx\mathmc\undefined
4294 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4295 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4296 \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)。

```

4297 \ifbxjs@jaspace@cmd
4298 \bxjs@protected\def\bxjs@choice@jathinspace{%
4299 \relax\ifmmode \mskip\medmuskip

```

```

4300 \else \jathinspace\ignorespaces
4301 \fi}
4302 \jsAtEndOfClass{%
4303 \ifjsWiththeTeX \let\>\bxjs@choice@jathinspace
4304 \else \def\>{\protect\bxjs@choice@jathinspace}%
4305 \fi}
4306 \fi

```

■和文・和欧文間空白の初期値

```

4307 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4308 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4309 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4310 \fi

```

以上で終わり。

```
4311 %</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。

```

4312 %<*modern>
4313 \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。

```

4314 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4315 \def\encodingdefault{T1}%
4316 \input{t1enc.def}%
4317 \fontencoding\encodingdefault\selectfont
4318 \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。

```

4319 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4320 \renewcommand{\rmdefault}{lmr}
4321 \renewcommand{\sfdefault}{lmss}
4322 \renewcommand{\ttdefault}{lmtt}
4323 \fi

```

大型演算子用の数式フォントの設定。

※`amsfonts` パッケージと同等にする。

```
4324 \DeclareFontShape{OMX}{cmex}{m}{n}{%
```

```

4325 <-7.5>cmex7<7.5-8.5>cmex8%
4326 <8.5-9.5>cmex9<9.5->cmex10}{}%
4327 \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。
4328 \def\cmex@opt{10}

```

D.2 fixltx2e 読込

※fixltx2e 廃止前の L^AT_EX カーネルの場合。

```

4329 \ifx\@IncludeInRelease\@undefined
4330 \RequirePackage{fixltx2e}
4331 \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。

```

4332 \RequirePackage{bxjscssjkat}

```

D.4 完了

おしまい。

```

4333 %</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。

```

4334 %<*pandoc>
4335 \input{bxjsja-standard.def}
    bxjspandoc パッケージを読み込む。
4336 \RequirePackage{bxjspandoc}

```

`\bxjs@endpreamble@hook` etoolbox の `\AtEndPreamble` で実行される BXJS クラス用のフック。

※ ε -T_EX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```

4337 \@onlypreamble\bxjs@endpreamble@hook
4338 \let\bxjs@endpreamble@hook\empty
4339 \ifjsWitheTeX
4340 \RequirePackage{etoolbox}[2010/08/21]% v2.0
4341 \AtEndPreamble{\bxjs@endpreamble@hook}
4342 \fi

```

E.1 dupload システム

パッケージが重複して読み込まれたときに“option clash”の検査をスキップする。この時に何らかのコードを実行させることができる。

`\bxjs@set@dupload@proc` `\bxjs@set@dupload@proc{〈ファイル名〉}{〈定義本体〉}` 特定のファイルの読込が `\@filewithoptions` で指示されて、しかもそのファイルが読込済である場合に、オプション重複検査をスキップして、代わりに〈定義本体〉のコードを実行する。このコード中で #1 は渡されたオプション列のテキストに置換される。

```
4343 \@onlypreamble\bxjs@set@dupload@proc
4344 \def\bxjs@set@dupload@proc#1{%
4345   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
4346 \@onlypreamble\bxjs@set@dupload@proc@a
4347 \def\bxjs@set@dupload@proc@a#1{%
4348   \@onlypreamble#1\def#1##1}
4349 \def\bxjs@unset@dupload@proc#1{%
4350   \expandafter\let\csname bxjs@dlp/#1\endcsname\@undefined}
```

`\@if@options` `\@if@options` の再定義。

```
4351 \@onlypreamble\bxjs@org@if@options
4352 \let\bxjs@org@if@options\@if@options
4353 \@onlypreamble\bxjs@org@reset@options
4354 \let\bxjs@org@reset@options\relax
4355 \def\@if@options#1#2#3{%
4356   \let\bxjs@next\@secondoftwo
4357   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@current}%
4358   \ifx\bxjs@tmpa\bxjs@tmpb
4359     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4360       \let\bxjs@next\@firstoftwo \fi
4361   \fi
4362   \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4363 \g@addto@macro\bxjs@begin@document@hook{%
4364   \let\@if@options\bxjs@org@if@options}
4365 \@onlypreamble\bxjs@do@dupload@proc
4366 \def\bxjs@do@dupload@proc#1#2#3{%
4367   \ifx\bxjs@org@reset@options\relax
4368     \let\bxjs@org@reset@options\@reset@options
4369   \fi
4370   \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4371   \def\@reset@options{%
4372     \let\@reset@options\bxjs@org@reset@options
4373     \@reset@options
4374     \bxjs@next{#3}}%
4375   \@firstoftwo}
```

E.2 lang 変数

`lang=ja` という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐための対策を行う。

`\bxjs@polyglossia@options` Polyglossia のオプション列のテキスト。“実際には読み込まれていない” 場合は `\relax` になる。

```
4376 \let\bxjs@polyglossia@options\relax
```

`\bxjs@babel@options` Babel のオプション列のテキスト。“実際には読み込まれていない” 場合は `\relax` になる。

```
4377 \let\bxjs@babel@options\relax
```

■ Polyglossia について つまり $X_{\text{d}}\text{T}_{\text{E}}\text{X}$ · $\text{L}\text{u}\text{a}\text{T}_{\text{E}}\text{X}$ の場合。

※この場合 `etoolbox` が使用可能になっている。

```
4378 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0
```

パッケージの読込を検知するため読込済のマークを付けて `dupload` の処理を仕込む。

```
4379 \pandocSkipLoadPackage{polyglossia}
4380 \bxjs@set@dupload@proc{polyglossia.sty}{%
4381   \bxjs@unset@dupload@proc{polyglossia.sty}%
4382   \ClassWarning\bxjs@clsname
4383   {Package polyglossia is required}%
4384   \def\bxjs@polyglossia@options{#1}%
```

`polyglossia` の読込が指示された場合、直後に `\setmainlanguage` が実行されることを想定して、フック用の `\setmainlanguage` を定義する。

※先に `\setmainlanguage` 以外が実行された場合はエラーになる。

```
4385 \newcommand*\setmainlanguage[2] []{%
```

もし、言語名が空の `\setmainlanguage{}` が実行された場合は、`lang=ja` が指定されたと見なす。

```
4386   \ifblank{##2}{%
4387     \ClassWarning\bxjs@clsname
4388     {Main language is invalid, thus the fallback\MessageBreak
4389     definitions will be employed}%
4390     \bxjs@pandoc@lang@ja
```

言語名が非空ならば、本来の `polyglossia` の処理を実行する。

```
4391   }{%else
4392     \ClassWarning\bxjs@clsname
4393     {Main language is valid (##2),\MessageBreak
4394     thus polyglossia will be loaded}%
4395     \csundef{ver@polyglossia.sty}%
4396     \edef\bxjs@next{%
4397       \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia} []%
4398     }\bxjs@next
4399     \setmainlanguage[##1]{##2}%
4400   }}}
```

プレアンブルで polyglossia の読込が指示されなかった場合、Polyglossia と連携するパッケージの誤動作を防ぐため、読込済マークを外す。

```
4401 \g@addto@macro\bxjs@endpreamble@hook{%
4402   \ifx\bxjs@polyglossia@options\relax
4403     \csundef{ver@polyglossia.sty}%
4404   \fi}
```

`\bxjs@pandoc@lang@ja` Pandoc 側で lang=ja が指定されていた場合の処理。この場合は Polyglossia の処理を無効化するためにダミーの定義を行う。その時点でダミーの `\setotherlanguage(s)` を定義する。

```
4405 \@onlypreamble\bxjs@pandoc@lang@ja
4406 \def\bxjs@pandoc@lang@ja{%
4407   \renewcommand*\setmainlanguage[2] [] {}%
4408   \newcommand*\setotherlanguage[2] [] {}%
4409   \ifblank{##2}{-}{%else
4410     \cslet{##2}\@empty \cslet{end##2}\@empty
4411     \cslet{text##2}\@firstofone}}%
4412   \newcommand*\setotherlanguages[2] [] {}%
4413   \@for\bxjs@tmpa:={##2}\do{%
4414     \setotherlangauge{\bxjs@tmpa}}%
```

Polyglossia の読込済マークは外れるようにしておく。

```
4415 \let\bxjs@polyglossia@options\relax}%
4416 \fi
```

■Babel について Xe_{La}TeX 以外の場合。

※ Pandoc 2.6 版において、Lua_{TeX} で用いられる多言語パッケージが Polyglossia から Babel に変更された。

```
4417 \if x\jsEngine\else
```

パッケージの読込を検知するため読込済のマークを付けて dupload の処理を仕込む。

```
4418 \pandocSkipLoadPackage{babel}
4419 \bxjs@set@dupload@proc{babel.sty}{%
4420   \bxjs@unset@dupload@proc{babel.sty}%
4421   \ClassWarning\bxjs@clsname
4422   {Package babel is required}}%
```

パッケージオプションに言語名が空の main= があるかを調べる。ある場合は lang=ja 対策を実行する。

※`\bxjs@babel@options` には main= を除いたオプション列を格納する。

```
4423 \@tempwafalse \let\bxjs@babel@options\@empty
4424 \def\bxjs@tmpb{main=}
4425 \@for\bxjs@tmpa:=#1\do{%
4426   \ifx\bxjs@tmpa\bxjs@tmpb \@tempwatruel
4427   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4428   \fi}%
4429 \if@tempswa
```



```

4430 \ClassWarning\bxjs@clsname
4431   {Main language is invalid, so fallback\MessageBreak
4432     definitions will be employed}%
4433 \bxjs@pandoc@lang@ja

```

ない場合は、本来の `babel` の処理を実行する。

```

4434 \else
4435 \ClassWarning\bxjs@clsname
4436   {Main language is valid,\MessageBreak
4437     thus babel will be loaded}%
4438 \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4439 \RequirePackage[#1]{babel}[]%
4440 \fi}

```

プレアンブルで `babel` の読込が指示されなかった場合、読込済マークを外す。

```

4441 \g@addto@macro\bxjs@endpreamble@hook{%
4442 \ifx\bxjs@babel@options\relax
4443 \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4444 \fi}

```

`\bxjs@pandoc@lang@ja` Pandoc 側で `lang=ja` が指定されていた場合の処理。

```

4445 \@onlypreamble\bxjs@pandoc@lang@ja
4446 \def\bxjs@pandoc@lang@ja{%
4447 \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4448 \edef\bxjs@next{%
4449 \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4450 }\bxjs@next
4451 \if j\jsEngine
4452 \RequirePackage[main=japanese]{pxbabel}[]%
4453 \else
4454 \RequirePackage{bxorigcapt}[]%
4455 \fi}

```

`lang` 対策はこれで終わり。

```
4456 \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。

```

4457 \bxjs@set@dupload@proc{geometry.sty}{%
4458 \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。

```

4459 \if l\jsEngine
4460   \pandocSkipLoadPackage{xCJK}
4461   \providecommand*{\setCJKmainfont}{\setmainfont}
4462 \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。

```

4463 \g@addto@macro\bxjs@begin@document@hook{%
4464   \@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。

```

4465   \ifx\oldparagraph\undefined\else
4466     \@tempwatrue
4467   \fi

```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```

4468   \ifnum\c@secnumdepth>3
4469     \@tempwatrue
4470   \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。

```

4471   \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4472     \let\jsParagraphMark\empty
4473   \fi\fi}

```

E.6 全角空白文字

```

4474 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4475   \catcode"3000=\active
4476   \begingroup \catcode`\!=7
4477   \protected\def!!!3000{\zwspace}
4478   \endgroup
4479 \else\ifx\DeclareUnicodeCharacter\undefined\else
4480   \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4481   \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4482 \fi\fi

```

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

```
4483 \if j\jsEngine
4484   \bxjs@fix@hyperref@unicode{false}
4485 \else
4486   \bxjs@fix@hyperref@unicode{true}
4487 \fi
```

E.7 完了

おしまい。

```
4488 %</pandoc>
```

和文ドライバ実装はここまで。

```
4489 %</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 : ムニヤムニヤムニヤ。
- bxjscjcat : 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。

```
4490 %<*anc>
```

付録 G 補助パッケージ : bxjscompat

ムニヤムニヤムニヤ……。

G.1 準備

```
4491 %<*compat>
4492 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4493 \let\bxac@engine=n
4494 \def\bxac@do#1#2{%
4495   \edef\bxac@tmpa{\string#1}%
4496   \edef\bxac@tmpb{\meaning#1}%
4497   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
4498 \bxac@do\XeTeXversion{\let\bxac@engine=x}
4499 \bxac@do\luatexversion{\let\bxac@engine=1}
```

`\bxac@delayed@if@bxjs` もし BXJS クラスの読込中でこのパッケージが読み込まれているならば、BXJS のクラスの終わりまで実行を遅延する。

```

4500 \ifx\jsAtEndOfClass\@undefined
4501 \let\bxac@delayed@if@bxjs\@firstofone
4502 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
4503 \fi

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 4504 \newif\ifbxac@in@old@behavior
4505 \let\ImposeOldLuaTeXBehavior\relax
4506 \let\RevokeOldLuaTeXBehavior\relax

```

G.2 XeTeX 部分

```

4507 \ifx x\bxac@engine
    XeTeX 文字クラスのムニヤムニヤ。
4508 \@onlypreamble\bxac@adjust@charclass
4509 \bxac@delayed@if@bxjs{%
4510 \@ifpackageloaded{xCJK}{-}{%else
4511 \ifx\Xe@alloc@intercharclass\@undefined\else
4512 \ifnum\Xe@alloc@intercharclass=\z@
4513 \PackageInfo\bxac@pkgname
4514 {Setting up interchar class for CJK...\@gobble}%
4515 \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4516 \Xe@alloc@intercharclass=3
4517 }{%else
4518 \PackageWarning\bxac@pkgname
4519 {Cannot find file 'load-unicode-xetex-classes.tex'%
4520 \@gobble}%
4521 }%
4522 \fi\fi
4523 \ifnum\XeTeXcharclass"3041=\z@
4524 \PackageInfo\bxac@pkgname
4525 {Adjusting interchar class for CJK...\@gobble}%
4526 \@for\bxac@x={%
4527 3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4528 3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4529 30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4530 31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4531 31FF%
4532 }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4533 \fi
4534 }%
4535 }
    以上。
4536 \fi

```

G.3 LuaTeX 部分

```

4537 \ifx l\bxac@engine

```

ムニヤムニヤ。

```
4538 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4539 \chardef\pdftexversion=200
4540 \def\pdftexrevision{0}
4541 \let\pdftexbanner\luatexbanner
4542 \fi\fi
```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```
\RevokeOldLuaTeXBehavior 4543 \begingroup\expandafter\expandafter\expandafter\endgroup
4544 \expandafter\ifx\csname outputmode\endcsname\relax\else
4545 \def\bxac@ob@list{%
4546 \do{\let}\pdfoutput{\outputmode}%
4547 \do{\let}\pdfpagewidth{\pagewidth}%
4548 \do{\let}\pdfpageheight{\pageheight}%
4549 \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4550 \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}%
4551 \def\bxac@ob@do#1#2{\begingroup
4552 \expandafter\bxac@ob@do@a\csname bxac@string#2\endcsname{#1}#2}
4553 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4554 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4555 \else \let#3#1\relax \let#1\@undefined
4556 \fi}
4557 \protected\def\ImposeOldLuaTeXBehavior{%
4558 \unless\ifbxac@in@old@behavior
4559 \bxac@in@old@behaviortrue
4560 \let\do\bxac@ob@do \bxac@ob@list
4561 \fi}
4562 \protected\def\RevokeOldLuaTeXBehavior{%
4563 \ifbxac@in@old@behavior
4564 \bxac@in@old@behaviorfalse
4565 \let\do\bxac@ob@do \bxac@ob@list
4566 \fi}
4567 \fi
```

漢字および完成形ハングルのカテゴリコードのムニヤムニヤ。

```
4568 \ifnum\luatexversion>64 \directlua{
4569 local function range(cs, ce, cc, ff)
4570 if ff or not tex.getcatcode(cs) == cc then
4571 local setcc = tex.setcatcode
4572 for c = cs, ce do setcc(c, cc) end
4573 end
4574 end
4575 range(0x3400, 0x4DB5, 11, false)
4576 range(0x4DB5, 0x4DBF, 11, true)
4577 range(0x4E00, 0x9FCC, 11, false)
4578 range(0x9FCD, 0x9FFF, 11, true)
4579 range(0xAC00, 0xD7A3, 11, false)
4580 range(0x20000, 0x2A6D6, 11, false)
4581 range(0x2A6D7, 0x2A6FF, 11, true)
```

```

4582   range(0x2A700, 0x2B734, 11, false)
4583   range(0x2B735, 0x2B73F, 11, true)
4584   range(0x2B740, 0x2B81D, 11, false)
4585   range(0x2B81E, 0x2B81F, 11, true)
4586   range(0x2B820, 0x2CEA1, 11, false)
4587   range(0x2CEA2, 0x2FFFD, 11, true)
4588 } \fi

```

以上。

```
4589 \fi
```

G.4 完了

おしまい。

```
4590 %</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

```

4591 %<*cjkat>
4592 \def\bxjx@pkgname{bxjscjkat}
4593 \newcount\bxjx@canta
4594 \@onlypreamble\bxjx@tmpdo
4595 \@onlypreamble\bxjx@tmpdo@a
4596 \@onlypreamble\bxjx@tmpdo@b

```

\bxjx@engine エンジンの種別。

```

4597 \let\bxjx@engine=n
4598 \def\bxjx@tmpdo#1#2{%
4599   \edef\bxjx@tmpa{\string#1}%
4600   \edef\bxjx@tmpb{\meaning#1}%
4601   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4602 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4603 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4604 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4605 \bxjx@tmpdo\pdftexversion{\let\bxjx@engine=p}
4606 \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。

```

4607 \def\bxjx@tmpdo#1#2{%
4608   \if#1\bxjx@engine
4609     \@ifpackageloaded{#2}{}{%else
4610       \PackageError\bxjx@pkgname

```

```

4611     {Package '#2' must be loaded}%
4612     {Package loading is aborted.\MessageBreak\@ehc}%
4613     \endinput}
4614     \fi}
4615 \bxjx@tmpdo{p}{bxcjkatype}
4616 \bxjx@tmpdo{x}{xeCJK}
4617 \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。

```

4618 \ifx\TextOrMath\@undefined
4619     \RequirePackage{fixltx2e}
4620 \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。

```

4621 \if u\bxjx@engine
4622 \@for\bxjx@tmpa:={%
4623 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4624 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4625 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4626 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4627 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4628 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4629 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4630 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4631 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4632 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4633 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4634 FE70,FFF0,%
4635 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4636 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4637 10480,104B0,10500,10530,10600,10800,10840,10860,%
4638 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4639 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4640 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4641 11200,11280,112B0,11300,11400,11480,11580,11600,%
4642 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4643 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4644 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4645 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4646 1EE00,1F000,1F030,1FOA0,1F300,1F600,1F650,1F680,%
4647 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4648 00C0%
4649 }\do{%
4650 \@tempcnta="\bxjx@tmpa\relax

```

```

4651 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4652 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4653 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4654 \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 (LuaTeX-ja 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。

```

4655 \@onlypreamble\bxjx@grkcyr@list
4656 \def\bxjx@grkcyr@list{%
4657 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4658 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4659 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4660 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4661 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4662 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4663 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
4664 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4665 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4666 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4667 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4668 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4669 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
4670 \do{039E}{LGR}{\textXi}{\Xi}%          % GR. C. L. XI
4671 \do{039F}{LGR}{\textOmicron}{O}%      % GR. C. L. OMICRON
4672 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
4673 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
4674 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4675 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
4676 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4677 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
4678 \do{03A7}{LGR}{\textChi}{X}%           % GR. C. L. CHI
4679 \do{03A8}{LGR}{\textPsi}{\Psi}%        % GR. C. L. PSI
4680 \do{03A9}{LGR}{\textOmega}{\Omega}%    % GR. C. L. OMEGA
4681 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
4682 \do{03B2}{LGR}{\textbeta}{\beta}%      % GR. S. L. BETA
4683 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA
4684 \do{03B4}{LGR}{\textdelta}{\delta}%    % GR. S. L. DELTA

```


4685 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
4686 \do{03B6}{LGR}{\textzeta}{\zeta}% % GR. S. L. ZETA
4687 \do{03B7}{LGR}{\texteta}{\eta}% % GR. S. L. ETA
4688 \do{03B8}{LGR}{\texttheta}{\theta}% % GR. S. L. THETA
4689 \do{03B9}{LGR}{\textiota}{\iota}% % GR. S. L. IOTA
4690 \do{03BA}{LGR}{\textkappa}{\kappa}% % GR. S. L. KAPPA
4691 \do{03BB}{LGR}{\textlambda}{\lambda}% % GR. S. L. LAMDA
4692 \do{03BC}{LGR}{\textmu}{\mu}% % GR. S. L. MU
4693 \do{03BD}{LGR}{\textnu}{\nu}% % GR. S. L. NU
4694 \do{03BE}{LGR}{\textxi}{\xi}% % GR. S. L. XI
4695 \do{03BF}{LGR}{\textomicron}{\o}% % GR. S. L. OMICRON
4696 \do{03C0}{LGR}{\textpi}{\pi}% % GR. S. L. PI
4697 \do{03C1}{LGR}{\textrho}{\rho}% % GR. S. L. RHO
4698 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
4699 \do{03C3}{LGR}{\textsigma}{\sigma}% % GR. S. L. SIGMA
4700 \do{03C4}{LGR}{\texttau}{\tau}% % GR. S. L. TAU
4701 \do{03C5}{LGR}{\textupsilon}{\upsilon}% % GR. S. L. UPSILON
4702 \do{03C6}{LGR}{\textphi}{\phi}% % GR. S. L. PHI
4703 \do{03C7}{LGR}{\textchi}{\chi}% % GR. S. L. CHI
4704 \do{03C8}{LGR}{\textpsi}{\psi}% % GR. S. L. PSI
4705 \do{03C9}{LGR}{\textomega}{\omega}% % GR. S. L. OMEGA
4706 \do{0401}{T2A}{\CYRYO}{\O}% % CY. C. L. IO
4707 \do{0410}{T2A}{\CYRA}{\A}% % CY. C. L. A
4708 \do{0411}{T2A}{\CYRB}{\B}% % CY. C. L. BE
4709 \do{0412}{T2A}{\CYRV}{\V}% % CY. C. L. VE
4710 \do{0413}{T2A}{\CYRG}{\G}% % CY. C. L. GHE
4711 \do{0414}{T2A}{\CYRD}{\D}% % CY. C. L. DE
4712 \do{0415}{T2A}{\CYRE}{\E}% % CY. C. L. IE
4713 \do{0416}{T2A}{\CYRZH}{\ZH}% % CY. C. L. ZHE
4714 \do{0417}{T2A}{\CYRZ}{\Z}% % CY. C. L. ZE
4715 \do{0418}{T2A}{\CYRI}{\I}% % CY. C. L. I
4716 \do{0419}{T2A}{\CYRISHRT}{\I}% % CY. C. L. SHORT I
4717 \do{041A}{T2A}{\CYRK}{\K}% % CY. C. L. KA
4718 \do{041B}{T2A}{\CYRL}{\L}% % CY. C. L. EL
4719 \do{041C}{T2A}{\CYRM}{\M}% % CY. C. L. EM
4720 \do{041D}{T2A}{\CYRN}{\N}% % CY. C. L. EN
4721 \do{041E}{T2A}{\CYRO}{\O}% % CY. C. L. O
4722 \do{041F}{T2A}{\CYRP}{\P}% % CY. C. L. PE
4723 \do{0420}{T2A}{\CYRR}{\R}% % CY. C. L. ER
4724 \do{0421}{T2A}{\CYRS}{\S}% % CY. C. L. ES
4725 \do{0422}{T2A}{\CYRT}{\T}% % CY. C. L. TE
4726 \do{0423}{T2A}{\CYRU}{\U}% % CY. C. L. U
4727 \do{0424}{T2A}{\CYRF}{\F}% % CY. C. L. EF
4728 \do{0425}{T2A}{\CYRH}{\H}% % CY. C. L. HA
4729 \do{0426}{T2A}{\CYRC}{\C}% % CY. C. L. TSE
4730 \do{0427}{T2A}{\CYRCH}{\C}% % CY. C. L. CHE
4731 \do{0428}{T2A}{\CYRSH}{\S}% % CY. C. L. SHA
4732 \do{0429}{T2A}{\CYRSHCH}{\S}% % CY. C. L. SHCHA
4733 \do{042A}{T2A}{\CYRHRDSN}{\S}% % CY. C. L. HARD SIGN

4734	<code>\do{042B}{T2A}{\CYRERY}{}</code>	% CY. C. L. YERU
4735	<code>\do{042C}{T2A}{\CYRSFTSN}{}</code>	% CY. C. L. SOFT SIGN
4736	<code>\do{042D}{T2A}{\CYREREV}{}</code>	% CY. C. L. E
4737	<code>\do{042E}{T2A}{\CYRYU}{}</code>	% CY. C. L. YU
4738	<code>\do{042F}{T2A}{\CYRYA}{}</code>	% CY. C. L. YA
4739	<code>\do{0430}{T2A}{\cyra}{}</code>	% CY. S. L. A
4740	<code>\do{0431}{T2A}{\cyrb}{}</code>	% CY. S. L. BE
4741	<code>\do{0432}{T2A}{\cyrv}{}</code>	% CY. S. L. VE
4742	<code>\do{0433}{T2A}{\cyrg}{}</code>	% CY. S. L. GHE
4743	<code>\do{0434}{T2A}{\cyrd}{}</code>	% CY. S. L. DE
4744	<code>\do{0435}{T2A}{\cyre}{}</code>	% CY. S. L. IE
4745	<code>\do{0436}{T2A}{\cyrzh}{}</code>	% CY. S. L. ZHE
4746	<code>\do{0437}{T2A}{\cyrz}{}</code>	% CY. S. L. ZE
4747	<code>\do{0438}{T2A}{\cyri}{}</code>	% CY. S. L. I
4748	<code>\do{0439}{T2A}{\cyrishrt}{}</code>	% CY. S. L. SHORT I
4749	<code>\do{043A}{T2A}{\cyrk}{}</code>	% CY. S. L. KA
4750	<code>\do{043B}{T2A}{\cyrl}{}</code>	% CY. S. L. EL
4751	<code>\do{043C}{T2A}{\cyrm}{}</code>	% CY. S. L. EM
4752	<code>\do{043D}{T2A}{\cyrn}{}</code>	% CY. S. L. EN
4753	<code>\do{043E}{T2A}{\cyro}{}</code>	% CY. S. L. O
4754	<code>\do{043F}{T2A}{\cyrp}{}</code>	% CY. S. L. PE
4755	<code>\do{0440}{T2A}{\cyrr}{}</code>	% CY. S. L. ER
4756	<code>\do{0441}{T2A}{\cyrs}{}</code>	% CY. S. L. ES
4757	<code>\do{0442}{T2A}{\cyrt}{}</code>	% CY. S. L. TE
4758	<code>\do{0443}{T2A}{\cyru}{}</code>	% CY. S. L. U
4759	<code>\do{0444}{T2A}{\cyrf}{}</code>	% CY. S. L. EF
4760	<code>\do{0445}{T2A}{\cyrh}{}</code>	% CY. S. L. HA
4761	<code>\do{0446}{T2A}{\cyrc}{}</code>	% CY. S. L. TSE
4762	<code>\do{0447}{T2A}{\cyrch}{}</code>	% CY. S. L. CHE
4763	<code>\do{0448}{T2A}{\cyrrsh}{}</code>	% CY. S. L. SHA
4764	<code>\do{0449}{T2A}{\cyrshch}{}</code>	% CY. S. L. SHCHA
4765	<code>\do{044A}{T2A}{\cyrhrdsn}{}</code>	% CY. S. L. HARD SIGN
4766	<code>\do{044B}{T2A}{\cyrery}{}</code>	% CY. S. L. YERU
4767	<code>\do{044C}{T2A}{\cyrsftsn}{}</code>	% CY. S. L. SOFT SIGN
4768	<code>\do{044D}{T2A}{\cyrerev}{}</code>	% CY. S. L. E
4769	<code>\do{044E}{T2A}{\cyryu}{}</code>	% CY. S. L. YU
4770	<code>\do{044F}{T2A}{\cyrya}{}</code>	% CY. S. L. YA
4771	<code>\do{0451}{T2A}{\cyryo}{}</code>	% CY. S. L. IO
4772	<code>\do{00A7}{TS1}{\textsection}{\mathsection}</code>	% SECTION SYMBOL
4773	<code>\do{00A8}{TS1}{\textasciidieresis}{}</code>	% DIAERESIS
4774	<code>\do{00B0}{TS1}{\textdegree}{\mathdegree}</code>	% DEGREE SIGN
4775	<code>\do{00B1}{TS1}{\textpm}{\pm}</code>	% PLUS-MINUS SIGN
4776	<code>\do{00B4}{TS1}{\textasciicute}{}</code>	% ACUTE ACCENT
4777	<code>\do{00B6}{TS1}{\textparagraph}{\mathparagraph}</code>	% PILCROW SIGN
4778	<code>\do{00D7}{TS1}{\texttimes}{\times}</code>	% MULTIPLICATION SIGN
4779	<code>\do{00F7}{TS1}{\textdiv}{\div}</code>	% DIVISION SIGN
4780	}	

`\mathdegree` 面倒なので補っておく。

```
4781 \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。

```
4782 \newif\ifbxjx@gcc@cjk
```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```
4783 \newcommand*\greekasCJK{%
```

```
4784   \bxjx@gcc@cjktrue}
```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```
4785 \newcommand*\nogreekasCJK{%
```

```
4786   \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。

```
4787 \def\bxjx@tmpdo#1\relax{%
```

```
4788   \def\bxjx@fake@grk##1##2{%
```

```
4789     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
```

```
4790 \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
```

```
4791   \ifx\##1\\\%
```

```
4792     \bxjx@canta##4\divide\bxjx@canta\@cclvi
```

```
4793     \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
```

```
4794     \mathchar\bxjx@canta
```

```
4795     \else ##3\fi}
```

```
4796 }\expandafter\bxjx@tmpdo\string\mathchar\relax
```

■ pdf \TeX ・up \TeX の場合

```
4797 \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 \LaTeX においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```
4798 \@ifpackageloaded{inputenc}{}\fi\else
```

```
4799   \RequirePackage[utf8]{inputenc}}
```

```
4800 \def\bxjx@tmpa{utf8}
```

```
4801 \ifx\bxjx@tmpa\inputencdoingname
```

```
4802   \PackageWarningNoLine\bxjx@pkgname
```

```

4803   {Input encoding changed to utf8}%
4804   \inputencoding{utf8}%
4805 \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。

```

4806 \if u\bxjx@engine
4807 \kcatcode"0370=15
4808 \kcatcode"0400=15
4809 \kcatcode"0500=15
4810 \fi

```

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

```

4811 \def\bxjx@tmpdo#1{%
4812   \@tempcnta="#1\relax
4813   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4814 \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。

```

4815   \ifx\#5\%
4816     \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。

```

4817   \else\ifcat A\noexpand#5%
4818     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4819       {\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。

```

4820   \else \def\bxjx@tmpa{#5}%
4821   \fi\fi
4822   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4823   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

4824 \if u\bxjx@engine
4825 \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。

```

4826   \kchardef#1=\@tempcnta
4827   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4828   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdfTeX の場合も処理はほとんど同じ。ただし、和文用の定義として `\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。

```
4829 \else\if p\bxjx@engine
4830 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4831   \mathchardef#1=\@tempcnta
4832   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4833   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4834 \fi\fi
```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```
4835 \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。

```
4836 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4837 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4838 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4839 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4840   \count@=#1\relax
4841   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4842     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4843   \else
4844     \wlog{ \space\space skipped defining Unicode char U+#1}%
4845   \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。

```
4846 \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。

```
4847 \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。

```
4848 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
```

```
4849 \else \UseTextSymbol{#2}{#3}%
```

```
4850 \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。

```
4851 \begingroup
4852 \toks@ \expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}
4853 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4854   \noexpand\bxjx@swap@DUC@cmd
4855   \the\toks@
4856   \noexpand\bxjx@swap@DUC@cmd}}
4857 \endgroup\next
4858 \def\bxjx@swap@DUC@cmd{%
4859   \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```

4860 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4861 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
4862 \let\bxjx@tmpa\relax}

```

以上。

■ $X_{\text{L}}\text{T}_{\text{E}}\text{X}$ ・ $\text{LuaT}_{\text{E}}\text{X}$ の場合

```

4863 \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。

```

4864 \def\bxjx@tmpdo#1{%
4865   \bxjx@cnta="#1\relax
4866   \begingroup
4867     \lccode`~=\bxjx@cnta
4868   \lowercase{\endgroup
4869     \bxjx@tmpdo@a{~}{#1}}
4870 \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、 $\text{upL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ の場合と同じ方法で“数式中の動作”を決定し、当該の文字を math active にしてその動作を設定する。

```

4871   \ifx\#5\\\let\bxjx@tmpa\relax
4872   \else\ifcat A\noexpand#5%
4873     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4874       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4875   \else \def\bxjx@tmpa{#5}%
4876   \fi\fi
4877   \ifx\bxjx@tmpa\relax\else
4878     \mathcode\bxjx@cnta"8000 \let#1\bxjx@tmpa
4879   \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。

```

4880 \mathchardef\bxjx@tmpa="119
4881 \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 $\backslash\text{no}[\text{greekasCJK}]$ で切り替わるようにする。

$\text{LuaT}_{\text{E}}\text{X}$ の場合は、 $\text{LuaT}_{\text{E}}\text{X}\text{-ja}$ 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。

```

4882 \if 1\bxjx@engine
4883   \protected\def\greekasCJK{%
4884     \bxjx@gcc@cjctrue
4885     \ltjsetparameter{jacharrange={+2, +8}}
4886   \protected\def\nogreekasCJK{%
4887     \bxjx@gcc@cjcfalse
4888     \ltjsetparameter{jacharrange={-2, -8}}
4889 \fi

```

$X_{\text{L}}\text{T}_{\text{E}}\text{X}$ の場合、 xeCJK は $X_{\text{L}}\text{T}_{\text{E}}\text{X}$ 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。

```

4890 \if x\bxjx@engine
4891 \let\bxjx@gcc@CJK@list\@empty
4892 \def\do#1#2#3#4{%
4893 \edef\bxjx@gcc@CJK@list{\bxjx@gcc@CJK@list
4894 \noexpand\XeTeXcharclass"#1\bxjx@canta}}
4895 \bxjx@grkcyr@list
4896 \protected\def\greekasCJK{%
4897 \bxjx@gcc@CJKtrue
4898 \bxjx@canta=\@ne \bxjx@gcc@CJK@list}
4899 \protected\def\nogreekasCJK{%
4900 \bxjx@gcc@CJKfalse
4901 \bxjx@canta=\z@ \bxjx@gcc@CJK@list}
4902 \fi

```

以上。

```

4903 \fi\fi

```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```

4904 \nogreekasCJK

```

H.5 完了

おしまい。

```

4905 %</cjkcat>

```

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

```

4906 %<*ancpandoc>
4907 %% このファイルは日本語文字を含みます。
4908 \def\bxjsp@pkgname{bxjscjkcat}

```

\bxjsp@engine エンジンの種別。

```

4909 \let\bxjsp@engine=n
4910 \@onlypreamble\bxjsp@do
4911 \def\bxjsp@do#1#2{%
4912 \edef\bxjsp@tmpa{\string#1}%

```

```

4913 \edef\bxjsp@tmpb{\meaning#1}%
4914 \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
4915 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
4916 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
4917 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
4918 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}

```

1.2 パッケージオプション

`english` オプションが指定されている場合、`\ldots` の調整を抑制する。
 ※つまり、「グローバルの `english` オプション」が指定されている場合も抑制の対象になる。BXJS クラスの英語モードを想定しているが、それ以外の場合でも、一般的な L^AT_EX の習慣として、グローバルの `english` は「その文書の基底言語が英語である」ことを示す。

```

4919 \newif\ifbxjsp@english
4920 \DeclareOption{english}{\bxjsp@englishtrue}

```

オプション定義はおしまい。

```

4921 \ProcessOptions*

```

1.3 パッケージ読込の阻止

`\pandocSkipLoadFile` `\pandocSkipLoadFile{〈ファイル名〉}` : 特定のファイルを (`\@filewithoptions` の処理に関して) 読込済であるとマークする。

```

4922 \@onlypreamble\pandocSkipLoadFile
4923 \newcommand*\pandocSkipLoadFile[1]{%
4924   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4925 \def\bxjsp@skip@load@file@a#1#2{%
4926   \ifx#1\relax
4927     \def#1{2001/01/01}%
4928     \PackageInfo\bxjsp@pkgname
4929       {File '#2' marked as loaded\@gobble}%
4930   \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。

```

4931 \@onlypreamble\pandocSkipLoadPackage
4932 \newcommand*\pandocSkipLoadPackage[1]{%
4933   \pandocSkipLoadFile{#1.sty}}

```

1.4 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近 (2015 年版以降) の L^AT_EX ではこれで警告が出る。これを抑制する。

L^AT_EX カーネルが新しい場合は `fixltx2e` を読込済にする。


```

4934 \ifx\@IncludeInRelease\@undefined\else
4935   \pandocSkipLoadPackage{fixltx2e}
4936 \fi

```

1.5 cmap パッケージ

エンジンが (u)pL^AT_EX 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。)

```

4937 \if j\bxjisp@engine
4938   \pandocSkipLoadPackage{cmap}
4939 \fi

```

1.6 microtype パッケージ

警告が多すぎなので消す。

```

4940 \if j\bxjisp@engine \else
4941   \PassOptionsToPackage{verbose=silent}{microtype}
4942 \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。

```

4943 \if j\bxjisp@engine
4944   \pandocSkipLoadPackage{microtype}
4945   \newcommand*\UseMicrotypeSet [2] [] {}
4946 \fi

```

1.7 Unicode 文字変換対策

Pandoc で L^AT_EX 形式に書き出す場合は、元データ中の一部の Unicode 文字を「L^AT_EX の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

```

…→\ldots{}   ‘→` ’→’ “→` ”→”

```

日本語 L^AT_EX では「L^AT_EX の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「\ldots を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の `\ldots` の実装。非数式である場合は代わりに … を実行する。

※以前は「Pandoc が必ず `\ldots{}` の形で書き出す」ことを利用して後続に `{}` があるかで「元が … であるか」を判断していた。ところが、Pandoc 2.7 版で `{}` を必ずしも付けなくなったため、1.9f 版で非数式の `\ldots` を全て … に戻す動作に変更した。

```

4947 \DeclareRobustCommand{\pandocLdots}{%
4948   \relax\ifmmode \expandafter\bxjsp@org@ldots
4949   \else \expandafter\bxjsp@ja@ellipsis
4950   \fi}
4951 \def\bxjsp@ja@ellipsis{…}
4952 \let\bxjsp@org@ldots\ldots

```

\ldots の実装を置き換える。

※ Benglish オプション指定時は置き換えない。

```

4953 \AtBeginDocument{%
4954   \let\bxjsp@org@ldots\ldots
4955   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}

```

\ldots の直後の文字が非英字の場合、Pandoc は「\ldots。」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが X_YT_EX・LuaT_EX は英字と見なす（または将来その可能性がある）」文字で、特に日本語文書に現れるものについて、非英字扱いしておく。

※ Pandoc は「Unicode 7.0 で GC が Letter」な文字を英字と判定している。

```

4956 \chardef\bxjsp@cc@other=12
4957 \@onlypreamble\bxjsp@makeother@range
4958 \def\bxjsp@makeother@range#1#2{%
4959   \@tempcnta"#1\relax \@tempcntb"#2\relax
4960   \loop\ifnum\@tempcnta<\@tempcntb
4961     \catcode\@tempcnta\bxjsp@cc@other
4962     \advance\@tempcnta\@ne
4963   \repeat}
4964 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0
4965   \catcode"1F23B=\bxjsp@cc@other
4966   \bxjsp@makeother@range{9FCD}{A000}
4967   \bxjsp@makeother@range{1B002}{1B170}
4968   \bxjsp@makeother@range{2B820}{2EBF0}
4969 \fi

```

1.8 Pandola モジュール

インストール済であれば読み込む。

```

4970 \IfFileExists{bxpandola.sty}{%
4971   \RequirePackage{bxpandola}\relax
4972   \PackageInfo\bxjsp@pkgname
4973   {Pandola module is loaded\@gobble}
4974 }{}

```

1.9 完了

おしまい。

```

4975 %</ancpandoc>

```

補助パッケージ実装はここまで。

4976 %</anc>