

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”)





v2.6 [2022/03/20]

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

目次

1	はじめに	3
2	オプション	11
3	和文フォントの変更	40
4	フォントサイズ	41
5	レイアウト	47
5.1	ページレイアウト	48
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	63
7	ページスタイル	64
8	文書のマークアップ	67
8.1	表題	67
8.2	章・節	72
8.3	リスト環境	85
8.4	パラメータの設定	92
8.5	フロート	94
8.6	キャプション	95
9	フォントコマンド	96

10	相互参照	99
10.1	目次の類	99
10.2	参考文献	104
10.3	索引	106
10.4	脚注	107
11	段落の頭へのグルー挿入禁止	110
12	いろいろなロゴ	114
13	amsmath との衝突の回避	114
14	初期設定	115
付録 A	和文ドライバの仕様 	119
付録 B	和文ドライバ : minimal 	120
B.1	補助マクロ	120
B.2	(u)pTeX 用の設定	122
B.3	pdfTeX 用の処理	127
B.4	X _Y TeX 用の処理	128
B.5	後処理 (エンジン共通)	129
付録 C	和文ドライバ : standard 	132
C.1	準備	132
C.2	和文ドライバパラメタ	132
C.3	共通処理 (1)	133
C.4	pTeX 用設定	141
C.5	pdfTeX 用設定 : CJK + bxcjkatype	145
C.6	X _Y TeX 用設定 : xeCJK + zxjatype	147
C.7	LuaTeX 用設定 : LuaTeX-ja	149
C.8	共通処理 (2)	153
付録 D	和文ドライバ : modern 	154
D.1	フォント設定	154
D.2	fixltx2e 読込	154
D.3	和文カテゴリコード	155
D.4	完了	155
付録 E	和文ドライバ : pandoc 	155
E.1	準備	155
E.2	和文ドライバパラメタ	156
E.3	dupload システム	157

E.4	lang 変数	158
E.5	geometry 変数	161
E.6	CJKmainfont 変数	161
E.7	Option clash 対策	161
E.8	レイアウト上書き禁止	162
E.9	paragraph のマーク	163
E.10	全角空白文字	163
E.11	hyperref 対策	164
E.12	Pandoc 要素に対する和文用の補正	164
E.13	ifPDFTeX スイッチ	165
E.14	完了	166
付録 F	補助パッケージ一覧 	167
付録 G	補助パッケージ : bxjscompat 	167
G.1	準備	167
G.2	X _q TeX 部分	167
G.3	LuaTeX 部分	168
G.4	完了	169
付録 H	補助パッケージ : bxjscjkat 	170
H.1	準備	170
H.2	和文カテゴリコードの設定	171
H.3	ギリシャ・キリル文字の扱い	171
H.4	初期設定	179
H.5	完了	179
付録 I	補助パッケージ : bxjspandoc 	179
I.1	準備	179
I.2	パッケージオプション	180
I.3	パッケージ読込の阻止	180
I.4	fixltx2e パッケージ	180
I.5	cmap パッケージ	181
I.6	microtype パッケージ	181
I.7	Unicode 文字変換対策	181
I.8	PandoLa モジュール	182
I.9	完了	183

1 はじめに

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

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

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「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_EX 3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいてもともと奥村晴彦により作成されたものです。現在は日本語 T_EX 開発コミュニティにより GitHub で管理されています。

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

[2002-12-19] いろいろなものに収録していただく際にライセンスを明確にする必要が生じてきました。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。

minijs は、jsclasses に似た設定を行うパッケージです。

```
1 %<*minijs>
2 %% if jsclasses loaded, abort loading this package
3 \ifx\@jsc@uplatextrue\@undefined\else
4   \PackageInfo{minijs}{jsclasses does not need minijs, exiting}
5   \expandafter\endinput
6 \fi
7 %% "fake" jsarticle
```

```

8 \expandafter\def\csname ver@jsarticle.cls\endcsname{}
9 %</minijs>

\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。
10 %<*class>
11 %% このファイルは日本語文字を含みます。
12 %<article>\def\bxjs@clsname{bxjsarticle}
13 %<book>\def\bxjs@clsname{bxjsbook}
14 %<report>\def\bxjs@clsname{bxjsreport}
15 %<slide>\def\bxjs@clsname{bxjsslide}

\ifjsc@needsp@tch [2016-08-22] 従来 jsclasses では、pLATEX や LATEX 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 pLATEX 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 pLATEX 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。
16 \newif\ifjsc@needsp@tch
17 \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。

```

18 \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。
19 \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。
20 \IfFileExists{bxjscompat.sty}{%
21   \let\jsAtEndOfClass\@gobble
22   \RequirePackage{bxjscompat}%
23 }{}

```

`\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`。

```

24 \let\jsArticle=a
25 \let\jsBook=b
26 \let\jsReport=r
27 \let\jsSlide=s
28 %<article>\let\jsDocClass\jsArticle
29 %<book>\let\jsDocClass\jsBook
30 %<report>\let\jsDocClass\jsReport
31 %<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` = pT_EX 系、`x` = X_TL_AT_EX、`p` =

pdf \TeX (含 DVI モード)、1 = Lua \TeX 、J = NTT j \TeX 、0 = Omega 系、n = 以上の何れでもない。

```

32 \let\jsEngine=n
33 \def\bxjs@test@engine#1#2{%
34   \edef\bxjs@tmpa{\string#1}%
35   \edef\bxjs@tmpb{\meaning#1}%
36   \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
37 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
38 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
39 \bxjs@test@engine\Omegaversion{\let\jsEngine=0}
40 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
41 \bxjs@test@engine\pdftexversion{\let\jsEngine=p}
42 \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) up \TeX であるか。

```

43 \newif\ifjsWithupTeX
44 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
45   \jsWithupTeXtrue
46 \fi\fi
47 \let\if@jsc@uplatex\ifjsWithupTeX

```

`\ifjsWithpTeXng` [スイッチ] エンジンが p \TeX -ng であるか。

```

48 \newif\ifjsWithpTeXng
49 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}

```

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

```

50 \newif\ifjsWitheTeX
51 \bxjs@test@engine\eTeXversion{\jsWitheTeXtrue}

```

非サポートのエンジンの場合は強制終了させる。

※ NTT j \TeX と Omega 系。

```

52 \let\bxjs@tmpa\relax
53 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
54 \ifx O\jsEngine \def\bxjs@tmpa{Omega}\fi
55 \ifx\bxjs@tmpa\relax \expandafter\@gobble
56 \else
57   \ClassError\bxjs@clsname
58   {The engine in use (\bxjs@tmpa) is not supported}
59   {It's a fatal error. I'll quit right now.}
60   \expandafter\@firstofone
61 \fi{\endinput\@@end}

```

Lua \TeX の場合、本クラス用の Lua モジュールを用意する。

```

62 \ifx l\jsEngine
63   \directlua{ bxjs = {} }
64 \fi

```

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

```

65 \ifjsWitheTeX \let\bxjs@protected\protected
66 \else \let\bxjs@protected\@empty
67 \fi

```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

```

68 \ifjsWitheTeX
69 \def\bxjs@robust@def{\protected\def}
70 \else
71 \def\bxjs@robust@def{\DeclareRobustCommand*}
72 \fi

```

`\ifjsInPdfMode` [スイッチ] pdf_TE_X / Lua_TE_X が PDF モードで動作しているか。
 ※ Lua_TE_X 0.8x 版でのプリミティブ名変更に対応。

```

73 \newif\ifjsInPdfMode
74 \@nameuse{ImposeOldLuaTeXBehavior}
75 \let\bxjs@tmpa\PackageWarningNoLine
76 \let\PackageWarningNoLine\PackageInfo % suppress warning
77 \RequirePackage{ifpdf}
78 \let\PackageWarningNoLine\bxjs@tmpa
79 \@nameuse{RevokeOldLuaTeXBehavior}
80 \let\ifjsInPdfMode\ifpdf

```

`\ifbxjs@explIII` [スイッチ] expl3 がカーネルに組み込まれているか。

```

81 \newif\ifbxjs@explIII
82 \ifl@t@r\fmtversion{2020/02/02}{\bxjs@explIIItrue}{}

```

`\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_L_AT_EX または Lua_TE_X 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。

```

83 \newif\ifbxjs@TUenc
84 \def\bxjs@tmpa{TU}\edef\bxjs@tmpb{\f@encoding}
85 \ifx\bxjs@tmpa\bxjs@tmpb
86 \bxjs@TUenctrue
87 \fi

```

`\ifbxjs@old@hook@system` [スイッチ] L_AT_EX の新しいフック管理システムが未導入であるか。

※カーネルの 2020/10/01 版で導入された。

```

88 \newif\ifbxjs@old@hook@system
89 \ifl@t@r\fmtversion{2020/10/01}{\bxjs@old@hook@systemtrue}{}

```

`\bxjs@CGHN` カーネルの 2021/11/15 版の改修で“要素の順が変わった”フック名について、新仕様の名前を“使用中の L_AT_EX において正しい名前”に変換する。

※例えば、“package/after/PKG”は新仕様で“package/PKG/after”に変更された。

```

90 \ifl@t@r\fmtversion{2021/11/15}{%
91 \def\bxjs@CGHN#1{#1}%

```

```

92 }{%else
93   \def\bxjs@CGHN#1{\bxjs@CGHN@a#1//}%
94   \def\bxjs@CGHN@a#1/#2/#3//{#1/#3/#2}}

\bxjs@cond \bxjs@cond\ifXXX……\fi{⟨真⟩}{⟨偽⟩}
  TEX の if-文 ( \ifXXX……⟨真⟩\else⟨偽⟩\fi ) を末尾呼出形式に変換するためのマクロ。
95 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
96   #1\expandafter\@firstoftwo
97   \else\expandafter\@secondoftwo
98   \fi}

```

```

\bxjs@cslet \bxjs@cslet{⟨名前 1⟩}\制御綴 :
99 \def\bxjs@cslet#1{%
100   \expandafter\let\csname#1\endcsname}

```

```

\bxjs@csletcs \bxjs@csletcs{⟨名前 1⟩}{⟨名前 2⟩} :
101 \def\bxjs@csletcs#1#2{%
102   \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。
103 \def\bxjs@catopt#1#2{%
104   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

```

```

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

```

```

\bxjs@trim \bxjs@trim\CS で、\CS の内容のトークン列を先頭と末尾の空白トークン群を除去したも
のに置き換える。
106 \def\bxjs@trim#1{\expandafter\bxjs@trim@a#1\@nil#1}
107 \def\bxjs@trim@a{\futurelet\bxjs@tmpb\bxjs@trim@b}
108 \def\bxjs@trim@b{\bxjs@cond\ifx\bxjs@tmpb\@sptoken\fi
109   {\bxjs@trim@c\bxjs@trim@a}{\bxjs@trim@d}}
110 \def\bxjs@trim@c#1 {#1}
111 \def\bxjs@trim@d#1\@nil{\bxjs@trim@e#1\@nil: \@nil\@nnil}
112 \def\bxjs@trim@e#1 \@nil#2\@nnil{\bxjs@cond\ifx\@nnil#2\@nnil\fi
113   {\bxjs@trim@f#1\@nnil}{\bxjs@trim@e#1\@nil: \@nil\@nnil}}
114 \def\bxjs@trim@f#1\@nil#2\@nnil#3{\def#3{#1}}

```

```

\bxjs@set@array@from@clist \bxjs@set@array@from@clist{⟨配列名接頭辞⟩}{⟨コンマ区切りリスト⟩} : コンマ区切
りの値のリストから擬似配列を生成する。

```

※各要素について、先頭・末尾の空白トークン群は除去される。

```

115 \def\bxjs@set@array@from@clist#1#2{%
116   \@tempcnta\z@
117   \@for\bxjs@tmpa:=\@empty#2\do{%
118     \bxjs@trim\bxjs@tmpa \bxjs@cslet{#1/\the\@tempcnta}\bxjs@tmpa
119     \advance\@tempcnta\@ne}
120   \bxjs@cslet{#1/\the\@tempcnta}\relax}

```


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

```
121 \let\c@bxjs@tempcnta\@tempcnta
122 \def\bxjs@gset@tempcnta{\setcounter{bxjs@tempcnta}}
```

`\jsSetQHLLength` `\jsSetQHLLength\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）。

```
123 \def\jsSetQHLLength#1#2{%
124   \begingroup
125     \bxjs@parse@qh{#2}%
126     \ifx\bxjs@tmpb\relax
127       \setlength\@tempdima{#2}%
128       \xdef\bxjs@g@tmpa{\the\@tempdima}%
129     \else \global\let\bxjs@g@tmpa\bxjs@tmpb
130     \fi
131   \endgroup
132   #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。

```
133 \if j\jsEngine \def\bxjs@parse@qh@units{zw,zh}
134 \else \def\bxjs@parse@qh@units{trueQ,trueH,Q,H,zw,zh}
135 \fi
136 \def\bxjs@parse@qh#1{%
137   \let\bxjs@tmpb\relax
138   \for\bxjs@tmpa:=\bxjs@parse@qh@units\do{%
139     \ifx\bxjs@tmpb\relax
140       \edef\bxjs@next{{\bxjs@tmpa}{#1}}%
141       \expandafter\bxjs@parse@qh@a\csname bxjs@unit@\bxjs@tmpa\expandafter
142         \endcsname\bxjs@next
143     \fi}}
144 \def\bxjs@parse@qh@a#1#2#3{%
145   \def\bxjs@next##1#2\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}#1}%
146   \bxjs@next#3\@nil#2\@nil\@nnil}
147 \def\bxjs@parse@qh@b#1#2#3{%
148   \ifx\@nnil#2\@nnil\else
149     \ifx#3\relax
150       \ClassError\bxjs@clsname
151       {You cannot use '\bxjs@tmpa' here}{\@ehc}%
152     \def\bxjs@tmpb{0pt}%
153   \else
154     \@tempdimb#3\relax \@tempdimb#1\@tempdimb
155     \edef\bxjs@tmpb{\the\@tempdimb}%
156   \fi
```

```
157 \fi}
```

今の段階では Q/H だけが使用可能。

```
158 \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。

```
159 \@onlypreamble\bxjs@begin@document@hook
```

```
160 \let\bxjs@begin@document@hook\@empty
```

```
161 \AtBeginDocument{\bxjs@begin@document@hook}
```

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

```
162 \@onlypreamble\bxjs@post@option@hook
```

```
163 \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。

```
164 \@onlypreamble\bxjs@pre@jadriver@hook
```

```
165 \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。)

```
166 \def\jsAtEndOfClass{%
```

```
167 \expandafter\g@addto\macro\csname\bxjs@clsname.cls-h@k\endcsname}
```

一時的な手続き用の制御綴。

```
168 \@onlypreamble\bxjs@tmpdo
```

```
169 \@onlypreamble\bxjs@tmpdo@a
```

```
170 \@onlypreamble\bxjs@tmpdo@b
```

```
171 \@onlypreamble\bxjs@tmpdo@c
```

```
172 \@onlypreamble\bxjs@tmpdo@d
```

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

※現在の Lua_{LaTeX} 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。

```
173 \if 1\jsEngine
```

```
174 \def\bxjs@tmpdo#1{%
```

```
175 \xdef\bxjs@pre@jadriver@hook{%
```

```
176 \bxjs@pre@jadriver@hook
```

```
177 \catcode`#1=\the\catcode`#1\relax}%
```

```
178 \catcode`#1=11\relax}
```

```
179 \@tfor\bxjs@tmpa:=和西暦\do
```

```
180 {\expandafter\bxjs@tmpdo\bxjs@tmpa}
```

```
181 \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。

```
182 \bxjs@robust\def\jsInhibitGlue{%
```

```
183 \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。

```
184 \if@compatibility
185   \ClassError\bxjs@clsname
186   {Something went chaotic!\MessageBreak
187    (How come '\string\documentstyle' is there?)\MessageBreak
188    I cannot go a single step further...}
189   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
190    then there'll still be hope....}
191   \expandafter\@firstofone
192 \else \expandafter\@gobble
193 \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。
194 \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。
195 \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。
196 %<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。
197 %<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。


---


198 %<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。


---


199 \newif\if@enablejfam \@enablejfamfalse

                以下で各オプションを宣言します。
```

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

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

デフォルトは `a4paper` です。

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

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

```

200 \onlypreamble\bxjs@setpaper
201 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
202 \newif\ifbxjs@iso@bsize
203 \DeclareOption{iso-bsize}{\bxjs@iso@bsizetrue}
204 \onlypreamble\bxjs@setpaper@bsize
205 \def\bxjs@setpaper@bsize#1{\edef\bxjs@param@paper{%
206   b#1\ifbxjs@iso@bsize paper\else jfi}}
207 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
208 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
209 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
210 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
211 \DeclareOption{b4paper}{\bxjs@setpaper@bsize{4}}
212 \DeclareOption{b5paper}{\bxjs@setpaper@bsize{5}}
213 \DeclareOption{b6paper}{\bxjs@setpaper@bsize{6}}
214 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
215 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
216 \DeclareOption{b4j}{\bxjs@setpaper{b4j}}
217 \DeclareOption{b5j}{\bxjs@setpaper{b5j}}
218 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
219 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
220 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
221 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
222 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

geometry の用紙サイズのオプション名を全てサポートする。
223 \@for\bxjs@tmpa:={%
224   a0,a1,a2,c0,c1,c2,c3,c4,c5,c6,ansia,ansib,ansic,ansid,ansie%
225 }\do{\edef\bxjs@next{%
226   \noexpand\DeclareOption{\bxjs@tmpa paper}%
227   {\noexpand\bxjs@setpaper{\bxjs@tmpa paper}}%
228 }\bxjs@next}

```

```
229 \DeclareOption{screen}{\bxjs@setpaper{screen}}
```

ただし `b?paper` は `iso-bsize` の指定に従い ISO と JIS の適切な方の B 列を選択する。

```
230 \@for\bxjs@tmpa:={0,1,2,3}\do{\edef\bxjs@next{%
231   \noexpand\DeclareOption{b\bxjs@tmpa paper}%
232   {\noexpand\bxjs@setpaper@bsize{\bxjs@tmpa}}%
233 }\bxjs@next}
```

Pandoc では用紙サイズ指定について「後ろに `paper` を付けた名前のオプション」を指定する。これに対処するため、後ろに `paper` をつけた形を用意する。さらに、「用紙サイズを `custom` とすると何も設定しない」ようにするため `custompaper` というオプションを用意する。

```
234 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
235 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
236 \DeclareOption{screenpaper}{\bxjs@setpaper{screen}}
237 \DeclareOption{custompaper}{}
```

■横置き 用紙の縦と横の長さを入れ換えます。

```
238 \newif\if@landscape
239 \@landscapefalse
240 \DeclareOption{landscape}{\@landscapetrue}
```

■slide オプション `slide` を新設しました。

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

```
241 \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。

```
242 %<!slide>\@slidefalse
243 %<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 へ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。

```
244 \newcommand{\@ptsize}{0}
245 \def\bxjs@param@basefontsize{10pt}
246 \def\jsUnusualPtSize{-20}
```

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

```
247 \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。

```
248 \jsSetQHLlength\@tempdima{#1}%
249 \edef\bxjs@param@basefontsize{\the\@tempdima}%
250 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
251 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
252 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
253 \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi
```

TODO: 恐らく 14pt と `base=14.4pt` 等の関係も全く等価であるべき。

```
254 \def\bxjs@setjbasefontsize#1{%
255 \setkeys{bxjs}{jbase=#1}}
```

`\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。

```
256 \newif\ifjsc@mag
257 \newif\ifjsc@mag@xreal
258 %\let\jsc@magscale\@undefined

259 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
260 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
261 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
262 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
263 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
264 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
265 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
266 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
267 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
268 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
269 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
270 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
271 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
272 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
```

```

273 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
274 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
275 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
276 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
277 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

278 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@@usemag}
279 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@@nomag}
280 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@@xreal}

```

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

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

```

281 \if j\jsEngine
282 \hour\time \divide\hour by 60\relax
283 \@tempcnta\hour \multiply\@tempcnta 60\relax
284 \minute\time \advance\minute-\@tempcnta
285 \DeclareOption{tombow}{%
286   \tombowtrue \tombowdatetrue
287   \setlength{\@tombowwidth}{.1\p@}%
288   \@bannertoken{%
289     \jobname\space(\number\year-\two@digits\month-\two@digits\day
290     \space\two@digits\hour:\two@digits\minute)}}%
291   \maketombowbox}
292 \DeclareOption{tombo}{%
293   \tombowtrue \tombowdatefalse
294   \setlength{\@tombowwidth}{.1\p@}%
295   \maketombowbox}
296 \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。

```

297 \if j\jsEngine
298 \DeclareOption{mentuke}{%
299   \tombowtrue \tombowdatefalse
300   \setlength{\@tombowwidth}{\z@}%
301   \maketombowbox}
302 \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。

```
303 \DeclareOption{oneside}{\@twosidefalse \mparswitchfalse}
304 \DeclareOption{twoside}{\@twosidetrue \mparswitchtrue}
305 \DeclareOption{vartwoside}{\@twosidetrue \mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
306 \DeclareOption{onecolumn}{\@twocolumnfalse}
307 \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。

```
308 \DeclareOption{titlepage}{\@titlepagetrue}
309 \DeclareOption{notitlepage}{\@titlepagefalse}
```

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

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

```
310 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
311 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
312 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}
```

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

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

```
313 \def\eqnarray{%
314   \stepcounter{equation}%
315   \def\@currentlabel{\p@equation\theequation}%
316   \global\@eqnswtrue
317   \m@th
318   \global\@eqcnt\z@
319   \tabskip\@centering
320   \let\\\@eqnocr
321   $$\everycr{}\halign to\displaywidth\bgroup
322     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
323     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
324     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
325     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
326     \tabskip\z@skip
327   \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。

```
328 \DeclareOption{leqno}{\input{leqno.clo}}
329 \DeclareOption{fleqn}{\input{fleqn.clo}}%
330 % fleqn 用の eqnarray 環境の再定義
331 \def\eqnarray{%
332   \stepcounter{equation}%
333   \def\@currentlabel{\p@equation\theequation}%
334   \global\@eqnswtrue\m@th
335   \global\@eqcnt\z@
336   \tabskip\mathindent
337   \let\@=\@eqnocr
338   \setlength\abovedisplayskip{\topsep}%
339   \ifvmode
340     \addtolength\abovedisplayskip{\partopsep}%
341   \fi
342   \addtolength\abovedisplayskip{\parskip}%
343   \setlength\belowdisplayskip{\abovedisplayskip}%
344   \setlength\belowdisplayshortskip{\abovedisplayskip}%
345   \setlength\abovedisplayshortskip{\abovedisplayskip}%
346   $$\everycr{}\halign to\linewidth% $$
347   \bgroup
348   \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
349   &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
350   &\global\@eqcnt\tw@
351   $\displaystyle{##}$\hfil \tabskip\@centering
352   &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
353   \tabskip\z@skip\cr
354   }}
```

■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。

```
355 % \DeclareOption{openbib}{%
356 %   \AtEndOfPackage{%
357 %     \renewcommand\@openbib@code{%
358 %       \advance\leftmargin\bibindent
359 %       \itemindent -\bibindent
360 %       \listparindent \itemindent
361 %       \parsep \z@}%
362 %     \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：

```
363 %\let\bxjs@enablejfam\undefined
```

`enablejfam` オプションの処理。

```
364 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
365 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
366 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
367 \define@key{bxjs}{enablejfam}[true]{%
368   \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。

```
369 \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` `draft` オプションが指定されているか。

※ JS クラスの `\ifdraft` が廃止されたので、BXJS クラスでも `\ifdraft` を 2.0 版で廃止した。

```
370 \newif\ifjsDraft
371 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
372 \DeclareOption{final}{\jsDraftfalse \overfullrule=0pt }
```

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

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

BXJS クラスではここは和文ドライバの管轄。

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

BXJS クラスでは geometry パッケージがこの処理を行う。

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

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

```
373 \newif\ifbxjs@papersize
374 \bxjs@papersizetrue
375 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
376 \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。

```
377 \newif\if@english
378 \@englishfalse
379 \DeclareOption{english}{\@englishttrue}
```

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


[2017-02-13] 従来は「jsreport 相当」を jsbook の report オプションで提供していましたが、新しく jsreport クラスも作りました。どちらでも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。

```
380 \newif\if@jslogo \@jslogofalse
381 \DeclareOption{jslogo}{\@jslogotrue}
382 \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。

```
383 \mathchardef\bxjs@isc@ll=128
384 \mathchardef\bxjs@isc@sl=259
385 \def\bxjs@isc@sl@h{65539 }
386 \def\bxjs@invscale#1#2{%
387   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
388   \ifdim\@tempdima<\bxjs@isc@ll\p@
389     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
390     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
391   \else
392     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
393     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
394   \fi
395   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
396   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
397   \@tempdimb\@tempcnta\@ne
398   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
399   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
400   \@whiledim\@tempdimb<\@tempdimc\do{%
401     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
402     \advance\@tempcntb\@ne \divide\@tempcntb\tw@
403     \ifdim #2\@tempcntb>\@tempdima
404       \advance\@tempcntb\@ne \@tempdimc=\@tempcntb\@ne
405     \else \@tempdimb=\@tempcntb\@ne \fi}%
406   \xdef\bxjs@gtmpa{\the\@tempdimb}%
407   \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 \LaTeX 用の既定テンプレートを用いて他形式から \LaTeX (および PDF) 形式に変換する用途に最適化した設定を与える。

```
408 \DeclareOption{pandoc}{%
409   \bxjs@apply@pandoc@opt}
410 \@onlypreamble\bxjs@apply@pandoc@opt
411 \def\bxjs@apply@pandoc@opt{%
```

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。

```
412 \g@addto@macro\bxjs@post@option@hook{%
413   \bxjs@oldfontcommandstrue
414   \setkeys{bxjs}{ja=pandoc}%
415   \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）。

```
416 \ifx\bxjs@driver@opt\@undefined
417   \def\bxjs@driver@opt{dvipdfmx}%
418   \bxjs@dvi@opttrue
419 \fi
420 \global\let\bxjs@apply@pandoc@opt\relax
```

`pandoc+` オプションは、`pandoc` と同じ設定をした上で、さらに和文パラメタの先頭に `_plus` を追加する。

```
421 \DeclareOption{pandoc+}{%
422   \g@addto@macro\bxjs@post@option@hook{%
423     \edef\jsJaParam{\bxjs@catopt{+_plus}\jsJaParam}}%
424 \ExecuteOptions{pandoc}}
```

■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。

```
425 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
426 %\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。

```
427 \DeclareOption{autodetect-engine}{%
428   \let\bxjs@engine@given=*}
429 \DeclareOption{latex}{%
430   \def\bxjs@engine@opt{latex}%
431   \let\bxjs@engine@given=n}
432 \DeclareOption{platex}{%
433   \def\bxjs@engine@opt{platex}%
434   \let\bxjs@engine@given=j}
435 \DeclareOption{uplatex}{%
436   \def\bxjs@engine@opt{uplatex}%
437   \let\bxjs@engine@given=u}
438 \DeclareOption{xelatex}{%
439   \def\bxjs@engine@opt{xelatex}%
440   \let\bxjs@engine@given=x}
441 \DeclareOption{pdflatex}{%
442   \def\bxjs@engine@opt{pdflatex}%
443   \let\bxjs@engine@given=p}
444 \DeclareOption{lualatex}{%
445   \def\bxjs@engine@opt{lualatex}%
446   \let\bxjs@engine@given=l}
```

```

447 \DeclareOption{platex-ng}{%
448   \def\bxjs@engine@opt{platex-ng}%
449   \let\bxjs@engine@given=g}
450 \DeclareOption{platex-ng*}{%
451   \def\bxjs@engine@opt{platex-ng*}%
452   \let\bxjs@platexng@nodrv=t%
453   \let\bxjs@engine@given=g}

```

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

```

454 %\let\bxjs@driver@given\@undefined
455 \let\bxjs@driver@@dvimode=0
456 \let\bxjs@driver@@dvi pdfmx=1
457 \let\bxjs@driver@@pdfmode=2
458 \let\bxjs@driver@@xetex=3
459 \let\bxjs@driver@@dvips=4
460 \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。

```

461 %\let\bxjs@driver@opt\@undefined

462 \DeclareOption{dvips}{%
463   \def\bxjs@driver@opt{dvips}%
464   \let\bxjs@driver@given\bxjs@driver@@dvips}
465 \DeclareOption{dviout}{%
466   \def\bxjs@driver@opt{dviout}%
467   \let\bxjs@driver@given\bxjs@driver@@dvimode}
468 \DeclareOption{xdvi}{%
469   \def\bxjs@driver@opt{xdvi}%
470   \let\bxjs@driver@given\bxjs@driver@@dvimode}
471 \DeclareOption{dvi pdfmx}{%
472   \def\bxjs@driver@opt{dvi pdfmx}%
473   \let\bxjs@driver@given\bxjs@driver@@dvi pdfmx}
474 \DeclareOption{nodvidriver}{%
475   \def\bxjs@driver@opt{nodvidriver}%
476   \let\bxjs@driver@given\bxjs@driver@@none}
477 \DeclareOption{pdftex}{%
478   \def\bxjs@driver@opt{pdftex}%
479   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
480 \DeclareOption{luatex}{%
481   \def\bxjs@driver@opt{luatex}%
482   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
483 \DeclareOption{xetex}{%
484   \def\bxjs@driver@opt{xetex}%
485   \let\bxjs@driver@given\bxjs@driver@@xetex}

dvipdfmx-if-dvi は 2.0 版より非推奨となった。
486 \DeclareOption{dvi pdfmx-if-dvi}{\bxjs@depreopt@do{dvi pdfmx-if-dvi}{dvi=dvi pdfmx}}

```

■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?)。

`\bxjs@depre@opt` 非推奨のオプションについて警告を出す。

```
\bxjs@depre@opt@do 487 \@onlypreamble\bxjs@depre@opt
488 \def\bxjs@depre@opt#1#2{%
489   \ClassWarningNoLine\bxjs@clsname
490     {The old option '#1' is DEPRECATED\MessageBreak
491       and may be abolished in future!\MessageBreak
492       You should instead write:\MessageBreak
493       \space\space #2}}
494 \@onlypreamble\bxjs@depre@opt@do
495 \def\bxjs@depre@opt@do#1#2{%
496   \bxjs@depre@opt{#1}{#2}%
497   \setkeys{bxjs}{#2}}
```

`\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、既定を真とした上で、オプションで指定することとする。

※ 2.0 版より、既定値を常に真とする。

```
498 \newif\ifbxjs@bigcode \bxjs@bigcodetrue
```

`nobigcode` / `bigcode` オプションの定義。

```
499 \DeclareOption{nobigcode}{%
500   \bxjs@bigcodefalse}
501 \DeclareOption{bigcode}{%
502   \bxjs@bigcodetrue}
```

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

```
503 \newif\ifbxjs@oldfontcommands
```

`nooldfontcommands`、`oldfontcommands` オプションの定義。

※`oldfontcommands` オプションの名前は memoir クラスに倣った。ちなみに KOMA-Script では `enabledeprecatedfontcommands` であるがこれはチョットアレなので避けた。

```
504 \DeclareOption{nooldfontcommands}{%
505   \bxjs@oldfontcommandsfalse}
506 \DeclareOption{oldfontcommands}{%
507   \bxjs@oldfontcommandstrue}
```

■JS クラスのオプションで無効なもの  `ltjsclasses` に倣って警告を出す。

```
508 \DeclareOption{winjis}{%
509   \ClassWarningNoLine\bxjs@clsname
510     {This class does not support `winjis' option}%
```

```

511 }
512 \DeclareOption{mingoth}{%
513   \ClassWarningNoLine{bxjs@clsname}
514   {This class does not support `mingoth' option}%
515 }
516 \DeclareOption{jis}{%
517   \ClassWarningNoLine{bxjs@clsname}
518   {This class does not support `jis' option}%
519 }

```

■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。

```

520 \DeclareOption*{%
521   \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
522   \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。

```

523 \def\bxjs@safe@setkeys#1#2{%
524   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
525   \setkeys{#1}{#2}%
526   \let\KV@errx\bxjs@save@KV@errx}

```

`\bxjs@declare@enum@option` `\bxjs@declare@enum@option{<オプション名>}{<enum 名>}{<初期値>}`

“<オプション名>=<値>” のオプション指定に対して、`\[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。

```

527 \@onlypreamble\bxjs@declare@enum@option
528 \def\bxjs@declare@enum@option#1#2#3{%
529   \bxjs@csletcs{bxjs@#2}{bxjs@#2@#3}%
530   \define@key{bxjs}{#1}{%
531     \expandafter\ifx\csname bxjs@#2@#1\endcsname\relax
532     \bxjs@error@keyval{#1}{##1}%
533     \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@#1}%
534     \fi}}

```

`\bxjs@declare@bool@option` `\bxjs@declare@bool@option{<オプション名>}{<スイッチ名>}{<初期値>}`

“<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。

```

535 \@onlypreamble\bxjs@declare@bool@option
536 \def\bxjs@declare@bool@option#1#2#3{%
537   \csname newif\expandafter\endcsname\csname ifbxjs@#2\endcsname
538   \@nameuse{bxjs@#2#3}%
539   \define@key{bxjs}{#1}[true]{%
540     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
541     \bxjs@error@keyval{#1}{##1}%
542     \else \@nameuse{bxjs@#2##1}%

```



```

543     \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。
544 \def\bxjs@set@keyval#1#2#3{%
545     \bxjs@csletcs{bxjs@next}{bxjs@kv@#1@#2}%
546     \ifx\bxjs@next\relax
547         \bxjs@error@keyval{#1}{#2}%
548         #3%
549     \else \bxjs@next
550     \fi}
551 \@onlypreamble\bxjs@error@keyval
552 \def\bxjs@error@keyval#1#2{%
553     \ClassError\bxjs@clsname
554     {Invalid value '#2' for option #1}\@ehc}

\jsScale 〔実数値マクロ〕 和文スケール値。
555 \def\jsScale{0.924715}

\bxjs@base@opt 明示された base オプションの値。
556 %\let\bxjs@base@opt\@undefined

    base オプションの処理。
557 \define@key{bxjs}{base}{%
558     \edef\bxjs@base@opt{#1}%
559     \bxjs@setbasefontsize{#1}}
560 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

\bxjs@jbase@opt 明示された jbase オプションの値。
561 %\let\bxjs@jbase@opt\@undefined

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

\bxjs@scale@opt 明示された scale オプションの値。
564 %\let\bxjs@scale@opt\@undefined

    scale オプションの処理。
565 \define@key{bxjs}{scale}{%
566     \edef\bxjs@scale@opt{#1}%
567     \let\jsScale\bxjs@scale@opt}
568 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

    noscale オプションの処理。
569 \DeclareOption{noscale}{\bxjs@depre@opt@do{noscale}{scale=1}}

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

```

mag オプションの処理。

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

paper オプションの処理。

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

\bxjs@jadriver 和文ドライバの名前。

```
573 \let\bxjs@jadriver\relax
574 %\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)。

```
575 \define@key{bxjs}{jadriver}{%
576   \bxjs@depre@opt{jadriver}{ja=#1}\edef\bxjs@jadriver@opt{#1}}
577 \define@key{bxjs}{ja}[\relax]{%
578   \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}
```

\jsJaFont 和文フォント設定の名前。

```
579 \let\jsJaFont\@empty
```

jafont オプションの処理。

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

\jsJaParam 和文ドライバパラメタの文字列。

```
581 \let\jsJaParam\@empty
```

japaram オプションの処理。

```
582 \define@key{bxjs}{japaram}{%
583   \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}
```

引数をもつ pandoc・pandoc+ オプションは、その引数を和文パラメタの指定と見なす。

```
584 \define@key{bxjs}{pandoc}[]{%
585   \ExecuteOptions{pandoc}%
586   \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}%
587 \define@key{bxjs}{pandoc+}[]{%
588   \ExecuteOptions{pandoc+}%
589   \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}%
```

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

```
590 \let\bxjs@magstyle@@mag=m
591 \let\bxjs@magstyle@@real=r
592 \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。

```
593 \let\bxjs@magstyle@@usemag\bxjs@magstyle@@mag
594 \let\bxjs@magstyle@@nomag\bxjs@magstyle@@real
595 \bxjs@cslet{bxjs@magstyle@@nomag*}\bxjs@magstyle@@xreal
```

`\bxjs@magstyle@@default` は既定の値を表す。

```
596 \let\bxjs@magstyle@@default\bxjs@magstyle@@usemag
597 \ifx l\jsEngine \ifnum\luatexversion>86
598   \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
599 \fi\fi
600 \ifjsWithpTeXng
601   \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
602 \fi
603 \let\bxjs@magstyle\bxjs@magstyle@@default
```

`magstyle` オプションの処理。

```
604 \define@key{bxjs}{magstyle}{%
605   \bxjs@csletcs{bxjs@magstyle}{bxjs@magstyle@@#1}%
606   \ifx\bxjs@magstyle\relax
607     \bxjs@error@keyval{magstyle}{#1}%
608     \let\bxjs@magstyle\bxjs@magstyle@@default
609   \fi}
```

`\bxjs@geometry geometry` オプションの指定値。

```
610 \let\bxjs@geometry@@class=c
611 \let\bxjs@geometry@@user=u
612 \bxjs@declare@enum@option{geometry}{geometry}{class}
```

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

```
613 \bxjs@declare@bool@option{fancyhdr}{fancyhdr}{true}
```

`\ifbxjs@dvi@opt` `dvi` オプションが指定されたか。

```
614 \newif\ifbxjs@dvi@opt
```

DVI モードのドライバとドライバ種別との対応。

```
615 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
616 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
617 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
618 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
619 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none
```

`dvi` オプションの処理。

```
620 \define@key{bxjs}{dvi}{%
621   \bxjs@csletcs{bxjs@tmpa}{bxjs@dvidriver@@#1}%
622   \ifx\bxjs@tmpa\relax
623     \bxjs@error@keyval{dvi}{#1}%
624   \else
```

`\bxjs@driver@given` を未定義にしていることに注意。

```
625   \def\bxjs@driver@opt{#1}%
626   \let\bxjs@driver@given\undefined
627   \bxjs@dvi@opttrue
628 \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。

```
629 \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。

```
630 \newif\ifbxjs@force@chapterabstract
```

```
631 %<book>\bxjs@force@chapterabstracttrue
```

layout オプションの処理。

```
632 \@namedef{bxjs@kv@layout@v1}{%
```

```
633 %<book>\bxjs@layout@buggyhmargintrue
```

```
634 %<report>\bxjs@force@chapterabstracttrue
```

```
635 }
```

```
636 \@namedef{bxjs@kv@layout@v2}{%
```

```
637 %<book>\bxjs@layout@buggyhmarginfalse
```

```
638 %<report>\bxjs@force@chapterabstractfalse
```

```
639 }
```

```
640 \define@key{bxjs}{layout}{%
```

```
641 \bxjs@set@keyval{layout}{#1}{}}
```

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

```
642 %\let\bxjs@textwidth@limit@opt\@undefined
```

```
643 \define@key{bxjs}{textwidth-limit}{%
```

```
644 \bxjs@depre@opt{textwidth-limit}{textwidth=#1zw}%
```

```
645 \edef\bxjs@textwidth@limit@opt{#1}}
```

\bxjs@textwidth@opt textwidth の指定値。

```
646 %\let\bxjs@textwidth@opt\@undefined
```

```
647 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
```

```
648 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}
```

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

```
649 %\let\bxjs@number@of@lines@opt\@undefined
```

```
650 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
```

```
651 \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。

```
652 %\let\bxjs@paragraph@mark\@undefined
```

```
653 \define@key{bxjs}{paragraph-mark}{%
```

```
654 \edef\bxjs@paragraph@mark{#1}}
```

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

```
655 \bxjs@declare@bool@option{whole-zw-lines}{whole@zw@lines}{true}
```

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

```
656 \bxjs@declare@bool@option{jaspacer-cmd}{jaspace@cmd}{true}
```

```
657 \define@key{bxjs}{xkanjiskip-cmd}[true]{\setkeys{bxjs}{jaspace-cmd=#1}}
```

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

```
658 \bxjs@declare@bool@option{fix-at-cmd}{fix@at@cmd}{true}
```

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

```
659 \bxjs@declare@bool@option{hyperref-enc}{hyperref@enc}{true}
```

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

```
660 \chardef\bxjs@everyparhook@@none=0
661 \chardef\bxjs@everyparhook@@compat=1
662 \chardef\bxjs@everyparhook@@modern=2
663 \bxjs@declare@enum@option{everyparhook}{everyparhook}{%
664   \if j\jsEngine compat\else modern\fi}
```

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

```
665 \chardef\bxjs@label@section@@none=0
666 \chardef\bxjs@label@section@@compat=1
667 \chardef\bxjs@label@section@@modern=2
668 \bxjs@declare@enum@option{label-section}{label@section}{compat}
```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```
669 \bxjs@declare@bool@option{use-zw}{usezw}{true}
670 \DeclareOption{nozw}{\bxjs@depre@opt@do{nozw}{use-zw=false}}
671 \DeclareOption{zw}{\bxjs@depre@opt@do{zw}{use-zw=true}}
```

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

```
672 \bxjs@declare@bool@option{disguise-js}{disguise@js}{true}
673 \DeclareOption{nojs}{\bxjs@depre@opt@do{nojs}{disguise-js=false}}
674 \DeclareOption{js}{\bxjs@depre@opt@do{js}{disguise-js=true}}
```

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

```
675 \bxjs@declare@bool@option{precise-text}{precisetext}{false}
676 \DeclareOption{noprecisetext}{\bxjs@depre@opt@do{noprecisetext}{precise-
  text=false}}
677 \DeclareOption{precisetext}{\bxjs@depre@opt@do{precisetext}{precise-
  text=true}}
```

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

```
678 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}{true}
679 \DeclareOption{nosimplejasetup}{\bxjs@depre@opt@do{nosimplejasetup}{simple-ja-
  setup=false}}
680 \DeclareOption{simplejasetup}{\bxjs@depre@opt@do{simplejasetup}{simple-ja-
  setup=true}}
```

`\ifbxjs@plautopatch` [スイッチ] `plautopatch` の指定値。

```
681 \bxjs@declare@bool@option{plautopatch}{plautopatch}{false}
682 \g@addto@macro\bxjs@plautopatchfalse{\def\bxjs@plautopatch@given{false}}
```

■ 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。

```
683 \let\bxjs@org@removeelement\@removeelement
684 \def\@removeelement#1#2#3{%
685   \def\reserved@a{#2}%
686   \ifx\reserved@a\@empty \let#3\@empty
687   \else \bxjs@org@removeelement{#1}{#2}{#3}%
688   \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。

```
689 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
690 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
691 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
692 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
693 \ProcessOptions\relax
694 \bxjs@post@option@hook
```

後処理

```
695 \if@slide
696   \def\maybblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
697 \fi
698 \if@landscape
699   \setlength\@tempdima {\paperheight}
700   \setlength\paperheight{\paperwidth}
701   \setlength\paperwidth {\@tempdima}
702 \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。

```
703 \def\bxjs@tmpdo{%
704   \def\bxjs@tmpa{\gobble}%
705   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
706   \let\@classoptionslist\bxjs@tmpa}
707 \def\bxjs@tmpdo@a#1,{%
708   \ifx\@nil#1\relax\else
709     \bxjs@tmpdo@b#1{\@nil
710       \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
711       \expandafter\bxjs@tmpdo@a
712     \fi}
```

```

713 \def\bxjs@tmpdo@b#1#{\bxjs@tmpdo@c}
714 \def\bxjs@tmpdo@c#1\@nil{%
715   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
716 \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。

```

717 \@expandtwoargs\@removeelement
718   {papersize}\@classoptionslist\@classoptionslist
719 \@expandtwoargs\@removeelement
720   {10pt}\@classoptionslist\@classoptionslist
721 \@expandtwoargs\@removeelement
722   {noscale}\@classoptionslist\@classoptionslist

```

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

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

このコードを削除。

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

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

```

723 \ifx\bxjs@jadriver@opt\@undefined\else
724   \let\bxjs@jadriver\bxjs@jadriver@opt
725 \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。

```

726 \let\bxjs@tmpb\jsEngine
727 \ifx j\bxjs@tmpb\ifjsWithpTeXng
728   \let\bxjs@tmpb=g
729 \fi\fi
730 \ifx j\bxjs@tmpb\ifjsWithupTeX
731   \let\bxjs@tmpb=u
732 \fi\fi
733 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
734   \let\bxjs@tmpb=n
735 \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。)


```
736 \ifx *\bxjs@engine@given
737   \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。

```
738 \ifx j\bxjs@engine@given
739   \g@addto@macro\@classoptionslist{,latex}
740 \else\ifx u\bxjs@engine@given
741   \g@addto@macro\@classoptionslist{,uplatex}
742 \fi\fi
743 \fi
744 \ifx\bxjs@engine@given\@undefined\else
745   \ifx\bxjs@engine@given\bxjs@tmpb\else
746     \ClassError\bxjs@clsname
747       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
748   \fi
749 \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。

```
750 \ifjsWithpTeXng
751   \g@addto@macro\@classoptionslist{,uplatex}
752 \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。

```
753 \@tempswatrue
754 \ifx \bxjs@driver@given\@undefined\else
755   \ifjsInPdfMode
756     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
757       \@tempswafalse
758     \fi
759   \else\ifx x\jsEngine
760     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
761       \@tempswafalse
762     \fi
763   \else
764     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
765       \@tempswafalse
766     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
767       \@tempswafalse
768     \fi\fi
769   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
770     \@tempswafalse
771   \fi\fi
772 \fi\fi
773 \fi
```



```

774 \if@tempswa\else
775   \ClassError\bxjs@clsname
776   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
777 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

778 \ifjsInPdfMode \@tempswafalse
779 \else\ifx x\jsEngine \@tempswafalse
780 \else\ifjsWithpTeXng \@tempswafalse
781 \else \@tempwatrue
782 \fi\fi\fi
783 \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)。

```

784   \ifx\bxjs@driver@opt\@undefined
785     \if \ifbxjs@explIII T\else\ifx\bxjs@jadriver@opt\@undefined F\else T\fi\fi T%
786     \ClassWarningNoLine\bxjs@clsname
787     {A driver option is MISSING!!\MessageBreak
788     You should properly specify one of the valid\MessageBreak
789     driver options according to the DVI driver\MessageBreak
790     that is in use:\MessageBreak
791     \@spaces dvips, dvipdfmx, dviout, xdvi,\MessageBreak
792     \@spaces nodvidriver}
793   \fi
794 \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。)

```

795   \ifbxjs@dvi@opt
796     \edef\bxjs@next{%
797       \let\noexpand\bxjs@driver@given
798       \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
799       \noexpand\g@addto@macro\noexpand\@classoptionslist
800       {,\bxjs@driver@opt}%
801     }\bxjs@next
802   \fi
803 \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。


```

804 \ifjsWithpTeXng
805   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
806     \let\bxjs@platexng@nodrv\@undefined
807   \else\ifx t\bxjs@platexng@nodrv\else
808     \g@addto@macro\@classoptionslist{,dvipdfmx}
809   \fi\fi
810 \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。

```
811 \ifx\bxjs@driver@given\bxjs@driver@@none
812   \bxjs@papersizefalse
813 \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。

※ (u)pTeX 以外で `ja` を省略するのは 2.0 版より非推奨となった。

```
814 \def\bxjs@@minimal{minimal}
815 \ifx\bxjs@jadriver\relax
816   \ifx j\jsEngine
817     \def\bxjs@jadriver{standard}
818   \else
819     \ClassWarningNoLine\bxjs@clsname
820     {The option 'ja' is MISSING!!\MessageBreak
821      So 'ja=minimal' is assumed as fallback, but\MessageBreak
822      such implicit setting is now DEPRECATED!\MessageBreak
823      You should write 'ja=minimal' explicitly,\MessageBreak
824      if it is intended}
825     \let\bxjs@jadriver\bxjs@@minimal
826   \fi
827 \fi
```

`plautopatch` が真の場合はここで `plautopatch` を読み込む。

※この時点で既に読み込まれているパッケージは、`calc`、`keyval`、`ifpdf`。

※ Pandoc モードでは `plautopatch` の既定値を真とする。

```
828 \def\bxjs@@pandoc{pandoc}
829 \ifx\bxjs@jadriver\bxjs@@pandoc \ifx\bxjs@plautopatch@given\@undefined
830   \ifjsWitheTeX
831     \bxjs@plautopatchtrue
832 \fi\fi\fi
833 \ifx j\jsEngine \ifbxjs@plautopatch
834   \RequirePackage{plautopatch}[2018/08/22]%v0.3
835 \fi\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。

```
836 \ifx\bxjs@jadriver@opt\@undefined\else
837   \ifx\bxjs@engine@given\@undefined
838     \ClassError\bxjs@clsname
839     {An engine option must be explicitly given}%
840     {When you use a Japanese-driver you must specify a correct\MessageBreak
841      engine option.\MessageBreak\@ehc}
842 \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。)

```

843 \ifx\bxjs@magstyle@@default\bxjs@magstyle@@mag\else
844   \ifx\bxjs@magstyle\bxjs@magstyle@@mag
845     \let\bxjs@magstyle\bxjs@magstyle@@default
846     \ClassError\bxjs@clsname
847       {The engine does not support 'magstyle=usemag'}%
848       {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
849       The default value 'nomag*' is used instead.\MessageBreak \@ehc}
850   \fi
851 \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。

```

852 \ifx\bxjs@jbase@opt\@undefined\else
853   \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。

```

854     \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
855     \bxjs@invscale\@tempdima\jsScale
856     \bxjs@setbasefontsize{\@tempdima}%
857   \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。

```

858     \ifx\bxjs@scale@opt\@undefined\else
859       \ClassWarningNoLine\bxjs@clsname
860         {Redundant 'scale' option is ignored}%
861     \fi
862     \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
863     \@tempdimb=\bxjs@param@basefontsize\relax
864     \edef\jsScale{\strip@pt\@tempdimb}%
865     \bxjs@invscale\@tempdima\jsScale
866     \edef\jsScale{\strip@pt\@tempdima}%
867   \fi
868 \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。

```

869 \let\Cjascale\jsScale

```

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

```

870 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
871   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
872     \catcode\@tempcnta\active
873     \advance\@tempcnta\@ne
874   \repeat
875 \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。

```
876 \ifbxjs@disguise@js
877 %<book|report>\def\bxjs@js@clsname{jsbook}
878 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
879 \@namedef{ver@bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
880 \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。

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

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

```
882 \ifbxjs@oldfontcommands
883 \AtEndOfClass{\allowoldfontcommands}
884 \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 dvipout 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。

[2020-10-04] L^AT_EX 2_ε 2020-10-01 でカーネルの \shipout コードが拡張され \AtBeginDvi の実行タイミングが変化したので、この時点で発行する \special の中身を展開しておくようにしました。こうしないと、用紙サイズ設定を間違ってしまう (Issue #72)。

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。

```

885 \newdimen\stockwidth \newdimen\stockheight
886 \begingroup\expandafter\expandafter\expandafter\endgroup
887 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
888 % \newdimen\stockwidth \newdimen\stockheight
889 \setlength{\stockwidth}{\paperwidth}
890 \setlength{\stockheight}{\paperheight}
891 \advance \stockwidth 2in
892 \advance \stockheight 2in
893 \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。

```

894 %<slide>\def\n@baseline{13}%
895 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
896 %<!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。

```

897 \ifx\bxjs@magstyle\bxjs@magstyle@@mag
898 \jsc@magtrue
899 \else\ifx\bxjs@magstyle\bxjs@magstyle@@xreal
900 \jsc@mag@xrealtrue
901 \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。

```

902 \ifx\bxjs@param@mag\relax
903 \@tempdima=\bxjs@param@basefontsize
904 \advance\@tempdima.001pt \multiply\@tempdima25
905 \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
906 \edef\bxjs@param@mag{\the\@tempcmta}
907 \else
908 % mag 値が直接指定された場合
909 \bxjs@gset@tempcmta{\bxjs@param@mag}

```

```

910 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
911 % 有効な mag 値の範囲は 1--32768
912 \edef\bxjs@param@mag{\the\@tempcnta}
913 \advance\@tempcnta100000
914 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
915 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
916 \edef\bxjs@param@basefontsize{\the\@tempdima}
917 \fi
918 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
919 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
920 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
921 \edef\jsc@magscale{\strip@pt\@tempdima}
922 \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。

```

923 \newdimen\jsc@mpt
924 \newdimen\jsc@mmm
925 \ifjsc@mag
926 \jsc@mpt=1\p@
927 \jsc@mmm=1mm
928 \catcode`\?=9 % \p@? read as \p@
929 \else
930 \jsc@mpt=\jsc@magscale\p@
931 \jsc@mmm=\jsc@magscale mm
932 \catcode`\?=11 \let\p@?\jsc@mpt
933 \fi
934 \chardef\bxjs@qmc=\catcode`\?\relax
935 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode`\?=12\relax}

```

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

noz 非指定時は \zw を \jsZw と同義にする。

```

936 \newdimen\jsZw
937 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
938 \ifbxjs@usezw
939 \providecommand*\zw{\jsZw}
940 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

942 \ifjsc@mag@xreal
943   \RequirePackage{type1cm}
944   \let\jsc@invscale\bxjs@invscale

ムニャムニャムニャ……。



---


945   \ifbxjs@TUenc
946     \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
947   \else
948     \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
949   \fi
950   \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
951   \let\jsc@get@external@font\get@external@font
952   \def\get@external@font{%
953     \jsc@preadjust@extract@font
954     \jsc@get@external@font}
955   \def\jsc@fstrunc#1{%
956     \edef\jsc@tmpa{\strip@pt#1}%
957     \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
958   \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
959     \if#5*\else
960       \edef\jsc@tmpa{#1%
961         \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
962     \fi}
963   \def\jsc@preadjust@extract@font{%
964     \let\jsc@req@size\f@size
965     \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
966     \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
967     \let\jsc@ref@size\jsc@tmpa
968     \let\f@size\jsc@ref@size}
969   \def\execute@size@function#1{%
970     \let\jsc@cref@size\f@size
971     \let\f@size\jsc@req@size
972     \csname s@fct@#1\endcsname}
973   \let\jsc@DeclareErrorFont\DeclareErrorFont
974   \def\DeclareErrorFont#1#2#3#4#5{%
975     \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
976     \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
977     \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
978   \def\gen@sfcnt{%
979     \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
980     \empty@sfcnt}
981   \def\genb@sfcnt{%
982     \edef\mandatory@arg{%
983       \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@{}}%
984     \empty@sfcnt}
985   \ifbxjs@TUenc\else
986     \DeclareErrorFont{OT1}{cmr}{m}{n}{10}

```

```

987 \fi
988 \fi

```

[2016-11-16] latex.ltx (ltspace.dtx) で定義されている `\smallskip` の、単位 `pt` を `\jsc@empt` 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 989 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 990 %\def\jsc@medskip{\vspace\jsc@medskipamount}
991 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount 992 \newskip\jsc@smallskipamount
\jsc@bigskipamount 993 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
994 %\newskip\jsc@medskipamount
995 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
996 %\newskip\jsc@bigskipamount
997 %\jsc@bigskipamoun =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

`\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] トンボオブションが指定されているとき「だけ」`\stockwidth`, `\stockheight` が定義されています。

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

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

```

998 % \ifpapersize
999 % \setlength{\@tempdima}{\paperwidth}
1000 % \setlength{\@tempdimb}{\paperheight}
1001 % \iftombow
1002 % \advance \@tempdima 2truein
1003 % \advance \@tempdimb 2truein
1004 % \fi
1005 % \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
1006 % \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, L^AT_EX の `\def\@{\spacefactor\@m}` という定義 (`\@m` は 1000) では

I watch TV\@. と書くと V とピリオドのペアカーニングが効かなくなります。そこで、次のような定義に直し、I watch TV.\@ と書くことにします。

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

BXJS クラスでの変更点：

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

```
1007 \chardef\bxjs@periodchar=`\.  
1008 \bxjs@protected\def\bxjs@SE{%  
1009   \ifnum\spacefactor<\@m \spacefactor\@m  
1010   \else \spacefactor\sfcode\bxjs@periodchar  
1011   \fi}  
1012 \ifbxjs@fix@at@cmd  
1013   \def\@{\bxjs@SE{}}  
1014 \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>\@viiipt</code>	8	<code>\@ixpt</code>	9	<code>\@xpt</code>	10
<code>\@xipt</code>	10.95	<code>\@xiipt</code>	12	<code>\@xivpt</code>	14.4

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

`\kanjiskip` は pL^AT_EX 2_ε で 0pt plus .4pt minus .5pt に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追

い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

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

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

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

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

```
1015 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
1016 \expandafter\bxjs@tmpa\expandafter{%
1017   \set@fontsize{#1}{#2}{#3}%
1018 % 末尾にコードを追加
1019   \expandafter\def\expandafter\size@update\expandafter{%
1020     \size@update
1021     \jsFontSizeChanged}%
1022 }
```

`\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。

```
1023 \newcommand*\jsFontSizeChanged{%
1024   \jsZw=\f@size\p@
1025   \jsZw=\jsScale \jsZw
1026   \ifdim\parindent>\z@
1027     \if@english \parindent=1em
1028     \else       \parindent=1\jsZw
1029     \fi
1030   \fi\relax
1031   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
1032 \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。

```
1033 \ifjsc@mag
1034   \let\jsc@setfontsize\@setfontsize
1035 \else
1036   \def\jsc@setfontsize#1#2#3{%
1037     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
1038 % microtype 対策
1039 \ifjsWithTeX\if j\jsEngine\else
```

```

1040 \def\jsc@setfontsize#1#2#3{%
1041 \edef\bxjs@sfs@next{%
1042 \unexpanded{\@setfontsize#1}%
1043 {\the\dimexpr#2\jsc@mpt\relax}{\the\dimexpr#3\jsc@mpt\relax}%
1044 }\bxjs@sfs@next}
1045 \fi\fi
1046 \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。

```

1047 \emergencystretch 3\jsZw

```

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

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

`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対して、しっぽ愛好家さんが次の修正を教えてくださいました。

[2008-02-18] `english` オプションで最初の段落のインデントをしないようにしました。

TODO: Hasumi さん [qa:54539] のご指摘は考慮中です。

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

```

1048 \newif\ifnarrowbaselines
1049 \if@english
1050 \narrowbaselinestrue
1051 \fi
1052 \def\narrowbaselines{%
1053 \narrowbaselinestrue
1054 \skip0=\abovedisplayskip
1055 \skip2=\abovedisplayshortskip
1056 \skip4=\belowdisplayskip
1057 \skip6=\belowdisplayshortskip
1058 % 一時的に警告を無効化する
1059 \let\bxjs@save@nomath\@nomath
1060 \let\@nomath\@gobble
1061 \@currsize\selectfont
1062 \let\@nomath\bxjs@save@nomath
1063 \abovedisplayskip=\skip0

```

```

1064 \abovedisplayskip=\skip2
1065 \belowdisplayskip=\skip4
1066 \belowdisplayshortskip=\skip6\relax}
1067 \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。

```

1068 \def\bxjs@if@narrowbaselines{%
1069   \ifnarrowbaselines\expandafter\@firstoftwo
1070   \else \expandafter\@secondoftwo
1071   \fi
1072 }

```

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

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

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

microtype 対策のため if 文を避ける。

```

1073 \renewcommand{\normalsize}{%
1074   \bxjs@if@narrowbaselines{%
1075     \jsc@setfontsize\normalsize\@xpt\@xipt
1076   }{%else
1077     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1078   }%

```

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

[2003-02-16] ちょっと変えました。

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

```

1079 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1080 \abovedisplayshortskip \z@ \@plus3\p@?
1081 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1082 \belowdisplayshortskip \belowdisplayskip

```

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

```
1083 \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。

```
1084 %</class>
1085 %<*class|minijs>
1086 %% initialize
1087 \normalsize
1088 %</class|minijs>
1089 %<*class>
```

`\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
`\Cdp` ラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (1zw) です。
`\Cwd` [2017-08-31] 基準とする文字を「全角空白」(EUC コード 0xA1A1) から「漢」(JIS コー
`\Cvs` ド 0x3441) へ変更しました。

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

```
1090 \ifx\Cht\@undefined \newdimen\Cht \fi
1091 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1092 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1093 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1094 \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。

```
1095 \setlength\Cht{0.88\jsZw}
1096 \setlength\Cdp{0.12\jsZw}
1097 \setlength\Cwd{1\jsZw}
1098 \setlength\Cvs{\baselineskip}
1099 \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。

```
1100 \newcommand{\small}{%
1101 \bxjs@if@narrowbaselines{%
1102 %<!kiyou> \jsc@setfontsize\small\@ixpt{11}%
1103 %<kiyou> \jsc@setfontsize\small{8.8888}{11}%
1104 }{%else
1105 %<!kiyou> \jsc@setfontsize\small\@ixpt{13}%
1106 %<kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
1107 }%
1108 \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@}
```

```

1109 \abovedisplaysshortskip \z@ \@plus3\p@?
1110 \belowdisplayskip \abovedisplayskip
1111 \belowdisplayshortskip \belowdisplayskip
1112 \def\@listi{\leftmargin\leftmargini
1113         \topsep \z@
1114         \parsep \z@
1115         \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。

```

1116 \newcommand{\footnotesize}{%
1117   \bxjs@if@narrowbaselines{%
1118     %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
1119     %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
1120   }{%else
1121     %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
1122     %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1123   }%
1124   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1125   \abovedisplayshortskip \z@ \@plus2\p@?
1126   \belowdisplayskip \abovedisplayskip
1127   \belowdisplayshortskip \belowdisplayskip
1128   \def\@listi{\leftmargin\leftmargini
1129         \topsep \z@
1130         \parsep \z@
1131         \itemsep \parsep}}

```

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

\LARGE [2004-11-03] \HUGE を追加。

```

\huge 1132 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1133 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 1134 \if@twocolumn
1135 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1136 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1137 \else
1138 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
1139 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1140 \fi
1141 %<!kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xivpt{21}}
1142 %<kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE{12.222}{21}}
1143 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1144 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1145 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1146 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

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

```
1147 \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。

```
1148 % \newcommand{\headfont}{\bfseries}
```

```
1149 \newcommand{\headfont}{\sffamily}
```

```
1150 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}
```

5 レイアウト

■二段組

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

```
1151 %<!kiyou>\setlength\columnsep{2\Cwd}
```

```
1152 %<kiyou>\setlength\columnsep{28truebp}
```

```
1153 \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 1154 \setlength\lineskip{1\jsc@mp}
```

```
\normallineskiplimit 1155 \setlength\normallineskip{1\jsc@mp}
```

```
1156 \setlength\lineskiplimit{1\jsc@mp}
```

```
1157 \setlength\normallineskiplimit{1\jsc@mp}
```

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

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

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

```
1158 \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。

```
1159 \setlength\parskip{\z@}
```

```
1160 \if@slide
```

```
1161 \setlength\parindent{0\p@}
```

```
1162 \else
```

```
1163 \setlength\parindent{1\Cwd}
```

```
1164 \fi
```

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

```
\@highpenalty 1165 \@lowpenalty 51
```

```
1166 \@medpenalty 151
```

```
1167 \@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。

```
1168 % \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。

```
1169 % \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。

```
1170 \@onlypreamble\bxjs@bd@pre@geometry@hook
```

```
1171 \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。

```
1172 \ifjsc@mag
```

```
1173 \mag=\bxjs@param@mag
```

```
1174 \fi
```

```
1175 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLlength` のための和文単位の定義。

```
1176 \def\bxjs@unit@trueQ{0.25trueem}\let\bxjs@unit@trueH\bxjs@unit@trueQ
```

```
1177 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw
```


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

```

1178 \@tempswafalse
1179 \def\bxjs@tmpdo{\@ifnextchar\bgroup\bxjs@tmpdo@a\remove@to@nnil}
1180 \def\bxjs@tmpdo@a#1{\edef\bxjs@tmpa{#1}%
1181   \@ifnextchar\bgroup\bxjs@tmpdo@b\remove@to@nnil}
1182 \def\bxjs@tmpdo@b#1{\edef\bxjs@tmpa{\bxjs@tmpa,#1}%
1183   \@ifnextchar\@nnil\bxjs@tmpdo@c\remove@to@nnil}
1184 \def\bxjs@tmpdo@c\@nnil{\@tempswatrue
1185   \edef\bxjs@param@paper{papersize={\bxjs@tmpa}}}
1186 \expandafter\bxjs@tmpdo\bxjs@param@paper\@nnil

```

`W,H` の形式について。

```

1187 \if@tempswa\else
1188   \def\bxjs@tmpa{\@nil,\@nil}
1189   \def\bxjs@tmpdo#1,#2,#3\@nnil{%
1190     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1191       \@tempswatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1192   \expandafter\bxjs@tmpdo\bxjs@param@paper,\@nil,\@nil\@nnil
1193 \fi

```

`W*H` の形式について。

```

1194 \if@tempswa\else
1195   \def\bxjs@tmpa{\@nil*\@nil}
1196   \def\bxjs@tmpdo#1*#2*#3\@nnil{%
1197     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1198       \@tempswatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1199   \expandafter\bxjs@tmpdo\bxjs@param@paper*\@nil*\@nil\@nnil
1200 \fi

```

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

```

1201 \edef\bxjs@layout@paper{%
1202   \ifjsc@mag truedimen,\fi
1203   \if@landscape landscape,\fi
1204   \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。

```

1205 %<*article|report>
1206 \def\bxjs@layout@base{%
1207   headheight=\topskip,footskip=0.03367\paperheight,%
1208   headsep=\footskip-\topskip,includeheadfoot,%
1209 }
1210 \edef\bxjs@layout{\bxjs@layout@base
1211   hscale=0.76,hmarginratio=1:1,%
1212   vscale=0.83,vmarginratio=1:1,%
1213 }
1214 %</article|report>
1215 %<*book>
1216 \def\bxjs@layout@base{%

```

```

1217 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1218 }
1219 \ifbxjs@layout@buggyhmargin %---
1220 % アレ
1221 \edef\bxjs@layout{\bxjs@layout@base
1222 hmargin=36\jsc@mmm,hmarginratio=1:1,%
1223 vscale=0.83,vmarginratio=1:1,%
1224 }
1225 \else %---
1226 % 非アレ
1227 \edef\bxjs@layout{\bxjs@layout@base
1228 hmargin=18\jsc@mmm,%
1229 vscale=0.83,vmarginratio=1:1,%
1230 }
1231 \fi %---
1232 %</book>
1233 %<*slide>
1234 \def\bxjs@layout@base{%
1235 noheadfoot,%
1236 }
1237 \edef\bxjs@layout{\bxjs@layout@base
1238 hscale=0.9,hmarginratio=1:1,%
1239 vscale=0.944,vmarginratio=1:1,%
1240 }
1241 %</slide>

```

textwidth オプションの設定を反映する。

```

1242 %<*!book>
1243 \ifx\bxjs@textwidth@opt\undefined\else
1244 \jsSetQHLlength\@tempdima{\bxjs@textwidth@opt}
1245 \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1246 \fi
1247 %</!book>
1248 \ifx\bxjs@number@of@lines@opt\undefined\else
1249 \bxjs@egset@tempcnta{\bxjs@number@of@lines@opt}
1250 \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1251 \fi

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1252 \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。

```

1253 %<*book>
1254 \newcommand\jsTextWidthLimit{40}
1255 \@tempdima=\jsTextWidthLimit\Cwd
1256 \ifx\bxjs@textwidth@limit@opt\undefined\else
1257 \bxjs@egset@tempcnta{\bxjs@textwidth@limit@opt}

```

```

1258 \tempdima=\@tempcnta\Cwd
1259 \fi
1260 \ifx\bxjs@textwidth@opt\undefined\else
1261 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1262 \fi
1263 \edef\bxjs@textwidth@limit{\the\@tempdima}
1264 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1265 \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1266 \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1267 \fi
1268 %</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。

```

1269 \def\bxjs@preproc@layout{%
1270 \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mppt}

```

\bxjs@postproc@layout geometry の後処理。

```

1271 \def\bxjs@postproc@layout{%
    geometry のドライバを再設定する。
1272 \ifx\bxjs@geometry@driver\relax\else
1273 \let\Gm@driver\bxjs@geometry@driver
1274 \fi
    \ht\strutbox の値を元に戻す。
1275 \ht\strutbox=\bxjs@save@ht@strutbox\relax
    \textwidth の値を補正する。
1276 \ifbxjs@whole@zw@lines
1277 \tempdimb=\textwidth
1278 \if@twocolumn \tempdima=2\Cwd \else \tempdima=1\Cwd \fi
1279 \advance\textwidth.005pt\relax
1280 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1281 \advance\@tempdimb-\textwidth
1282 \advance\oddsidemargin 0.5\@tempdimb
1283 \advance\evensidemargin 0.5\@tempdimb
1284 \fi
1285 \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。

```

1286 %<*book>
1287 \tempdima=\bxjs@textwidth@limit\relax
1288 \ifbxjs@whole@zw@lines
1289 \advance\@tempdima.005pt\relax
1290 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd

```

```

1291 \fi
1292 \ifdim\textwidth>\@tempdima
1293   \textwidth=\@tempdima
1294   \addtolength\evensidemargin{\fullwidth-\textwidth}
1295 \fi
1296 %</book>

\textheight 関連の調整。
1297 \@tempdimb=\textheight
1298 \advance\textheight-\topskip
1299 \advance\textheight.005pt\relax
1300 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1301 \advance\textheight\topskip
1302 \advance\@tempdimb-\textheight
1303 \advance\topmargin0.5\@tempdimb

\headheight 関連の調整。
1304 \@tempdima=\topskip
1305 \advance\headheight\@tempdima
1306 \advance\topmargin-\@tempdima

marginpar 関連の調整。
1307 \setlength\marginparsep{\columnsep}
1308 \setlength\marginparpush{\baselineskip}
1309 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1310   -\textwidth-10\jsc@mmm-\marginparsep}
1311 \ifbxjs@whole@zw@lines
1312   \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1313 \fi

連動する変数。
1314 \maxdepth=.5\topskip
1315 \stockwidth=\paperwidth
1316 \stockheight=\paperheight
1317 }

```

`\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。

```

1318 \edef\jsGeometryOptions{%
1319   \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージを読み込む

ムニャムニャ。

```

1320 \def\bxjs@geometry@name{geometry}
1321 \ifbxjs@old@hook@system
1322   \let\bxjs@apply@bd@pre@geometry@hook\AtBeginDocument
1323 \else

```

```

1324 \def\bxjs@apply@bd@pre@geometry@hook{%
1325   \AddToHook{begindocument}{\bxjs@geometry@name}}
1326 \fi

```

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

```

1327 \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。

```

1328 \ifbxjs@papersize
1329   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1330     \PassOptionsToPackage{dvipdfm}{geometry}
1331   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1332     \PassOptionsToPackage{dvipdfm}{geometry}
1333   \fi\fi
1334   \let\bxPapersizeSpecialDone=t
1335 \else
1336   \PassOptionsToPackage{driver=none}{geometry}
1337 \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。

```

1338 \bxjs@apply@bd@pre@geometry@hook
1339 \bxjs@apply@bd@pre@geometry@hook{\bxjs@bd@pre@geometry@hook}
1340 \bxjs@apply@bd@pre@geometry@hook{\ImposeOldLuaTeXBehavior}
1341 \bxjs@preproc@layout
1342 \edef\bxjs@next{%
1343   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1344 }\bxjs@next
1345 \bxjs@apply@bd@pre@geometry@hook{\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。

```

1346 \let\bxjs@geometry@driver\Gm@driver
1347 \bxjs@postproc@layout

```

geometry のドライバ自動判別に対する前処理。

```

1348 \g@addto@macro\bxjs@bd@pre@geometry@hook{%

```

BXJS2.0 版より、geometry の 4.x 版のサポートは廃止された。

```

1349   \ifpackage@later{geometry}{2010/02/12}{\else
1350     \PackageError\bxjs@clsname
1351       {Your 'geometry' package is too old (< v5.0)}%
1352     {\@ehc}%
1353     \let\Gm@driver\relax}%

```

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

```
1354 \ifjsWithpTeXng
1355 \ifx\Gm@driver\@empty
1356 \def\Gm@driver{pdftex}%
1357 \fi
1358 \fi}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```
1359 \def\setpagelayout{%
1360 \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1361 \@ifstar{\bxjs@setpagelayout@a@one}{\bxjs@setpagelayout@a@z@}}%
1362 \def\bxjs@setpagelayout@a#1#2{%
1363 \ifcase#1% modify
1364 \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1365 \or% reset(*)
1366 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1367 \or% semireset(+)
1368 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1369 \fi
1370 \bxjs@preproc@layout
1371 \edef\bxjs@next{%
1372 \noexpand\geometry{\bxjs@next}%
1373 }\bxjs@next
1374 \bxjs@postproc@layout}
```

■ geometry パッケージを読み込まない

`geometry=user` の場合の処理。

```
1375 \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。

```
1376 \g@addto@macro\bxjs@begin@document@hook{%
1377 \ifdim\textwidth=.5\maxdimen
1378 \ClassError\bxjs@clsname
1379 {Page layout is not properly set}%
1380 {\@ehd}%
1381 \fi}
1382 \def\jsUseMinimalPageLayout{%
1383 \setlength{\textwidth}{6.5in}%
1384 \setlength{\textheight}{8in}}
```

`\setpagelayout` はとりあえず無効にしておく。

```
1385 \let\bxjs@geometry@driver\relax
```

```

1386 \def\setpagelayout{%
1387   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1388     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1389 \def\bxjs@pagelayout@a#1{%
1390   \ClassError\bxjs@clsname
1391   {Command '\string\setpagelayout' is not supported,\MessageBreak
1392     because 'geometry' value is not 'class'}\@eha}
1393 %
1394 \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。

```
1395 %<*jsclasses>
```

■ 縦方向のスペース

`\headheight` `\topskip` は本文領域上端と本文 1 行目のベースラインとの距離です。あまりぎりぎりの値にすると、本文中に \int のような高い文字が入ったときに 1 行目のベースラインが他のページより下がってしまいます。ここでは本文の公称フォントサイズ (10pt) にします。

[2003-06-26] `\headheight` はヘッダの高さで、元は 12pt でしたが、新ドキュメントクラスでは `\topskip` と等しくしていました。ところが、`fancyhdr` パッケージで `\headheight` が小さいとおかしいことになるようですので、2 倍に増やしました。代わりに、版面の上下揃えの計算では `\headheight` ではなく `\topskip` を使うことにしました。

[2016-08-17] 圏点やルビが一行目に来た場合に下がるのを防ぐため、`\topskip` を 10pt から 1.38zw に増やしました。`\headheight` は従来と同じ 20pt のままとします。

```

1396 \setlength\topskip{1.38zw}% from 10\jsc@mpt (2016-08-17)
1397 \if@slide
1398   \setlength\headheight{0\jsc@mpt}
1399 \else
1400   \setlength\headheight{20\jsc@mpt}% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1401 \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。

```

1402 %<*article|kiyou>
1403 \if@slide
1404   \setlength\footskip{0pt}
1405 \else
1406   \setlength\footskip{0.03367\paperheight}
1407   \ifdim\footskip<\baselineskip
1408     \setlength\footskip{\baselineskip}
1409   \fi
1410 \fi
1411 %</article|kiyou>
1412 %<jspf>\setlength\footskip{9\jsc@mmm}
1413 %<*book>
1414 \if@report
1415   \setlength\footskip{0.03367\paperheight}
1416   \ifdim\footskip<\baselineskip
1417     \setlength\footskip{\baselineskip}
1418   \fi
1419 \else
1420   \setlength\footskip{0pt}
1421 \fi
1422 %</book>
1423 %<*report>
1424 \setlength\footskip{0.03367\paperheight}
1425 \ifdim\footskip<\baselineskip
1426   \setlength\footskip{\baselineskip}
1427 \fi
1428 %</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)。

```

1429 %<*article>
1430 \if@slide
1431   \setlength\headsep{0\jsc@empt}
1432   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1433   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1434 \else
1435   \setlength\headsep{\footskip}
1436   \addtolength\headsep{-\topskip}
1437 \fi
1438 %</article>
1439 %<*book>
1440 \if@report
1441   \setlength\headsep{\footskip}
1442   \addtolength\headsep{-\topskip}
1443 \else

```



```

1444 \setlength\headsep{6\jsc@mmm}
1445 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1446 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1447 \fi
1448 %</book>
1449 %<*report>
1450 \setlength\headsep{\footskip}
1451 \addtolength\headsep{-\topskip}
1452 %</report>
1453 %<*jspf>
1454 \setlength\headsep{9\jsc@mmm}
1455 \addtolength\headsep{-\topskip}
1456 %</jspf>
1457 %<*kiyou>
1458 \setlength\headheight{0\jsc@mpt}
1459 \setlength\headsep{0\jsc@mpt}
1460 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1461 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1462 %</kiyou>

```

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

```

1463 \setlength\maxdepth{.5\topskip}

```

■本文の幅と高さ

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

```

1464 \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。

```

1465 %<*article>
1466 \if@slide
1467 \setlength\fullwidth{0.9\paperwidth}
1468 \else
1469 \setlength\fullwidth{0.76\paperwidth}

```

```

1470 \fi
1471 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1472 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1473 \setlength\textwidth{\fullwidth}
1474 %</article>
1475 %<*book>
1476 \if@report
1477   \setlength\fullwidth{0.76\paperwidth}
1478 \else
1479   \setlength\fullwidth{\paperwidth}
1480   \addtolength\fullwidth{-36\jsc@mmm}
1481 \fi
1482 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1483 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1484 \setlength\textwidth{\fullwidth}
1485 \if@report \else
1486   \if@twocolumn \else
1487     \ifdim \fullwidth>40zw
1488       \setlength\textwidth{40zw}
1489     \fi
1490   \fi
1491 \fi
1492 %</book>
1493 %<*report>
1494 \setlength\fullwidth{0.76\paperwidth}
1495 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1496 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1497 \setlength\textwidth{\fullwidth}
1498 %</report>
1499 %<*jspf>
1500 \setlength\fullwidth{50zw}
1501 \addtolength\fullwidth{8\jsc@mmm}
1502 \setlength\textwidth{\fullwidth}
1503 %</jspf>
1504 %<*kiyou>
1505 \setlength\fullwidth{48zw}
1506 \addtolength\fullwidth{\columnsep}
1507 \setlength\textwidth{\fullwidth}
1508 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

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

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

```
1509 %<*article|book|report>
1510 \if@slide
1511   \setlength{\textheight}{0.95\paperheight}
1512 \else
1513   \setlength{\textheight}{0.83\paperheight}
1514 \fi
1515 \addtolength{\textheight}{-10\jsc@mp}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1516 \addtolength{\textheight}{-\headsep}
1517 \addtolength{\textheight}{-\footskip}
1518 \addtolength{\textheight}{-\topskip}
1519 \divide\textheight\baselineskip
1520 \multiply\textheight\baselineskip
1521 %</article|book|report>
1522 <jspf>\setlength{\textheight}{51\baselineskip}
1523 <kiyou>\setlength{\textheight}{47\baselineskip}
1524 \addtolength{\textheight}{\topskip}
1525 \addtolength{\textheight}{0.1\jsc@mp}
1526 <jspf>\setlength{\mathindent}{10\jsc@mm}
```

`\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。

```
1527 \def\flushbottom{%
1528   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mp}%
1529   \let\@texttop\relax}
```

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

```
1530 \setlength\marginparsep{\columnsep}
1531 \setlength\marginparpush{\baselineskip}
```

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

ペース（1truein ではなく）を挿入するので、場合分けしています。

```
1532 \setlength{\oddsidemargin}{\paperwidth}
1533 \addtolength{\oddsidemargin}{-\fullwidth}
1534 \setlength{\oddsidemargin}{.5\oddsidemargin}
1535 \iftombow
1536   \addtolength{\oddsidemargin}{-1in}
1537 \else
1538   \addtolength{\oddsidemargin}{-\inv@mag in}
1539 \fi
1540 \setlength{\evensidemargin}{\oddsidemargin}
1541 \if@mparswitch
1542   \addtolength{\evensidemargin}{\fullwidth}
1543   \addtolength{\evensidemargin}{-\textwidth}
1544 \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。

```
1545 \setlength\marginparwidth{\paperwidth}
1546 \addtolength\marginparwidth{-\oddsidemargin}
1547 \addtolength\marginparwidth{-\inv@mag in}
1548 \addtolength\marginparwidth{-\textwidth}
1549 \addtolength\marginparwidth{-10\jsc@mmm}
1550 \addtolength\marginparwidth{-\marginparsep}
1551 \@tempdima=1zw
1552 \divide\marginparwidth\@tempdima
1553 \multiply\marginparwidth\@tempdima
```

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

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

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

```
1554 \setlength\topmargin{\paperheight}
1555 \addtolength\topmargin{-\textheight}
1556 \if@slide
1557   \addtolength\topmargin{-\headheight}
1558 \else
1559   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
     \headheight (2003-06-26)
1560 \fi
1561 \addtolength\topmargin{-\headsep}
1562 \addtolength\topmargin{-\footskip}
1563 \setlength\topmargin{0.5\topmargin}
1564 %<kiyou>\setlength\topmargin{81truebp}
1565 \iftombow
```

```

1566 \addtolength\topmargin{-1in}
1567 \else
1568 \addtolength\topmargin{-\inv@mag in}
1569 \fi
1570 %</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。

```
1571 \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。

```
1572 \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。

```
1573 \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。

```
1574 \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。

```
1575 \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。

```
1576 \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。
 1577 `\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。
 1578 `\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。
 1579 `\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。
 1580 `\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。
 1581 `\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。
 1582 `\renewcommand{\dblfloatpagefraction}{.8}`

`\floatsep` `\floatsep` はページ上部・下部のフロート間の距離です。`\textfloatsep` はページ上部・
`\textfloatsep` 下部のフロートと本文との距離です。`\intextsep` は本文の途中に出力されるフロートと本
`\intextsep` 文との距離です。
 1583 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 1584 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`
 1585 `\setlength\intextsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

`\dblfloatsep` 二段組のときの段抜きのフロートについての値です。
`\dbltextfloatsep` 1586 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 1587 `\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` 1588 `\setlength\@fptop{0\p@? \@plus 1fil}`
 1589 `\setlength\@fpsep{8\p@? \@plus 2fil}`
 1590 `\setlength\@fpbot{0\p@? \@plus 1fil}`

`\@dblftop` 段抜きフロートについての値です。
`\@dblfpsep` 1591 `\setlength\@dblftop{0\p@? \@plus 1fil}`
`\@dblfpbot` 1592 `\setlength\@dblfpsep{8\p@? \@plus 2fil}`
 1593 `\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
`\pltx@cleartoleftpage` 加しました。

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

となっています。

```
1594 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1595 % \ifodd\c@page
1596 % \iftdir
1597 % \hbox{}\thispagestyle{empty}\newpage
1598 % \if@twocolumn\hbox{}\newpage\fi
1599 % \fi
1600 % \else
1601 % \ifydir
1602 % \hbox{}\thispagestyle{empty}\newpage
1603 % \if@twocolumn\hbox{}\newpage\fi
1604 % \fi
1605 % \fi\fi}
1606 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1607 % \ifodd\c@page
1608 % \ifydir
1609 % \hbox{}\thispagestyle{empty}\newpage
1610 % \if@twocolumn\hbox{}\newpage\fi
1611 % \fi
1612 % \else
1613 % \iftdir
1614 % \hbox{}\thispagestyle{empty}\newpage
1615 % \if@twocolumn\hbox{}\newpage\fi
1616 % \fi
1617 % \fi\fi}
1618 %\def\pltx@cleartooddpage{\clearpage\if@twoside
1619 % \ifodd\c@page\else
1620 % \hbox{}\thispagestyle{empty}\newpage
1621 % \if@twocolumn\hbox{}\newpage\fi
1622 % \fi\fi}
1623 %\def\pltx@cleartoevenpage{\clearpage\if@twoside
1624 % \ifodd\c@page
1625 % \hbox{}\thispagestyle{empty}\newpage
1626 % \if@twocolumn\hbox{}\newpage\fi
1627 % \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。

```
1628 \let\pltx@cleartorightpage\pltx@cleartooddpage
1629 \let\pltx@cleartoleftpage\pltx@cleartoevenpage
```

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

```
1630 %<*book|report>
1631 \if@openleft
1632   \let\cleardoublepage\pltx@cleartoleftpage
1633 \else\if@openright
1634   \let\cleardoublepage\pltx@cleartorightpage
1635 \fi\fi
1636 %</book|report>
```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

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

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

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

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


```

1637 % \def\ps@empty{%
1638 %   \let\@mkboth\@gobbletwo
1639 %   \let\@oddhead\@empty
1640 %   \let\@oddfoot\@empty
1641 %   \let\@evenhead\@empty
1642 %   \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。

```

1643 \def\ps@plainfoot{%
1644   \let\@mkboth\@gobbletwo
1645   \let\@oddhead\@empty
1646   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1647   \let\@evenhead\@empty
1648   \let\@evenfoot\@oddfoot}
1649 \def\ps@plainhead{%
1650   \let\@mkboth\@gobbletwo
1651   \let\@oddfoot\@empty
1652   \let\@evenfoot\@empty
1653   \def\@evenhead{%
1654     \if@mparswitch \hss \fi
1655     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1656     \if@mparswitch\else \hss \fi}%
1657   \def\@oddhead{%
1658     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1659 %<book>\let\ps@plain\ps@plainhead
1660 %<!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。

```

1661 %<*article|slide>
1662 \if@twoside
1663   \def\ps@headings{%
1664     \let\@oddfoot\@empty
1665     \let\@evenfoot\@empty
1666     \def\@evenhead{\if@mparswitch \hss \fi
1667       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1668     \if@mparswitch\else \hss \fi}%
1669     \def\@oddhead{%
1670       \underline{%
1671         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1672     \let\@mkboth\markboth
1673     \def\sectionmark##1{\markboth{%
1674       \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1675       ##1}\fi}%
1676     \def\subsectionmark##1{\markright{%

```

```

1677     \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1678     ##1}}}%
1679 }
1680 \else % if not twoside
1681     \def\ps@headings{%
1682         \let\@oddfoot\@empty
1683         \def\@oddhead{%
1684             \underline{%
1685                 \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1686         \let\@mkboth\markboth
1687         \def\sectionmark##1{\markright{%
1688             \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1689             ##1}}}%
1690 \fi
1691 %</article|slide>

```

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

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

```

1692 %<*book|report>
1693 \def\bxjs@maybe@autoxspacing{%
1694     \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1695 \newif\if@omit@number
1696 \def\ps@headings{%
1697     \let\@oddfoot\@empty
1698     \let\@evenfoot\@empty
1699     \def\@evenhead{%
1700         \if@mparswitch \hss \fi
1701         \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1702             \textbf{\thepage}\hfil\leftmark}}}%
1703         \if@mparswitch\else \hss \fi}%
1704     \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1705         {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1706     \let\@mkboth\markboth
1707     \def\chaptermark##1{\markboth{%
1708         \ifnum \c@secnumdepth >\m@ne
1709             \if@mainmatter
1710                 \if@omit@number\else
1711                     \@chapapp\thechapter\@chappos\hskip1\jsZw
1712                 \fi
1713             \fi
1714         \fi
1715         ##1-}}}%
1716     \def\sectionmark##1{\markright{%
1717         \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi

```

```

1718     ##1}}}%
1719 %</book|report>

    最後は学会誌の場合です。

1720 %<*jspf>
1721 \def\ps@headings{%
1722     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1723     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1724     \def\@oddhead{\normalfont\hfil \@title \hfil}
1725     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1726 %</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。

```

1727 \def\ps@myheadings{%
1728     \let\@oddfoot\@empty\let\@evenfoot\@empty
1729     \def\@evenhead{%
1730         \if@mparswitch \hss \fi%
1731         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1732         \if@mparswitch\else \hss \fi}%
1733     \def\@oddhead{%
1734         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1735     \let\@mkboth\@gobbletwo
1736 %<book|report> \let\chaptermark\@gobble
1737 \let\sectionmark\@gobble
1738 %<!book&!report> \let\subsectionmark\@gobble
1739 }

```

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。

```

\author
\date 1740 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1741 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1742 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1743 % \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。

```

1744 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1745 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1746 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1747 \g@addto@macro\bxjs@begin@document@hook{\bxjs@decl@subtitle}
1748 \def\bxjs@decl@subtitle{%
1749   \global\let\bxjs@decl@subtitle\relax
1750   \ifx\subtitle\@undefined
1751     \global\let\subtitle\jsSubtitle
1752   \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。

```

1753 \def\bxjs@annihilate@subtitle{%
1754   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1755   \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 1756 %<*jspf>
\keywords 1757 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1758 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
1759 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1760 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1761 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1762 %</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。

```

1763 \def\plainifnotempty{%
1764   \ifx \@oddhead \@empty
1765     \ifx \@oddfoot \@empty
1766     \else
1767       \thispagestyle{plainfoot}%
1768     \fi
1769   \else
1770     \thispagestyle{plainhead}%
1771   \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。

```
1772 %<*article|book|report|slide>
1773 \if@titlepage
1774   \newcommand{\maketitle}{%
1775     \begin{titlepage}%
1776       \let\footnotesize\small
1777       \let\footnoterule\relax
1778       \let\footnote\thanks
1779       \null\vfil
1780       \if@slide
1781         {\footnotesize \@date}%
1782         \begin{center}
1783           \mbox{} \\\[1\jsZw]
1784           \large
1785           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1786           \jsc@smallskip
1787           \@title
1788           \ifx\bxjs@subtitle\@undefined\else
1789             \par\vskip\z@
1790             {\small \bxjs@subtitle\par}
1791           \fi
1792           \jsc@smallskip
1793           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1794           \vfill
1795           {\small \@author}%
1796         \end{center}
1797       \else
1798       \vskip 60\p@?
1799       \begin{center}%
1800         {\LARGE \@title \par}%
1801         \ifx\bxjs@subtitle\@undefined\else
1802           \vskip5\p@?
1803           {\normalsize \bxjs@subtitle\par}
1804         \fi
1805         \vskip 3em%
1806         {\large
1807           \lineskip .75em
1808           \begin{tabular}[t]{c}%
1809             \@author
1810           \end{tabular}\par}%
1811         \vskip 1.5em
1812         {\large \@date \par}%
1813       \end{center}%
1814       \fi
1815       \par
1816       \@thanks\vfil\null
1817     \end{titlepage}%
1818     \setcounter{footnote}{0}%
```

```

1819 \global\let\thanks\relax
1820 \global\let\maketitle\relax
1821 \global\let\@thanks\@empty
1822 \global\let\@author\@empty
1823 \global\let\@date\@empty
1824 \global\let\@title\@empty
1825 \global\let\title\relax
1826 \global\let\author\relax
1827 \global\let\date\relax
1828 \global\let\and\relax
1829 \bxjs@annihilate@subtitle
1830 }%
1831 \else
1832 \newcommand{\maketitle}{\par
1833 \begin{group}
1834 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1835 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1836 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1837 \parindent 1\jsZw\noindent
1838 \llap{\@textsuperscript{\normalfont\@thefnmark}\hspace{0.3\jsZw}}##1}%
1839 \if@twocolumn
1840 \ifnum \col@number=\@ne
1841 \maketitle
1842 \else
1843 \twocolumn[\maketitle]%
1844 \fi
1845 \else
1846 \newpage
1847 \global\@topnum\z@ % Prevents figures from going at top of page.
1848 \maketitle
1849 \fi
1850 \plainifnotempty
1851 \@thanks
1852 \endgroup
1853 \setcounter{footnote}{0}%
1854 \global\let\thanks\relax
1855 \global\let\maketitle\relax
1856 \global\let\@thanks\@empty
1857 \global\let\@author\@empty
1858 \global\let\@date\@empty
1859 \global\let\@title\@empty
1860 \global\let\title\relax
1861 \global\let\author\relax
1862 \global\let\date\relax
1863 \global\let\and\relax
1864 \bxjs@annihilate@subtitle
1865 }

```

`\@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。

```
1866 \def\@maketitle{%
1867   \newpage\null
1868   \vskip 2em
1869   \begin{center}%
1870     \let\footnote\thanks
1871     {\LARGE \@title \par}%
1872     \ifx\bxjs@subtitle\@undefined\else
1873       \vskip3\p@?
1874       {\normalsize \bxjs@subtitle\par}
1875     \fi
1876     \vskip 1.5em
1877     {\large
1878       \lineskip .5em
1879       \begin{tabular}[t]{c}%
1880         \@author
1881       \end{tabular}\par}%
1882     \vskip 1em
1883     {\large \@date}%
1884   \end{center}%
1885   \par\vskip 1.5em
1886 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1887 }
1888 \fi
1889 %</article|book|report|slide>
1890 %<*jspf>
1891 \newcommand{\maketitle}{\par
1892   \begingroup
1893     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1894     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1895     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1896       \parindent 1\jsZw\noindent
1897       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1898     \twocolumn[\@maketitle]%
1899     \plainifnotempty
1900     \@thanks
1901   \endgroup
1902   \setcounter{footnote}{0}%
1903   \global\let\thanks\relax
1904   \global\let\maketitle\relax
1905   \global\let\@thanks\@empty
1906   \global\let\@author\@empty
1907   \global\let\@date\@empty
1908 % \global\let\@title\@empty % \@title は柱に使う
1909   \global\let\title\relax
1910   \global\let\author\relax
1911   \global\let\date\relax
1912   \global\let\and\relax
```

```

1913 \ifx\authors@mail\@undefined\else{%
1914 \def\@makefnctext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1915 \footnotetext[0]{\itshape\authors@mail}%
1916 }\fi
1917 \global\let\authors@mail\@undefined}
1918 \def\@maketitle{%
1919 \newpage\null
1920 \vskip 6em % used to be 2em
1921 \begin{center}
1922 \let\footnote\thanks
1923 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1924 \lineskip .5em
1925 \ifx\@author\@undefined\else
1926 \vskip 1em
1927 \begin{tabular}[t]{c}%
1928 \@author
1929 \end{tabular}\par
1930 \fi
1931 \ifx\@etitle\@undefined\else
1932 \vskip 1em
1933 {\large \@etitle \par}%
1934 \fi
1935 \ifx\@eauthor\@undefined\else
1936 \vskip 1em
1937 \begin{tabular}[t]{c}%
1938 \@eauthor
1939 \end{tabular}\par
1940 \fi
1941 \vskip 1em
1942 \@date
1943 \end{center}
1944 \vskip 1.5em
1945 \centerline{\box\@abstractbox}
1946 \ifx\@keywords\@undefined\else
1947 \vskip 1.5em
1948 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1949 \fi
1950 \vskip 1.5em}
1951 %</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。

```

1952 \def\bxjs@label@sect#1{%

```



```

1953 \expandafter\ifx\csname label#1\endcsname\relax
1954 \csname the#1\endcsname
1955 \else \csname label#1\endcsname
1956 \fi}
1957 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1958 \ifnum\bxjs@label@section=\bxjs@label@section@@compat\else
1959 \def\@secapp{\presectionname}
1960 \def\@secpos{\postsectionname}
1961 \fi

```

`\labelsection` 節番号の出力書式。

```

1962 \ifnum\bxjs@label@section=\bxjs@label@section@@modern
1963 \def\labelsection{\@secapp\thesection\@secpos}
1964 \fi

```

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

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

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```

1965 \def\@startsection#1#2#3#4#5#6{%
1966   \if@noskipsec \leavevmode \fi
1967   \par
1968 % 見出し上の空きを \@tempskipa にセットする
1969   \@tempskipa #4\relax
1970 % \@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
1971   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1972 % 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
1973   \ifdim \@tempskipa <\z@
1974     \@tempskipa -\@tempskipa \@afterindentfalse
1975   \fi
1976   \if@nobreak
1977 %   \everypar{\everyparhook}% これは間違い
1978     \everypar{}%
1979   \else
1980     \addpenalty\@secpenalty
1981 % 次の行は削除
1982 %   \addvspace\@tempskipa
1983 % 次の \noindent まで追加
1984     \ifdim \@tempskipa >\z@
1985       \if@slide\else
1986         \null
1987         \vspace*{-\baselineskip}%
1988       \fi
1989       \vskip\@tempskipa
1990     \fi
1991   \fi
1992   \noindent
1993 % 追加終わり
1994   \@ifstar
1995     {\@ssect{#3}{#4}{#5}{#6}}%
1996     {\@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。

```

1997 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
1998   \let\bxjs@if@ceph\@firstofone
1999 \else \let\bxjs@if@ceph\@gobble
2000 \fi

```

```

2001 \def\@sect#1#2#3#4#5#6[#7]#8{%
2002   \ifnum #2>\c@secnumdepth
2003     \let\@svsec\@empty
2004   \else

```

```

2005 \refstepcounter{#1}%
2006 \protected@edef\@svsec{\@seccntformat{#1}\relax}%
2007 \fi
2008 % 見出し後の空きを \@tempskipa にセット
2009 \@tempskipa #5\relax
2010 % 条件判断の順序を入れ替えました
2011 \ifdim \@tempskipa<\z@
2012 \def\@svsechd{%
2013 #6{\hskip #3\relax
2014 \@svsec #8}%
2015 \csname #1mark\endcsname{#7}%
2016 \addcontentsline{toc}{#1}{%
2017 \ifnum #2>\c@secnumdepth \else
2018 \protect\numberline{\bxjs@label@sect{#1}}%
2019 \fi
2020 #7}}% 目次にフルネームを載せるなら #8
2021 \else
2022 \begingroup
2023 \interlinepenalty \@M % 下から移動
2024 #6{%
2025 \@hangfrom{\hskip #3\relax\@svsec}%
2026 % \interlinepenalty \@M % 上に移動
2027 #8\@@par}%
2028 \endgroup
2029 \csname #1mark\endcsname{#7}%
2030 \addcontentsline{toc}{#1}{%
2031 \ifnum #2>\c@secnumdepth \else
2032 \protect\numberline{\bxjs@label@sect{#1}}%
2033 \fi
2034 #7}}% 目次にフルネームを載せるならここは #8
2035 \fi
2036 \@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。

```

2037 \def\@xsect#1{%
2038 % 見出しの後ろの空きを \@tempskipa にセット
2039 \@tempskipa #1\relax
2040 % 条件判断の順序を変えました
2041 \ifdim \@tempskipa<\z@
2042 \@nbreakfalse
2043 \global\@noskipsecttrue
2044 \everypar{%
2045 \if@noskipsec
2046 \global\@noskipsecfalse
2047 {\setbox\z@\lastbox}%

```

```

2048      \clubpenalty\@M
2049      \begingroup \@svsechd \endgroup
2050      \unskip
2051      \@tempskipa #1\relax
2052      \hskip -\@tempskipa
2053      \else
2054      \clubpenalty \@clubpenalty
2055      \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
2056      \fi\bxjs@if@ceph\everyparhook}%
2057 \else
2058   \par \nobreak
2059   \vskip \@tempskipa
2060   \@afterheading
2061 \fi
2062 \if@slide
2063   {\vskip\if@twocolumn-5\jsc@mpt\else-6\jsc@mpt\fi
2064    \maybeblue\hrule height0\jsc@mpt depth1\jsc@mpt
2065    \vskip\if@twocolumn 4\jsc@mpt\else 7\jsc@mpt\fi\relax}%
2066 \fi
2067 \par % 2000-12-18
2068 \ignorespaces}
2069 \def\@ssect#1#2#3#4#5{%
2070   \@tempskipa #3\relax
2071   \ifdim \@tempskipa<\z@
2072     \def\@svsechd{#4{\hskip #1\relax #5}}%
2073   \else
2074     \begingroup
2075     #4{%
2076       \@hangfrom{\hskip #1}%
2077       \interlinepenalty \@M #5\@@par}%
2078     \endgroup
2079   \fi
2080   \@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 \LaTeX 本体で定義済みです。

```

\subsectionmark 2081 \newcommand*\chaptermark[1]{}
\subsubsectionmark 2082 % \newcommand*\sectionmark[1]{}
2083 % \newcommand*\subsectionmark[1]{}
\paragraphmark 2084 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 2085 % \newcommand*\paragraphmark[1]{}
2086 % \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。

```
2087 %<!book&!report>\setcounter{secnumdepth}{3}
2088 %<book|report>\setcounter{secnumdepth}{2}
```

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

```
\c@subsection 2089 \newcounter{part}
2090 %<book|report>\newcounter{chapter}
\c@subsubsection 2091 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2092 %<!book&!report>\newcounter{section}
\c@subparagraph 2093 \newcounter{subsection}[section]
2094 \newcounter{subsubsection}[subsection]
2095 \newcounter{paragraph}[subsubsection]
2096 \newcounter{subparagraph}[paragraph]
```

`\thepart` カウンタの値を出力する命令 `\the` 何々 を定義します。

`\thechapter` カウンタを出力するコマンドには次のものがあります。

<code>\thesection</code>	<code>\arabic{COUNTER}</code>	1, 2, 3, ...
<code>\thesubsection</code>	<code>\roman{COUNTER}</code>	i, ii, iii, ...
<code>\thesubsubsection</code>	<code>\Roman{COUNTER}</code>	I, II, III, ...
<code>\theparagraph</code>	<code>\alph{COUNTER}</code>	a, b, c, ...
<code>\thesubparagraph</code>	<code>\Alph{COUNTER}</code>	A, B, C, ...
	<code>\kansuji{COUNTER}</code>	一, 二, 三, ...

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```
2097 \renewcommand{\thepart}{\@Roman\c@part}
2098 %<*!book&!report>
2099 \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2100 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2101 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2102 \else
2103 \renewcommand{\thesection}{\@arabic\c@section}
2104 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2105 \fi
2106 %</!book&!report>
2107 %<*book|report>
2108 \renewcommand{\thechapter}{\@arabic\c@chapter}
2109 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2110 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2111 %</book|report>
2112 \renewcommand{\thesubsubsection}{\%
2113   \thesubsection.\@arabic\c@subsubsection}
2114 \renewcommand{\theparagraph}{\%
2115   \thesubsubsection.\@arabic\c@paragraph}
2116 \renewcommand{\thesubparagraph}{\%
2117   \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。

```
2118 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2119 %<book|report>\newcommand{\@chappos}{\postchaptername}
```

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

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

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

```
2120 %<*book|report>
2121 \newcommand\frontmatter{%
2122   \pltx@cleartooddpage
2123   \@mainmatterfalse
2124   \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。

```
2125 \newcommand\mainmatter{%
2126   \pltx@cleartooddpage
2127   \@mainmattertrue
2128   \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。

```
2129 \newcommand\backmatter{%
2130   \if@openleft
2131     \cleardoublepage
2132   \else\if@openright
2133     \cleardoublepage
2134   \else
2135     \clearpage
2136   \fi\fi
2137   \@mainmatterfalse}
2138 %</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。

```
2139 %<!*book&!report>
2140 \newcommand\part{%
2141   \if@noskipsec \leavevmode \fi
2142   \par
2143   \addvspace{4ex}%
2144   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2145   \secdef\@part\@spart}
2146 %</!*book&!report>
```

`book` および `report` クラスの場合は、少し複雑です。

```
2147 %<*book|report>
2148 \newcommand\part{%
2149   \if@openleft
2150     \cleardoublepage
2151   \else\if@openright
2152     \cleardoublepage
2153   \else
2154     \clearpage
2155   \fi\fi
2156   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2157   \if@twocolumn
2158     \onecolumn
2159     \@restonecoltrue
2160   \else
2161     \@restonecolfalse
2162   \fi
2163   \null\vfil
2164   \secdef\@part\@spart}
2165 %</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。

```
2166 %<!*book&!report>
2167 \def\@part[#1]#2{%
2168   \ifnum \c@secnumdepth >\m@ne
2169     \refstepcounter{part}%
2170     \addcontentsline{toc}{part}{%
2171       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2172   \else
```

```

2173 \addcontentsline{toc}{part}{#1}%
2174 \fi
2175 \markboth{}{}%
2176 {\parindent\z@
2177 \raggedright
2178 \interlinepenalty \@M
2179 \normalfont
2180 \ifnum \c@secnumdepth >\m@ne
2181 \Large\headfont\prepartname\thepart\postpartname
2182 \par\nobreak
2183 \fi
2184 \huge \headfont #2%
2185 \markboth{}{}\par}%
2186 \nobreak
2187 \vskip 3ex
2188 \@afterheading}
2189 %</*!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。

```

2190 %<*book|report>
2191 \def\@part[#1]#2{%
2192 \ifnum \c@secnumdepth >-2\relax
2193 \refstepcounter{part}%
2194 \addcontentsline{toc}{part}{%
2195 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2196 \else
2197 \addcontentsline{toc}{part}{#1}%
2198 \fi
2199 \markboth{}{}%
2200 {\centering
2201 \interlinepenalty \@M
2202 \normalfont
2203 \ifnum \c@secnumdepth >-2\relax
2204 \huge\headfont \prepartname\thepart\postpartname
2205 \par\vskip20\p@?
2206 \fi
2207 \Huge \headfont #2\par}%
2208 \@endpart}
2209 %</book|report>

```

\@spart 番号を付けない部です。

```

2210 %<*!book&!report>
2211 \def\@spart#1{%
2212 \parindent \z@ \raggedright
2213 \interlinepenalty \@M
2214 \normalfont
2215 \huge \headfont #1\par}%
2216 \nobreak
2217 \vskip 3ex

```



```

2218 \afterheading}
2219 %</*!book&!report>
2220 %<*book|report>
2221 \def\@spart#1{%
2222     \centering
2223     \interlinepenalty \@M
2224     \normalfont
2225     \Huge \headfont #1\par}%
2226 \@endpart}
2227 %</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。

```

2228 %<*book|report>
2229 \def\@endpart{\vfil\newpage
2230     \if@twoside
2231     \if@openleft %% added (2017/02/24)
2232     \null\thispagestyle{empty}\newpage
2233     \else\if@openright %% added (2016/12/13)
2234     \null\thispagestyle{empty}\newpage
2235     \fi\fi %% added (2016/12/13, 2017/02/24)
2236     \fi
2237     \if@restonecol
2238     \twocolumn
2239     \fi}
2240 %</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。

```

2241 %<*book|report>
2242 \newcommand{\chapter}{%
2243     \if@openleft\cleardoublepage\else
2244     \if@openright\cleardoublepage\else\clearpage\fi\fi
2245     \plainifnotempty % 元: \thispagestyle{plain}
2246     \global\@topnum\z@
2247     \if@english \@afterindentfalse \else \@afterindenttrue \fi
2248     \secdef
2249     {\@omit@numberfalse\@chapter}%
2250     {\@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。

```

2251 \def\@chapter[#1]#2{%
2252   \ifnum \c@secnumdepth >\m@ne
2253     \if@mainmatter
2254       \refstepcounter{chapter}%
2255       \typeout{\@chapapp\thechapter\@chappos}%
2256       \addcontentsline{toc}{chapter}%
2257         {\protect\numberline
2258 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2259       {\@chapapp\thechapter\@chappos}%
2260       #1}%
2261     \else\addcontentsline{toc}{chapter}{#1}\fi
2262   \else
2263     \addcontentsline{toc}{chapter}{#1}%
2264   \fi
2265   \chaptermark{#1}%
2266   \addtocontents{lof}{\protect\addvspace{10\jsc@mp}}%
2267   \addtocontents{lot}{\protect\addvspace{10\jsc@mp}}%
2268   \if@twocolumn
2269     \@topnewpage[\@makechapterhead{#2}]%
2270   \else
2271     \@makechapterhead{#2}%
2272     \@afterheading
2273   \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。

```

2274 \def\@makechapterhead#1{%
2275   \vspace*{2\Cvs}% 欧文は 50pt
2276   {\parindent \z@ \raggedright \normalfont
2277     \ifnum \c@secnumdepth >\m@ne
2278       \if@mainmatter
2279         \huge\headfont \@chapapp\thechapter\@chappos
2280         \par\nobreak
2281         \vskip \Cvs % 欧文は 20pt
2282       \fi
2283     \fi
2284     \interlinepenalty\@M
2285     \Huge \headfont #1\par\nobreak
2286     \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。

```

2287 \def\@schapter#1{%
2288   \chaptermark{#1}%
2289   \if@twocolumn
2290     \@topnewpage[\@makeschapterhead{#1}]%
2291   \else
2292     \@makeschapterhead{#1}\@afterheading
2293   \fi}

```

\@makeschapterhead 番号なしの章見出しです。

```

2294 \def\@makeschapterhead#1{%
2295   \vspace*{2\Cvs}% 欧文は 50pt
2296   {\parindent \z@ \raggedright
2297     \normalfont
2298     \interlinepenalty\@M
2299     \Huge \headfont #1\par\nobreak
2300     \vskip 3\Cvs}} % 欧文は 40pt
2301 %</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。

```

2302 \if@twocolumn
2303   \newcommand{\section}{%
2304     %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2305     \@startsection{section}{1}{\z@}%
2306     %<!kiyou>    {0.6\Cvs}{0.4\Cvs}%
2307     %<kiyou>    {\Cvs}{0.5\Cvs}%
2308     %    {\normalfont\large\headfont\@secapp}%
2309     {\normalfont\large\headfont\raggedright}}
2310 \else
2311   \newcommand{\section}{%
2312     \if@slide\clearpage\fi
2313     \@startsection{section}{1}{\z@}%
2314     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2315     {.5\Cvs \@plus.3\Cdp}% 後アキ
2316     %    {\normalfont\Large\headfont\@secapp}%
2317     {\normalfont\Large\headfont\raggedright}}
2318 \fi

```

`\subsection` 同上です。

```

2319 \if@twocolumn
2320   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2321     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2322     {\normalfont\normalsize\headfont}}
2323 \else
2324   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2325     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2326     {.5\Cvs \@plus.3\Cdp}% 後アキ
2327     {\normalfont\large\headfont}}
2328 \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)。

```

2329 \if@twocolumn
2330   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%

```

```

2331    {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2332    {\normalfont\normalsize\headfont}}
2333 \else
2334    \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2335    {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2336    {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2337    {\normalfont\normalsize\headfont}}
2338 \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 `\jachar` と同義になるが、それ以外は何もしない。

```

2339 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2340 \let\bxjs@org@paragraph@mark\jsParagraphMark
2341 \ifx\bxjs@paragraph@mark\@empty
2342   \let\jsParagraphMark\@empty
2343 \else\ifx\bxjs@paragraph@mark\undefined\else
2344   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2345 \fi\fi
2346 \let\jsJaChar\@empty
2347 \if@twocolumn
2348   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2349   {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2350 %<jspf>    {\normalfont\normalsize\headfont}}
2351 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2352 \else
2353   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2354   {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2355   {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2356 %<jspf>    {\normalfont\normalsize\headfont}}
2357 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2358 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```
2359 \if@twocolumn
```

```

2360 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2361     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2362     {\normalfont\normalsize\headfont}}
2363 \else
2364 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2365     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2366     {\normalfont\normalsize\headfont}}
2367 \fi

```

8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2368 \if@slide
2369 \setlength\leftmargini{1\jsZw}
2370 \else
2371 \if@twocolumn
2372 \setlength\leftmargini{2\jsZw}
2373 \else
2374 \setlength\leftmargini{3\jsZw}
2375 \fi
2376 \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 2377 \if@slide
\leftmarginiv 2378 \setlength\leftmarginii {1\jsZw}
\leftmarginiv 2379 \setlength\leftmarginiii {1\jsZw}
\leftmarginiv 2380 \setlength\leftmarginiv {1\jsZw}
2381 \setlength\leftmarginv {1\jsZw}
2382 \setlength\leftmarginvi {1\jsZw}
2383 \else
2384 \setlength\leftmarginii {2\jsZw}
2385 \setlength\leftmarginiii {2\jsZw}
2386 \setlength\leftmarginiv {2\jsZw}
2387 \setlength\leftmarginv {1\jsZw}
2388 \setlength\leftmarginvi {1\jsZw}
2389 \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。

```

2390 \setlength \labelsep {0.5\jsZw} % .5em

```

```

2391 \setlength \labelwidth{\leftmarginI}
2392 \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。

```

2393 \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 2394 \@beginparpenalty -\@lowpenalty
\@itempenalty 2395 \@endparpenalty -\@lowpenalty
2396 \@itempenalty -\@lowpenalty

```

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

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

```

2397 \def\@listI{\leftmargin\leftmarginI
2398   \parsep \z@
2399   \topsep 0.5\baselineskip
2400   \itemsep \z@ \relax}
2401 \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）。

```

2402 \@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 2403 \def\@listII{\leftmargin\leftmarginII
\@listIV 2404   \labelwidth\leftmarginII \advance\labelwidth-\labelsep
2405   \topsep \z@
\@listV 2406   \parsep \z@
\@listVI 2407   \itemsep\parsep}
2408 \def\@listIII{\leftmargin\leftmarginIII
2409   \labelwidth\leftmarginIII \advance\labelwidth-\labelsep
2410   \topsep \z@
2411   \parsep \z@
2412   \itemsep\parsep}
2413 \def\@listIV {\leftmargin\leftmarginIV
2414   \labelwidth\leftmarginIV
2415   \advance\labelwidth-\labelsep}
2416 \def\@listV {\leftmargin\leftmarginV
2417   \labelwidth\leftmarginV
2418   \advance\labelwidth-\labelsep}
2419 \def\@listVI {\leftmargin\leftmarginVI

```

```
2420 \labelwidth\leftmarginvi
2421 \advance\labelwidth-\labelsep}
```

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

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

```
2422 \renewcommand{\theenumi}{\@arabic\c@enumi}
2423 \renewcommand{\theenumii}{\@alph\c@enumii}
2424 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2425 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

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

`\labelenumiv`

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

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

```
2426 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2427 \newcommand*{\jsInJaParen}[1]{%
2428   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2429 \newcommand{\labelenumi}{\theenumi.}
2430 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2431 \newcommand{\labelenumiii}{\theenumiii.}
2432 \newcommand{\labelenumiv}{\theenumiv.}
```

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

```
\p@enumiv 2433 \renewcommand{\p@enumii}{\theenumi}
2434 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2435 \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 2436 \newcommand\labelitemi{\textbullet}
2437 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 2438 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2439 \newcommand\labelitemiv{\textperiodcentered}
```

■description 環境

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

```
2440 \newenvironment{description}{%
2441   \list{}{%
2442     \labelwidth=\leftmargin
2443     \labelsep=1\jsZw
2444     \advance \labelwidth by -\labelsep
2445     \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。

```
2446 \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。

```
2447 %<*book|report>
2448 \newenvironment{chapterabstract}{%
2449   \begin{list}{}{%
2450     \listparindent=1\jsZw
2451     \itemindent=\listparindent
2452     \rightmargin=0pt
2453     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2454 %</book|report>
```

“普通の” **abstract** 環境の定義。

```
2455 %<*article|report|slide>
2456 \newbox\@abstractbox
2457 \if@titlepage
2458   \newenvironment{abstract}{%
```



```

2459 \titlepage
2460 \null\vfil
2461 \@beginparpenalty\@lowpenalty
2462 \begin{center}%
2463 \headfont \abstractname
2464 \endparpenalty\@M
2465 \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。

```

2466 \par}%
2467 {\par\vfil\null\endtitlepage}
2468 \else
2469 \newenvironment{abstract}{%
2470 \if@twocolumn
2471 \ifx\maketitle\relax
2472 \section*{\abstractname}%
2473 \else
2474 \global\setbox\@abstractbox\hbox\bgroup
2475 \begin{minipage}[b]{\textwidth}
2476 \small\parindent1\jsZw
2477 \begin{center}%
2478 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2479 \end{center}%
2480 \list{}{%
2481 \listparindent\parindent
2482 \itemindent \listparindent
2483 \rightmargin \leftmargin}%
2484 \item\relax
2485 \fi
2486 \else
2487 \small
2488 \begin{center}%
2489 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2490 \end{center}%
2491 \list{}{%
2492 \listparindent\parindent
2493 \itemindent \listparindent
2494 \rightmargin \leftmargin}%
2495 \item\relax
2496 \fi}{\if@twocolumn
2497 \ifx\maketitle\relax
2498 \else
2499 \endlist\end{minipage}\egroup
2500 \fi
2501 \else
2502 \endlist
2503 \fi}
2504 \fi
2505 %</article|report|slide>

```

```

2506 %<*jspf>
2507 \newbox\@abstractbox
2508 \newenvironment{abstract}{%
2509   \global\setbox\@abstractbox\hbox\bgroup
2510   \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2511     \small
2512     \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2513   {\end{minipage}\egroup}
2514 %</jspf>

```

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

```

2515 %<*book|report>
2516 \ifbxjs@force@chapterabstract
2517   \let\abstract\chapterabstract
2518   \let\endabstract\endchapterabstract
2519 \fi
2520 %</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。

```

2521 %<*jspf>
2522 %\newbox\@keywordsbox
2523 %\newenvironment{keywords}{%
2524 %  \global\setbox\@keywordsbox\hbox\bgroup
2525 %  \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2526 %    \small\parindent0\jsZw}%
2527 %  {\end{minipage}\egroup}
2528 %</jspf>

```

■verse 環境

verse 詩のための `verse` 環境です。

```

2529 \newenvironment{verse}{%
2530   \let \\\=\@centercr
2531   \list{}{%
2532     \itemsep \z@
2533     \itemindent -2\jsZw % 元: -1.5em
2534     \listparindent\itemindent
2535     \rightmargin \z@
2536     \advance\leftmargin 2\jsZw}% 元: 1.5em
2537   \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。

```
2538 \newenvironment{quotation}{%
2539   \list{}{%
2540     \listparindent\parindent
2541     \itemindent\listparindent
2542     \rightmargin \z@}%
2543   \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。

```
2544 \newenvironment{quote}%
2545   {\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。

```
2546 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2547   \item[\hskip \labelsep{\headfont #1\ #2}]}
2548 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2549   \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。

```
2550 \newenvironment{titlepage}{%
2551   %<book>   \pltx@cleartooddpage %% 2017-02-24
2552   \if@twocolumn
2553     \@restonecoltrue\onecolumn
2554   \else
2555     \@restonecolfalse\newpage
2556   \fi
2557   \thispagestyle{empty}}%
```

```

2558 \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2559 }%
2560 {\if@restonecol\twocolumn \else \newpage \fi
2561 \if@twoside\else
2562 \setcounter{page}\@ne
2563 \fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```

2564 %<*&book&!report>
2565 \newcommand{\appendix}{\par
2566 \setcounter{section}{0}%
2567 \setcounter{subsection}{0}%
2568 \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2569 \gdef\presectionname{\appendixname}%
2570 \gdef\postsectionname{}}%
2571 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2572 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2573 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2574 \else
2575 \gdef\@secapp{\appendixname}%
2576 \gdef\@secpos{}}%
2577 \gdef\thesection{\@Alph\c@section}%
2578 \fi}
2579 %</&book&!report>
2580 %<*&book|report>
2581 \newcommand{\appendix}{\par
2582 \setcounter{chapter}{0}%
2583 \setcounter{section}{0}%
2584 \gdef\@chapapp{\appendixname}%
2585 \gdef\@chappos{}}%
2586 \gdef\thechapter{\@Alph\c@chapter}}
2587 %</&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。

```
2588 \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。

```
2589 \setlength\tabcolsep{6\p@?}
```

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

```
2590 \setlength\arrayrulewidth{.4\p@}
```

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

```
2591 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \ ' コマンドで入るアキです。

```
2592 \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。

```
2593 \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。

```
2594 \setlength\fboxsep{3\p@?}
```

```
2595 \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。

```
2596 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```
2597 %<*book|report>
```

```
2598 \@addtoreset{equation}{chapter}
```

```
2599 \renewcommand\theequation
```

```
2600 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

```
2601 %</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。

```
2602 % \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。

```
2603 % \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。

```
2604 % \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。

```
2605 %<!*book&!report>
2606 \newcounter{figure}
2607 \renewcommand \thefigure {\@arabic\c@figure}
2608 %<!/book&!report>
2609 %<*book|report>
2610 \newcounter{figure}[chapter]
2611 \renewcommand \thefigure
2612     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2613 %</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 2614 \def\fps@figure{tbp}
2615 \def\ftype@figure{1}
\fnum@figure 2616 \def\ext@figure{lof}
2617 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2618 \newenvironment{figure}%
2619     {\@float{figure}}%
2620     {\end@float}
2621 \newenvironment{figure*}%
2622     {\@dblfloat{figure}}%
2623     {\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。

```

2624 %<!*book&!report>
2625 \newcounter{table}
2626 \renewcommand\thetable{\@arabic\c@table}
2627 %</!book&!report>
2628 %<*book|report>
2629 \newcounter{table}[chapter]
2630 \renewcommand \thetable
2631     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2632 %</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 2633 \def\fps@table{tbp}
2634 \def\ftype@table{2}
\fnun@table 2635 \def\ext@table{lot}
2636 \def\fnun@table{\tablename\nobreak\thetable}

```

table * は段抜きのフロートです。

```

table* 2637 \newenvironment{table}%
2638     {\@float{table}}%
2639     {\end@float}
2640 \newenvironment{table*}%
2641     {\@dblfloat{table}}%
2642     {\end@dblfloat}

```

8.6 キャプション

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

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

```

2643 \newlength\abovecaptionskip
2644 \newlength\belowcaptionskip
2645 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2646 \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。

```

2647 %<!*jspf>
2648 % \long\def\@makecaption#1#2{\small
2649 %     \advance\leftskip10\jsc@mmm
2650 %     \advance\rightskip10\jsc@mmm
2651 %     \vskip\abovecaptionskip
2652 %     \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%

```

```

2653 % \ifdim \wd\@tempboxa >\hsize
2654 % #1\hskip1\jsZw\relax #2\par
2655 % \else
2656 % \global \@minipagefalse
2657 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2658 % \fi
2659 % \vskip\belowcaptionskip}}
2660 \long\def\@makecaption#1#2{\small
2661 \advance\leftskip .0628\linewidth
2662 \advance\rightskip .0628\linewidth
2663 \vskip\abovecaptionskip
2664 \sbox\@tempboxa{#1\zwspace#2}%
2665 \ifdim \wd\@tempboxa <\hsize \centering \fi
2666 #1\zwspace#2\par
2667 \vskip\belowcaptionskip}}
2668 %</!jspf>
2669 %<*jspf>
2670 \long\def\@makecaption#1#2{%
2671 \vskip\abovecaptionskip
2672 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2673 \ifdim \wd\@tempboxa >\hsize
2674 {\small\sffamily
2675 \list{#1}{%
2676 \renewcommand{\makelabel}[1]{##1\hfil}
2677 \itemsep \z@
2678 \itemindent \z@
2679 \labelsep \z@
2680 \labelwidth 11\jsc@mmm
2681 \listparindent\z@
2682 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2683 \else
2684 \global \@minipagefalse
2685 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2686 \fi
2687 \vskip\belowcaptionskip}
2688 %</jspf>

```

9 フォントコマンド

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

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

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

```

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

```

```

2689 \newif\if@jsc@warnoldfontcmd
2690 \@jsc@warnoldfontcmdtrue
2691 \newif\if@jsc@warnoldfontcmdexception
2692 \@jsc@warnoldfontcmdexceptionfalse

```

```

\jsc@DeclareOldFontCommand

```

```

2693 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2694   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2695   \DeclareOldFontCommand{#1}{%
2696     \bxjs@oldfontcmd{#1}#2%
2697   }{%
2698     \bxjs@oldfontcmd{#1}#3%
2699   }%
2700 }
2701 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2702   \ClassInfo\bxjs@clsname
2703   {Old font command '\string#1' is used!!\MessageBreak
2704     The first occurrence is}%
2705 }

```

```

\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。

```

```

2706 \newcommand*{\allowoldfontcommands}{%
2707   \@jsc@warnoldfontcmdexceptiontrue}
2708 \newcommand*{\disallowoldfontcommands}{%
2709   \@jsc@warnoldfontcmdexceptionfalse}

2710 \let\bxjs@oldfontcmd@list\@empty
2711 \def\bxjs@oldfontcmd#1{%
2712   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2713 \def\bxjs@oldfontcmd@a#1#2{%
2714   \if@jsc@warnoldfontcmdexception\else
2715     \global\@jsc@warnoldfontcmdfalse
2716     \ifx#1\relax
2717       \global\let#1=t%
2718       \jsc@warnoldfontcmd{#2}%
2719     \fi
2720   \fi}
2721 \def\bxjs@warnoldfontcmd@final{%
2722 % \par
2723   \global\let\bxjs@warnoldfontcmd@final\@empty

```

```

2724 \let\@tempa\@empty
2725 \def\do##1{%
2726   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2727     \edef\@tempa{\@tempa \space\string##1}\fi}
2728 \bxjs@oldfontcmd@list
2729 \ifx\@tempa\@empty\else
2730   \ClassWarningNoLine\bxjs@clsname
2731     {Some old font commands were used in text:\MessageBreak
2732       \space\@tempa\MessageBreak
2733       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2734       new font selection scheme called NFSS2 with several\MessageBreak
2735       new, combinable font commands. The
2736       class provides\MessageBreak
2737       the old font commands only for compatibility}
2738 \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。

```

2739 \def\bxjs@warnoldfontcmd@kick@final{%
2740   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2741 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2742 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2743 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 2744 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf 2745 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2746 \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。

```

2747 \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。

```

2748 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2749 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2750 \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 2751 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2752 \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。

```
2753 \newcommand\@pnumwidth{1.55em}
2754 \newcommand\@tocrmarg{2.55em}
2755 \newcommand\@dotsep{4.5}
2756 %<!book&!report>\setcounter{tocdepth}{2}
2757 %<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)

```
2758 \newdimen\jsc@tocl@width
2759 \newcommand{\tableofcontents}{%
2760 %<*book|report>
2761   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2762   \settowidth\@tempdima{\headfont\appendixname}%
2763   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2764   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2765   \if@twocolumn
2766     \@restonecoltrue\onecolumn
2767   \else
2768     \@restonecolfalse
2769   \fi
2770   \chapter*{\contentsname}%
2771   \@mkboth{\contentsname}{}%
2772 %</book|report>
2773 %<*!book&!report>
2774   \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2775   \settowidth\@tempdima{\headfont\appendixname}%
2776   \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2777   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2778   \section*{\contentsname}%
2779   \@mkboth{\contentsname}{\contentsname}%
2780 %</!book&!report>
2781   \@starttoc{toc}%
2782 %<book|report> \if@restonecol\twocolumn\fi
2783 }
```

`\l@part` 部の目次です。

```
2784 \newcommand*{\l@part}[2]{%
2785   \ifnum \c@tocdepth >-2\relax
2786 %<!book&!report> \addpenalty\@secpenalty
2787 %<book|report> \addpenalty{-\@highpenalty}%
2788   \addvspace{2.25em \@plus\p@}%
2789   \begingroup
2790     \parindent \z@
2791 % \@pnumwidth should be \@tocrmarg
2792 % \rightskip \@pnumwidth
2793 \rightskip \@tocrmarg
2794 \parfillskip -\rightskip
2795 {\leavevmode
2796   \large \headfont
2797   \setlength\@lnumwidth{4\jsZw}%
2798   #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2799   \nobreak
2800 %<book|report> \global\@nobreaktrue
2801 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
```

```

2802 \endgroup
2803 \fi}

```

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

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

```

2804 %<*book|report>
2805 \newcommand*{\l@chapter}[2]{%
2806 \ifnum \c@tocdepth >\m@ne
2807 \addpenalty{-\@highpenalty}%
2808 \addvspace{1.0em \@plus\p@?}
2809 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2810 \begingroup
2811 \parindent\z@
2812 % \rightskip\@pnumwidth
2813 \rightskip\@tocrmarg
2814 \parfillskip-\rightskip
2815 \leavevmode\headfont
2816 % % \if@english\setlength\@l@numwidth{5.5em}\else\setlength\@l@numwidth{4.683\jsZw}\fi
2817 \setlength\@l@numwidth{\jsc@tocl@width}\advance\@l@numwidth 2.683\jsZw
2818 \advance\leftskip\@l@numwidth \hskip-\leftskip
2819 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2820 \penalty\@highpenalty
2821 \endgroup
2822 \fi}
2823 %</book|report>

```

`\l@section` 節の目次です。

```

2824 %<!*book&!report>
2825 \newcommand*{\l@section}[2]{%
2826 \ifnum \c@tocdepth >\z@
2827 \addpenalty{\@secpenalty}%
2828 \addvspace{1.0em \@plus\p@?}%
2829 \begingroup
2830 \parindent\z@
2831 % \rightskip\@pnumwidth
2832 \rightskip\@tocrmarg
2833 \parfillskip-\rightskip
2834 \leavevmode\headfont
2835 % % \setlength\@l@numwidth{4\jsZw}% 元 1.5em [2003-03-02]
2836 \setlength\@l@numwidth{\jsc@tocl@width}\advance\@l@numwidth 2\jsZw
2837 \advance\leftskip\@l@numwidth \hskip-\leftskip
2838 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2839 \endgroup
2840 \fi}
2841 %</!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。

```

2842 %<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@section さらに下位レベルの目次項目の体裁です。あまり使ったことがありませんので、要修正かも
\l@subsubsection しれません。

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

```
\l@subparagraph 2843 %<!*book&!report>
2844 % \newcommand*{\l@section} {\@dottedtocline{2}{1.5em}{2.3em}}
2845 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2846 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2847 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2848 %
2849 % \newcommand*{\l@section} {\@dottedtocline{2}{1zw}{3zw}}
2850 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2851 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2852 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2853 %
2854 \newcommand*{\l@section}{%
2855     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2856     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2857 \newcommand*{\l@subsubsection}{%
2858     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2859     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2860 \newcommand*{\l@paragraph}{%
2861     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2862     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2863 \newcommand*{\l@subparagraph}{%
2864     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2865     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2866 %</!*book&!report>
2867 %<!*book|report>
2868 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2869 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2870 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2871 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2872 \newcommand*{\l@section}{%
2873     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2874     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2875 \newcommand*{\l@section}{%
2876     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2877     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2878 \newcommand*{\l@subsubsection}{%
2879     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2880     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2881 \newcommand*{\l@paragraph}{%
2882     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2883     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2884 \newcommand*{\l@subparagraph}{%
2885     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
```

```

2886 \dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2887 %</book|report>

```

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

```

2888 \newdimen\@lnumwidth
2889 \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。

```

2890 \def\jsTocLine{\leaders\hbox{%
2891   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$\}\hfill}
2892 \def\dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2893   \vskip \z@ \@plus.2\p@?
2894   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2895    \parindent #2\relax\@afterindenttrue
2896    \interlinepenalty\@M
2897    \leavevmode
2898    \@lnumwidth #3\relax
2899    \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2900    {#4}\nobreak
2901    \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2902      \hfil\normalfont \normalcolor #5\}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2903 \newcommand{\listoffigures}{%
2904 %<*book|report>
2905 \if@twocolumn\@restonecoltrue\onecolumn
2906 \else\@restonecolfalse\fi
2907 \chapter*{\listfigurename}%
2908 \mkboth{\listfigurename}{}%
2909 %</book|report>
2910 %<!*book&!report>
2911 \section*{\listfigurename}%
2912 \mkboth{\listfigurename}{\listfigurename}%
2913 %</*!book&!report>
2914 \@starttoc{lof}%

```

```

2915 %<book|report> \if@restonecol\twocolumn\fi
2916 }

```

`\l@figure` 図目次の項目を出力します。

```

2917 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2918 \newcommand{\listoftables}{%
2919 %<*book|report>
2920 \if@twocolumn\@restonecoltrue\onecolumn
2921 \else\@restonecolfalse\fi
2922 \chapter*{\listtablename}%
2923 \@mkboth{\listtablename}{}%
2924 %</book|report>
2925 %<!*book&!report>
2926 \section*{\listtablename}%
2927 \@mkboth{\listtablename}{\listtablename}%
2928 %<!/book&!report>
2929 \@starttoc{lot}%
2930 %<book|report> \if@restonecol\twocolumn\fi
2931 }

```

`\l@table` 表目次は図目次と同じです。

```

2932 \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。

```

2933 \newdimen\bibindent
2934 \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。

```

2935 \newenvironment{thebibliography}[1]{%
2936 \@jsc@warnoldfontcmdexceptiontrue
2937 \global\let\presectionname\relax
2938 \global\let\postsectionname\relax
2939 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2940 %<*kiyou>
2941 \vspace{1.5\baselineskip}
2942 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2943 \vspace{0.5\baselineskip}
2944 %</kiyou>
2945 %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2946 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%

```



```

2947 \list{\@biblabel{\@arabic\c@enumiv}}%
2948 {\settowidth\labelwidth{\@biblabel{#1}}%
2949 \leftmargin\labelwidth
2950 \advance\leftmargin\labelsep
2951 \@openbib@code
2952 \usecounter{enumiv}%
2953 \let\p@enumiv\@empty
2954 \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2955 %<kiyou> \small
2956 \sloppy
2957 \clubpenalty4000
2958 \@clubpenalty\clubpenalty
2959 \widowpenalty4000%
2960 \sfcode`.\@m}
2961 {\def\@noitemerr
2962 {\@latex@warning{Empty `thebibliography' environment}}}%
2963 \endlist}

```

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

```

2964 \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。

```

2965 \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。

```

2966 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

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

```

2967 % \def\@citex[#1]#2{\leavevmode
2968 % \let\@citea\@empty
2969 % \@cite{\@for\@citeb:=#2\do
2970 % {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2971 % \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2972 % \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2973 % \ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2974 % \G@refundefinedtrue
2975 % \@latex@warning
2976 % {Citation `{\@citeb' on page \thepage \space undefined}}%
2977 % {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2978 % \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。

```
2979 % \DeclareRobustCommand\cite{\unskip
2980 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2981 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempwa
2982 %   , \jsInhibitGlue\ #2\fi}} }$}
```

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)。

```
2983 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2984   \if@twocolumn
2985     \onecolumn\@restonecolfalse
2986   \else
2987     \clearpage\@restonecoltrue
2988   \fi
2989   \columnseprule.4pt \columnsep 2\jsZw
2990   \ifx\multicols\@undefined
2991 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2992 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2993 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2994 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2995   \else
2996     \ifdim\textwidth<\fullwidth
2997       \setlength{\evensidemargin}{\oddsidemargin}
2998       \setlength{\textwidth}{\fullwidth}
2999       \setlength{\linewidth}{\fullwidth}
3000 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
3001 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
3002 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
3003 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
3004   \else
3005 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}%
3006 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
3007 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
3008 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
3009   \fi
3010   \fi
3011 %<book|report>      \@mkboth{\indexname}{}%
3012 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
3013   \plainifnotempty % \thispagestyle{plain}
3014   \parindent\z@
3015   \parskip\z@ \@plus .3\p@?\relax
3016   \let\item\@idxitem
3017   \raggedright
3018   \footnotesize\narrowbaselines
3019 }{
```

```

3020 \ifx\multicols\@undefined
3021 \if@restonecol\onecolumn\fi
3022 \else
3023 \end{multicols}
3024 \fi
3025 \clearpage
3026 }

```

`\@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 3027 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 3028 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
3029 \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。

```

3030 \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* 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。

```

3031 \newcommand\seename{\if@english see\else →\fi}
3032 \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。

```

3033 \ifx\pltx@foot@penalty\@undefined
3034 \let\footnotes@ve=\footnote
3035 \def\footnote{\jsInhibitGlue\footnotes@ve}
3036 \let\footnotemarks@ve=\footnotemark
3037 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
3038 \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 p_AT_EX の変更に従いました (Thanks: 角藤さん)。p_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。

p_TE_X 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
3039 \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。

```
3040 % \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。

```
3041 \renewcommand{\footnoterule}{%
3042   \kern-2.6\p@? \kern-.4\p@
3043   \hrule width .4\columnwidth
3044   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
3045 %<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 のバグ修正に従いました。

[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。

```
3046 \long\def\@footnotetext{%
3047   \insert\footins\bgroup
3048   \normalfont\footnotesize
3049   \interlinepenalty\interfootnotelinepenalty
3050   \splittopskip\footnotesep
3051   \splitmaxdepth \dp\strutbox \floatingpenalty \MM
3052   \hsize\columnwidth \@parboxrestore
3053   \protected@edef\@currentlabel{%
3054     \csname p@footnote\endcsname\@thefnmark
```

```

3055 }%
3056 \color@begingroup
3057 \makefnintext{%
3058 \rule{z@footnotesep\ignorespaces}%
3059 \futurelet\jsc@next\jsc@fo@t}
3060 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@f@t
3061 \else \let\jsc@next\jsc@f@t\fi \jsc@next}
3062 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3063 \def\jsc@f@t#1{#1\jsc@@foot}
3064 \def\jsc@@foot{\finalstrut\strutbox\color@endgroup\egroup
3065 \ifx\pltx@foot@penalty\undefined\else
3066 \ifhmode\null\fi
3067 \ifnum\pltx@foot@penalty=z@\else
3068 \penalty\pltx@foot@penalty
3069 \pltx@foot@penaltyz@
3070 \fi
3071 \fi}

```

`\makefnintext` 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。

```

3072 \newcommand\makefnintext[1]{%
3073 \advance\leftskip 3\jsZw
3074 \parindent 1\jsZw
3075 \noindent
3076 \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。

```

3077 % \def\xfootnotenext[#1]{%
3078 % \begingroup
3079 % \ifnum#1>z@
3080 % \csname c@\mpfn\endcsname #1\relax
3081 % \unrestored@protected@xdef\thefnmark{\thempfn}%
3082 % \else
3083 % \unrestored@protected@xdef\thefnmark{}%
3084 % \fi
3085 % \endgroup
3086 % \@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。

```
3087 \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。

```
3088 \def\everyparhook{\jsInhibitGlueAtParTop}
3089 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
3090 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3091 \fi
```

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

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

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

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

```
3092 \ifx j\jsEngine
3093 \def\@inhibitglue{%
3094   \futurelet\@let@token\@@inhibitglue}
3095 \begingroup
3096 \let\GDEF=\gdef
3097 \let\CATCODE=\catcode
3098 \let\ENDGROUP=\endgroup
3099 \CATCODE`k=12
3100 \CATCODE`a=12
3101 \CATCODE`n=12
3102 \CATCODE`j=12
3103 \CATCODE`i=12
3104 \CATCODE`c=12
3105 \CATCODE`h=12
3106 \CATCODE`r=12
3107 \CATCODE`t=12
```

```

3108 \CATCODE`e=12
3109 \GDEF\KANJI@CHARACTER{kanji character }
3110 \ENDGROUP
3111 \def\@@inhibitglue{%
3112   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
3113 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3114   \def\jsc@ig@temp{#1}%
3115   \ifx\jsc@ig@temp\@empty
3116     \ifnum\the\inhibitxspcode`#2=2\relax
3117       \inhibitglue
3118     \fi
3119   \fi}
3120 \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。

```

3121 \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。

```

3122 \def\@doendpe{%
3123   \@endpetrue
3124   \def\par{%
3125     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3126   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3127 \def\@setminipage{%
3128   \@minipagetrue
3129   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3130 }

```

`\item` 命令の直後です。

```

3131 \def\@item[#1]{%
3132   \if@noparitem
3133     \@donoparitem
3134   \else
3135     \if@inlabel
3136       \indent \par
3137     \fi
3138     \ifhmode
3139       \unskip\unskip \par
3140     \fi
3141     \if@newlist

```

```

3142     \if@nobreak
3143     \@nbitem
3144     \else
3145         \addpenalty\@beginparpenalty
3146         \addvspace\@topsep
3147         \addvspace{-\parskip}%
3148     \fi
3149     \else
3150         \addpenalty\@itempenalty
3151         \addvspace\itemsep
3152     \fi
3153     \global\@inlabeltrue
3154 \fi
3155 \everypar{%
3156     \@minipagefalse
3157     \global\@newlistfalse
3158     \if@inlabel
3159         \global\@inlabelfalse
3160         {\setbox\z@\lastbox
3161         \ifvoid\z@
3162             \kern-\itemindent
3163         \fi}%
3164     \box\@labels
3165     \penalty\z@
3166 \fi
3167 \if@nobreak
3168     \@nobreakfalse
3169     \clubpenalty \@M
3170 \else
3171     \clubpenalty \@clubpenalty
3172     \everypar{\everyparhook}%
3173 \fi
3174 \everyparhook}%
3175 \if@noitemarg
3176     \@noitemargfalse
3177     \if@nmbrlist
3178         \refstepcounter\@listctr
3179     \fi
3180 \fi
3181 \sbox\@tempboxa{\makelabel{#1}}%
3182 \global\setbox\@labels\hbox{%
3183     \unhbox\@labels
3184     \hskip \itemindent
3185     \hskip -\labelwidth
3186     \hskip -\labelsep
3187     \ifdim \wd\@tempboxa >\labelwidth
3188         \box\@tempboxa
3189     \else
3190         \hbox to\labelwidth {\unhbox\@tempboxa}%

```



```

3191 \fi
3192 \hskip \labelsep}%
3193 \ignorespaces}

```

二つ挿入した `\everyparhook` のうち後者が `\section` 類の直後に 2 回、前者が 3 回目以降に実行されます。

```

3194 \def\@afterheading{%
3195 \@nobreaktrue
3196 \everypar{%
3197 \if@nobreak
3198 \@nobreakfalse
3199 \clubpenalty \@M
3200 \if@afterindent \else
3201 {\setbox\z@\lastbox}%
3202 \fi
3203 \else
3204 \clubpenalty \@clubpenalty
3205 \everypar{\everyparhook}%
3206 \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

```

3207 \fi

```

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

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

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

```

3208 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none\else
3209 \AtEndOfPackage{%
3210 \def\@gnewline #1{%
3211 \ifvmode
3212 \@nolnerr
3213 \else
3214 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3215 \jsInhibitGlue \ignorespaces
3216 \fi}
3217 }
3218 \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。

```
3219 \if@jslogo
3220   \IfFileExists{jslogo.sty}{%
3221     \RequirePackage{jslogo}%
3222   }{%
3223     \ClassWarningNoLine{bxjs@clsname}
3224     {The package 'jslogo' is not installed.\MessageBreak
3225      It is included in the recent release of\MessageBreak
3226      the 'jsclasses' bundle}
3227   }
3228 \fi
```

13 amsmath との衝突の回避

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

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

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

```
3229 %\let\ltx@ifnextchar\@ifnextchar
3230 %\def\ProvidesFile#1{%
3231 %   \begingroup
3232 %     \catcode`\ 10 %
3233 %     \ifnum \endlinechar<256 %
3234 %       \ifnum \endlinechar>\m@ne
3235 %         \catcode\endlinechar 10 %
3236 %       \fi
3237 %     \fi
3238 %     \@makeother\/%
```

```

3239 % \makeother\&%
3240 % \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 3241 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3242 \newcommand{\postpartname}{\if@english\else 部\fi}
3243 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3244 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3245 \newcommand{\presectionname}{}% 第
\postsectionname 3246 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3247 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3248 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3249 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3250 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3251 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3252 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3253 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3254 %<jspf>\newcommand{\figurename}{Fig.~}
3255 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3256 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3257 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3258 \newcommand{\appendixname}{\if@english \else 付録\fi}
3259 %<book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

環境変数 `SOURCE_DATE_EPOCH` / `FORCE_SOURCE_DATE` が設定されている場合は“今日”が過去・未来の日付になる可能性がある。BXJS クラスでは、和暦の扱いは `bxwareki` パッケージに任せる。

※ 2.0 版より、完全に `bxwareki` に任せる。

```

3260 \onlypreamble\bxjs@decl@Seireki@cmds

```

```

3261 \@tempswafalse
3262 \if p\jsEngine \@tempswatrue \fi
3263 \if n\jsEngine \@tempswatrue \fi
3264 \bxjs@cond\if@tempswa\fi{%
3265 % 欧文 8bitTeX の場合
3266 \newif\ifjsSeireki \jsSeirekitrue
3267 \def\bxjs@decl@Seireki@cmds{%
3268   \def\西暦{\jsSeirekitrue}%
3269   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3270 \def\Seireki{\jsSeirekitrue}
3271 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3272 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3273 \def\bxjs@iai{\noexpand~}
3274 }{%
3275 \newif\if 西暦 \西暦 true
3276 \def\bxjs@decl@Seireki@cmds{%
3277   \def\西暦{\西暦 true}%
3278   \def\和暦{\西暦 false\bxjs@wareki@used}}
3279 \def\Seireki{\西暦 true}
3280 \def\Wareki{\西暦 false\bxjs@wareki@used}
3281 \def\bxjs@if@use@seireki{\bxjs@cond\if 西暦\fi}
3282 \let\bxjs@iai@empty
3283 }
3284 \bxjs@decl@Seireki@cmds
3285 \let\bxjs@unxp@firstofone \let\bxjs@onxp@firstofone
3286 \bxjs@test@engine\unexpanded{%
3287   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}

```

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

```

3288 \newif\ifbxjs@bxwareki@avail
3289 \IfFileExists{bxwareki.sty}{%
3290   \RequirePackage{bxwareki}[]%
3291   \bxjs@bxwareki@availtrue}{%

```

`\bxjs@wareki@used` 和暦が非対応の場合に警告を出す。

```

3292 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\empty
3293 \else
3294   \bxjs@robust@def\bxjs@wareki@used{%
3295     \global\let\bxjs@wareki@used\empty
3296     \ClassWarning\bxjs@clsname
3297       {Wareki mode is not supported, since\MessageBreak
3298         'bxwareki' is unavailable, found}}
3299   \g@addto@macro\bxjs@begin@document@hook{%
3300     \let\bxjs@wareki@used\empty}
3301 \fi

```

`\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。

```

3302 \ifbxjs@bxwareki@avail
3303   \let\jyear\warekiyear
3304   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3305     \newcount\heisei \heisei=\value{warekiyear}
3306   \fi

    bxwareki が使えない場合は西暦表示にする。

3307 \else
3308   \edef\jyear{\the\year \bxjs@iai}
3309 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3310 \let\bxjs@next\relax
3311 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3312   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3313 \fi\fi
3314 \edef\bxjs@today{%
3315   \if@english
3316     \ifcase\month\or
3317       January\or February\or March\or April\or May\or June\or
3318       July\or August\or September\or October\or November\or December\fi
3319     \space\number\day, \number\year
3320   \else
3321     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3322     \else \noexpand\bxjs@if@use@seireki
3323     \fi {%
3324       \number\year\bxjs@iai\bxjs@unxp{年}%
3325       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3326       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3327     }\bxjs@next}%
3328   \fi}
3329 \let\today\bxjs@today

```

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

```

3330 \g@addto@macro\bxjs@begin@document@hook{%
3331   \ifx\bbl@jpn@maybekansuji\@undefined\else
3332     \bxjs@decl@Seireki@cmds
3333     \g@addto@macro\datejapanese{%
3334       \let\today\bxjs@today}%
3335   \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: english)


```

3336 \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。

```
3337 %<slide>\pagestyle{empty}%
3338 %<article|report>\pagestyle{plain}%
3339 %<book>\pagestyle{headings}%
3340 \pagenumbering{arabic}
3341 \if@twocolumn
3342   \twocolumn
3343   \sloppy
3344   \flushbottom
3345 \else
3346   \onecolumn
3347   \raggedbottom
3348 \fi
3349 %<*slide>
3350 \renewcommand\familydefault{\sfdefault}
3351 \raggedright
3352 %</slide>
```

■BXJS 独自の追加処理 

フックを実行する。

```
3353 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3354 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

```
3355 %</class>
```

以上です。

付録 A 和文ドライバの仕様

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

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

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

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

以下で標準で用意されている和文ドライバの実装を示す。

```
3356 %<*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

```
3357 %<*minimal>
```

```
3358 %% 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.
```

```
3359 \def\DeclareJaTextFontCommand#1#2{%
```

```
3360   \DeclareRobustCommand#1[1]{%
```

```
3361     \relax
```

```
3362     \ifmmode \expandafter\nfss@text \fi
```

```
3363     {#2##1}}%
```

```
3364 }
```

```
\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。
```

```
3365 \def\DeclareJaMathFontCommand#1#2{%
```

```
3366   \DeclareRobustCommand#1[1]{%
```

```
3367     \relax
```

```
3368     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3369     \nfss@text{\fontfamily\familydefault
```

```
3370       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3371       #2##1}}%
```



```

3372 }%
3373 }

```

`\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。

```

3374 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3375 \@onlypreamble\bxjs@if@sf@default
3376 \def\bxjs@if@sf@default#1{%
3377   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3378   \g@addto@macro\bxjs@begin@document@hook{%
3379     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3380 }

```

`\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。

```

3381 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3382 \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。

```

3383 \def\jsLetHeadChar#1#2{%
3384   \begingroup
3385     \escapechar=`\ %
3386     \let\bxjs@tmpa={% brace-match-hack
3387       \bxjs@let@hchar@exp#2}%
3388   \endgroup
3389   \let#1\bxjs@g@tmpa}
3390 \def\bxjs@let@hchar@exp{%
3391   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3392 \def\bxjs@let@hchar@exp@a{%
3393   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3394   \bxjs@let@hchar@out\let\relax
3395   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3396   \bxjs@let@hchar@out\let\space%
3397   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3398   \bxjs@let@hchar@out\let\@backslashchar
3399   }{\bxjs@let@hchar@exp@b}}}%
3400 \def\bxjs@let@hchar@exp@b#1{%
3401   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3402 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3403   %\message{<#1#2>}%
3404   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3405 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3406 \bxjs@let@hchar@out\let\relax
3407 }{%else
3408 \expandafter\bxjs@let@hchar@exp
3409 }%
3410 }{%else
3411 \bxjs@let@hchar@chr#1%
3412 }}
3413 \def\bxjs@let@hchar@chr#1{%
3414 \bxjs@let@hchar@out\def{{#1}}}}
3415 \def\bxjs@let@hchar@out#1#2{%
3416 \global#1\bxjs@g@tmpa#2\relax
3417 \toks@{\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3418 \chardef\bxjs@let@hchar@csta=128
3419 \chardef\bxjs@let@hchar@cstb=192
3420 \chardef\bxjs@let@hchar@cstc=224
3421 \chardef\bxjs@let@hchar@cstd=240
3422 \chardef\bxjs@let@hchar@cste=248
3423 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3424 \def\bxjs@let@hchar@chr@ue#1{%
3425 \@tempcnta=`#1\relax
3426 %\message{\the\@tempcnta}%
3427 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3428 \bxjs@let@hchar@chr@ue@a#1%
3429 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3430 \bxjs@let@hchar@out\let\relax
3431 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3432 \bxjs@let@hchar@chr@ue@b
3433 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3434 \bxjs@let@hchar@chr@ue@c
3435 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3436 \bxjs@let@hchar@chr@ue@d
3437 }{%else
3438 \bxjs@let@hchar@out\let\relax
3439 }}}}}
3440 \def\bxjs@let@hchar@chr@ue@a#1{%
3441 \bxjs@let@hchar@out\def{{#1}}}}
3442 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3443 \bxjs@let@hchar@out\def{{#1#2}}}}
3444 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3445 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3446 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3447 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
3448 \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。

```
3449 \def\bxjs@let@hchar@chr@pp#1#2{%
3450   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3451 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3452   \%message{(\meaning#3:\meaning#4)}%
3453   \bxjs@cond\if#1k\fi{%
3454     \bxjs@let@hchar@out\def{#4}}%
3455   }{%else
3456     \bxjs@let@hchar@chr@ue#3#4%
3457   }}
3458 \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。

```
3459 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3460 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3461 \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。

```
3462 \@onlypreamble\bxjs@declarefontshape
3463 \ifjsWithupTeX
3464 \def\bxjs@declarefontshape{%
3465   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3466   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3467   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3468   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3469 }
3470 \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。

```
3471 \else
3472 \def\bxjs@declarefontshape{%
3473   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3474   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3475   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3476 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3477 }
3478 \def\bxjs@sizereference{jis}
3479 \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。

```

3480 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3481   \def\bxjs@tmpb{#5}}
3482 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3483 \else
3484 \expandafter\expandafter\expandafter\bxjs@next
3485 \expandafter\string\the\jfont\relax
3486 \fi
3487 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3488                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3489   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3490    \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。

```

3491 \begingroup
3492 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3493 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3494 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3495 % 幅が丁度 10pt なら補正は不要
3496 \ifdim\wd\z@=10pt
3497   \global\let\bxjs@scale\jsScale
3498 \else
3499 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3500   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3501   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3502   \bxjs@invscale\@tempdima\bxjs@tmpa
3503   \xdef\bxjs@scale{\strip@pt\@tempdima}
3504 \fi
3505 \endgroup
3506 %\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。

```

3507 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3508 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3509 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3510 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3511 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3512 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3513 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3514 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3515 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{f}
3516 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{f}
3517 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{f}
3518 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{f}
3519 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{f}
3520 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{f}
3521 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{f}
3522 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{f}
3523 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3524 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3525 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3526 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3527 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3528 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3529 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3530 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{f}
3531 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{f}
3532 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{f}
3533 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{f}
3534 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{f}
3535 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{f}
3536 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{f}
3537 \DeclareFontShape{\jsc@JTn}{gt}{b}{sl}{<->ssub*gt/bx/n}{f}

```

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

※ 2020-02-02 の NFSS の改修に対する jsclasses の対策を取り入れた。

```

3538 \@ifl@t@r\fmtversion{2020/10/01}
3539   {\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
3540 \ifjsc@needspace@tch          % --- for 2020-02-02 or older BEGIN
3541 \ifx\@rmfamilyhook\@undefined % old
3542 \DeclareRobustCommand\rmfamily
3543   {\not@math@alphabet\rmfamily\mathrm
3544    \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3545 \DeclareRobustCommand\sffamily
3546   {\not@math@alphabet\sffamily\mathsf
3547    \romanfamily\sffamily\kanjifamily\gtdefault\selectfont}
3548 \DeclareRobustCommand\ttfamily
3549   {\not@math@alphabet\ttfamily\mathtt
3550    \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3551 \g@addto@macro\bxjs@begin@document@hook{%
3552   \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3553     % my definitions above should have been overwritten, recover it!
3554     % \selectfont is executed twice but I don't care about speed...
3555     \expandafter\g@addto@macro\csname rmfamily \endcsname

```

```

3556      {\kanjifamily\mcdefault\selectfont}%
3557      \expandafter\g@addto@macro\csname sffamily \endcsname
3558      {\kanjifamily\gtdefault\selectfont}%
3559      \expandafter\g@addto@macro\csname ttfamily \endcsname
3560      {\kanjifamily\gtdefault\selectfont}%
3561      \fi}
3562 \else                                     % 2020-02-02
3563 \g@addto@macro\@rmfamilyhook
3564   {\prepare@family@series@update@kanji{mc}\mcdefault}
3565 \g@addto@macro\@sffamilyhook
3566   {\prepare@family@series@update@kanji{gt}\gtdefault}
3567 \g@addto@macro\@ttfamilyhook
3568   {\prepare@family@series@update@kanji{gt}\gtdefault}
3569 \fi
3570 \else % --- for 2020-02-02 or older END & for 2020-10-01 BEGIN
3571 \AddToHook{rmfamily}%
3572   {\prepare@family@series@update@kanji{mc}\mcdefault}
3573 \AddToHook{sffamily}%
3574   {\prepare@family@series@update@kanji{gt}\gtdefault}
3575 \AddToHook{ttfamily}%
3576   {\prepare@family@series@update@kanji{gt}\gtdefault}
3577 \fi % --- for 2020-10-01 END
3578 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3579 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3580 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3581 \fi
3582 \bxjs@if@sf@default{%
3583   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3584 \selectfont

```

`\bxjs@parse@qh` の処理は p_T_EX 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）。

```

3585 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3586 \let\bxjs@parse@qh@a\undefined
3587 \let\bxjs@parse@qh@b\undefined

```

■パラメタの設定

```

3588 \prebreakpenalty\jis"2147=10000
3589 \postbreakpenalty\jis"2148=10000
3590 \prebreakpenalty\jis"2149=10000
3591 \inhibitxspcode`!=1
3592 \inhibitxspcode`〒=2
3593 \xspcode`+=3
3594 \xspcode`\%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3595 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%

```

```
3596 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
```

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

```
3597 \let\jsInhibitGlueAtParTop\@inhibitglue
```

`\jsResetDimen` は空のままでよい。

■組方向依存の処理 組方向判定の `\if?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。

```
3598 \begingroup
```

```
3599 \catcode`\!=0
```

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

```
3600 \gdef\bxjs@ptex@dir{%
```

```
3601 !iftdir t%
```

```
3602 !else!ifydir y%
```

```
3603 !else ?%
```

```
3604 !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。

```
3605 % 古い \@makefnmark の定義
```

```
3606 \long\def\bxjs@tmpa{\hbox{%
```

```
3607 !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
```

```
3608 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
```

```
3609 \ifx\@makefnmark\bxjs@tmpa
```

```
3610 \long\gdef\@makefnmark{%
```

```
3611 !ifydir \hbox{}\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}%
```

```
3612 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
```

```
3613 \fi
```

エスケープ文字の変更はここまで。

```
3614 \endgroup
```

■`minijs` パッケージのブロック やっておく。

```
3615 \@namedef{ver@minijs.sty}{}
```

B.3 pdfTeX 用の処理

```
3616 \else\ifx p\jsEngine
```

```
3617 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

```
3618 \@onlypreamble\bxjs@cjk@loaded
```

```
3619 \def\bxjs@cjk@loaded{%
```

```
3620 \def\@footnotemark{%
```

```
3621 \leavevmode
```

```
3622 \ifhmode
```

```

3623 \edef\x@sf{\the\spacefactor}%
3624 \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3625 \unkern\unkern
3626 \ifdim\lastskip>\z@ \unskip \fi
3627 \fi\fi
3628 \nobreak
3629 \fi
3630 \@makefnmark
3631 \ifhmode \spacefactor\x@sf \fi
3632 \relax}%
3633 \let\bxjs@cjk@loaded\relax
3634 }
3635 \g@addto@macro\bxjs@begin@document@hook{%
3636 \ifpackageloaded{CJK}{%
3637 \bxjs@cjk@loaded
3638 }{}%
3639 }

```

B.4 X₃TEX 用の処理

```

3640 \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。

```

3641 \def\bxjs@let@hchar@chr#1{%
3642 \tempcnta`#1\relax \divide\tempcnta"800\relax
3643 \bxjs@cond\ifnum\tempcnta=27 \fi{%
3644 \bxjs@let@hchar@chr@xe
3645 }{\bxjs@let@hchar@out\def{{#1}}}}
3646 \def\bxjs@let@hchar@chr@xe#1{%
3647 \lccode`0=`#1\relax
3648 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3649 \@onlypreamble\bxjs@do@precisetext
3650 \ifx\XeTeXgenerateactualtext\@undefined\else
3651 \def\bxjs@do@precisetext{%
3652 \XeTeXgenerateactualtext=\@ne}
3653 \fi

```

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

```

3654 \@onlypreamble\bxjs@do@simplejasetup
3655 \def\bxjs@do@simplejasetup{%
3656 \ifnum\XeTeXinterchartokenstate>\z@
3657 \else\ifnum\stricmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3658 \jsSimpleJaSetup
3659 \ClassInfo\bxjs@clsname
3660 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3661 \fi\fi}

```


`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```
3662 \newcommand*{\jsSimpleJaSetup}{%
3663   \XeTeXlinebreaklocale "ja"\relax
3664   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3665   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

```
3666 \fi\fi\fi
```

`simplejasetup` オプションの処理。

```
3667 \ifx\bxjs@do@simplejasetup\@undefined\else
3668   \g@addto@macro\bxjs@begin@document@hook{%
3669     \ifbxjs@simplejasetup
3670       \bxjs@do@simplejasetup
3671     \fi}
3672 \fi

precisetext オプションの処理。
3673 \ifbxjs@precisetext
3674   \ifx\bxjs@do@precisetext\@undefined
3675     \ClassWarning\bxjs@clsname
3676       {The current engine does not support the\MessageBreak
3677        'precise-text' option\@gobble}
3678   \else
3679     \bxjs@do@precisetext
3680   \fi
3681 \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。

```
3682 \g@addto@macro\bxjs@begin@document@hook{%
3683   \ifx\jsInhibitGlueAtParTop\@empty
3684     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3685     \ifx\everyparhook\bxjs@tmpa
3686       \let\everyparhook\@empty
3687     \fi
3688   \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。

```
3689 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@modern

まず \everypar を“乗っ取る” 処理を行う。
3690 \let\bxjs@everypar\everypar
3691 \newtoks\everypar
3692 \everypar\bxjs@everypar
```

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

```

3693 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3694 \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。

```

3695 \ifbxjs@fancyhdr

```

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

```

3696 \@onlypreamble\bxjs@adjust@fancyhdr
3697 \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。

```

3698 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3699 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3700 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3701 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3702 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3703 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3704 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3705 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3706 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3707 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3708 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3709 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3710 \def\bxjs@tmpa{\rm\thepage\strut}%
3711 \def\bxjs@tmpb{\thepage\strut}%
3712 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3713 \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。

```

3714 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth
3715 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3716 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3717 }\bxjs@tmpa
3718 \fi\fi
3719 \PackageInfo\bxjs@clsname
3720 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3721 \def\bxjs@pagestyle@hook{%
3722 \@ifpackageloaded{fancyhdr}{%

```

```

3723 \bxjs@adjust@fancyhdr
3724 \global\let\bxjs@adjust@fancyhdr\relax
3725 }{}

```

\pagestyle にフックを入れ込む。

```

3726 \let\bxjs@org@pagestyle\pagestyle
3727 \def\pagestyle{%
3728 \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。

```

3729 \g@addto@macro\bxjs@begin@document@hook{%
3730 \bxjs@pagestyle@hook
3731 \global\let\bxjs@pagestyle@hook\relax}

```

■和文空白命令

```

3732 \ifbxjs@jaspace@cmd

```

\jaenspace 半角幅の水平空き。

```

3733 \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。

```

3734 \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。

```

3735 \def\_ {\zwspace}

```

\jaspace jlreq クラスと互換の命令。

```

3736 \DeclareRobustCommand*{\jaspace}[1]{%
3737 \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3738 \ClassError\bxjs@clsname
3739 {Unknown jaspac: #1}{\@eha}%
3740 \else
3741 \csname bxjs@jaspace@@#1\endcsname
3742 \fi}
3743 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3744 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3745 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}

```

```

3746 \fi

```

終わり。

```

3747 \fi

```

以上で終わり。

```

3748 %</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

C.1 準備

まず `minimal` ドライバを読み込む。

```
3749 %<*standard>
3750 %% このファイルは日本語文字を含みます
3751 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3752 \bxjs@simplejasetupfalse
```

C.2 和文ドライバパラメタ

`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` 2004JIS 字形を優先させるか。

```
3753 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3754 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3755 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3756 \define@key{bxjsStd}{jis2004}[true]{%
3757   \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。

```
3758 \newif\ifbxjs@jp@units

units オプションの処理。
```

```

3759 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3760 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3761 \define@key{bxjsStd}{units}[true]{%
3762   \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。

```

3763 \let\bxjs@jp@font\@empty

font オプションの処理。
3764 \define@key{bxjsStd}{font}{%
3765   \edef\bxjs@jp@font{#1}}

```

`\ifbxjs@jp@strong@cmd` `\strong` 命令を補填するか。

```

3766 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue

strong-cmd オプションの処理。
3767 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
3768 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
3769 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}

```

実際の `japaram` の値を適用する。

```

3770 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3771 \expandafter\bxjs@next\expandafter{\jsJaParam}

```

C.3 共通処理 (1)

■`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。

```

3772 \@onlypreamble\bxjs@apply@mmiv
3773 \def\bxjs@apply@mmiv{%
3774   \g@addto@macro\@classoptionslist{,jis2004}
3775 % \@ifpackagewith 判定への対策
3776   \PassOptionsToPackage{jis2004}{otf}
3777   \global\let\bxjs@apply@mmiv\relax}
3778 \ifbxjs@jp@jismmiv \bxjs@apply@mmiv \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。

```

3779 \if j\jsEngine
3780   \bxjs@jp@unitsfalse
3781 \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。

```

3782 \ifbxjs@jp@units
3783   \IfFileExists{bxcalc.sty}{%
3784     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3785     \ifx\usepTeXunits\@undefined

```

```

3786 \PackageWarningNoLine\bxjs@clsname
3787 {Cannot support pTeX units (zw etc.), since\MessageBreak
3788 the package 'bxcalc' is too old}%
3789 \bxjs@jp@unitsfalse
3790 \else \usepTeXunits
3791 \fi
3792 }{%else
3793 \PackageWarningNoLine\bxjs@clsname
3794 {Cannot support pTeX units (zw etc.), since\MessageBreak
3795 the package 'bxcalc' is unavailable}%
3796 \bxjs@jp@unitsfalse
3797 }
3798 \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。

```

3799 \ifbxjs@jp@units
3800 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3801 \let\bxjs@parse@qh@a\@undefined
3802 \let\bxjs@parse@qh@b\@undefined
3803 \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。

```

3804 \ifbxjs@jp@units
3805 \def\bxjs@let@lenexpr#1#2{%
3806 \edef#1{#2}%
3807 \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3808 \else
3809 \def\bxjs@let@lenexpr{\edef}
3810 \fi

```

■\strong 命令の補填

\strong fontspec で提供される \strong 命令と strongenv 環境を全てのエンジンで使えるよう strongenv にする。

※

```

3811 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3812 \ifx\strong\@undefined\ifx\strongenv\@undefined
3813 \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3814 \DeclareTextFontCommand{\strong}{\strongenv}%

```

fontspec と互換の \strongfontdeclare 命令も提供する。ただし、BX.JS での \strong は多段階には対応しないので、引数のコンマ区切りリストのうちの先頭の項目だけが意味をもつ。既定の設定は \bfseries (太字) である。

※\strongfontdeclare は試験的機能とする。

```

3815 \newcommand*{\strongfontdeclare}{\bxjs@strongfontdeclare}%
3816 \newcount\bxjs@strong@level

```

```

3817 \bxjs@protected\def\bxjs@strongfontdeclare#1{%
3818 \bxjs@set@array@from@cclist{bxjs@strong}{#1}%
3819 \bxjs@strong@level\z0}%
3820 \bxjs@strongfontdeclare{\bfseries}%
3821 \def\bxjs@strong@font{%
3822 \bxjs@csletcs{bxjs@tmpa}{bxjs@strong/\the\bxjs@strong@level}%
3823 \ifx\bxjs@tmpa\relax
3824 \advance\bxjs@strong@level\m@ne \bxjs@strong@font
3825 \else \advance\bxjs@strong@level\@ne \bxjs@tmpa
3826 \fi}%
3827 \fi\fi
3828 }\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。

```

3829 \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。

```

\jH3830 \@tempdima=0.25mm
3831 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3832 \let\jH\jQ

```

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

```

\trueH3833 \ifjsc@mag
3834 \@tempdimb=\jsBaseFontSize\relax
3835 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3836 \@tempdima=2.5mm
3837 \bxjs@invscale\@tempdima\bxjs@tmpa
3838 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3839 \@tempdima=10pt
3840 \bxjs@invscale\@tempdima\bxjs@tmpa
3841 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3842 \else \let\trueQ\jQ \let\bxjs@truept\p@
3843 \fi
3844 \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。

```

3845 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3846 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3847 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3848 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3849 \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。

```

3850 \def\jafontsize#1#2{%
3851   \begingroup
3852     \bxjs@jafontsize@a{#1}%
3853     \@tempdimb\jsInverseScale\@tempdima
3854     \bxjs@jafontsize@a{#2}%
3855     \xdef\bxjs@g@tmpa{%
3856       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3857   \endgroup\bxjs@g@tmpa}
3858 \def\bxjs@jafontsize@a#1{%
3859   \bxjs@parse@qh{#1}%
3860   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3861   \@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。

```

3862 \def\bxjs@kanjiskip{Opt}

```

`\setkanjiskip` 和文間空白の量を設定する。

```

3863 \newcommand*\setkanjiskip[1]{%
3864   \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3865   \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3866 \newcommand*\getkanjiskip{%
3867   \bxjs@kanjiskip}

```

`\ifbxjs@kanjiskip@enabled` 和文間空白の挿入が有効か。ただし pT_EX では自身の `\(no)autospacing` での制御を用いるのでこの変数は常に真とする。

```

3868 \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 3869 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3870   \bxjs@kanjiskip@enabledtrue
3871   \bxjs@reset@kanjiskip}
3872 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3873   \bxjs@kanjiskip@enabledfalse
3874   \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。

```

3875 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3876   \ifbxjs@kanjiskip@enabled
3877     \setlength{\@tempkipa}{\bxjs@kanjiskip}%
3878   \else \@tempkipa\z@
3879   \fi
3880   \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 3881 \def\bxjs@xkanjiskip{Opt}
\getxkanjiskip 3882 \newcommand*\setxkanjiskip[1]{%
3883   \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3884   \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3885 \newcommand*\getxkanjiskip{%
3886   \bxjs@xkanjiskip}
\bxjs@disable@xkanjiskip 3887 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3888 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3889   \bxjs@xkanjiskip@enabledtrue
3890   \bxjs@reset@xkanjiskip}
3891 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3892   \bxjs@xkanjiskip@enabledfalse
3893   \bxjs@reset@xkanjiskip}
3894 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3895   \ifbxjs@xkanjiskip@enabled
3896     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3897   \else \@tempskipa\z@
3898   \fi
3899   \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。

```
3900 \g@addto@macro\jsResetDimen{%
3901   \bxjs@reset@kanjiskip
3902   \bxjs@reset@xkanjiskip}
3903 \let\bxjs@apply@kanjiskip\relax
3904 \let\bxjs@apply@xkanjiskip\relax
```

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

`\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。

```
3905 \@onlypreamble\bxjs@adjust@jafont
3906 \def\bxjs@adjust@jafont#1{%
3907   \ifx\jsJaFont\bxjs@@auto
3908     \bxjs@get@kanjiEmbed
3909     \ifx\bxjs@jaEmbed\relax
3910       \let\bxjs@tmpa\@empty
3911     \else
3912       \let\bxjs@tmpa\bxjs@jaEmbed
3913       \ifx\bxjs@jaVariant\bxjs@@hziv
3914         \bxjs@apply@mmiv
```

```

3915     \fi
3916     \fi
3917   \else
3918     \let\bxjs@tmpa\jsJaFont
3919   \fi
3920   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3921     \ClassWarningNoLine\bxjs@clsname
3922       {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3923         not available on the current situation}%
3924     \let\bxjs@tmpa@empty
3925   \fi\fi
3926 }
3927 \def\bxjs@@auto{auto}
3928 \def\bxjs@@noEmbed{noEmbed}
3929 \def\bxjs@@hziv{-04}

```

\bxjs@jaEmbed 現在の updmap の jaEmbed・jaVariant パラメタの値。 \bxjs@get@kanjiEmbed により実
 \bxjs@jaVariant 際の設定値が取得されてここに設定される。

※古い版の updmap では kanjiEmbed・kanjiVariant であった。

```

3930 \let\bxjs@jaEmbed\relax
3931 \let\bxjs@jaVariant\relax

```

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

```

3932 \@onlypreamble\bxjs@get@kanjiEmbed
3933 \def\bxjs@get@kanjiEmbed{%
3934   \begingroup\setbox\z@=\hbox{%
3935     \global\let\bxjs@tmpdo@empty
3936     \def\bxjs@next##1##2##3{%
3937       \def##1####1##3 ####2\@nil####3\@nnil{%
3938         \ifx$####1$\gdef##2{####2}\fi}%
3939       \g@addto@macro\bxjs@tmpdo{%
3940         \expandafter##1\bxjs@tmpa\@nil##3 \@nil\@nnil}}%
3941       \bxjs@next\bxjs@tmpdo@a\bxjs@g@tmpa{kanjiEmbed}%
3942       \bxjs@next\bxjs@tmpdo@b\bxjs@g@tmpa{jaEmbed}%
3943       \bxjs@next\bxjs@tmpdo@c\bxjs@g@tmpb{kanjiVariant}%
3944       \bxjs@next\bxjs@tmpdo@d\bxjs@g@tmpb{jaVariant}%
3945   %
3946   \global\let\bxjs@g@tmpa\relax
3947   \global\let\bxjs@g@tmpb\relax
3948   \endlinechar\m@ne
3949   \let\do\@makeother\dospecials
3950   \catcode32=10 \catcode12=10 %form-feed
3951   \let\bxjs@tmpa@empty
3952   \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3953   \ifeof\@inputcheck\else
3954     \read\@inputcheck to\bxjs@tmpa
3955     \closein\@inputcheck
3956   \fi
3957   \ifx\bxjs@tmpa@empty\else

```

```

3958 \openin\@inputcheck="\bxjs@tmpa"\relax
3959 \@tempswatrue
3960 \loop\if@tempswa
3961 \read\@inputcheck to\bxjs@tmpa
3962 \bxjs@tmpdo
3963 \ifeof\@inputcheck \@tempswafalse \fi
3964 \repeat
3965 \fi
3966 }\endgroup
3967 \let\bxjs@jaEmbed\bxjs@g@tmpa
3968 \let\bxjs@jaVariant\bxjs@g@tmpb
3969 }

```

`\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。

```

3970 \@onlypreamble\bxjs@resolve@jafont@paren
3971 \def\bxjs@resolve@jafont@paren#1{%
3972 \def\bxjs@tmpb{\let#1}%
3973 \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\@nnil#1}
3974 \@onlypreamble\bxjs@resolve@jafont@paren@a
3975 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\@nnil#5{%
3976 \ifx\relax#4\relax \bxjs@tmpb#5%
3977 \else
3978 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3979 \bxjs@tmpb\bxjs@tmpa
3980 \fi}

```

■和文として出力 「欧文扱い」となっている文字を和文として出力するための機能。

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

```

3981 \newcommand*\jachar[1]{%
3982 \begingroup
3983 \jsLetHeadChar\bxjs@tmpa{#1}%
3984 \ifx\bxjs@tmpa\relax
3985 \ClassWarningNoLine\bxjs@clsname
3986 {Illegal argument given to \string\jachar}%
3987 \else
3988 \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3989 \fi
3990 \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```
3991 \let\jsJaChar\jachar
```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
3992 \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。

```
3993 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3994 \@onlypreamble\bxjs@fix@hyperref@unicode
3995 \def\bxjs@fix@hyperref@unicode#1{%
3996   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3997   \@namedef{KV@Hyp@bxjs/hook}##1{%
3998     \KV@Hyp@unicode{##1}%
3999     \def\KV@Hyp@unicode###1{%
4000       \expandafter\ifx\csname if##1\expandafter\endcsname
4001       \csname if###1\endcsname\else
4002       \ClassWarningNoLine{bxjs@clsname
4003       {Blcked hyperref option 'unicode=###1'}}%
4004       \fi
4005     }%
4006   }%
4007 }
```

\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。

```
4008 \@onlypreamble\jsCheckHyperrefUnicode
4009 \let\jsCheckHyperrefUnicode\@empty
4010 \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。

```
4011 \@onlypreamble\bxjs@check@hyperref@unicode
4012 \def\bxjs@check@hyperref@unicode#1{%
4013   \g@addto@macro\jsCheckHyperrefUnicode{%
4014     \@tempswafalse
4015     \begingroup
4016     \expandafter\ifx\csname ifHy@unicode\endcsname\relax
4017     \aftergroup\@tempwattrue \fi
4018     \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
4019     \csname if#1\endcsname
4020     \aftergroup\@tempwattrue \fi
4021   \endgroup
4022   \if@tempswa\else
4023     \ClassError{bxjs@clsname
4024     {The value of hyperref 'unicode' key is not suitable\MessageBreak
4025     for the present engine (must be #1)}}%
4026     {\@ehc}%
4027   \fi}}
```

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

```
4028 \@onlypreamble\bxjs@urgent@special
4029 \@onlypreamble\bxjs@urgent@special@a
```

L^AT_EX カーネルの新フック管理が導入済かを調べる。未導入の古い版である場合。

```
4030 \ifbxjs@old@hook@system
4031 \def\bxjs@urgent@special#1{%
4032   \AtBeginDvi{\special{#1}}%
4033   \g@addto@macro\bxjs@begin@document@hook{%
4034     \@ifpackageloaded{atbegshi}{%
4035       \begingroup
4036         \toks\z@{\special{#1}}%
4037         \toks\tw@{\expandafter\AtBegShi@HookFirst}%
4038         \xdef\AtBegShi@HookFirst{\the\toks@the\toks\tw@}%
4039       \endgroup
4040     }{}%
4041   }%
4042 }
```

導入済の場合。

※自分が先行する必要がある対象のパッケージを適宜追加する。

※pxjahyper パッケージの処理と合わせる。

```
4043 \else
4044   \def\bxjs@urgent@special#1{%
4045     \bxjs@urgent@special@a
4046     \AddToHook{shipout/firstpage}[pxjahyper/enc]{\special{#1}}
4047   \def\bxjs@urgent@special@a{%
4048     \DeclareHookRule{shipout/firstpage}{pxjahyper/enc}{<}{hyperref}%
4049     \global\let\bxjs@urgent@special@a\relax
4050 \fi
```

C.4 pT_EX 用設定

```
4051 \if j\jsEngine
```

■ 共通命令の実装

```
4052 \def\bxjs@apply@kanjiskip{%
4053   \kanjiskip\@tempskipa}
4054 \def\bxjs@apply@xkanjiskip{%
4055   \xkanjiskip\@tempskipa}
```

\jaJaChar のサブマクロ。

```
4056 \def\bxjs@jachar#1{%
4057   \bxjs@jachar@a#1...\@nil}
4058 \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。

```
4059   \ifx.#2#1%
```

引数が複数トークンの場合は、UTF-8 のバイト列であると見なし、そのスカラー値を \@tempcnta に代入する。

```
4060   \else\ifx.#3%
4061     \@tempcnta`#1 \multiply\@tempcnta64
```

```

4062 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
4063 \bxjs@jachar@b
4064 \else\ifx.#4%
4065 \@tempcnta`#1 \multiply\@tempcnta64
4066 \advance\@tempcnta`#2 \multiply\@tempcnta64
4067 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
4068 \bxjs@jachar@b
4069 \else
4070 \@tempcnta`#1 \multiply\@tempcnta64
4071 \advance\@tempcnta`#2 \multiply\@tempcnta64
4072 \advance\@tempcnta`#3 \multiply\@tempcnta64
4073 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
4074 \bxjs@jachar@b
4075 \fi\fi\fi}

```

符号値が `\@tempcnta` の和文文字を出力する処理。

```

4076 \ifjsWithupTeX
4077 \def\bxjs@jachar@b{\kchar\@tempcnta}
4078 \else
4079 \def\bxjs@jachar@b{%
4080 \ifx\bxUInt\@undefined\else
4081 \bxUInt{\@tempcnta}%
4082 \fi}
4083 \fi

```

和欧文間空白の命令 `\jathinspace` の実装。

```

4084 \ifbxjs@jaspace@cmd
4085 \def\jathinspace{\hskip\xkanjiskip}
4086 \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。

```

4087 \ifbxjs@jp@jismmiv
4088 \PassOptionsToPackage{prefer2004jis}{pxchfon}
4089 \PassOptionsToPackage{prefer2004jis}{pxbabel}
4090 \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` が指定されいる場合は、その値を `pxchfon` のオプションに追加する。

```

4091 \let\bxjs@jafont@paren\@firstofone
4092 \let\bxjs@tmpa\jsJaFont
4093 \ifx\bxjs@tmpa\bxjs@@auto
4094 \let\bxjs@tmpa\@empty
4095 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
4096 \def\bxjs@tmpa{noembed}
4097 \fi\fi

```

```

4098 \bxjs@resolve@jafont@paren\bxjs@tmpa
4099 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4100 \ifx\bxjs@tmpa\@empty\else
4101   \edef\bxjs@next{%
4102     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4103   }\bxjs@next
4104 \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……。)

```

4105 \begingroup
4106   \global\let\bxjs@g@tmpa\relax
4107   \catcode`\|=0 \catcode`\|=12
4108   |def|bxjs@tmpdo@a#1|@nil{%
4109     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4110   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4111     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4112   |catcode`\|=0 \catcode`\|=12
4113   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4114     \ifx$#2$\else
4115       \xdef\bxjs@g@tmpa{%
4116         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4117     \fi}
4118 \@firstofone{%
4119   \catcode10=12 \endlinechar\m@ne
4120   \let\do\@makeother \dospecials \catcode32=10
4121   \openin\@inputcheck=otf.sty\relax
4122   \@tempwattrue
4123   \loop\if@tempwa
4124     \ifeof\@inputcheck \@tempwafalse \fi
4125     \if@tempwa
4126       \read\@inputcheck to\bxjs@next
4127       \expandafter\bxjs@tmpdo\bxjs@next\@nil
4128     \fi
4129   \repeat
4130   \closein\@inputcheck
4131 \endgroup}
4132 \bxjs@g@tmpa

```

■hyperref 対策 unicode にしてはいけない。

```

4133 \ifbxjs@hyperref@enc
4134   \bxjs@check@hyperref@unicode{false}

```

暫定的なナニカ。

```

4135   \ifjsWithupTeX\ifbxjs@old@hook@system\else

```

```

4136 \IfFileExists{pxjahyper-uni.def}{%
4137 \AddToHook{bxjs@CGHN{package/hyperref/after}}{\input{pxjahyper-
uni.def}}
4138 }{}
4139 \fi\fi
4140 \fi

```

tounicode special 命令を出力する。

```

4141 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4142 \else\ifjsWithpTeXng T\else F\fi\fi T%
4143 \IfFileExists{pxjahyper-enc.sty}{\@tempswatrue}{\@tempswafalse}
4144 \if@tempswa
4145 \RequirePackage{pxjahyper-enc}[2020/10/05]%v0.6
4146 \ifbxjs@bigcode\else \suppressbigcode \fi
4147 \else
4148 \ifnum\jis"2121="A1A1 %euc
4149 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4150 \else\ifnum\jis"2121="8140 %sjis
4151 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4152 \else\ifnum\jis"2121="3000 %uptex
4153 \ifbxjs@bigcode
4154 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4155 \PassOptionsToPackage{bigcode}{pxjahyper}
4156 \else
4157 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4158 \PassOptionsToPackage{nobigcode}{pxjahyper}
4159 \fi
4160 \fi\fi\fi
4161 \let\bxToUnicodeSpecialDone=t
4162 \fi
4163 \fi

```

■和文数式ファミリー 和文数式ファミリー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。

```

4164 \ifx f\bxjs@enablejfam\else
4165 \@enablejfamtrue
4166 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

4167 \if@enablejfam
4168 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4169 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4170 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4171 \jfam\symmincho
4172 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4173 \g@addto@macro\bxjs@begin@document@hook{%
4174 \ifx\reDeclareMathAlphabet\undefined\else
4175 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4176 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%

```



```

4177      \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}{%
4178      \fi}
4179 \fi

```

C.5 pdfTeX 用設定 : CJK + bxcjkatype

```

4180 \else\if p\jsEngine

```

■bxcjkatype パッケージの読込 \jsJaFont が指定されている場合は、その値を bxcjkatype のオプション (プリセット指定) に渡す。(auto ならば \bxjs@get@kanjiEmbed を実行する。) スケール値 (\jsScale) の反映は bxcjkatype の側で行われる。

※ Pandoc モードでは autotilde を指定しない。

```

4181 \bxjs@adjust@jafont{f}
4182 \let\bxjs@jafont@paren\@firstofone
4183 \bxjs@resolve@jafont@paren\bxjs@tmpa
4184 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4185 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4186 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4187   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4188 \fi
4189 \edef\bxjs@next{%
4190   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4191 }\bxjs@next
4192 \bxjs@cjk@loaded

```

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

※取りあえず固定はしない。

```

4193 \ifbxjs@hyperref@enc
4194   \PassOptionsToPackage{unicode}{hyperref}
4195 \fi

```

\hypersetup 命令で (CJK* 環境に入れなくても) 日本語文字を含む文書情報を設定できるようにするための細工。

※ bxcjkatype を whole 付きで使っていることが前提。

※パッケージオプションでの指定に対応するのは、「アクティブな高位バイトトークンがその場で展開されてしまう」ため困難である。

```

4196 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
4197 \begingroup
4198   \CJK@input{UTF8.bdg}
4199 \endgroup
4200 \g@addto@macro\pdfstringdefPreHook{%
4201   \@nameuse{CJK@UTF8Binding}%
4202 }
4203 \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。

```

4204 \ifx\bxCJKattypeHyperrefPatchDone\@undefined
4205 \g@addto@macro\pdfstringdefPreHook{%
4206   \ifx~\bxjs@CJKTilde
4207     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4208     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4209     \let~\@empty
4210   \fi
4211 }
4212 \def\bxjs@CJKTilde{\CJKecglue\ignorespaces}
4213 \def\bxjs@tildecmd{~}
4214 \def\bxjs@LetUnexpandableSpace#1{%
4215   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@tildecmd\else
4216     \bxjs@org@LetUnexpandableSpace#1%
4217   \fi}
4218 \fi

```

■ 共通命令の実装

```

4219 \newskip\jsKanjiSkip
4220 \newskip\jsXKanjiSkip
4221 \ifx\CJKecglue\@undefined
4222   \def\CJKTilde{\CJKglobal\def~{\CJKecglue\ignorespaces}}
4223 \fi
4224 \let\autospacing\bxjs@enable@kanjiskip
4225 \let\noautospacing\bxjs@disable@kanjiskip
4226 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4227 \def\bxjs@apply@kanjiskip{%
4228   \jsKanjiSkip\@tempskipa
4229   \let\CJKglue\bxjs@CJKglue}
4230 \let\autoxspacing\bxjs@enable@xkanjiskip
4231 \let\noautoxspacing\bxjs@disable@xkanjiskip
4232 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4233 \def\bxjs@apply@xkanjiskip{%
4234   \jsXKanjiSkip\@tempskipa
4235   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4236 \def\bxjs@jachar#1{%
4237   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4238 \ifbxjs@jaspace@cmd
4239   \protected\def\jathinspace{\CJKecglue}
4240 \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。

```

4241 \ifx t\bxjs@enablejfam
4242   \ClassWarningNoLine\bxjs@clsname
4243   {You cannot use 'enablejfam=true', since the\MessageBreak

```

```

4244 CJK package does not support Japanese math}
4245 \fi

```

C.6 Xe_{La}TeX 用設定：xeCJK + zxjatype

```

4246 \else\if x\jsEngine

```

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

```

4247 \RequirePackage{zxjatype}
4248 \PassOptionsToPackage{no-math}{fontspec}%!
4249 \PassOptionsToPackage{xetex}{graphicx}%!
4250 \PassOptionsToPackage{xetex}{graphics}%!
4251 \ifx\zxJaFamilyName\undefined
4252   \ClassError{bxjs}{clsname}
4253   {xeCJK or zxjatype is too old}\@ehc
4254 \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。非指定の場合は原ノ味フォントを使用する。

※ 2.0 版より既定を IPAex から原ノ味に変更。

```

4255 \bxjs@adjust@jafont{f}
4256 \let\bxjs@jafont@paren\@gobble
4257 \bxjs@resolve@jafont@paren\bxjs@tmpa
4258 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4259 \ifx\bxjs@tmpa\@empty
4260   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4261   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4262 \else
4263   \edef\bxjs@next{%
4264     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4265   }\bxjs@next
4266 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

Xe_{La}TeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、Xe_{La}TeX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「Xe_{La}TeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```

4267 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne

```

```

4268 \ifbxjs@hyperref@enc
4269 \PassOptionsToPackage{unicode}{hyperref}
4270 \fi
4271 \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。

```

4272 \AtEndOfPackage{%
4273 \def\@gnewline #1{%
4274 \ifvmode \@nolnerr
4275 \else
4276 \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
4277 \nobreak \hskip-1sp\hskip1sp\relax
4278 \ignorespaces
4279 \fi}
4280 }

```

■共通命令の実装

```

4281 \newskip\jsKanjiSkip
4282 \newskip\jsXKanjiSkip
4283 \ifx\CJKecglue\@undefined
4284 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4285 \fi
4286 \let\autospacing\bxjs@enable@kanjiskip
4287 \let\noautospacing\bxjs@disable@kanjiskip
4288 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4289 \def\bxjs@apply@kanjiskip{%
4290 \jsKanjiSkip\@tempkipa
4291 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4292 \let\autoxspacing\bxjs@enable@xkanjiskip
4293 \let\noautoxspacing\bxjs@disable@xkanjiskip
4294 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4295 \def\bxjs@apply@xkanjiskip{%
4296 \jsXKanjiSkip\@tempkipa
4297 \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。

```

4298 \ifx\mcfamily\@undefined
4299 \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4300 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4301 \fi

```

`\jachar` のサブマクロの実装。

```

4302 \def\bxjs@jachar#1{%
4303   \xeCJKDeclareCharClass{CJK}{`#1}\relax
4304   #1}

\jathinspace の実装。
4305 \ifbxjs@jaspace@cmd
4306   \protected\def\jathinspace{\CJKecglue}
4307 \fi

```

■和文数式ファミリー 和文数式ファミリーは既定で無効とする。すなわち `enablejfam=true` の場合にのみ `@enablejfam` を真にする。

```

4308 \ifx t\bxjs@enablejfam
4309   \@enablejfamtrue
4310 \fi

```

実際に和文用の数式ファミリーの設定を行う。

※ FIXME: 要検討。

```

4311 \if@enablejfam
4312   \xeCJKsetup{CJKmath=true}
4313 \fi

```

C.7 Lua_TE_X 用設定 : Lua_TE_X-ja

```

4314 \else\if 1\jsEngine

```

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

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

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

```

4315 \let\zw\@undefined
4316 \RequirePackage{luatexja}
4317 \edef\bxjs@next{%
4318   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4319 }\bxjs@next

```

フォント代替の明示的定義。

```

4320 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4321 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4322 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4323 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4324 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4325 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4326 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4327 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4328 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }

```

```

4329 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4330 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4331 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4332 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4333 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4334 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4335 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4336 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4337 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4338 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4339 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4340 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4341 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4342 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4343 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4344 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4345 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4346 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4347 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4348 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4349 \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。非指定の場合は原ノ味フォントを指定する (luatexja-preset は読み込まない)。

※ 2.0 版より既定を IPAex から原ノ味に変更。

```

4350 \bxjs@adjust@jafont{t}
4351 \ifx\bxjs@tmpa\bxjs@noEmbed
4352   \def\bxjs@tmpa{noembed}
4353 \fi
4354 \let\bxjs@jafont@paren\gobble
4355 \bxjs@resolve@jafont@paren\bxjs@tmpa
4356 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4357 \ifx\bxjs@tmpa\empty
4358   \defaultjfontfeatures{ Kerning=Off }
4359   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4360   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4361 \else
4362   \edef\bxjs@next{%
4363     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4364   }\bxjs@next
4365 \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。

```

4366 \ifpackageafter{luatexja}{2016/03/31}{ }{\%else

```

```

4367 \DeclareRobustCommand\rmfamily
4368   {\not@math@alphabet\rmfamily\mathrm
4369     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4370 \DeclareRobustCommand\sffamily
4371   {\not@math@alphabet\sffamily\mathsf
4372     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4373 \DeclareRobustCommand\ttfamily
4374   {\not@math@alphabet\ttfamily\mathtt
4375     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4376 }
4377 \long\def\jttdefault{\gtdefault}
4378 \unless\ifx\@ltj@match@familytrue\@undefined
4379   \@ltj@match@familytrue
4380 \fi
4381 \g@addto@macro\bxjs@begin@document@hook{%
4382   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4383   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4384   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4385 \bxjs@if@sf@default{%
4386   \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

4387 % 次の3つは既定値の通り
4388 %\ltjsetparameter{prebreakpenalty={`',10000}}
4389 %\ltjsetparameter{postbreakpenalty={`",10000}}
4390 %\ltjsetparameter{prebreakpenalty={`",10000}}
4391 \ltjsetparameter{jaxspmode={`!',1}}
4392 \ltjsetparameter{jaxspmode={`〒,2}}
4393 \ltjsetparameter{alxspmode={`+,3}}
4394 \ltjsetparameter{alxspmode={`%,3}}

```

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

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の `LuaTeX-j` 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。

```

4395 \ifx\ltjfakeparbegin\@undefined
4396   \protected\def\ltjfakeparbegin{%
4397     \ifhmode
4398       \relax\directlua{%
4399         luatexja.jfmglue.create_beginpar_node()}}
4400   \fi}
4401 \fi

```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```

4402 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none
4403 \begingroup
4404   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}

```

```

4405 \gobble\if\def\bxjs@tmpa{\@{\everypar{}\fi}}
4406 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
4407 \gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}\fi}}\fi
4408 \directlua{
4409   local function patchcmd(cs, code, from, to)
4410     tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4411       :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1)..")")
4412   end
4413   patchcmd(\@xsect, [[\meaning\xsect]],
4414     \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4415   patchcmd(\@item, [[\meaning@item]],
4416     \bxjs@tmpa, \@ltjfakeparbegin)}
4417 \endgroup
4418 \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。

```

4419 \ifbxjs@hyperref@enc
4420 \PassOptionsToPackage{unicode}{hyperref}
4421 \bxjs@check@hyperref@unicode{true}
4422 \fi

```

■共通命令の実装

```

4423 \protected\def\autospacing{%
4424   \ltjsetparameter{autospacing=true}}
4425 \protected\def\noautospacing{%
4426   \ltjsetparameter{autospacing=false}}
4427 \protected\def\autoxspacing{%
4428   \ltjsetparameter{autoxspacing=true}}
4429 \protected\def\noautoxspacing{%
4430   \ltjsetparameter{autoxspacing=false}}
4431 \def\bxjs@apply@kanjiskip{%
4432   \ltjsetparameter{kanjiskip={\@tempskipa}}}
4433 \def\bxjs@apply@xkanjiskip{%
4434   \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4435 \def\bxjs@jachar#1{%
4436   \ltjjachar`#1\relax}

```

\jathinspace の実装。

```

4437 \ifbxjs@jaspace@cmd
4438 \protected\def\jathinspace{%
4439   \hskip\ltjgetparameter{xkanjiskip}\relax}
4440 \fi

```

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


```

4441 \ifx f\bxjs@enablejfam
4442   \ClassWarningNoLine\bxjs@clsname
4443   {You cannot use 'enablejfam=false', since the\MessageBreak
4444     LuaTeX-ja always provides Japanese math families}
4445 \fi

```

C.8 共通処理 (2)

```

4446 \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 4447 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4448   \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4449   \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4450 \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。

```

4451 \ifx\mathmc\@undefined
4452   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4453   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4454 \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)。

```

4455 \ifbxjs@jaspace@cmd
4456   \bxjs@protected\def\bxjs@choice@jathinspace{%
4457     \relax\ifmmode \mskip\medmuskip
4458     \else \jathinspace\ignorespaces
4459     \fi}
4460   \jsAtEndOfClass{%
4461     \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4462     \else \def\>{\protect\bxjs@choice@jathinspace}%
4463     \fi}
4464 \fi

```

■和文・和欧文間空白の初期値

```

4465 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4466 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4467 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4468 \fi

```

以上で終わり。

```

4469 %</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。

```
4470 %<*modern>
4471 \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。

```
4472 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4473 \def\encodingdefault{T1}%
4474 \input{t1enc.def}%
4475 \fontencoding\encodingdefault\selectfont
4476 \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。

```
4477 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4478 \renewcommand{\rmdefault}{lmr}
4479 \renewcommand{\sfdefault}{lms}
4480 \renewcommand{\ttdefault}{lmtt}
4481 \fi
```

大型演算子用の数式フォントの設定。

※`amsmaths` パッケージと同等にする。

```
4482 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4483   <-7.5>cmex7<7.5-8.5>cmex8%
4484   <8.5-9.5>cmex9<9.5->cmex10}{}%
4485 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
```

`amsmath` 読み込時に上書きされるのを防ぐ。

```
4486 \def\cmex@opt{10}
```

D.2 fixltx2e 読み込

※`fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```
4487 \ifx\@IncludeInRelease\@undefined
4488 \RequirePackage{fixltx2e}
4489 \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。

```
4490 \RequirePackage{bxjscjkat}
```

D.4 完了

おしまい。

```
4491 %</modern>
```

付録 E 和文ドライバ : pandoc

「Pandoc モード」で使用される和文ドライバ。standard ドライバの機能を継承するが、「Pandoc の既定の latex テンプレート」が使われることを前提として、それと BXJS の設定を整合させるための措置を加えている。

E.1 準備

standard ドライバの設定を引き継ぐ。

```
4492 %<*pandoc>
```

```
4493 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4494 \RequirePackage{bxjspandoc}
```

ϵ -TeX ではない場合に警告を出す。

※近い将来に ϵ -TeX 拡張を必須にする予定。

```
4495 \ifjsWitheTeX\else
```

```
4496   \ClassWarningNoLine{bxjs}{clsname}
```

```
4497   {!!!!!!! WARNING !!!!!!!\MessageBreak
```

```
4498   This engine does not support e-TeX extension!\MessageBreak
```

```
4499   Some feature might not work properly}
```

```
4500 \fi
```

`\ifbxjs@bxghost@available` [スイッチ] bxghost パッケージが利用できるか。

```
4501 \newif\ifbxjs@bxghost@available
```

```
4502 \ifjsWitheTeX
```

```
4503   \RequirePackage{pdftexcmds}[2009/09/22]% v0.5
```

```
4504   \IfFileExists{bxghost.sty}{%
```

```
4505     \bxjs@bxghost@availabletrue
```

```
4506     \@namedef{bxjs@bgbv/79E70A0991967E27981832C84DB5DF99}{1}%v0.2.0
```

```
4507     \ifx\pdf@filemdfivesum\@undefined\else
```

```
4508       \expandafter\ifx\csname bxjs@bgbv/\pdf@filemdfivesum{bxghost.sty}%
```

```
4509         \endcsname\relax\else \bxjs@bxghost@availablefalse \fi
```

```
4510   \fi
```

```

4511 }{}
4512 \fi

```

`\bxjs@endpreamble@hook` etoolbox の `\AtEndPreamble` で実行される BXJS クラス用のフック。

※ ε -TeX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```

4513 \@onlypreamble\bxjs@endpreamble@hook
4514 \let\bxjs@endpreamble@hook\@empty

```

パッケージ読込。

```

4515 \RequirePackage{iftex}[2013/04/04]% v0.2
4516 \ifjsWithTeX
4517 \RequirePackage{etoolbox}[2010/08/21]% v2.0
4518 \AtEndPreamble{\bxjs@endpreamble@hook}
4519 \RequirePackage{filehook}[2011/10/12]% v0.5d
4520 \fi

```

E.2 和文ドライバパラメタ

`keyval` のファミリーは `bxjsPan` とする。

`\ifbxjs@jp@fix@strong` 重要要素を補正するか。

```

4521 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue

```

`fix-strong` オプションの処理。

```

4522 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
4523 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
4524 \define@key{bxjsPan}{fix-strong}[true]{%
4525 \bxjs@set@keyval{fixstrong}{#1}{}}

```

`\ifbxjs@jp@fix@code` インラインコード要素を補正するか。

```

4526 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue

```

`fix-code` オプションの処理。

```

4527 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
4528 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
4529 \define@key{bxjsPan}{fix-code}[true]{%
4530 \bxjs@set@keyval{fixcode}{#1}{}}

```

`\bxjs@jp@strong` 重要要素に適用される書体変更の種類。

```

4531 \chardef\bxjs@jp@strong=0

```

`strong` オプションの処理。

```

4532 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
4533 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
4534 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }
4535 \define@key{bxjsPan}{strong}{%
4536 \bxjs@set@keyval{strong}{#1}{}}

```

`\ifbxjs@jp@or@indent` プレアンブルでのレイアウト上書きを許可するか。既定値は真。

`\ifbxjs@jp@or@secnumdepth` 4537 `\newif\ifbxjs@jp@or@indent \bxjs@jp@or@indenttrue`

`\ifbxjs@jp@or@block@heading` 4538 `\newif\ifbxjs@jp@or@secnumdepth \bxjs@jp@or@secnumdepthtrue`

4539 `\newif\ifbxjs@jp@or@block@heading \bxjs@jp@or@block@headingtrue`

クラスで `pandoc+` が指定された場合、内部和文パラメタ `_plus` が和文ドライバに渡される。この場合、レイアウト上書きを禁止する。

※`_plus` は必ずパラメタ列の先頭にあるので、個別のパラメタ設定の方が常に優先される。

4540 `\define@key{bxjsPan}{_plus}[]{%`

4541 `\bxjs@jp@or@indentfalse`

4542 `\bxjs@jp@or@secnumdepthfalse`

4543 `\bxjs@jp@or@block@headingfalse}`

レイアウト上書き許可オプション (`or-indent`・`or-secnumdepth`・`or-block-heading`) の処理。

4544 `\let\bxjs@kv@orindent@true\bxjs@jp@or@indenttrue`

4545 `\let\bxjs@kv@orindent@false\bxjs@jp@or@indentfalse`

4546 `\define@key{bxjsPan}{or-indent}[true]{%`

4547 `\bxjs@set@keyval{orindent}{#1}{}}`

4548 `\let\bxjs@kv@orsecnumdepth@true\bxjs@jp@or@secnumdepthtrue`

4549 `\let\bxjs@kv@orsecnumdepth@false\bxjs@jp@or@secnumdepthfalse`

4550 `\define@key{bxjsPan}{or-secnumdepth}[true]{%`

4551 `\bxjs@set@keyval{orsecnumdepth}{#1}{}}`

4552 `\let\bxjs@kv@orblockheading@true\bxjs@jp@or@block@headingtrue`

4553 `\let\bxjs@kv@orblockheading@false\bxjs@jp@or@block@headingfalse`

4554 `\define@key{bxjsPan}{or-block-heading}[true]{%`

4555 `\bxjs@set@keyval{blockheading}{#1}{}}`

実際の `japaram` の値を適用する。

4556 `\def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}`

4557 `\expandafter\bxjs@next\expandafter{\jsJaParam}`

E.3 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〉}` : 指定の名前の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。

4558 `\@onlypreamble\bxjs@set@dupload@proc`

4559 `\def\bxjs@set@dupload@proc#1{%`

4560 `\expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}`

4561 `\@onlypreamble\bxjs@set@dupload@proc@a`

4562 `\def\bxjs@set@dupload@proc@a#1{%`

4563 `\@onlypreamble#1\def#1##1}`

```

4564 \def\bxjs@unset@dupload@proc#1{%
4565   \bxjs@cslet{bxjs@dlp/#1}\@undefined}

```

\@if@ptions \@if@ptions の再定義。

```

4566 \@onlypreamble\bxjs@org@if@ptions
4567 \let\bxjs@org@if@ptions\@if@ptions
4568 \@onlypreamble\bxjs@org@reset@ptions
4569 \let\bxjs@org@reset@ptions\relax
4570 \def\@if@ptions#1#2#3{%
4571   \let\bxjs@next\@secondoftwo
4572   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4573   \ifx\bxjs@tmpa\bxjs@tmpb
4574     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4575       \let\bxjs@next\@firstoftwo \fi
4576   \fi
4577   \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@ptions{#1}{#2}{#3}}
4578 \g@addto@macro\bxjs@begin@document@hook{%
4579   \let\@if@ptions\bxjs@org@if@ptions}
4580 \@onlypreamble\bxjs@do@dupload@proc
4581 \def\bxjs@do@dupload@proc#1#2#3{%
4582   \ifx\bxjs@org@reset@ptions\relax
4583     \let\bxjs@org@reset@ptions\@reset@ptions
4584   \fi
4585   \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4586   \def\@reset@ptions{%
4587     \let\@reset@ptions\bxjs@org@reset@ptions
4588     \@reset@ptions
4589     \bxjs@next{#3}}%
4590   \@firstoftwo}

```

E.4 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。

※ Pandoc 2.12 版で lang=ja 指定に対応し、正しく L^AT_EX 側の言語名 `japanese` に変換されるようになった。しかし、日本語指定の場合は相変わらず調整処理が必要である。

\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。

```

4591 \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。

```

4592 \let\bxjs@babel@options\relax

```

■ Polyglossia について つまり X_ƎL^AT_EX および Lua_TE_X (古い Pandoc で) の場合。

※この場合 etoolbox が使用可能になっている。

```

4593 \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。

```
4594 \pandocSkipLoadPackage{polyglossia}
4595 \bxjs@set@dupload@proc{polyglossia.sty}{%
4596   \bxjs@unset@dupload@proc{polyglossia.sty}%
4597   \ClassWarning\bxjs@clsname
4598   {Package polyglossia is requested}%
4599   \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。

```
4600 \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。言語名が japanese だった場合も同様。

```
4601   \ifbool{test{\ifblank{##2}}or test{\ifstrequal{##2}{japanese}}}{%
4602     \ClassWarning\bxjs@clsname
4603     {Main language is 'japanese', thus fallback\MessageBreak
4604     definitions will be employed}%
4605     \bxjs@pandoc@polyglossia@ja
```

それ以外は、改めて polyglossia を読み込んで、本来の処理を実行する。

```
4606   }{%else
4607     \ClassWarning\bxjs@clsname
4608     {Main language is '##2',\MessageBreak
4609     thus polyglossia will be loaded}%
4610     \csundef{ver@polyglossia.sty}%
4611     \edef\bxjs@next{%
4612       \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4613     }\bxjs@next
4614     \setmainlanguage[##1]{##2}%
4615   }}
```

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。

```
4616 \g@addto@macro\bxjs@endpreamble@hook{%
4617   \ifx\bxjs@polyglossia@options\relax
4618     \csundef{ver@polyglossia.sty}%
4619   \fi}
```

\bxjs@pandoc@polyglossia@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。

※現在では Polyglossia の日本語用の定義ファイル (gloss-japanese.ldf) が存在するので、本来なら普通に処理できるはずであるが、現状の定義ファイルはアレなので回避したい。

```
4620 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4621 \def\bxjs@pandoc@polyglossia@ja{%
4622   \renewcommand*\setmainlanguage[2][]{%
```

```

4623 \newcommand*\setotherlanguage[2] [] {%
4624   \ifblank{##2}{-}{%else
4625     \cslet{##2}\@empty \cslet{end##2}\@empty
4626     \cslet{text##2}\@firstofone}}%
4627 \newcommand*\setotherlanguages[2] [] {%
4628   \@for\bxjs@tmpa:={##2}\do{%
4629     \setotherlangauge{\bxjs@tmpa}}}%

```

Polyglossia の読込済マークは外れるようにしておく。

```

4630 \let\bxjs@polyglossia@options\relax}%
4631 \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。

```

4632 \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。

```

4633 \pandocSkipLoadPackage{babel}
4634 \bxjs@set@dupload@proc{babel.sty}{%
4635   \bxjs@unset@dupload@proc{babel.sty}%
4636   \ClassWarning\bxjs@clsname
4637   {Package babel is requested}%

```

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。

```

4638 \@tempwafalse \let\bxjs@babel@options\@empty
4639 \def\bxjs@tmpb{main=}%
4640 \def\bxjs@next{main=japanese}%
4641 \@for\bxjs@tmpa:=#1\do{%
4642   \ifx\bxjs@tmpa\bxjs@tmpb \@tempwattrue
4643   \else\ifx\bxjs@tmpa\bxjs@next \@tempwattrue
4644   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4645   \fi\fi}%
4646 \if@tempwa
4647   \ClassWarning\bxjs@clsname
4648   {Main language is 'japanese', thus fallback\MessageBreak
4649     definitions will be employed}%
4650   \bxjs@pandoc@babel@ja

```

ない場合は、本来の babel の処理を実行する。

```

4651 \else
4652   \ClassWarning\bxjs@clsname
4653   {Main language is not 'japanese',\MessageBreak
4654     thus babel will be loaded}%
4655   \bxjs@cslet{ver@babel.sty}\@undefined
4656   \RequirePackage[#1]{babel}[]%
4657 \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。

```
4658 \g@addto@macro\bxjs@endpreamble@hook{%
4659   \ifx\bxjs@babel@options\relax
4660     \bxjs@cslet{ver@babel.sty}\@undefined
4661   \fi}
```

`\bxjs@pandoc@babel@ja` Pandoc 側で `lang=ja` が指定されていた場合の処理。

```
4662 \@onlypreamble\bxjs@pandoc@babel@ja
4663 \def\bxjs@pandoc@babel@ja{%
4664   \bxjs@cslet{ver@babel.sty}\@undefined
4665   \edef\bxjs@next{%
4666     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4667   }\bxjs@next
4668   \if j\jsEngine
4669     \RequirePackage[main=japanese]{pxbabel}[]%
4670   \else
4671     \RequirePackage{bxorigcapt}[]%
4672   \fi}
```

`lang` 対策はこれで終わり。

```
4673 \fi
```

E.5 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。

```
4674 \bxjs@set@dupload@proc{geometry.sty}{%
4675   \setpagelayout*{#1}}
```

E.6 CJKmainfont 変数

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

```
4676 \if l\jsEngine
4677   \pandocSkipLoadPackage{xeCJK}
4678   \providecommand*\setCJKmainfont{\setmainfont}
4679 \fi
```

E.7 Option clash 対策

`xeCJK` パッケージについて。

※`xeCJK` はクラス内で既に読み込まれているので、`space` は（意図通りに）無効になる。

※ v2.8～v2.9.2 の間。

```
4680 \if x\jsEngine
4681   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
```

```

4682     ,space}
4683 \fi

```

E.8 レイアウト上書き禁止

レイアウト上書き禁止の実装は `etoolbox` の機能を使う。

```

4684 \ifjsWithTeX
4685 \@onlypreamble\bxjs@info@or@ban
4686 \def\bxjs@info@or@ban#1{%
4687   \PackageInfo\bxjs@clsname
4688   {Freeze layout on '#1',\MessageBreak detected}}

```

■`indent` について `indent` 変数を指定しない場合に「段落表現形式をインデント方式に変更する」動作を抑止する。

```

4689 \unless\ifbxjs@jp@or@indent
4690   \bxjs@info@or@ban{indent}

```

`parskip` がある場合はそれを読み込もうとするため、`parskip` の読込をブロックする。

```

4691   \IfFileExists{parskip.sty}{%
4692     \pandocSkipLoadPackage{parskip}%

```

`parskip` がない場合はパラメタを変更しようとするため、該当のパラメタを復帰させる。

```

4693   }{%else
4694     \eappto\bxjs@endpreamble@hook{%
4695       \parindent=\the\parindent\relax
4696       \parskip=\the\parskip\relax}}
4697 \fi

```

■`secnumdepth` について `secnumdepth` の値を決めるのは `numbersections` 変数 (`-N/--number-sections` オプションに連動する) や `secnumdepth` 変数であるが、何れにしても `secnumdepth` の値は書き換えられる。そのため、`secnumdepth` を復帰させる。

```

4698 \ifbxjs@jp@or@secnumdepth\else
4699   \bxjs@info@or@ban{secnumdepth}
4700   \eappto\bxjs@endpreamble@hook{%
4701     \c@secnumdepth=\the\c@secnumdepth\relax}
4702 \fi

```

■`block-heading` について `\paragraph`、`\subparagraph` を別行見出しに変える処理を抑止する。

※ 2.7.1 版以前では別行見出し変更が既定で有効であった。

```

4703 \ifbxjs@jp@or@block@heading\else
4704   \let\bxjs@frozen@paragraph\paragraph
4705   \let\bxjs@frozen@subparagraph\subparagraph
4706   \bxjs@info@or@ban{block-heading}
4707   \appto\bxjs@endpreamble@hook{%
4708     \let\oldparagraph\undefined
4709     \let\paragraph\bxjs@frozen@paragraph

```

```

4710 \let\subparagraph\bxjs@frozen@subparagraph}
4711 \fi

    以上。
4712 \fi

```

E.9 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。

```

4713 \g@addto@macro\bxjs@begin@document@hook{%
4714   \@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。

```

4715   \ifx\oldparagraph\@undefined\else
4716     \@tempwattrue
4717   \fi

```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```

4718   \ifnum\c@secnumdepth>3
4719     \@tempwattrue
4720   \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。

```

4721   \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4722     \let\jsParagraphMark\@empty
4723   \fi\fi}

```

E.10 全角空白文字

L^AT_EX でない入力では、全角空きを入れるために全角空白文字（U+3000）が使われる可能性があるので、全角空白文字を和文文字でなく空きとして扱うようにしておく。

※ (u)pL^AT_EX では対応できないので対象外。

`\pandocZWSpace` 全角空白文字の入力で実行されるコード。

```

4724 \def\pandocZWSpace{\zwspace}

```

全角空白文字の入力で `\pandocZWSpace` が実行されるようにする。

```

4725 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
4726 \catcode"3000=\active
4727 \begingroup \catcode`\!=7
4728 \protected\gdef!!!!3000{\pandocZWSpace}
4729 \endgroup
4730 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4731 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4732 \bxjs@protected\def\bxjs@zsp@char{\pandocZWSpace}
4733 \fi\fi

```

E.11 hyperref 対策

hyperref の unicode オプションの固定を行う。

```

4734 \if j\jsEngine
4735 \bxjs@fix@hyperref@unicode{false}
4736 \else
4737 \bxjs@fix@hyperref@unicode{true}
4738 \fi

```

E.12 Pandoc 要素に対する和文用の補正

■重要要素 重要 (Strong) 要素に対する L^AT_EX 出力は `\textbf` となるが、代わりに `\strong` を使いたいため、`\textbf` を書き換えてしまう (うわあ)。

```

4739 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
4740 \let\orgtextbf\textbf
4741 \DeclareRobustCommand\pandocTextbf[1]{%
4742 \begingroup
4743 \let\textbf\orgtextbf
4744 \strong{#1}%
4745 \endgroup}%
4746 \g@addto@macro\bxjs@begin@document@hook{%
4747 \let\textbf\pandocTextbf}
4748 \fi\fi

```

`\strong` の書体を設定する。

```

4749 \jsAtEndOfClass{%
4750 \ifx\strongfontdeclare\@undefined\else
4751 \ifcase\bxjs@jp@strong
4752 \or \strongfontdeclare{\sffamily}%
4753 \or \strongfontdeclare{\sffamily\bfseries}%
4754 \fi
4755 \fi}

```

■インラインコード要素 インラインコード (Code) 要素に対する L^AT_EX 出力は `\texttt` となる。`\texttt` の両端に欧文ゴーストが入るようにする。さらに `\verb` の外側にも欧文ゴーストが入るようにする。

```

4756 \ifbxjs@jp@fix@code

```

bxghost パッケージが利用できる場合はその機能を利用する。使えない場合は自前実装を用いる。

```

4757 \ifbxjs@bxghost@available
4758   \RequirePackage[verb]{bxghost}[2020/01/31]% v0.3.0
4759   \let\bxjs@eghostguarded\eghostguarded
4760 \else
4761 \chardef\bxjs@eghost@c=23
4762 \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4763 \else\ifx l\jsEngine \ltjsetparameter{alxspmode={\bxjs@eghost@c,3}}
4764 \else\ifx x\jsEngine %no-op
4765 \else \let\bxjs@eghost@c\@undefined
4766 \fi\fi\fi
4767 \ifx\bxjs@eghost@c\@undefined\else
4768   \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4769   \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4770   \def\bxjs@eghostguarded#1{%
4771     \bxjs@pan@eghost\null#1\null\bxjs@pan@eghost}
4772 \fi
4773 \fi
4774 \ifx\bxjs@eghostguarded\@undefined\else
4775   \let\orgtexttt\texttt
4776   \DeclareRobustCommand\pandocTexttt[1]{%
4777     \ifmmode \nfss@text{\ttfamily #1}%
4778     \else
4779       \ifvmode \leavevmode \fi
4780       \bxjs@eghostguarded{\begingroup\ttfamily#1\endgroup}%
4781     \fi}
4782   \g@addto@macro\bxjs@begin@document@hook{%
4783     \let\texttt\pandocTexttt}

```

bxghost を使わない場合の \verb の処理。

※bxghost の実装を参考にした。

```

4784 \ifbxjs@bxghost@available\else
4785   \expandafter\def\expandafter\verb\expandafter{%
4786     \expandafter\bxjs@pan@eghost\verb}
4787   \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4788   \fi
4789 \fi
4790 \fi

```

E.13 ifPDFTeX スイッチ

Pandoc モードでは Pandoc の既定テンプレートを（無理やり）(u)pTeX に対応させることを目的にしている。

旧版のテンプレートではifxetex とifluatex パッケージを読み込んだ上で「XeTeX でも LuaTeX でもないものは pdfTeX」という前提の動作をしていた。よって、(u)pTeX に対応させる際には「pdfTeX 用の処理が実行される」ことを前提にすればよかった。

ところが、Pandoc の 2.12 版では `iftex` パッケージが導入されて「`pdfTeX` の判定を直接 `\ifPDFTeX` で行う」ように改修された。このため、`(u)pTeX` での実行でどのコードが実行されるかを予測することが困難になってしまった。

これに対処するため、「文書ファイルのプレアンブル実行中に限って `\ifPDFTeX` が（実際とは異なり）真になるようにする」という細工を施すことで、従来通り「`pdfTeX` 用の処理が実行される」前提が維持されるようにする。

```
4791 \if j\jsEngine
```

`\bxjs@check@frontier` `\bxjs@check@frontier\CS` は現在のパッケージ読込ネストレベルが丁度 1 であるときのみ `\CS` を実行する。

```
4792 \def\bxjs@check@frontier{%
4793   \expandafter\bxjs@check@frontier@a\@currnamestack\noindent...\@nil}
4794 \def\bxjs@check@frontier@a#1#2#3#4#5\@nil#6{%
4795   \ifx\noindent#4#6\fi}
```

`\bxjs@unforge@ifPDFTeX` `\ifPDFTeX` を偽（正しい値）にする。

```
4796 \@onlypreamble\bxjs@unforge@ifPDFTeX
4797 \def\bxjs@unforge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iffalse}}
```

`\bxjs@forge@ifPDFTeX` `\ifPDFTeX` を真（偽装した値）にする。

```
4798 \@onlypreamble\bxjs@forge@ifPDFTeX
4799 \def\bxjs@forge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iftrue}}
```

`\bxjs@unload@forge@ifPDFTeX` `\ifPDFTeX` に対する細工を無効化する。

```
4800 \def\bxjs@unload@forge@ifPDFTeX{%
4801   \bxjs@unforge@ifPDFTeX
4802   \global\let\bxjs@check@frontier\@gobble}
```

プレアンブル開始時は `\ifPDFTeX` は真で、終了時に偽装を無効化する。filehook のフックで「パッケージ読込中は偽装を解除する」ことを実現している。

```
4803 \jsAtEndOfClass{\bxjs@forge@ifPDFTeX}
4804 \ifjsWithTeX
4805   \AtBeginOfEveryFile{\bxjs@check@frontier\bxjs@unforge@ifPDFTeX}%
4806   \AtEndOfEveryFile{\bxjs@check@frontier\bxjs@forge@ifPDFTeX}%
4807   \g@addto@macro\bxjs@endpreamble@hook{\bxjs@unload@forge@ifPDFTeX}
4808 \else
4809   \g@addto@macro\bxjs@begin@document@hook{\bxjs@unload@forge@ifPDFTeX}
4810 \fi
4811 \fi
```

E.14 完了

おしまい。

```
4812 %</pandoc>
```

和文ドライバ実装はここまで。

```
4813 %</drv>
```

付録 F 補助パッケージ一覧

BXJS クラスの機能を実現するために用意されたものだが、他のクラスの文書で読み込んで利用することもできる。

- bxjscompat : ムニャムニャムニャ。
- bxjscjkat : modern ドライバ用の和文カテゴリを適用する。
- bxjspandoc : Pandoc 用のナニカ。

```
4814 %<*anc>
```

付録 G 補助パッケージ : bxjscompat

ムニャムニャムニャ……。

G.1 準備

```
4815 %<*compat>
```

```
4816 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
4817 \let\bxac@engine=n
```

```
4818 \def\bxac@do#1#2{%
```

```
4819   \edef\bxac@tmpa{\string#1}%
```

```
4820   \edef\bxac@tmpb{\meaning#1}%
```

```
4821   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
4822 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
4823 \bxac@do\luatexversion{\let\bxac@engine=l}
```

`\bxac@delayed@if@bxjs` もし BXJS クラスの読込中でこのパッケージが読み込まれているならば、BXJS のクラスの終わりまで実行を遅延する。

```
4824 \ifx\jsAtEndOfClass\undefined
```

```
4825   \let\bxac@delayed@if@bxjs\@firstofone
```

```
4826 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
4827 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4828 \newif\ifbxac@in@old@behavior
```

```
4829 \let\ImposeOldLuaTeXBehavior\relax
```

```
4830 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 X_YTEX 部分

```
4831 \ifx x\bxac@engine
```

X_YTEX 文字クラスのムニャムニャ。

```

4832 \@onlypreamble\bxac@adjust@charclass
4833 \bxac@delayed@if@bxjs{%
4834   \@ifpackageloaded{xeCJK}{-}{%else
4835     \ifx\XeTeXalloc@intercharclass\undefined\else
4836       \ifnum\XeTeXalloc@intercharclass=\z@
4837         \PackageInfo\bxac@pkgname
4838         {Setting up interchar class for CJK...\@gobble}%
4839         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4840           \XeTeXalloc@intercharclass=3
4841         }{%else
4842           \PackageWarning\bxac@pkgname
4843           {Cannot find file 'load-unicode-xetex-classes.tex'%
4844             \@gobble}%
4845         }%
4846       \fi\fi
4847       \ifnum\XeTeXcharclass"3041=\z@
4848         \PackageInfo\bxac@pkgname
4849         {Adjusting interchar class for CJK...\@gobble}%
4850         \@for\bxac@x:={%
4851           3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4852           3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4853           30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4854           31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4855           31FF%
4856         }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4857       \fi
4858     }%
4859 }

```

以上。

```
4860 \fi
```

G.3 LuaTeX 部分

```

4861 \ifx l\bxac@engine
      ムニヤムニヤ。
4862 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4863   \chardef\pdfTeXrevision=200
4864   \def\pdfTeXrevision{0}
4865   \let\pdfTeXbanner\luatexbanner
4866 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior 4867 \begingroup\expandafter\expandafter\expandafter\endgroup
4868 \expandafter\ifx\csgname outputmode\endcsgname\relax\else
4869 \def\bxac@ob@list{%
4870   \do{\let}\pdfoutput{\outputmode}%
4871   \do{\let}\pdfpagewidth{\pagewidth}%
4872   \do{\let}\pdfpageheight{\pageheight}%
4873   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%

```



```

4874 \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}}
4875 \def\bxac@ob@do#1#2{\begingroup
4876 \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4877 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4878 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4879 \else \let#3#1\relax \let#1\@undefined
4880 \fi}
4881 \protected\def\ImposeOldLuaTeXBehavior{%
4882 \unless\ifbxac@in@old@behavior
4883 \bxac@in@old@behaviortrue
4884 \let\do\bxac@ob@do \bxac@ob@list
4885 \fi}
4886 \protected\def\RevokeOldLuaTeXBehavior{%
4887 \ifbxac@in@old@behavior
4888 \bxac@in@old@behaviorfalse
4889 \let\do\bxac@ob@do \bxac@ob@list
4890 \fi}
4891 \fi

```

漢字および完成形ハングルのカテゴリコードのムニャムニャ。

```

4892 \ifnum\luatexversion>64 \directlua{
4893   local function range(cs, ce, cc, ff)
4894     if ff or not tex.getcatcode(cs) == cc then
4895       local setcc = tex.setcatcode
4896       for c = cs, ce do setcc(c, cc) end
4897     end
4898   end
4899   range(0x3400, 0x4DB5, 11, false)
4900   range(0x4DB5, 0x4DBF, 11, true)
4901   range(0x4E00, 0x9FCC, 11, false)
4902   range(0x9FCD, 0x9FFF, 11, true)
4903   range(0xAC00, 0xD7A3, 11, false)
4904   range(0x20000, 0x2A6D6, 11, false)
4905   range(0x2A6D7, 0x2A6FF, 11, true)
4906   range(0x2A700, 0x2B734, 11, false)
4907   range(0x2B735, 0x2B73F, 11, true)
4908   range(0x2B740, 0x2B81D, 11, false)
4909   range(0x2B81E, 0x2B81F, 11, true)
4910   range(0x2B820, 0x2CEA1, 11, false)
4911   range(0x2CEA2, 0x2FFFD, 11, true)
4912 } \fi

```

以上。

```

4913 \fi

```

G.4 完了

おしまい。

```

4914 %</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

```
4915 %<*cjkcat>
4916 \def\bxjx@pkgname{bxjscjkat}
4917 \newcount\bxjx@cna
4918 \@onlypreamble\bxjx@tmpdo
4919 \@onlypreamble\bxjx@tmpdo@a
4920 \@onlypreamble\bxjx@tmpdo@b
```

\bxjx@engine エンジンの種別。

```
4921 \let\bxjx@engine=n
4922 \def\bxjx@tmpdo#1#2{%
4923   \edef\bxjx@tmpa{\string#1}%
4924   \edef\bxjx@tmpb{\meaning#1}%
4925   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4926 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4927 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4928 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4929 \bxjx@tmpdo\pdfTeXversion{\let\bxjx@engine=p}
4930 \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。

```
4931 \def\bxjx@tmpdo#1#2{%
4932   \if#1\bxjx@engine
4933     \@ifpackageloaded{#2}{\}%else
4934     \PackageError\bxjx@pkgname
4935       {Package '#2' must be loaded}%
4936       {Package loading is aborted.\MessageBreak\@ehc}%
4937     \endinput}
4938 \fi}
4939 \bxjx@tmpdo{p}{bxcjkatype}
4940 \bxjx@tmpdo{x}{xeCJK}
4941 \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。

```
4942 \ifx\TextOrMath\@undefined
4943   \RequirePackage{fixltx2e}
4944 \fi
```

H.2 和文カテゴリコードの設定

upLaTeX の場合、和文カテゴリコードの設定を LuaTeX-já と（ほぼ）等価なものに変更する。

※ LuaTeX-já との相違点：A830、A960、1B000。

```
4945 \if u\bxjx@engine
4946 \@for\bxjx@tmpa:={%
4947 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4948 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4949 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4950 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4951 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4952 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4953 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4954 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4955 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4956 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4957 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4958 FE70,FFF0,%
4959 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4960 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4961 10480,104B0,10500,10530,10600,10800,10840,10860,%
4962 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4963 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4964 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4965 11200,11280,112B0,11300,11400,11480,11580,11600,%
4966 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4967 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4968 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4969 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4970 1EE00,1F000,1F030,1FOA0,1F300,1F600,1F650,1F680,%
4971 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4972 00C0%
4973 }\do{%
4974 \@tempcnta="\bxjx@tmpa\relax
4975 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4976 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4977 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4978 \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-jā 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。

```

4979 \@onlypreamble\bxjx@grkcyr@list
4980 \def\bxjx@grkcyr@list{%
4981 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4982 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4983 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4984 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4985 \do{0395}{LGR}{\textEpsilon}{E}%       % GR. C. L. EPSILON
4986 \do{0396}{LGR}{\textZeta}{Z}%          % GR. C. L. ZETA
4987 \do{0397}{LGR}{\textEta}{H}%           % GR. C. L. ETA
4988 \do{0398}{LGR}{\textTheta}{\Theta}%    % GR. C. L. THETA
4989 \do{0399}{LGR}{\textIota}{I}%          % GR. C. L. IOTA
4990 \do{039A}{LGR}{\textKappa}{K}%         % GR. C. L. KAPPA
4991 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4992 \do{039C}{LGR}{\textMu}{M}%           % GR. C. L. MU
4993 \do{039D}{LGR}{\textNu}{N}%           % GR. C. L. NU
4994 \do{039E}{LGR}{\textXi}{\Xi}%         % GR. C. L. XI
4995 \do{039F}{LGR}{\textOmicron}{O}%      % GR. C. L. OMICRON
4996 \do{03A0}{LGR}{\textPi}{\Pi}%         % GR. C. L. PI
4997 \do{03A1}{LGR}{\textRho}{P}%          % GR. C. L. RHO
4998 \do{03A3}{LGR}{\textSigma}{\Sigma}%    % GR. C. L. SIGMA
4999 \do{03A4}{LGR}{\textTau}{T}%          % GR. C. L. TAU
5000 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
5001 \do{03A6}{LGR}{\textPhi}{\Phi}%       % GR. C. L. PHI
5002 \do{03A7}{LGR}{\textChi}{X}%          % GR. C. L. CHI
5003 \do{03A8}{LGR}{\textPsi}{\Psi}%       % GR. C. L. PSI
5004 \do{03A9}{LGR}{\textOmega}{\Omega}%    % GR. C. L. OMEGA
5005 \do{03B1}{LGR}{\textalpha}{\alpha}%    % GR. S. L. ALPHA
5006 \do{03B2}{LGR}{\textbeta}{\beta}%     % GR. S. L. BETA
5007 \do{03B3}{LGR}{\textgamma}{\gamma}%    % GR. S. L. GAMMA
5008 \do{03B4}{LGR}{\textdelta}{\delta}%    % GR. S. L. DELTA
5009 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
5010 \do{03B6}{LGR}{\textzeta}{\zeta}%      % GR. S. L. ZETA
5011 \do{03B7}{LGR}{\texteta}{\eta}%        % GR. S. L. ETA
5012 \do{03B8}{LGR}{\texttheta}{\theta}%    % GR. S. L. THETA
5013 \do{03B9}{LGR}{\textiota}{\iota}%      % GR. S. L. IOTA
5014 \do{03BA}{LGR}{\textkappa}{\kappa}%    % GR. S. L. KAPPA
5015 \do{03BB}{LGR}{\textlambda}{\lambda}%   % GR. S. L. LAMDA
5016 \do{03BC}{LGR}{\textmu}{\mu}%          % GR. S. L. MU
5017 \do{03BD}{LGR}{\textnu}{\nu}%          % GR. S. L. NU
5018 \do{03BE}{LGR}{\textxi}{\xi}%         % GR. S. L. XI
5019 \do{03BF}{LGR}{\textomicron}{o}%        % GR. S. L. OMICRON
5020 \do{03C0}{LGR}{\textpi}{\pi}%         % GR. S. L. PI

```

5021 \do{03C1}{LGR}{\textrho}{\rho}% % GR. S. L. RHO
5022 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
5023 \do{03C3}{LGR}{\textsigma}{\sigma}% % GR. S. L. SIGMA
5024 \do{03C4}{LGR}{\texttau}{\tau}% % GR. S. L. TAU
5025 \do{03C5}{LGR}{\textupsilon}{\upsilon}% % GR. S. L. UPSILON
5026 \do{03C6}{LGR}{\textphi}{\phi}% % GR. S. L. PHI
5027 \do{03C7}{LGR}{\textchi}{\chi}% % GR. S. L. CHI
5028 \do{03C8}{LGR}{\textpsi}{\psi}% % GR. S. L. PSI
5029 \do{03C9}{LGR}{\textomega}{\omega}% % GR. S. L. OMEGA
5030 \do{0401}{T2A}{\CYRYO}{}% % CY. C. L. IO
5031 \do{0410}{T2A}{\CYRA}{}% % CY. C. L. A
5032 \do{0411}{T2A}{\CYRB}{}% % CY. C. L. BE
5033 \do{0412}{T2A}{\CYRV}{}% % CY. C. L. VE
5034 \do{0413}{T2A}{\CYRG}{}% % CY. C. L. GHE
5035 \do{0414}{T2A}{\CYRD}{}% % CY. C. L. DE
5036 \do{0415}{T2A}{\CYRE}{}% % CY. C. L. IE
5037 \do{0416}{T2A}{\CYRZH}{}% % CY. C. L. ZHE
5038 \do{0417}{T2A}{\CYRZ}{}% % CY. C. L. ZE
5039 \do{0418}{T2A}{\CYRI}{}% % CY. C. L. I
5040 \do{0419}{T2A}{\CYRISHRT}{}% % CY. C. L. SHORT I
5041 \do{041A}{T2A}{\CYRK}{}% % CY. C. L. KA
5042 \do{041B}{T2A}{\CYRL}{}% % CY. C. L. EL
5043 \do{041C}{T2A}{\CYRM}{}% % CY. C. L. EM
5044 \do{041D}{T2A}{\CYRN}{}% % CY. C. L. EN
5045 \do{041E}{T2A}{\CYRO}{}% % CY. C. L. O
5046 \do{041F}{T2A}{\CYRP}{}% % CY. C. L. PE
5047 \do{0420}{T2A}{\CYRR}{}% % CY. C. L. ER
5048 \do{0421}{T2A}{\CYRS}{}% % CY. C. L. ES
5049 \do{0422}{T2A}{\CYRT}{}% % CY. C. L. TE
5050 \do{0423}{T2A}{\CYRU}{}% % CY. C. L. U
5051 \do{0424}{T2A}{\CYRF}{}% % CY. C. L. EF
5052 \do{0425}{T2A}{\CYRH}{}% % CY. C. L. HA
5053 \do{0426}{T2A}{\CYRC}{}% % CY. C. L. TSE
5054 \do{0427}{T2A}{\CYRCH}{}% % CY. C. L. CHE
5055 \do{0428}{T2A}{\CYRSH}{}% % CY. C. L. SHA
5056 \do{0429}{T2A}{\CYRSHCH}{}% % CY. C. L. SHCHA
5057 \do{042A}{T2A}{\CYRHRDSN}{}% % CY. C. L. HARD SIGN
5058 \do{042B}{T2A}{\CYRERY}{}% % CY. C. L. YERU
5059 \do{042C}{T2A}{\CYRSFTSN}{}% % CY. C. L. SOFT SIGN
5060 \do{042D}{T2A}{\CYREREV}{}% % CY. C. L. E
5061 \do{042E}{T2A}{\CYRYU}{}% % CY. C. L. YU
5062 \do{042F}{T2A}{\CYRYA}{}% % CY. C. L. YA
5063 \do{0430}{T2A}{\cyra}{}% % CY. S. L. A
5064 \do{0431}{T2A}{\cyrb}{}% % CY. S. L. BE
5065 \do{0432}{T2A}{\cyrv}{}% % CY. S. L. VE
5066 \do{0433}{T2A}{\cyrg}{}% % CY. S. L. GHE
5067 \do{0434}{T2A}{\cyrd}{}% % CY. S. L. DE
5068 \do{0435}{T2A}{\cyre}{}% % CY. S. L. IE
5069 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE

```

5070 \do{0437}{T2A}{\cyrz}{}%           % CY. S. L. ZE
5071 \do{0438}{T2A}{\cyri}{}%           % CY. S. L. I
5072 \do{0439}{T2A}{\cyrishrt}{}%       % CY. S. L. SHORT I
5073 \do{043A}{T2A}{\cyrk}{}%           % CY. S. L. KA
5074 \do{043B}{T2A}{\cyrl}{}%           % CY. S. L. EL
5075 \do{043C}{T2A}{\cyrm}{}%           % CY. S. L. EM
5076 \do{043D}{T2A}{\cyrn}{}%           % CY. S. L. EN
5077 \do{043E}{T2A}{\cyro}{}%           % CY. S. L. O
5078 \do{043F}{T2A}{\cyrp}{}%           % CY. S. L. PE
5079 \do{0440}{T2A}{\cyrr}{}%           % CY. S. L. ER
5080 \do{0441}{T2A}{\cyrs}{}%           % CY. S. L. ES
5081 \do{0442}{T2A}{\cyrt}{}%           % CY. S. L. TE
5082 \do{0443}{T2A}{\cyru}{}%           % CY. S. L. U
5083 \do{0444}{T2A}{\cyrf}{}%           % CY. S. L. EF
5084 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
5085 \do{0446}{T2A}{\cyrc}{}%           % CY. S. L. TSE
5086 \do{0447}{T2A}{\cyrch}{}%           % CY. S. L. CHE
5087 \do{0448}{T2A}{\cyrsh}{}%           % CY. S. L. SHA
5088 \do{0449}{T2A}{\cyrshch}{}%        % CY. S. L. SHCHA
5089 \do{044A}{T2A}{\cyrhrdsn}{}%       % CY. S. L. HARD SIGN
5090 \do{044B}{T2A}{\cyrrery}{}%        % CY. S. L. YERU
5091 \do{044C}{T2A}{\cyrsftsn}{}%       % CY. S. L. SOFT SIGN
5092 \do{044D}{T2A}{\cyrrerev}{}%       % CY. S. L. E
5093 \do{044E}{T2A}{\cyryu}{}%          % CY. S. L. YU
5094 \do{044F}{T2A}{\cyrya}{}%          % CY. S. L. YA
5095 \do{0451}{T2A}{\cyryo}{}%          % CY. S. L. IO
5096 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
5097 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
5098 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
5099 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
5100 \do{00B4}{TS1}{\textasciicute}{}%  % ACUTE ACCENT
5101 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
5102 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
5103 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
5104 }

```

`\mathdegree` 面倒なので補っておく。

```
5105 \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。

```
5106 \newif\ifbxjx@gcc@CJK
```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```
5107 \newcommand*\greekasCJK{%
```

```
5108   \bxjx@gcc@CJKtrue}
```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```
5109 \newcommand*\nogreekasCJK{%
```

```
5110   \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。

```

5111 \def\bxjx@tmpdo#1\relax{%
5112   \def\bxjx@fake@grk##1##2{%
5113     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
5114   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
5115     \ifx\\##1\\%
5116       \bxjx@ccta##4\divide\bxjx@ccta\@cclvi
5117       \multiply\bxjx@ccta\@cclvi \advance\bxjx@ccta`##3\relax
5118       \mathchar\bxjx@ccta
5119     \else ##3\fi}
5120 } \expandafter\bxjx@tmpdo\string\mathchar\relax

```

■pdfTeX・upTeX の場合

```

5121 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0

```

- `\[bxjx@KC/〈符号値〉]` : その文字が「特定曖昧 CJK 文字」に該当する場合に定義済になる。

まず `inputenc` を読み込んで入力エンコーディングを `utf8` に変更する。

※「既定 UTF-8 化」後の L^AT_EX においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```

5122 \@ifpackageloaded{inputenc}{}{%else
5123   \RequirePackage[utf8]{inputenc}}
5124 \def\bxjx@tmpa{utf8}
5125 \ifx\bxjx@tmpa\inputencdoingname
5126   \PackageWarningNoLine\bxjx@pkcname
5127   {Input encoding changed to utf8}%
5128   \inputencoding{utf8}%
5129 \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。

```

5130 \if u\bxjx@engine
5131 \kcatcode"0370=15
5132 \kcatcode"0400=15
5133 \kcatcode"0500=15
5134 \fi

```

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

```

5135 \def\bxjx@tmpdo#1{%
5136   \@tempccta="#1\relax

```

```

5137 \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
5138 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

引数 = \backslash [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。

```

5139 \ifx\#5\%
5140 \def\bxjx@tmpa{\@inmathwarn#4}%

```

〈数式 LICR〉が英字である場合は \backslash bxjx@fake@grk で出力する。大文字なら \backslash Pi、小文字なら \backslash pi を基準文字にする。

```

5141 \else\ifcat A\noexpand#5%
5142 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5143 { \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。

```

5144 \else \def\bxjx@tmpa{#5}%
5145 \fi\fi
5146 \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
5147 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

5148 \if u\bxjx@engine
5149 \def\bxjx@tmpdo@b#1#2#3#4#5{%

```

引数 = \backslash [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 \backslash 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 \backslash [bxjx@KC/...] を流用している。

```

5150 \kchardef#1=\@tempcnta
5151 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
5152 \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 \backslash UTF{〈符号値〉} を使う（ \backslash UTF は bxcjkatype の命令）。 \backslash [bxjx@KC/...] は使わないが定義済にする必要がある。

```

5153 \else\if p\bxjx@engine
5154 \def\bxjx@tmpdo@b#1#2#3#4#5{%
5155 \mathchardef#1=\@tempcnta
5156 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
5157 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
5158 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

5159 \let\do\bxjx@tmpdo \bxjx@grkcyr@list

```

\backslash bxjx@DeclareUnicodeCharacter \backslash 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。

```

5160 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter

```



```

5161 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
5162 \@onlypreamble\bxjx@DeclareUnicodeCharacter
5163 \def\bxjx@DeclareUnicodeCharacter#1#2{%
5164   \count@=#1\relax
5165   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
5166     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
5167   \else
5168     \wlog{ \space\space skipped defining Unicode char U+#1}%
5169   \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。

```

5170 \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。

```

5171   \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。

```

5172   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
5173   \else \UseTextSymbol{#2}{#3}%
5174   \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。

```

5175 \begingroup
5176 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
5177 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
5178   \noexpand\bxjx@swap@DUC@cmd
5179   \the\toks@
5180   \noexpand\bxjx@swap@DUC@cmd}}
5181 \endgroup\next
5182 \def\bxjx@swap@DUC@cmd{%
5183   \let\bxjx@tmpa\DeclareUnicodeCharacter
5184   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
5185   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
5186   \let\bxjx@tmpa\relax}

```

以上。

■ $\mathrm{X\TeX}$ ・ $\mathrm{Lua\TeX}$ の場合

```

5187 \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。

```

5188 \def\bxjx@tmpdo#1{%
5189   \bxjx@cmta=#1\relax
5190   \begingroup
5191     \lccode`~=\bxjx@cmta
5192   \lowercase{\endgroup

```

```

5193 \bxjx@tmpdo@a{~}{#1}}
5194 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

〈数式 LICR〉が空なら何もしない。空でない場合、up $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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。

```

5195 \ifx\#5\\\let\bxjx@tmpa\relax
5196 \else\ifcat A\noexpand#5%
5197 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5198 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
5199 \else \def\bxjx@tmpa{#5}%
5200 \fi\fi
5201 \ifx\bxjx@tmpa\relax\else
5202 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
5203 \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。

```

5204 \mathchardef\bxjx@tmpa="119
5205 \ifx\bxjx@tmpa\pi \let\do\bxjx@tmpdo \bxjx@grkcyr@list \fi

```

次に、テキストにおいて「特定 CJK 曖昧文字」の扱いが `\[no]greekasCJK` で切り替わるようにする。

Lua $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の場合は、Lua $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -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。

```

5206 \if 1\bxjx@engine
5207 \protected\def\greekasCJK{%
5208 \bxjx@gcc@cjctrue
5209 \ltjsetparameter{jacharrange={+2, +8}}
5210 \protected\def\nogreekasCJK{%
5211 \bxjx@gcc@cjkfalse
5212 \ltjsetparameter{jacharrange={-2, -8}}
5213 \fi

```

X $\mathrm{E}_{\mathrm{L}}\mathrm{A}_{\mathrm{T}}\mathrm{E}_{\mathrm{X}}$ の場合、xeCJK は X $\mathrm{E}_{\mathrm{L}}\mathrm{A}_{\mathrm{T}}\mathrm{E}_{\mathrm{X}}$ 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。

```

5214 \if x\bxjx@engine
5215 \let\bxjx@gcc@cjk@list\@empty
5216 \def\do#1#2#3#4{%
5217 \edef\bxjx@gcc@cjk@list{\bxjx@gcc@cjk@list
5218 \noexpand\XeTeXcharclass"#1\bxjx@cmta}}
5219 \bxjx@grkcyr@list
5220 \protected\def\greekasCJK{%
5221 \bxjx@gcc@cjctrue
5222 \bxjx@cmta=\@ne \bxjx@gcc@cjk@list}
5223 \protected\def\nogreekasCJK{%
5224 \bxjx@gcc@cjkfalse
5225 \bxjx@cmta=\z@ \bxjx@gcc@cjk@list}
5226 \fi

```

以上。

```
5227 \fi\fi
```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5228 \nogreekasCJK
```

H.5 完了

おしまい。

```
5229 %</cjkat>
```

付録 I 補助パッケージ : bxjspandoc

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

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

I.1 準備

```
5230 %<*ancpandoc>
```

```
5231 %% このファイルは日本語文字を含みます。
```

```
5232 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```
5233 \let\bxjsp@engine=n
```

```
5234 \@onlypreamble\bxjsp@do
```

```
5235 \def\bxjsp@do#1#2{%
```

```
5236   \edef\bxjsp@tmpa{\string#1}%
```

```
5237   \edef\bxjsp@tmpb{\meaning#1}%
```

```
5238   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
```

```
5239 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
```

```
5240 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
```

```
5241 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
```

```
5242 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

\bxjsp@begin@document@hook 文書本体開始時フック。

```
5243 \@onlypreamble\bxjsp@begin@document@hook
```

```
5244 \let\bxjsp@begin@document@hook\empty
```

```
5245 \AtBeginDocument{\bxjsp@begin@document@hook}
```

\ifbxjsp@babel@used [スイッチ] Babel が読み込まれたか。

```

5246 \newif\ifbxjsp@babel@used
5247 \g@addto@macro\bxjsp@begin@document@hook{%
5248   \@ifpackageloaded{babel}{\bxjsp@babel@usedtrue}{}}

```

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` の調整を抑止する。
 ※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止の対象になる。
 BXS 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。

```

5249 \newif\ifbxjsp@english
5250 \DeclareOption{english}{\bxjsp@englishttrue}

```

オプション定義はおしまい。

```

5251 \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。

```

5252 \@onlypreamble\pandocSkipLoadFile
5253 \newcommand*\pandocSkipLoadFile[1]{%
5254   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5255 \def\bxjsp@skip@load@file@a#1#2{%
5256   \ifx#1\relax
5257     \def#1{2001/01/01}%
5258     \PackageInfo\bxjsp@pkgname
5259       {File '#2' marked as loaded\@gobble}%
5260   \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。

```

5261 \@onlypreamble\pandocSkipLoadPackage
5262 \newcommand*\pandocSkipLoadPackage[1]{%
5263   \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。これを抑止する。

L^AT_EX カーネルが新しい場合は `fixltx2e` を読込済にする。

```

5264 \ifx\@IncludeInRelease\@undefined\else
5265   \pandocSkipLoadPackage{fixltx2e}
5266 \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。)

```
5267 \if j\bxjsp@engine
5268   \pandocSkipLoadPackage{cmap}
5269 \fi
```

1.6 microtype パッケージ

警告が多すぎなので消す。

```
5270 \if j\bxjsp@engine \else
5271   \PassOptionsToPackage{verbose=silent}{microtype}
5272 \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。

```
5273 \if j\bxjsp@engine
5274   \pandocSkipLoadPackage{microtype}
5275   \newcommand*{\UseMicrotypeSet}[2] [] {}
5276 \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。

```
5277 \DeclareRobustCommand{\pandocLdots}{%
5278   \let\bxjsp@do\bxjsp@ja@ellipsis
5279   \ifmmode \let\bxjsp@do\bxjsp@org@ldots
```

```

5280 \else\ifbxjsp@babel@used
5281 \expandafter\ifx\csname bxjsp@ld/\language\endcsname\relax
5282 \let\bxjsp@do\bxjsp@org@ldots \fi
5283 \fi\fi \bxjsp@do}
5284 \@namedef{bxjsp@ld/japanese}{1}
5285 \def\bxjsp@ja@ellipsis{…}
5286 \let\bxjsp@org@ldots\ldots

```

`\ldots` の実装を `\pandocLdots` に置き換える。

```

5287 \g@addto@macro\bxjsp@begin@document@hook{%
5288 \let\bxjsp@org@ldots\pandocLdots

```

もしここで `\newcommand\pandocLdots{\ldots}` という定義である場合は置き換えない。

```

5289 \long\def\bxjsp@tmpa{\ldots}%
5290 \ifx\pandocLdots\bxjsp@tmpa\else

```

english オプションが指定されていてかつ Babel が読み込まれていない場合も置き換えない。

```

5291 \ifnum0\ifbxjsp@english\ifbxjsp@babel@used\else1\fi\fi=0
5292 \let\ldots\pandocLdots
5293 \fi
5294 \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 $X_{\text{T}}\text{E}_{\text{X}}$ ・Lua $\text{T}_{\text{E}}\text{X}$ 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、非英字扱いにしておく。

※ Pandoc は「Unicode 7.0 で GC が Letter」な文字を英字と判定している。

```

5295 \chardef\bxjsp@cc@other=12
5296 \@onlypreamble\bxjsp@makeother@range
5297 \def\bxjsp@makeother@range#1#2{%
5298 \@tempcnta"#1\relax \@tempcntb"#2\relax
5299 \loop\ifnum\@tempcnta<\@tempcntb
5300 \catcode\@tempcnta\bxjsp@cc@other
5301 \advance\@tempcnta\@ne
5302 \repeat}
5303 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
5304 \catcode"1F23B=\bxjsp@cc@other
5305 \bxjsp@makeother@range{9FCD}{A000}
5306 \bxjsp@makeother@range{1B002}{1B170}
5307 \bxjsp@makeother@range{2B820}{2EBF0}
5308 \fi

```

1.8 Pandola モジュール

インストール済であれば読み込む。

```

5309 \IfFileExists{bxpandola.sty}{%
5310 \RequirePackage{bxpandola}\relax

```

```
5311 \PackageInfo\bxjisp@pkgname
5312 {PandoLa module is loaded\@gobble}
5313 }{}
```

1.9 完了

おしまい。

```
5314 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
5315 %</anc>
```