

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”)

v2.2a [2020/09/22]

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

目次

1	はじめに	3
2	オプション	10
3	和文フォントの変更	39
4	フォントサイズ	40
5	レイアウト	45
5.1	ページレイアウト	46
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)	60
7	ページスタイル	62
8	文書のマークアップ	65
8.1	表題	65
8.2	章・節	70
8.3	リスト環境	82
8.4	パラメータの設定	90
8.5	フロート	91
8.6	キャプション	93
9	フォントコマンド	94

10	相互参照	96
10.1	目次の類	96
10.2	参考文献	102
10.3	索引	103
10.4	脚注	105
11	段落の頭へのグルー挿入禁止	107
12	いろいろなロゴ	111
13	amsmath との衝突の回避	112
14	初期設定	112
付録 A	和文ドライバの仕様 	117
付録 B	和文ドライバ : minimal 	118
B.1	補助マクロ	118
B.2	(u)pTeX 用の設定	120
B.3	pdfTeX 用の処理	125
B.4	X _Y TeX 用の処理	126
B.5	後処理 (エンジン共通)	126
付録 C	和文ドライバ : standard 	129
C.1	準備	130
C.2	和文ドライバパラメタ	130
C.3	共通処理 (1)	131
C.4	pTeX 用設定	138
C.5	pdfTeX 用設定 : CJK + bxcjkatype	141
C.6	X _Y TeX 用設定 : xeCJK + zxjatype	143
C.7	LuaTeX 用設定 : LuaTeX-ja	146
C.8	共通処理 (2)	149
付録 D	和文ドライバ : modern 	150
D.1	フォント設定	150
D.2	fixltx2e 読込	151
D.3	和文カテゴリコード	151
D.4	完了	151
付録 E	和文ドライバ : pandoc 	152
E.1	準備	152
E.2	和文ドライバパラメタ	152
E.3	dupload システム	153

E.4	lang 変数	154
E.5	geometry 変数	157
E.6	CJKmainfont 変数	157
E.7	Option clash 対策	157
E.8	paragraph のマーク	157
E.9	全角空白文字	158
E.10	hyperref 対策	158
E.11	Pandoc 要素に対する和文用の補正	158
E.12	完了	159
付録 F	補助パッケージ一覧 罫	160
付録 G	補助パッケージ : bxjscompat 罫	160
G.1	準備	160
G.2	X _Y TeX 部分	160
G.3	LuaTeX 部分	161
G.4	完了	162
付録 H	補助パッケージ : bxjscjkat 罫	163
H.1	準備	163
H.2	和文カテゴリコードの設定	164
H.3	ギリシャ・キリル文字の扱い	164
H.4	初期設定	172
H.5	完了	172
付録 I	補助パッケージ : bxjspandoc 罫	172
I.1	準備	172
I.2	パッケージオプション	172
I.3	パッケージ読込の阻止	173
I.4	fixltx2e パッケージ	173
I.5	cmap パッケージ	173
I.6	microtype パッケージ	174
I.7	Unicode 文字変換対策	174
I.8	PandoLa モジュール	175
I.9	完了	175

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_EX3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいてもともと奥村晴彦により作成されたものです。現在は日本語 T_EX 開発コミュニティにより GitHub で管理されています。

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

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

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

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

ここでは次のドキュメントクラス（スタイルファイル）を作ります。

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

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

以下では実際のコードに即して説明します。

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

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

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

`\ifjsc@needspatch` [2016-08-22] 従来 jsclasses では、pL^AT_EX や L^AT_EX の不都合な点に対して、クラスファイル内で独自に対策を施していました。しかし、2016 年以降、コミュニティ版 pL^AT_EX が次第に対策コードをカーネル内に取り込むようになりました。そこで、新しい pL^AT_EX カーネ

ルと衝突しないように、日付が古い場合だけパッチをあてる場合があります。この処理に使用するフラグを定義します。

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

■ BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため calc を読み込む。

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

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

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

クラスの本体ではこの他に geometry パッケージが読み込まれる。

TODO: 依存パッケージの情報。

互換性のための補助パッケージを読み込む。

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

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

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

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

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

`\ifjsWithupTeX` [スイッチ] エンジンが (内部漢字コードが Unicode の) `upTeX` であるか。

```
34 \newif\ifjsWithupTeX
35 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
36   \jsWithupTeXtrue
37 \fi\fi
38 \let\if@jsc@uplatex\ifjsWithupTeX
```

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

```
39 \newif\ifjsWithpTeXng
40 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}
```

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

```
41 \newif\ifjsWitheTeX
42 \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 `jTeX` と `Omega` 系。

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

`LuaTeX` の場合、本クラス用の `Lua` モジュールを用意する。

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

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

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

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

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

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

※ `LuaTeX` 0.8x 版でのプリミティブ名変更に対応。

```
64 \newif\ifjsInPdfMode
65 \@nameuse{ImposeOldLuaTeXBehavior}
```

```

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

\ifbxjs@TUenc 〔スイッチ〕 LATEX 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 LATEX カーネルにおいて「Unicode を表す LATEX 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 LATEX を XYLATEX または LuaTEX で動かしている場合は、既定のエンコーディングが TU になる。それ以外の場合は、既定のエンコーディングは OT1 である。
72 \newif\ifbxjs@TUenc
73 \def\bxjs@tmpa{TU}\edef\bxjs@tmpb{\f@encoding}
74 \ifx\bxjs@tmpa\bxjs@tmpb
75   \bxjs@TUenctrue
76 \fi

\bxjs@cond  \bxjs@cond\ifXXX……\fi{〈真〉}{〈偽〉}
      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。
77 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
78   #1\expandafter\@firstoftwo
79   \else\expandafter\@secondoftwo
80   \fi}

\bxjs@cslet  \bxjs@cslet{〈名前 1〉}\制御綴：
81 \def\bxjs@cslet#1{%
82   \expandafter\let\csname#1\endcsname}

\bxjs@csletcs  \bxjs@csletcs{〈名前 1〉}{〈名前 2〉}：
83 \def\bxjs@csletcs#1#2{%
84   \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}

\bxjs@catopt  \bxjs@catopt{〈文字列 1〉}{〈文字列 2〉}： 2 つの文字列を , で繋いだ文字列。ただし片方が空の場合は , を入れない。完全展開可能。
85 \def\bxjs@catopt#1#2{%
86   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

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

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

\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）。

```

```

90 \def\jsSetQHLength#1#2{%
91   \begingroup
92     \bxjs@parse@qh{#2}%
93     \ifx\bxjs@tmpb\relax
94       \setlength\@tempdima{#2}%
95       \xdef\bxjs@g@tmpa{\the\@tempdima}%
96     \else \global\let\bxjs@g@tmpa\bxjs@tmpb
97     \fi
98   \endgroup
99   #1=\bxjs@g@tmpa\relax}

```

\bxjs@parse@qh #1 が Q/H/trueQ/trueH/zw/zh で終わる場合、単位用の寸法値マクロ \bxjs@unit@XXX が定義済なら、\bxjs@tmpb に #1 に等しい寸法の表現を返し、そうでないならエラーを出す。それ以外では、\bxjs@tmpb は \relax になる。

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

```

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

```

今の段階では Q/H だけが使用可能。

```

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

```

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

```

126 \onlypreamble\bxjs@begin@document@hook
127 \let\bxjs@begin@document@hook\empty

```



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

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

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

```
130 \let\bxjs@post@option@hook\@empty
```

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

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

```
132 \let\bxjs@pre@jadriver@hook\@empty
```

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

```
133 \def\jsAtEndOfClass{%
```

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

一時的な手続き用の制御綴。

```
135 \onlypreamble\bxjs@tmpdo
```

```
136 \onlypreamble\bxjs@tmpdo@a
```

```
137 \onlypreamble\bxjs@tmpdo@b
```

```
138 \onlypreamble\bxjs@tmpdo@c
```

```
139 \onlypreamble\bxjs@tmpdo@d
```

LuaTeX 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。クラス読み込終了時点で元に戻される。

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

```
140 \if 1\jsEngine
```

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

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

```
143     \bxjs@pre@jadriver@hook
```

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

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

```
146 \tfor\bxjs@tmpa:=和西暦\do
```

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

```
148 \fi
```

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

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

```
150   \ifx\inhibitglue\@undefined\else \inhibitglue \fi}
```

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

```
151 \if@compatibility
```

```
152   \ClassError\bxjs@clsname
```

```
153     {Something went chaotic!\MessageBreak
```

```
154     (How come '\string\documentstyle' is there?)\MessageBreak
```

```
155     I cannot go a single step further...}
```

```
156   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
```

```

157     then there'll still be hope....}
158   \expandafter\@firstofone
159 \else \expandafter\@gobble
160 \fi{\typeout{Farewell!}}\endinput\@@end}

```

2 オプション

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

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

`\if@restonecol` 段組のときに真になる論理変数です。

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

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

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

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

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

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

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

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

BXJS では report 系でも定義されることに注意。

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

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

JS クラスと異なり、初期値は偽とする。

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

以下で各オプションを宣言します。

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

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

が、 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X} 2_{\epsilon}$ の `b5paper` は $257\text{ mm} \times 182\text{ mm}$ になっています。ここでは $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X} 2_{\epsilon}$ にならって JIS に従いました。

デフォルトは `a4paper` です。

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

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

```

167 \@onlypreamble\bxjs@setpaper
168 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
169 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
170 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
171 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
172 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
173 \DeclareOption{b4paper}{\bxjs@setpaper{b4j}}
174 \DeclareOption{b5paper}{\bxjs@setpaper{b5j}}
175 \DeclareOption{b6paper}{\bxjs@setpaper{b6j}}
176 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
177 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
178 \DeclareOption{b4j}{\bxjs@setpaper{b4j}}
179 \DeclareOption{b5j}{\bxjs@setpaper{b5j}}
180 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
181 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
182 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
183 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
184 \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。
185 \def\bxjs@tmpb#1#2{\DeclareOption{#1}{\bxjs@setpaper{#2}}}
186 \@for\bxjs@tmpa:={%
187   a0,a1,a2,c0,c1,c2,c3,c4,c5,c6,ansia,ansib,ansic,ansid,ansie%
188 }\do{\edef\bxjs@next{%
189   \noexpand\bxjs@tmpb{\bxjs@tmpa paper}{\bxjs@tmpa paper}%
190 }\bxjs@next}
191 \bxjs@tmpb{screen}{screen}

```

ただし `b?paper` は JIS の B 列に従う。

```

192 \@for\bxjs@tmpa:={0,1,2,3}\do{\edef\bxjs@next{%
193   \noexpand\bxjs@tmpb{b\bxjs@tmpa paper}{b\bxjs@tmpa j}%
194 }\bxjs@next}

```

`B?paper` で ISO の B 列を指定できるようにする。

```

195 \@for\bxjs@tmpa:={0,1,2,3,4,5,6}\do{\edef\bxjs@next{%
196   \noexpand\bxjs@tmpb{A\bxjs@tmpa paper}{a\bxjs@tmpa paper}%
197   \noexpand\bxjs@tmpb{B\bxjs@tmpa paper}{b\bxjs@tmpa paper}%
198 }\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。

```
199 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}  
200 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}  
201 \DeclareOption{screenpaper}{\bxjs@setpaper{screen}}  
202 \DeclareOption{custompaper}{}
```

■横置き 用紙の縦と横の長さを入れ換えます。

```
203 \newif\if@landscape  
204 \@landscapefalse  
205 \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。

```
206 \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。

```
207 %<!slide>\@slidefalse  
208 %<slide>\@slidetrue
```

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

[2016-07-24] オプティカルサイズを調整するために NFSS ヘパッチを当てるオプション `nomag*` を新設しました。

`\@ptsize` は 10pt, 11pt, 12pt が指定された時のみ従来と同じ値とし、それ以外は `\jsUnusualPtSize` (= -20) にする。

```
209 \newcommand{\@ptsize}{0}  
210 \def\bxjs@param@basefontsize{10pt}
```

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

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

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

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

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

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

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

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

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

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

```
218 \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。

```
219 \def\bxjs@setjbasefontsize#1{%
```

```
220 \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。

```
221 \newif\ifjsc@mag
```

```
222 \newif\ifjsc@mag@xreal
```

```
223 %\let\jsc@magscale\@undefined
```

```
224 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
```

```
225 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
```

```
226 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
```

```
227 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
```

```
228 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
```

```
229 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
```

```
230 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
```

```
231 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
```

```
232 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
```

```
233 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
```

```
234 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
```

```
235 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
```

```
236 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
```

```
237 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
```

```
238 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
```

```
239 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
```

```
240 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
```

```
241 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
```

```
242 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}
```

JS クラス互換の `magstyle` 設定オプション。

```
243 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
```

```
244 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
```

```
245 \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。

```
246 \if j\jsEngine
247 \hour\time \divide\hour by 60\relax
248 \@tempcnta\hour \multiply\@tempcnta 60\relax
249 \minute\time \advance\minute-\@tempcnta
250 \DeclareOption{tombow}{%
251   \tombowtrue \tombowdatetrue
252   \setlength{\@tombowwidth}{.1\p}%
253   \@bannertoken{%
254     \jobname\space(\number\year-\two@digits\month-\two@digits\day
255     \space\two@digits\hour:\two@digits\minute)}}%
256   \maketombowbox}
257 \DeclareOption{tombo}{%
258   \tombowtrue \tombowdatefalse
259   \setlength{\@tombowwidth}{.1\p}%
260   \maketombowbox}
261 \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。

```
262 \if j\jsEngine
263 \DeclareOption{mentuke}{%
264   \tombowtrue \tombowdatefalse
265   \setlength{\@tombowwidth}{\z@}%
266   \maketombowbox}
267 \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。

```
268 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
269 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
270 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
271 \DeclareOption{onecolumn}{\@twocolumnfalse}
272 \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。

```
273 \DeclareOption{titlepage}{\@titlepagetrue}
274 \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。

```
275 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
276 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
277 %<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。

```
278 \def\eqnarray{%
279   \stepcounter{equation}%
280   \def\@currentlabel{\p@equation\theequation}%
281   \global\@eqnswtrue
282   \m@th
283   \global\@eqcnt\z@
284   \tabskip\@centering
285   \let\\\@eqnocr
286   $$\everycr{}\halign to\displaywidth\bgroup
287     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
288     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
289     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
290     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
291     \tabskip\z@skip
292   \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。

```
293 \DeclareOption{leqno}{\input{leqno.clo}}
294 \DeclareOption{fleqn}{\input{fleqn.clo}}
295 % fleqn 用の eqnarray 環境の再定義
296 \def\eqnarray{%
297   \stepcounter{equation}%
298   \def\@currentlabel{\p@equation\theequation}%
299   \global\@eqnswtrue\m@th
300   \global\@eqcnt\z@
```

```

301 \tabskip\mathindent
302 \let\=\@eqnocr
303 \setlength\abovedisplayskip{\topsep}%
304 \ifvmode
305 \addtolength\abovedisplayskip{\partopsep}%
306 \fi
307 \addtolength\abovedisplayskip{\parskip}%
308 \setlength\belowdisplayskip{\abovedisplayskip}%
309 \setlength\belowdisplayshortskip{\abovedisplayskip}%
310 \setlength\abovedisplayshortskip{\abovedisplayskip}%
311 $$\everycr{}\halign to\linewidth% $$
312 \bgroup
313 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
314 &\global\@eqcnt\@ne \hfil$\displaystyle{##}\hfil
315 &\global\@eqcnt\tw@
316 $\displaystyle{##}$\hfil \tabskip\@centering
317 &\global\@eqcnt\thr@@ \hbext@z@\bgroup\hss#\egroup
318 \tabskip\z@skip\cr
319 }}

```

■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。

```

320 % \DeclareOption{openbib}{%
321 % \AtEndOfPackage{%
322 % \renewcommand\@openbib@code{%
323 % \advance\leftmargin\bibindent
324 % \itemindent -\bibindent
325 % \listparindent \itemindent
326 % \parsep \z@}%
327 % \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：

```

328 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。
329 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
330 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}

```



```

331 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\@undefined}
332 \define@key{bxjs}{enablejfam}[true]{%
333   \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。

```

334 \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。

```

335 \newif\ifjsDraft
336 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
337 \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。

```

338 \newif\ifbxjs@papersize
339 \bxjs@papersizetrue
340 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
341 \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。

```

342 \newif\if@english
343 \@englishfalse
344 \DeclareOption{english}{\@englishttrue}

```

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

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

`BXJS` では当初から `bxjsreport` クラスが用意されている。


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

`BXJS` クラスでは、`nojslogo` を既定とする。

```

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

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

※元の長さが 128 pt 以上の場合でも動作するように修正した。

```

348 \mathchardef\bxjs@isc@ll=128
349 \mathchardef\bxjs@isc@sl=259
350 \def\bxjs@isc@sl@h{65539 }
351 \def\bxjs@invscale#1#2{%
352   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax

```

```

353 \ifdim\@tempdima<\bxjs@isc@ll\p@
354 \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
355 \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
356 \else
357 \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
358 \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
359 \fi
360 \@tempcntb\p@ \divide\@tempcntb\@tempdimb
361 \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
362 \@tempdimb\@tempcnta\@ne
363 \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
364 \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
365 \@whiledim\@tempdimb<\@tempdimc\do{%
366 \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
367 \advance\@tempcntb\@ne \divide\@tempcntb\tw@
368 \ifdim #2\@tempcntb>\@tempdima
369 \advance\@tempcntb\@ne \@tempdimc=\@tempcntb\@ne
370 \else \@tempdimb=\@tempcntb\@ne \fi}%
371 \xdef\bxjs@gtmpa{\the\@tempdimb}%
372 \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。

```
373 \DeclareOption{pandoc}{%
```

和文ドライバを `pandoc` に、エンジン指定を `autodetect-engine` に変更する。

※実際の和文ドライバ・エンジン設定より優先される。

```

374 \g@addto@macro\bxjs@post@option@hook{%
375 \bxjs@oldfontcommandstrue
376 \setkeys{bxjs}{ja=pandoc}%
377 \let\bxjs@engine@given=*}%

```

ドライバオプションを `dvi=dvipdfmx` 相当に変更する。

※これは実際のドライバ設定で上書きできる (オプション宣言順に注意)。

TODO: できない気がする…。

```

378 \def\bxjs@driver@opt{dvipdfmx}%
379 \bxjs@dvi@opttrue}

```

■エンジン・ドライバオプション

`\bxjs@engine@given` オプションで明示されたエンジンの種別。

```
380 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
381 %\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。

```
382 \DeclareOption{autodetect-engine}{%
383   \let\bxjs@engine@given=*}
384 \DeclareOption{latex}{%
385   \def\bxjs@engine@opt{latex}%
386   \let\bxjs@engine@given=n}
387 \DeclareOption{platex}{%
388   \def\bxjs@engine@opt{platex}%
389   \let\bxjs@engine@given=j}
390 \DeclareOption{uplatex}{%
391   \def\bxjs@engine@opt{uplatex}%
392   \let\bxjs@engine@given=u}
393 \DeclareOption{xelatex}{%
394   \def\bxjs@engine@opt{xelatex}%
395   \let\bxjs@engine@given=x}
396 \DeclareOption{pdflatex}{%
397   \def\bxjs@engine@opt{pdflatex}%
398   \let\bxjs@engine@given=p}
399 \DeclareOption{lualatex}{%
400   \def\bxjs@engine@opt{lualatex}%
401   \let\bxjs@engine@given=l}
402 \DeclareOption{platex-ng}{%
403   \def\bxjs@engine@opt{platex-ng}%
404   \let\bxjs@engine@given=g}
405 \DeclareOption{platex-ng*}{%
406   \def\bxjs@engine@opt{platex-ng*}%
407   \let\bxjs@platexng@nodrv=t%
408   \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。

```
409 %\let\bxjs@driver@given\@undefined
410 \let\bxjs@driver@@dvimode=0
411 \let\bxjs@driver@@dvipdfmx=1
412 \let\bxjs@driver@@pdfmode=2
413 \let\bxjs@driver@@xetex=3
414 \let\bxjs@driver@@dvips=4
415 \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。

```
416 %\let\bxjs@driver@opt\@undefined
```

```
417 \DeclareOption{dvips}{%
```

```

418 \def\bxjs@driver@opt{dvips}%
419 \let\bxjs@driver@given\bxjs@driver@@dvips}
420 \DeclareOption{dviout}{%
421 \def\bxjs@driver@opt{dviout}%
422 \let\bxjs@driver@given\bxjs@driver@@dvimode}
423 \DeclareOption{xdvi}{%
424 \def\bxjs@driver@opt{xdvi}%
425 \let\bxjs@driver@given\bxjs@driver@@dvimode}
426 \DeclareOption{dviPDFmx}{%
427 \def\bxjs@driver@opt{dviPDFmx}%
428 \let\bxjs@driver@given\bxjs@driver@@dviPDFmx}
429 \DeclareOption{nodvidriver}{%
430 \def\bxjs@driver@opt{nodvidriver}%
431 \let\bxjs@driver@given\bxjs@driver@@none}
432 \DeclareOption{pdftex}{%
433 \def\bxjs@driver@opt{pdftex}%
434 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
435 \DeclareOption{luatex}{%
436 \def\bxjs@driver@opt{luatex}%
437 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
438 \DeclareOption{xetex}{%
439 \def\bxjs@driver@opt{xetex}%
440 \let\bxjs@driver@given\bxjs@driver@@xetex}

```

dviPDFmx-if-dvi は互換性のためのオプションで、dvi=dviPDFmx と同値である。

※ 2.0 版より dviPDFmx-if-dvi は非推奨となった。

```

441 \DeclareOption{dviPDFmx-if-dvi}{%
442 \ClassWarningNoLine\bxjs@clsname
443 {The old option 'dviPDFmx-if-dvi' is DEPRECATED\MessageBreak
444 and may be abolished in future!\MessageBreak
445 You should write 'dvi=dviPDFmx' instead}%
446 \setkeys{bxjs}{dvi=dviPDFmx}}
447

```

■その他の BXJS 独自オプション

TODO: 互換用オプションを分離する (2.0 版で?)。

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

```

448 \newif\ifbxjs@bigcode \bxjs@bigcodetrue

```

nobigcode / bigcode オプションの定義。

```

449 \DeclareOption{nobigcode}{%
450   \bxjs@bigcodefalse}
451 \DeclareOption{bigcode}{%
452   \bxjs@bigcodetrue}

```

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

```

453 \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。

```

454 \DeclareOption{nooldfontcommands}{%
455   \bxjs@oldfontcommandsfalse}
456 \DeclareOption{oldfontcommands}{%
457   \bxjs@oldfontcommandstrue}


```

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

```

458 \DeclareOption{winjis}{%
459   \ClassWarningNoLine\bxjs@clsname
460     {This class does not support `winjis' option}%
461 }
462 \DeclareOption{mingoth}{%
463   \ClassWarningNoLine\bxjs@clsname
464     {This class does not support `mingoth' option}%
465 }
466 \DeclareOption{jis}{%
467   \ClassWarningNoLine\bxjs@clsname
468     {This class does not support `jis' option}%
469 }

```

■`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。

```

470 \DeclareOption*{%
471   \def\bxjs@next{\bxjs@safe@setkeys\bxjs}%
472   \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。

```

473 \def\bxjs@safe@setkeys#1#2{%
474   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
475   \setkeys{#1}{#2}%
476   \let\KV@errx\bxjs@save@KV@errx}

```

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

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

```
477 \@onlypreamble\bxjs@declare@enum@option
478 \def\bxjs@declare@enum@option#1#2{%
479   \define@key{bxjs}{#1}{%
480     \expandafter\ifx\csname bxjs@#2@@##1\endcsname\relax
481     \bxjs@error@keyval{#1}{##1}%
482   \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@@##1}%
483   \fi}}
```

`\bxjs@declare@bool@option` `\bxjs@declare@bool@option{〈オプション名〉}{〈スイッチ名〉}`

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

```
484 \@onlypreamble\bxjs@declare@bool@option
485 \def\bxjs@declare@bool@option#1#2{%
486   \define@key{bxjs}{#1}[true]{%
487     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
488     \bxjs@error@keyval{#1}{##1}%
489   \else \@nameuse{bxjs@#2##1}%
490   \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。

```
491 \def\bxjs@set@keyval#1#2#3{%
492   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
493   \ifx\bxjs@next\relax
494     \bxjs@error@keyval{#1}{#2}%
495     #3%
496   \else \bxjs@next
497   \fi}
498 \@onlypreamble\bxjs@error@keyval
499 \def\bxjs@error@keyval#1#2{%
500   \ClassError\bxjs@clsname
501   {Invalid value '#2' for option #1}\@ehc}
```

`\jsScale` 〔実数値マクロ〕 和文スケール値。

```
502 \def\jsScale{0.924715}
```

`\bxjs@base@opt` 明示された base オプションの値。

```
503 %\let\bxjs@base@opt\@undefined
```

base オプションの処理。

```
504 \define@key{bxjs}{base}{%
505   \edef\bxjs@base@opt{#1}%
506   \bxjs@setbasefontsize{#1}}
507 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}
```

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

```
508 %\let\bxjs@jbase@opt\@undefined
```

jbase オプションの処理。

```

509 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
510 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}

```

\bxjs@scale@opt 明示された scale オプションの値。

```

511 %\let\bxjs@scale@opt\@undefined

```

scale オプションの処理。

```

512 \define@key{bxjs}{scale}{%
513   \edef\bxjs@scale@opt{#1}%
514   \let\jsScale\bxjs@scale@opt}
515 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

```

noscale オプションの処理。

```

516 \DeclareOption{noscale}{\setkeys{bxjs}{scale=1}}

```

\bxjs@param@mag mag オプションの値。

```

517 \let\bxjs@param@mag\relax

```

mag オプションの処理。

```

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

```

paper オプションの処理。

```

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

```

\bxjs@jadriver 和文ドライバの名前。

```

520 \let\bxjs@jadriver\relax
521 %\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）。

```

522 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver@opt{#1}}
523 \define@key{bxjs}{ja}[\relax]{%
524   \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}

```

\jsJaFont 和文フォント設定の名前。

```

525 \let\jsJaFont\@empty

```

jafont オプションの処理。

```

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

```

\jsJaParam 和文ドライバパラメタの文字列。

```

527 \let\jsJaParam\@empty

```

japaram オプションの処理。

```

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

```


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

```
529 \let\bxjs@magstyle@mag=m
530 \let\bxjs@magstyle@real=r
531 \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。

```
532 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
533 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
534 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal

\bxjs@magstyle@default は既定の値を表す。
```

```
535 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
536 \ifx \ljsEngine \ifnum\luatexversion>86
537   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
538 \fi\fi
539 \ifjsWithpTeXng
540   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
541 \fi
542 \let\bxjs@magstyle\bxjs@magstyle@default
```

magstyle オプションの処理。

```
543 \define@key{bxjs}{magstyle}{%
544   \expandafter\let\expandafter\bxjs@magstyle\csname
545     bxjs@magstyle@#1\endcsname
546   \ifx\bxjs@magstyle\relax
547     \ClassError\bxjs{clsname
548       {Invalid value '#1' for option magstyle}}\@ehc
549     \let\bxjs@magstyle\bxjs@magstyle@default
550   \fi}
```

`\bxjs@geometry` geometry オプションの値。

```
551 \let\bxjs@geometry@class=c
552 \let\bxjs@geometry@user=u
553 \let\bxjs@geometry\bxjs@geometry@class
```

geometry オプションの処理。

```
554 \define@key{bxjs}{geometry}{%
555   \expandafter\let\expandafter\bxjs@geometry\csname
556     bxjs@geometry@#1\endcsname
557   \ifx\bxjs@geometry\relax
558     \ClassError\bxjs{clsname
559       {Invalid value '#1' for option geometry}}\@ehc
560     \let\bxjs@geometry\bxjs@geometry@class
561   \fi}
```

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

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

fancyhdr オプションの処理。

```
563 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
564 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
565 \define@key{bxjs}{fancyhdr}[true]{%
566   \bxjs@set@keyval{fancyhdr}{#1}{}}
```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```
567 \newif\ifbxjs@dvi@opt
```

DVI モードのドライバとドライバ種別との対応。

```
568 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
569 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
570 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
571 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
572 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none
```

dvi オプションの処理。

```
573 \define@key{bxjs}{dvi}{%
574   \expandafter\let\expandafter\bxjs@tmpa\csname
575     bxjs@dvidriver@@#1\endcsname
576   \ifx\bxjs@tmpa\relax
577     \ClassError\bxjs@clsname
578     {Invalid value '#1' for option dvi}\@ehc
579   \else
```

\bxjs@driver@given を未定義にしていることに注意。

```
580   \def\bxjs@driver@opt{#1}%
581   \let\bxjs@driver@given\undefined
582   \bxjs@dvi@opttrue
583 \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。

```
584 \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。

```
585 \newif\ifbxjs@force@chapterabstract
586 %<book>\bxjs@force@chapterabstracttrue
```

layout オプションの処理。

```
587 \@namedef{bxjs@kv@layout@v1}{%
588 %<book>\bxjs@layout@buggyhmargintrue
589 %<report>\bxjs@force@chapterabstracttrue
590 }
591 \@namedef{bxjs@kv@layout@v2}{%
592 %<book>\bxjs@layout@buggyhmarginfalse
593 %<report>\bxjs@force@chapterabstractfalse
594 }
```

```

595 \define@key{bxjs}{layout}{%
596   \bxjs@set@keyval{layout}{#1}{}}

```

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

```

597 %\let\bxjs@textwidth@limit@opt\@undefined
598 \define@key{bxjs}{textwidth-limit}{%
599   \edef\bxjs@textwidth@limit@opt{#1}}

```

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

```

600 %\let\bxjs@textwidth@opt\@undefined
601 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
602 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

```

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

```

603 %\let\bxjs@number@of@lines@opt\@undefined
604 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
605 \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。

```

606 %\let\bxjs@paragraph@mark\@undefined
607 \define@key{bxjs}{paragraph-mark}{%
608   \edef\bxjs@paragraph@mark{#1}}

```

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

```

609 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
610 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
611 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
612 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}

```

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

```

613 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
614 \let\bxjs@kv@jaspacecmd@true\bxjs@jaspace@cmdtrue
615 \let\bxjs@kv@jaspacecmd@false\bxjs@jaspace@cmdfalse
616 \define@key{bxjs}{jaspace-cmd}[true]{\bxjs@set@keyval{jaspacecmd}{#1}{}}
617 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{jaspacecmd}{#1}{}}

```

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

```

618 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
619 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
620 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
621 \define@key{bxjs}{fix-at-cmd}[true]{\bxjs@set@keyval{fixatcmd}{#1}{}}

```

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

```

622 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
623 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
624 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
625 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}

```

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

```

626 \chardef\bxjs@everyparhook@none=0
627 \chardef\bxjs@everyparhook@compat=1
628 \chardef\bxjs@everyparhook@modern=2
629 \if j\jsEngine
630 \let\bxjs@everyparhook\bxjs@everyparhook@compat
631 \else
632 \let\bxjs@everyparhook\bxjs@everyparhook@modern
633 \fi
634 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
635 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
636 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
637 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}

```

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

```

638 \chardef\bxjs@label@section@none=0
639 \chardef\bxjs@label@section@compat=1
640 \chardef\bxjs@label@section@modern=2
641 \let\bxjs@label@section\bxjs@label@section@compat
642 \def\bxjs@kv@labelsection@none{\let\bxjs@label@section\bxjs@label@section@none}
643 \def\bxjs@kv@labelsection@compat{\let\bxjs@label@section\bxjs@label@section@compat}
644 \def\bxjs@kv@labelsection@modern{\let\bxjs@label@section\bxjs@label@section@modern}
645 \define@key{bxjs}{label-section}{\bxjs@set@keyval{labelsection}{#1}{}}

```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```

646 \newif\ifbxjs@usezw \bxjs@usezwtrue
647 \bxjs@declare@bool@option{use-zw}{usezw}
648 \DeclareOption{noz}{\setkeys{bxjs}{use-zw=false}}
649 \DeclareOption{zw}{\setkeys{bxjs}{use-zw=true}}

```

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

```

650 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
651 \bxjs@declare@bool@option{disguise-js}{disguise@js}
652 \DeclareOption{nojs}{\setkeys{bxjs}{disguise-js=false}}
653 \DeclareOption{js}{\setkeys{bxjs}{disguise-js=true}}

```

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

```

654 \newif\ifbxjs@precisetext
655 \bxjs@declare@bool@option{precise-text}{precisetext}
656 \DeclareOption{noprecisetext}{\setkeys{bxjs}{precise-text=false}}
657 \DeclareOption{precisetext}{\setkeys{bxjs}{precise-text=true}}

```

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

```

658 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
659 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}
660 \DeclareOption{nosimplejasetup}{\setkeys{bxjs}{simple-ja-setup=false}}
661 \DeclareOption{simplejasetup}{\setkeys{bxjs}{simple-ja-setup=true}}

```

■ オプションの実行

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

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

```
662 \let\bxjs@org@removeelement\@removeelement
663 \def\@removeelement#1#2#3{%
664   \def\reserved@a{#2}%
665   \ifx\reserved@a\@empty \let#3\@empty
666   \else \bxjs@org@removeelement{#1}{#2}{#3}%
667   \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。

```
668 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
669 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
670 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
671 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
672 \ProcessOptions\relax
673 \bxjs@post@option@hook
```

後処理

```
674 \if@slide
675   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
676 \fi
677 \if@landscape
678   \setlength\@tempdima {\paperheight}
679   \setlength\paperheight{\paperwidth}
680   \setlength\paperwidth {\@tempdima}
681 \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。

```
682 \def\bxjs@tmpdo{%
683   \def\bxjs@tmpa{\gobble}%
684   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
685   \let\@classoptionslist\bxjs@tmpa}
686 \def\bxjs@tmpdo@a#1,{%
687   \ifx\@nil#1\relax\else
688     \bxjs@tmpdo@b#1{\@nil
```

```

689 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
690 \expandafter\bxjs@tmpdo@a
691 \fi}
692 \def\bxjs@tmpdo@b#1#{\bxjs@tmpdo@c}
693 \def\bxjs@tmpdo@c#1\@nil{%
694 \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
695 \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。

```

696 \@expandtwoargs\@removeelement
697 {papersize}\@classoptionslist\@classoptionslist
698 \@expandtwoargs\@removeelement
699 {10pt}\@classoptionslist\@classoptionslist
700 \@expandtwoargs\@removeelement
701 {noscale}\@classoptionslist\@classoptionslist

```

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

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

ここのコードを削除。

[2016-11-11] pL^AT_EX の場合は、オプション `uplatex` が指定されていれば必ずエラーを出します。`autodetect-engine` が有効になって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。

```

702 \ifx\bxjs@jadriver@opt\@undefined\else
703 \let\bxjs@jadriver\bxjs@jadriver@opt
704 \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。

```

705 \let\bxjs@tmpb\jsEngine
706 \ifx j\bxjs@tmpb\ifjsWithpTeXng
707 \let\bxjs@tmpb=g
708 \fi\fi
709 \ifx j\bxjs@tmpb\ifjsWithupTeX
710 \let\bxjs@tmpb=u
711 \fi\fi
712 \ifx p\bxjs@tmpb\ifjsInPdfMode\else

```

```
713 \let\bxjs@tmpb=n
```

```
714 \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。)

```
715 \ifx *\bxjs@engine@given
```

```
716 \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。

```
717 \ifx j\bxjs@engine@given
```

```
718 \g@addto@macro\@classoptionslist{,latex}
```

```
719 \else\ifx u\bxjs@engine@given
```

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

```
721 \fi\fi
```

```
722 \fi
```

```
723 \ifx\bxjs@engine@given\@undefined\else
```

```
724 \ifx\bxjs@engine@given\bxjs@tmpb\else
```

```
725 \ClassError\bxjs@clsname
```

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

```
727 \fi
```


```
728 \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。

```
729 \ifjsWithpTeXng
```

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

```
731 \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。

```
732 \@tempswatrue
```

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

```
734 \ifjsInPdfMode
```

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

```
736 \@tempswafalse
```

```
737 \fi
```

```
738 \else\ifx x\jsEngine
```

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

```
740 \@tempswafalse
```

```
741 \fi
```

```
742 \else
```

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

```
744 \@tempswafalse
```

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

```
746 \@tempswafalse
```

```
747 \fi\fi
```

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

```
749 \@tempswafalse
```

```
750 \fi\fi
```

```

751 \fi\fi
752 \fi
753 \if@tempswa\else
754 \ClassError\bxjs@clsname
755 {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
756 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

757 \ifjsInPdfMode \@tempswafalse
758 \else\ifx x\jsEngine \@tempswafalse
759 \else\ifjsWithpTeXng \@tempswafalse
760 \else \@tempswatrue
761 \fi\fi\fi
762 \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)。

```

763 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@opt\@undefined\else
764 \ClassWarningNoLine\bxjs@clsname
765 {No driver option is given}
766 \fi\fi

```

dvi=XXX が指定されていた場合は、XXX が指定された時と同じ動作にする。(グローバルオプションに XXX を追加する。)

```

767 \ifbxjs@dvi@opt
768 \edef\bxjs@next{%
769 \let\noexpand\bxjs@driver@given
770 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
771 \noexpand\g@addto@macro\noexpand\@classoptionslist
772 {,\bxjs@driver@opt}%
773 }\bxjs@next
774 \fi
775 \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。

```

776 \ifjsWithpTeXng
777 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
778 \let\bxjs@platexng@nodrv\@undefined
779 \else\ifx t\bxjs@platexng@nodrv\else
780 \g@addto@macro\@classoptionslist{,dvipdfmx}
781 \fi\fi
782 \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。

```

783 \ifx\bxjs@driver@given\bxjs@driver@@none
784 \bxjs@papersizefalse
785 \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。

```

786 \def\bxjs@@minimal{minimal}
787 \ifx\bxjs@jadriver\relax
788   \ifx j\jsEngine
789     \def\bxjs@jadriver{standard}
790   \else
791     \ClassWarningNoLine\bxjs@clsname
792     {The option 'ja' is MISSING!!\MessageBreak
793     So 'ja=minimal' is assumed as fallback, but\MessageBreak
794     such implicit setting is now DEPRECATED!\MessageBreak
795     You should write 'ja=minimal' explicitly,\MessageBreak
796     if it is intended}
797     \let\bxjs@jadriver\bxjs@@minimal
798   \fi
799 \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。

```

800 \ifx\bxjs@jadriver@opt\@undefined\else
801   \ifx\bxjs@engine@given\@undefined
802     \ClassError\bxjs@clsname
803     {An engine option must be explicitly given}%
804     {When you use a Japanese-driver you must specify a correct\MessageBreak
805     engine option.\MessageBreak\@ehc}
806 \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。)

```

807 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
808   \ifx\bxjs@magstyle\bxjs@magstyle@mag
809     \let\bxjs@magstyle\bxjs@magstyle@default
810     \ClassError\bxjs@clsname
811     {The engine does not support 'magstyle=usemag'}%
812     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
813     The default value 'nomag*' is used instead.\MessageBreak \@ehc}
814   \fi
815 \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も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。

```

816 \ifx\bxjs@jbase@opt\@undefined\else
817   \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。

```
818 \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
819 \bxjs@invscale\@tempdima\jsScale
820 \bxjs@setbasefontsize{\@tempdima}%
821 \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。
822 \ifx\bxjs@scale@opt\@undefined\else
823 \ClassWarningNoLine\bxjs@clsname
824 {Redundant 'scale' option is ignored}%
825 \fi
826 \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
827 \@tempdimb=\bxjs@param@basefontsize\relax
828 \edef\jsScale{\strip@pt\@tempdimb}%
829 \bxjs@invscale\@tempdima\jsScale
830 \edef\jsScale{\strip@pt\@tempdima}%
831 \fi
832 \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。

```
833 \let\Cjascale\jsScale
```

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

```
834 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
835 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
836 \catcode\@tempcnta\active
837 \advance\@tempcnta\@ne
838 \repeat
839 \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。

```
840 \ifbxjs@disguise@js
841 %<book|report>\def\bxjs@js@clsname{jsbook}
842 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
843 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
844 \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。

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

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

```
846 \ifbxjs@oldfontcommands
847 \AtEndOfClass{\allowoldfontcommands}
```

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

トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが pL^AT_EX 2_ε はトンボ出力幅を両側に 1 インチとっていますので、dvips 使用時に

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

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

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

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

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

BXJS では出力用紙サイズ記録は geometry パッケージが行う。

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

```
849 \newdimen\stockwidth \newdimen\stockheight
850 \begingroup\expandafter\expandafter\expandafter\endgroup
851 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
852 % \newdimen\stockwidth \newdimen\stockheight
853 \setlength{\stockwidth}{\paperwidth}
854 \setlength{\stockheight}{\paperheight}
855 \advance \stockwidth 2in
856 \advance \stockheight 2in
857 \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。

```
858 %<slide>\def\n@baseline{13}%
859 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
860 %<!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。

```

861 \ifx\bxjs@magstyle\bxjs@magstyle@mag
862   \jsc@magtrue
863 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
864   \jsc@mag@xrealtrue
865 \fi\fi

```

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

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

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

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

```

866 \ifx\bxjs@param@mag\relax
867   \@tempdima=\bxjs@param@basefontsize
868   \advance\@tempdima.001pt \multiply\@tempdima25
869   \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
870   \edef\bxjs@param@mag{\the\@tempcmta}
871 \else
872 % mag 値が直接指定された場合
873   \bxjs@gset@tempcmta{\bxjs@param@mag}
874   \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
875 % 有効な mag 値の範囲は 1--32768
876   \edef\bxjs@param@mag{\the\@tempcmta}
877   \advance\@tempcmta100000
878   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
879   \expandafter\bxjs@tmpa\the\@tempcmta\@nil
880   \edef\bxjs@param@basefontsize{\the\@tempdima}
881 \fi
882 \@tempcmta\bxjs@param@mag \advance\@tempcmta100000
883 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
884 \expandafter\bxjs@tmpa\the\@tempcmta\@nil
885 \edef\jsc@magscale{\strip@pt\@tempdima}
886 \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 \p@ 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。

```

887 \newdimen\jsc@mpt
888 \newdimen\jsc@mmm
889 \ifjsc@mag
890   \jsc@mpt=1\p@
891   \jsc@mmm=1mm
892   \catcode`\?=9 % \p@? read as \p@
893 \else
894   \jsc@mpt=\jsc@magscale\p@
895   \jsc@mmm=\jsc@magscale mm
896   \catcode`\?=11 \let\p@?\jsc@mpt
897 \fi
898 \chardef\bxjs@qmc=\catcode`\?\relax
899 \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。

```

900 \newdimen\jsZw
901 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
902 \ifbxjs@usezw
903   \providecommand*\zw{\jsZw}
904 \fi

```

`\zwspace` 全角幅の水平空き。

```

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

```

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

```

906 \ifjsc@mag@xreal
907   \RequirePackage{type1cm}
908   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

909 \ifbxjs@TUenc
910   \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
911 \else
912   \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
913 \fi
914 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
915 \let\jsc@get@external@font\get@external@font
916 \def\get@external@font{%
917   \jsc@preadjust@extract@font
918   \jsc@get@external@font}
919 \def\jsc@fstrunc#1{%
920   \edef\jsc@tmpa{\strip@pt#1}%
921   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}

```

```

922 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
923   \if#5*\else
924     \edef\jsc@tmpa{#1%
925       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
926     \fi}
927 \def\jsc@preadjust@extract@font{%
928   \let\jsc@req@size\fontsize
929   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
930   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
931   \let\jsc@ref@size\jsc@tmpa
932   \let\fontsize\jsc@ref@size}
933 \def\execute@size@function#1{%
934   \let\jsc@cref@size\fontsize
935   \let\fontsize\jsc@req@size
936   \csname s@fct@#1\endcsname}
937 \let\jsc@DeclareErrorFont\DeclareErrorFont
938 \def\DeclareErrorFont#1#2#3#4#5{%
939   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
940   \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
941   \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
942 \def\gen@sfcnt{%
943   \def\mandatory@arg{\mandatory@arg\jsc@cref@size}%
944   \empty@sfcnt}
945 \def\genb@sfcnt{%
946   \edef\mandatory@arg{%
947     \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@{}}%
948   \empty@sfcnt}
949 \ifbxjs@TUenc\else
950   \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
951 \fi
952 \fi

```

[2016-11-16] latex.ltx (ltspace.dtx) で定義されている `\smallskip` の、単位 `pt` を `\jsc@mpt` に置き換えた `\jsc@smallskip` を定義します。これは `\maketitle` で用いられます。`\jsc@medskip` と `\jsc@bigskip` は必要ないのでコメントアウトしています。

```

\jsc@smallskip
\jsc@medskip 953 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 954 %\def\jsc@medskip{\vspace\jsc@medskipamount}
955 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 956 \newskip\jsc@smallskipamount
957 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
\jsc@bigskipamount 958 %\newskip\jsc@medskipamount
959 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
960 %\newskip\jsc@bigskipamount
961 %\jsc@bigskipamount =12\jsc@mpt plus 4\jsc@mpt minus 4\jsc@mpt

```

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

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

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

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

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

```
962 % \ifpapersize
963 %   \setlength{\@tempdima}{\paperwidth}
964 %   \setlength{\@tempdimb}{\paperheight}
965 %   \iftombow
966 %     \advance \@tempdima 2truein
967 %     \advance \@tempdimb 2truein
968 %   \fi
969 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
970 % \fi
```

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

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

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

BXJS クラスでの変更点：

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

```
971 \chardef\bxjs@periodchar=`\
972 \bxjs@protected\def\bxjs@SE{%
973   \ifnum\spacefactor<\@m \spacefactor\@m
974   \else \spacefactor\sfcode\bxjs@periodchar
975   \fi}
976 \ifbxjs@fix@at@cmd
977   \def\@{\bxjs@SE{}}
```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

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

```
979 \def\bxjs@tmpa{\def\setfontsize##1##2##3}
980 \expandafter\bxjs@tmpa\expandafter{%
981   \setfontsize{#1}{#2}{#3}%
982 % 末尾にコードを追加
983   \expandafter\def\expandafter\size@update\expandafter{%
984     \size@update
985     \jsFontSizeChanged}%
```


986 }

`\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。

```
987 \newcommand*\jsFontSizeChanged{%
988   \jsZw=\f@size\p@
989   \jsZw=\jsScale \jsZw
990   \ifdim\parindent>\z@
991     \if@english \parindent=1em
992     \else       \parindent=1\jsZw
993   \fi
994   \fi\relax
995   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
996 \newcommand*\jsResetDimen{}
```

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

```
997 \ifjsc@mag
998   \let\jsc@setfontsize\@setfontsize
999 \else
1000   \def\jsc@setfontsize#1#2#3{%
1001     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
1002 % microtype 対策
1003 \ifjsWithTeX\if j\jsEngine\else
1004   \def\jsc@setfontsize#1#2#3{%
1005     \edef\bxjs@sfs@next{%
1006       \unexpanded{\@setfontsize#1}%
1007       {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
1008     }\bxjs@sfs@next}
1009   \fi\fi
1010 \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。

```
1011 \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。

```
1012 \newif\ifnarrowbaselines
1013 \if@english
1014   \narrowbaselinestrue
1015 \fi
1016 \def\narrowbaselines{%
1017   \narrowbaselinestrue
1018   \skip0=\abovedisplayskip
1019   \skip2=\abovedisplayshortskip
1020   \skip4=\belowdisplayskip
1021   \skip6=\belowdisplayshortskip
1022 % 一時的に警告を無効化する
1023   \let\bxjs@save@nomath\@nomath
1024   \let\@nomath\@gobble
1025   \@currsize\selectfont
1026   \let\@nomath\bxjs@save@nomath
1027   \abovedisplayskip=\skip0
1028   \abovedisplayshortskip=\skip2
1029   \belowdisplayskip=\skip4
1030   \belowdisplayshortskip=\skip6\relax}
1031 \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。

```
1032 \def\bxjs@ifnarrowbaselines{%
1033   \ifnarrowbaselines\expandafter\@firstoftwo
1034   \else \expandafter\@secondoftwo
1035   \fi
1036 }
```

`\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。

```

1037 \renewcommand{\normalsize}{%
1038   \bxjs@if@narrowbaselines{%
1039     \jsc@setfontsize\normalsize\@xpt\@xiipt
1040   }{%else
1041     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1042   }%

```

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。

```

1043 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1044 \abovedisplayshortskip \z@ \@plus3\p@?
1045 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1046 \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。

```

1047 \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。

```

1048 \normalsize

```

\Cht 基準となる長さの設定をします。pT_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。

```

1049 \ifx\Cht\@undefined \newdimen\Cht \fi
1050 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1051 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1052 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1053 \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。

```

1054 \setlength\Cht{0.88\jsZw}

```

```

1055 \setlength\Cdp{0.12\jsZw}
1056 \setlength\Cwd{1\jsZw}
1057 \setlength\Cvs{\baselineskip}
1058 \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。

```

1059 \newcommand{\small}{%
1060   \bxjs@if@narrowbaselines{%
1061     <!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1062     <kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1063   }{%else
1064     <!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1065     <kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1066   }%
1067   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1068   \abovedisplayshortskip \z@ \@plus3\p@?
1069   \belowdisplayskip \abovedisplayskip
1070   \belowdisplayshortskip \belowdisplayskip
1071   \def\@listi{\leftmargin\leftmargini
1072             \topsep \z@
1073             \parsep \z@
1074             \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。

```

1075 \newcommand{\footnotesize}{%
1076   \bxjs@if@narrowbaselines{%
1077     <!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1078     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1079   }{%else
1080     <!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1081     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1082   }%
1083   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1084   \abovedisplayshortskip \z@ \@plus2\p@?
1085   \belowdisplayskip \abovedisplayskip
1086   \belowdisplayshortskip \belowdisplayskip
1087   \def\@listi{\leftmargin\leftmargini
1088             \topsep \z@
1089             \parsep \z@
1090             \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、
`\large` 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
`\Large` 行が揃うようにします。

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1091 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1092 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 1093 \if@twocolumn
\LARGE 1094 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
1095 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1096 \else
1097 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
1098 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1099 \fi
1100 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1101 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1102 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1103 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1104 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1105 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

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

```
1106 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}
```

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

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

```

1107 % \newcommand{\headfont}{\bfseries}
1108 \newcommand{\headfont}{\sffamily}
1109 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

5 レイアウト

■二段組

`\columnsep` `\columnsep` は二段組のときの左右の段間の幅です。元は 10pt でしたが、2zw にしました。
`\columnseprule`

このスペースの中央に `\columnseprule` の幅の罫線が引かれます。

```
1110 %<!kiyou>\setlength\columnsep{2\Cwd}
1111 %<kiyou>\setlength\columnsep{28truebp}
1112 \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 でしたが 1pt に変更しました。normal... の付いた方は保存用です。

```
\lineskiplimit 1113 \setlength\lineskip{1\jsc@empt}
\normallineskiplimit 1114 \setlength\normallineskip{1\jsc@empt}
1115 \setlength\lineskiplimit{1\jsc@empt}
1116 \setlength\normallineskiplimit{1\jsc@empt}
```

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

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

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

```
1117 \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。

```
1118 \setlength\parskip{\z@}
1119 \if@slide
1120 \setlength\parindent{0\p@}
1121 \else
1122 \setlength\parindent{1\Cwd}
1123 \fi
```

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

```
\@highpenalty 1124 \@lowpenalty 51
1125 \@medpenalty 151
1126 \@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。

```
1127 % \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。

```
1128 % \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。

```
1129 \onlypreamble\bxjs@bd@pre@geometry@hook
1130 \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。

```
1131 \ifjsc@mag
1132 \mag=\bxjs@param@mag
1133 \fi
1134 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLLength` のための和文単位の定義。

```
1135 \def\bxjs@unit@trueQ{0.25truemm}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1136 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw
```

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

```
1137 \def\bxjs@tmpdo{\futurelet\bxjs@tmpa\bxjs@tmpdo@a}
1138 \def\bxjs@tmpdo@a{%
1139   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@tmpdo@b
1140   \else \expandafter\bxjs@tmpdo@c \fi}
1141 \def\bxjs@tmpdo@b#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1142 \def\bxjs@tmpdo@c#1\@nil{\bxjs@tmpdo@d#1,,\@nil}
1143 \def\bxjs@tmpdo@d#1,#2,#3\@nil{%
1144   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1145 \expandafter\bxjs@tmpdo\bxjs@param@paper\@empty\@empty\@nil
```

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

```
1146 \edef\bxjs@layout@paper{%
1147   \ifjsc@mag truedimen,\fi
1148   \if@landscape landscape,\fi
1149   \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。

```
1150 %<*article|report>
1151 \def\bxjs@layout@base{%
1152   headheight=\topskip,footskip=0.03367\paperheight,%
1153   headsep=\footskip-\topskip,includeheadfoot,%
1154 }
1155 \edef\bxjs@layout{\bxjs@layout@base
1156   hscale=0.76,hmarginratio=1:1,%
1157   vscale=0.83,vmarginratio=1:1,%
```

```

1158 }
1159 %</article|report>
1160 %<*book>
1161 \def\bxjs@layout@base{%
1162   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1163 }
1164 \ifbxjs@layout@buggyhmargin    %---
1165 % アレ
1166 \edef\bxjs@layout{\bxjs@layout@base
1167   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1168   vscale=0.83,vmarginratio=1:1,%
1169 }
1170 \else    %---
1171 % 非アレ
1172 \edef\bxjs@layout{\bxjs@layout@base
1173   hmargin=18\jsc@mmm,%
1174   vscale=0.83,vmarginratio=1:1,%
1175 }
1176 \fi    %---
1177 %</book>
1178 %<*slide>
1179 \def\bxjs@layout@base{%
1180   noheadfoot,%
1181 }
1182 \edef\bxjs@layout{\bxjs@layout@base
1183   hscale=0.9,hmarginratio=1:1,%
1184   vscale=0.944,vmarginratio=1:1,%
1185 }
1186 %</slide>

```

textwidth オプションの設定を反映する。

```

1187 %<!*book>
1188 \ifx\bxjs@textwidth@opt\@undefined\else
1189   \jsSetQHLengh\@tempdima{\bxjs@textwidth@opt}
1190   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1191 \fi
1192 %</!book>
1193 \ifx\bxjs@number@of@lines@opt\@undefined\else
1194   \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1195   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1196 \fi

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1197 \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。

```
1198 %<*book>
```



```

1199 \newcommand\jsTextWidthLimit{40}
1200 \@tempdima=\jsTextWidthLimit\Cwd
1201 \ifx\bxjs@textwidth@limit@opt\@undefined\else
1202   \bxjs@gset@tempcnta{\bxjs@textwidth@limit@opt}
1203   \@tempdima=\@tempcnta\Cwd
1204 \fi
1205 \ifx\bxjs@textwidth@opt\@undefined\else
1206   \jsSetQHLengh\@tempdima{\bxjs@textwidth@opt}
1207 \fi
1208 \edef\bxjs@textwidth@limit{\the\@tempdima}
1209 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1210   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1211   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1212 \fi
1213 %</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。

```

1214 \def\bxjs@preproc@layout{%
1215   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mpt}

```

\bxjs@postproc@layout geometry の後処理。

```

1216 \def\bxjs@postproc@layout{%
    geometry のドライバを再設定する。
1217   \ifx\bxjs@geometry@driver\relax\else
1218     \let\Gm@driver\bxjs@geometry@driver
1219   \fi
    \ht\strutbox の値を元に戻す。
1220   \ht\strutbox=\bxjs@save@ht@strutbox\relax
    \textwidth の値を補正する。
1221   \ifbxjs@whole@zw@lines
1222     \@tempdimb=\textwidth
1223     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1224     \advance\textwidth.005pt\relax
1225     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1226     \advance\@tempdimb-\textwidth
1227     \advance\oddsidemargin 0.5\@tempdimb
1228     \advance\evensidemargin 0.5\@tempdimb
1229   \fi
1230   \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。

```

1231 %<*book>

```

```

1232 \@tempdima=\bxjs@textwidth@limit\relax
1233 \ifbxjs@whole@zw@lines
1234 \advance\@tempdima.005pt\relax
1235 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1236 \fi
1237 \ifdim\textwidth>\@tempdima
1238 \textwidth=\@tempdima
1239 \addtolength\evensidemargin{\fullwidth-\textwidth}
1240 \fi
1241 %</book>

\textheight 関連の調整。
1242 \@tempdimb=\textheight
1243 \advance\textheight-\topskip
1244 \advance\textheight.005pt\relax
1245 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1246 \advance\textheight\topskip
1247 \advance\@tempdimb-\textheight
1248 \advance\topmargin0.5\@tempdimb

\headheight 関連の調整。
1249 \@tempdima=\topskip
1250 \advance\headheight\@tempdima
1251 \advance\topmargin-\@tempdima

marginpar 関連の調整。
1252 \setlength\marginparsep{\columnsep}
1253 \setlength\marginparpush{\baselineskip}
1254 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1255 -\textwidth-10\jsc@mmm-\marginparsep}
1256 \ifbxjs@whole@zw@lines
1257 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1258 \fi

連動する変数。
1259 \maxdepth=.5\topskip
1260 \stockwidth=\paperwidth
1261 \stockheight=\paperheight
1262 }

```

`\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。

```

1263 \edef\jsGeometryOptions{%
1264 \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージを読み込む

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

```
1265 \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。

```
1266 \ifbxjs@papersize
1267   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1268     \PassOptionsToPackage{dvipdfm}{geometry}
1269   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1270     \PassOptionsToPackage{dvipdfm}{geometry}
1271   \fi\fi
1272   \let\bxPapersizeSpecialDone=t
1273 \else
1274   \PassOptionsToPackage{driver=none}{geometry}
1275 \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。

```
1276 \AtBeginDocument{\bxjs@bd@pre@geometry@hook}
1277 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1278 \bxjs@preproc@layout
1279 \edef\bxjs@next{%
1280   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1281 }\bxjs@next
1282 \AtBeginDocument{\RevokeOldLuaTeXBehavior}
```

`\bxjs@geometry@driver` `geometry` が用いるドライバの名前。

※この値は一度決めた後は変わってほしくないので、`\bxjs@postproc@layout` において書き戻す処理を入れている。

```
1283 \let\bxjs@geometry@driver\Gm@driver
1284 \bxjs@postproc@layout
```

`geometry` のドライバ自動判別に対する前処理。

```
1285 \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。

```
1286   \@ifpackagelater{geometry}{2010/02/12}{\relax}{\else
1287     \PackageError\bxjs@clsname
1288       {Your 'geometry' package is too old (< v5.0)}%
1289       {\@ehc}%
1290     \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。

```
1291   \ifjsWithpTeXng
1292     \ifx\Gm@driver\@empty
1293       \def\Gm@driver{pdftex}%
1294     \fi
1295   \fi}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```
1296 \def\setpagelayout{%
1297   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1298     \@ifstar{\bxjs@setpagelayout@a@ne}{\bxjs@setpagelayout@a@z@}}
1299 \def\bxjs@setpagelayout@a#1#2{%
1300   \ifcase#1% modify
1301     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1302   \or% reset(*)
1303     \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1304   \or% semireset(+)
1305     \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1306   \fi
1307   \bxjs@preproc@layout
1308   \edef\bxjs@next{%
1309     \noexpand\geometry{\bxjs@next}%
1310   }\bxjs@next
1311   \bxjs@postproc@layout}
```

■ `geometry` パッケージを読み込まない

`geometry=user` の場合の処理。

```
1312 \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。

```
1313 \g@addto@macro\bxjs@begin@document@hook{%
1314   \ifdim\textwidth=.5\maxdimen
1315     \ClassError\bxjs@clsname
1316       {Page layout is not properly set}%
1317     {\@ehd}%
1318   \fi}
1319 \def\jsUseMinimalPageLayout{%
1320   \setlength{\textwidth}{6.5in}%
1321   \setlength{\textheight}{8in}}
```

`\setpagelayout` はとりあえず無効にしておく。

```
1322 \let\bxjs@geometry@driver\relax
1323 \def\setpagelayout{%
1324   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1325     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1326 \def\bxjs@pagelayout@a#1{%
1327   \ClassError\bxjs@clsname
1328     {Command '\string\setpagelayout' is not supported,\MessageBreak
1329     because 'geometry' value is not 'class'}\@eha}
```

```
1330 %
1331 \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。

```
1332 %<*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。

```
1333 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1334 \if@slide
1335   \setlength\headheight{0\jsc@mpt}
1336 \else
1337   \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
    06-26)
1338 \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。

```
1339 %<*article|kiyou>
1340 \if@slide
1341   \setlength\footskip{0pt}
1342 \else
1343   \setlength\footskip{0.03367\paperheight}
1344   \ifdim\footskip<\baselineskip
1345     \setlength\footskip{\baselineskip}
```

```

1346 \fi
1347 \fi
1348 %</article|kiyou>
1349 %<jspf>\setlength\footskip{9\jsc@mmm}
1350 %<*book>
1351 \if@report
1352 \setlength\footskip{0.03367\paperheight}
1353 \ifdim\footskip<\baselineskip
1354 \setlength\footskip{\baselineskip}
1355 \fi
1356 \else
1357 \setlength\footskip{0pt}
1358 \fi
1359 %</book>
1360 %<*report>
1361 \setlength\footskip{0.03367\paperheight}
1362 \ifdim\footskip<\baselineskip
1363 \setlength\footskip{\baselineskip}
1364 \fi
1365 %</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)。

```

1366 %<*article>
1367 \if@slide
1368 \setlength\headsep{0\jsc@empt}
1369 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1370 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1371 \else
1372 \setlength\headsep{\footskip}
1373 \addtolength\headsep{-\topskip}
1374 \fi
1375 %</article>
1376 %<*book>
1377 \if@report
1378 \setlength\headsep{\footskip}
1379 \addtolength\headsep{-\topskip}
1380 \else
1381 \setlength\headsep{6\jsc@mmm}
1382 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1383 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1384 \fi
1385 %</book>
1386 %<*report>
1387 \setlength\headsep{\footskip}

```

```

1388 \addtolength\headsep{-\topskip}
1389 %</report>
1390 %<*jspf>
1391 \setlength\headsep{9\jsc@mmm}
1392 \addtolength\headsep{-\topskip}
1393 %</jspf>
1394 %<*kiyou>
1395 \setlength\headheight{0\jsc@mpt}
1396 \setlength\headsep{0\jsc@mpt}
1397 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1398 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1399 %</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。

```

1400 \setlength\maxdepth{.5\topskip}

```

■本文の幅と高さ

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

```

1401 \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。

```

1402 %<*article>
1403 \if@slide
1404   \setlength\fullwidth{0.9\paperwidth}
1405 \else
1406   \setlength\fullwidth{0.76\paperwidth}
1407 \fi
1408 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1409 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1410 \setlength\textwidth{\fullwidth}
1411 %</article>
1412 %<*book>
1413 \if@report

```

```

1414 \setlength\fullwidth{0.76\paperwidth}
1415 \else
1416 \setlength\fullwidth{\paperwidth}
1417 \addtolength\fullwidth{-36\jsc@mmm}
1418 \fi
1419 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1420 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1421 \setlength\textwidth{\fullwidth}
1422 \if@report \else
1423 \if@twocolumn \else
1424 \ifdim \fullwidth>40zw
1425 \setlength\textwidth{40zw}
1426 \fi
1427 \fi
1428 \fi
1429 %</book>
1430 %<*report>
1431 \setlength\fullwidth{0.76\paperwidth}
1432 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1433 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1434 \setlength\textwidth{\fullwidth}
1435 %</report>
1436 %<*jspf>
1437 \setlength\fullwidth{50zw}
1438 \addtolength\fullwidth{8\jsc@mmm}
1439 \setlength\textwidth{\fullwidth}
1440 %</jspf>
1441 %<*kiyou>
1442 \setlength\fullwidth{48zw}
1443 \addtolength\fullwidth{\columnsep}
1444 \setlength\textwidth{\fullwidth}
1445 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

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

[2016-10-08] article の slide のときに `\headheight` はゼロなので、さらに修正しました

(2016-08-17 での修正漏れ)。

```
1446 %<*article|book|report>
1447 \if@slide
1448   \setlength{\textheight}{0.95\paperheight}
1449 \else
1450   \setlength{\textheight}{0.83\paperheight}
1451 \fi
1452 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1453 \addtolength{\textheight}{-\headsep}
1454 \addtolength{\textheight}{-\footskip}
1455 \addtolength{\textheight}{-\topskip}
1456 \divide\textheight\baselineskip
1457 \multiply\textheight\baselineskip
1458 %</article|book|report>
1459 %<jspf>\setlength{\textheight}{51\baselineskip}
1460 %<kiyou>\setlength{\textheight}{47\baselineskip}
1461 \addtolength{\textheight}{\topskip}
1462 \addtolength{\textheight}{0.1\jsc@mpt}
1463 %<jspf>\setlength{\mathindent}{10\jsc@mmm}
```

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

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

ですが、次のようにします。

```
1464 \def\flushbottom{%
1465   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1466   \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
`\marginparpush` みどうしの最小の間隔です。

```
1467 \setlength\marginparsep{\columnsep}
1468 \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。

```
1469 \setlength{\oddsidemargin}{\paperwidth}
1470 \addtolength{\oddsidemargin}{-\fullwidth}
1471 \setlength{\oddsidemargin}{.5\oddsidemargin}
1472 \iftombow
1473   \addtolength{\oddsidemargin}{-1in}
1474 \else
```

```

1475 \addtolength{\oddsidemargin}{-\inv@mag in}
1476 \fi
1477 \setlength{\evensidemargin}{\oddsidemargin}
1478 \if@mparswitch
1479 \addtolength{\evensidemargin}{\fullwidth}
1480 \addtolength{\evensidemargin}{-\textwidth}
1481 \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。

```

1482 \setlength\marginparwidth{\paperwidth}
1483 \addtolength\marginparwidth{-\oddsidemargin}
1484 \addtolength\marginparwidth{-\inv@mag in}
1485 \addtolength\marginparwidth{-\textwidth}
1486 \addtolength\marginparwidth{-10\jsc@mmm}
1487 \addtolength\marginparwidth{-\marginparsep}
1488 \@tempdima=1zw
1489 \divide\marginparwidth\@tempdima
1490 \multiply\marginparwidth\@tempdima

```

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

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

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

```

1491 \setlength\topmargin{\paperheight}
1492 \addtolength\topmargin{-\textheight}
1493 \if@slide
1494 \addtolength\topmargin{-\headheight}
1495 \else
1496 \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1497 \fi
1498 \addtolength\topmargin{-\headsep}
1499 \addtolength\topmargin{-\footskip}
1500 \setlength\topmargin{0.5\topmargin}
1501 %<kiyou>\setlength\topmargin{81truebp}
1502 \iftombow
1503 \addtolength\topmargin{-1in}
1504 \else
1505 \addtolength\topmargin{-\inv@mag in}
1506 \fi
1507 %</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。

```
1508 \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。

```
1509 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

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

[2003-08-23] ちょっと増やしました。

```
1510 \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。

```
1511 \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。

```
1512 \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。

```
1513 \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。

```
1514 \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。

```
1515 \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。

```
1516 \renewcommand{\floatpagefraction}{.8}
```

`\cdbltopnumber` 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。

```
1517 \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。

```
1518 \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。

```
1519 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1520 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1521 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1522 \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 1523 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1524 \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 1525 \setlength\@fptop{0\p@? \@plus 1fil}
1526 \setlength\@fpsep{8\p@? \@plus 2fil}
1527 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
\@dblfpsep 1528 \setlength\@dblfpsep{0\p@? \@plus 1fil}
1529 \setlength\@dblfpsep{8\p@? \@plus 2fil}
\@dblfpbot 1530 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

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

`\pltx@cleartoleftpage`

`\pltx@cleartooddpage`

`\pltx@cleartoevenpage`

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

となっています。

```
1531 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1532 % \ifodd\c@page
1533 % \iftdir
1534 % \hbox{}\thispagestyle{empty}\newpage
1535 % \if@twocolumn\hbox{}\newpage\fi
1536 % \fi
1537 % \else
1538 % \ifydir
1539 % \hbox{}\thispagestyle{empty}\newpage
1540 % \if@twocolumn\hbox{}\newpage\fi
1541 % \fi
1542 % \fi\fi}
1543 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1544 % \ifodd\c@page
1545 % \ifydir
1546 % \hbox{}\thispagestyle{empty}\newpage
1547 % \if@twocolumn\hbox{}\newpage\fi
1548 % \fi
1549 % \else
1550 % \iftdir
1551 % \hbox{}\thispagestyle{empty}\newpage
1552 % \if@twocolumn\hbox{}\newpage\fi
1553 % \fi
1554 % \fi\fi}
1555 \def\pltx@cleartooddpage{\clearpage\if@twoside
1556 \ifodd\c@page\else
1557 \hbox{}\thispagestyle{empty}\newpage
1558 \if@twocolumn\hbox{}\newpage\fi
1559 \fi\fi}
1560 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1561 \ifodd\c@page
1562 \hbox{}\thispagestyle{empty}\newpage
1563 \if@twocolumn\hbox{}\newpage\fi
1564 \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。

```
1565 \let\pltx@cleartorightpage\pltx@cleartooddpage
1566 \let\pltx@cleartoleftpage\pltx@cleartoevenpage
```

`\cleardoublepage` [2017-02-24] コミュニティ版 p_AT_EX 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。

```
1567 %<*book|report>
1568 \if@openleft
1569 \let\cleardoublepage\pltx@cleartoleftpage
1570 \else\if@openright
```

```

1571 \let\cleardoublepage\pltx@cleartorightpage
1572 \fi\fi
1573 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

```

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

```

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

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

`\ps@empty` `empty` ページスタイルの定義です。 \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。

```

1574 % \def\ps@empty{%
1575 %   \let\mkboth\gobbletwo
1576 %   \let\oddhead\empty
1577 %   \let\oddfoot\empty
1578 %   \let\evenhead\empty
1579 %   \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。

```

1580 \def\ps@plainfoot{%

```

```

1581 \let\@mkboth\@gobbletwo
1582 \let\@oddhead\@empty
1583 \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1584 \let\@evenhead\@empty
1585 \let\@evenfoot\@oddfoot}
1586 \def\ps@plainhead{%
1587 \let\@mkboth\@gobbletwo
1588 \let\@oddfoot\@empty
1589 \let\@evenfoot\@empty
1590 \def\@evenhead{%
1591 \if@mparswitch \hss \fi
1592 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1593 \if@mparswitch\else \hss \fi}%
1594 \def\@oddhead{%
1595 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1596 %<book>\let\ps@plain\ps@plainhead
1597 %<!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。

```

1598 %<*article|slide>
1599 \if@twoside
1600 \def\ps@headings{%
1601 \let\@oddfoot\@empty
1602 \let\@evenfoot\@empty
1603 \def\@evenhead{\if@mparswitch \hss \fi
1604 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1605 \if@mparswitch\else \hss \fi}%
1606 \def\@oddhead{%
1607 \underline{%
1608 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1609 \let\@mkboth\markboth
1610 \def\sectionmark##1{\markboth{%
1611 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1612 ##1}{}}%
1613 \def\subsectionmark##1{\markright{%
1614 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1615 ##1}}%
1616 }
1617 \else % if not twoside
1618 \def\ps@headings{%
1619 \let\@oddfoot\@empty
1620 \def\@oddhead{%
1621 \underline{%
1622 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1623 \let\@mkboth\markboth
1624 \def\sectionmark##1{\markright{%

```

```

1625 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1626 ##1}}}}
1627 \fi
1628 %</article|slide>

```

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

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

```

1629 %<*book|report>
1630 \def\bxjs@maybe@autoxspacing{%
1631 \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1632 \newif\if@omit@number
1633 \def\ps@headings{%
1634 \let\@oddfoot\@empty
1635 \let\@evenfoot\@empty
1636 \def\@evenhead{%
1637 \if@mparswitch \hss \fi
1638 \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1639 \textbf{\thepage}\hfil\leftmark}}}%
1640 \if@mparswitch\else \hss \fi}%
1641 \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1642 {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1643 \let\@mkboth\markboth
1644 \def\chaptermark##1{\markboth{%
1645 \ifnum \c@secnumdepth >\m@ne
1646 \if@mainmatter
1647 \if@omit@number\else
1648 \@chapapp\thechapter\@chappos\hskip1\jsZw
1649 \fi
1650 \fi
1651 \fi
1652 ##1-}}}%
1653 \def\sectionmark##1{\markright{%
1654 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1655 ##1}}}%
1656 %</book|report>

```

最後は学会誌の場合です。

```

1657 %<*jspf>
1658 \def\ps@headings{%
1659 \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1660 \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1661 \def\@oddhead{\normalfont\hfil \@title \hfil}
1662 \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1663 %</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。

```
1664 \def\ps@myheadings{%
1665   \let\@oddfoot\@empty\let\@evenfoot\@empty
1666   \def\@evenhead{%
1667     \if@mparswitch \hss \fi%
1668     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1669     \if@mparswitch\else \hss \fi}%
1670   \def\@oddhead{%
1671     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1672   \let\mkboth\@gobbletwo
1673 %<book|report> \let\chaptermark\@gobble
1674   \let\sectionmark\@gobble
1675 %<!book&!report> \let\subsectionmark\@gobble
1676 }
```

8 文書のマークアップ

8.1 表題

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

```
\date 1677 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1678 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1679 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1680 % \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。

```
1681 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1682 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1683 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1684 \AtBeginDocument{\bxjs@decl@subtitle}
1685 \def\bxjs@decl@subtitle{%
1686   \global\let\bxjs@decl@subtitle\relax
1687   \ifx\subtitle\@undefined
1688     \global\let\subtitle\jsSubtitle
1689   \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。

```
1690 \def\bxjs@annihilate@subtitle{%
1691   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1692   \global\let\jsSubtitle\relax}
```

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

```
\eauthor 1693 %<*jspf>
1694 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
\keywords 1695 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
1696 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1697 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1698 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1699 %</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。

```
1700 \def\plainifnotempty{%
1701   \ifx \@oddhead \@empty
1702     \ifx \@oddfoot \@empty
1703       \else
1704         \thispagestyle{plainfoot}%
1705       \fi
1706     \else
1707       \thispagestyle{plainhead}%
1708     \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。

```
1709 %<*article|book|report|slide>
1710 \if@titlepage
1711   \newcommand*{\maketitle}{%
1712     \begin{titlepage}%
1713       \let\footnotesize\small
1714       \let\footnoterule\relax
1715       \let\footnote\thanks
1716       \null\vfil
1717       \if@slide
1718         {\footnotesize \@date}%
```

```

1719     \begin{center}
1720         \mbox{} \\\[1\jsZw]
1721         \large
1722         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1723         \jsc@smallskip
1724         \@title
1725         \ifx\bxjs@subtitle\@undefined\else
1726             \par\vskip\z@
1727             {\small \bxjs@subtitle\par}
1728         \fi
1729         \jsc@smallskip
1730         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1731         \vfill
1732         {\small \@author}%
1733     \end{center}
1734 \else
1735     \vskip 60\p@?
1736     \begin{center}%
1737         {\LARGE \@title \par}%
1738         \ifx\bxjs@subtitle\@undefined\else
1739             \vskip5\p@?
1740             {\normalsize \bxjs@subtitle\par}
1741         \fi
1742         \vskip 3em%
1743         {\large
1744             \lineskip .75em
1745             \begin{tabular}[t]{c}%
1746                 \@author
1747             \end{tabular}\par}%
1748         \vskip 1.5em
1749         {\large \@date \par}%
1750     \end{center}%
1751     \fi
1752     \par
1753     \@thanks\vfil\null
1754 \end{titlepage}%
1755 \setcounter{footnote}{0}%
1756 \global\let\thanks\relax
1757 \global\let\maketitle\relax
1758 \global\let\@thanks\@empty
1759 \global\let\@author\@empty
1760 \global\let\@date\@empty
1761 \global\let\@title\@empty
1762 \global\let\title\relax
1763 \global\let\author\relax
1764 \global\let\date\relax
1765 \global\let\and\relax
1766 \bxjs@annihilate@subtitle
1767 }%

```

```

1768 \else
1769   \newcommand{\maketitle}{\par
1770     \begin{group}
1771       \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1772       \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1773       \long\def\@makefnmark##1{\advance\leftskip 3\jsZw
1774         \parindent 1\jsZw\noindent
1775         \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1776       \if@twocolumn
1777         \ifnum \col@number=\@one
1778           \@maketitle
1779         \else
1780           \twocolumn[\@maketitle]%
1781         \fi
1782       \else
1783         \newpage
1784         \global\@topnum\z@ % Prevents figures from going at top of page.
1785         \@maketitle
1786       \fi
1787       \plainifnotempty
1788       \@thanks
1789     \end{group}
1790     \setcounter{footnote}{0}%
1791     \global\let\thanks\relax
1792     \global\let\maketitle\relax
1793     \global\let\@thanks\@empty
1794     \global\let\@author\@empty
1795     \global\let\@date\@empty
1796     \global\let\@title\@empty
1797     \global\let\title\relax
1798     \global\let\author\relax
1799     \global\let\date\relax
1800     \global\let\and\relax
1801     \bxjs@annihilate@subtitle
1802   }

```

`\@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。

```

1803 \def\@maketitle{%
1804   \newpage\null
1805   \vskip 2em
1806   \begin{center}%
1807     \let\footnote\thanks
1808     {\LARGE \@title \par}%
1809     \ifx\bxjs@subtitle\@undefined\else
1810       \vskip3\p@?
1811       {\normalsize \bxjs@subtitle\par}
1812     \fi
1813     \vskip 1.5em
1814     {\large

```

```

1815         \lineskip .5em
1816         \begin{tabular}[t]{c}%
1817             \@author
1818         \end{tabular}\par}%
1819     \vskip 1em
1820     {\large \@date}%
1821 \end{center}%
1822 \par\vskip 1.5em
1823 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1824 }
1825 \fi
1826 %</article|book|report|slide>
1827 %<*jspf>
1828 \newcommand\maketitle{\par
1829     \begingroup
1830         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1831         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1832         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1833             \parindent 1\jsZw\noindent
1834             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1835         \twocolumn[\@maketitle]%
1836         \plainifnotempty
1837         \@thanks
1838     \endgroup
1839     \setcounter{footnote}{0}%
1840     \global\let\thanks\relax
1841     \global\let\maketitle\relax
1842     \global\let\@thanks\@empty
1843     \global\let\@author\@empty
1844     \global\let\@date\@empty
1845     % \global\let\@title\@empty % \@title は柱に使う
1846     \global\let\title\relax
1847     \global\let\author\relax
1848     \global\let\date\relax
1849     \global\let\and\relax
1850     \ifx\authors@mail\@undefined\else{%
1851         \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1852         \footnotetext[0]{\itshape\authors@mail}%
1853     }\fi
1854     \global\let\authors@mail\@undefined}
1855 \def\@maketitle{%
1856     \newpage\null
1857     \vskip 6em % used to be 2em
1858     \begin{center}
1859         \let\footnote\thanks
1860         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1861         \lineskip .5em
1862         \ifx\@author\@undefined\else
1863             \vskip 1em

```

```

1864 \begin{tabular}[t]{c}%
1865 \@author
1866 \end{tabular}\par
1867 \fi
1868 \ifx\@etitle\@undefined\else
1869 \vskip 1em
1870 {\large \@etitle \par}%
1871 \fi
1872 \ifx\@eauthor\@undefined\else
1873 \vskip 1em
1874 \begin{tabular}[t]{c}%
1875 \@eauthor
1876 \end{tabular}\par
1877 \fi
1878 \vskip 1em
1879 \@date
1880 \end{center}
1881 \vskip 1.5em
1882 \centerline{\box\@abstractbox}
1883 \ifx\@keywords\@undefined\else
1884 \vskip 1.5em
1885 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\small\@keywords}}
1886 \fi
1887 \vskip 1.5em}
1888 %</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。

```

1889 \def\bxjs@label@sect#1{%
1890 \expandafter\ifx\csname label#1\endcsname\relax
1891 \csname the#1\endcsname
1892 \else \csname label#1\endcsname
1893 \fi}
1894 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1895 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1896 \def\@secapp{\presectionname}
1897 \def\@secpos{\postsectionname}
1898 \fi

```

`\labelsection` 節番号の出力書式。

```
1899 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1900 \def\labelsection{\@secapp\thesection\@secpos}
1901 \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。

```
1902 \def\@startsection#1#2#3#4#5#6{%
1903   \if@noskipsec \leavevmode \fi
1904   \par
1905 % 見出し上の空きを \@tempskipa にセットする
1906   \@tempskipa #4\relax
1907 % \@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
1908   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1909 % 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
1910   \ifdim \@tempskipa <\z@
1911     \@tempskipa -\@tempskipa \@afterindentfalse
1912   \fi
1913   \if@nobreak
1914 %   \everypar{\everyparhook}% これは間違い
1915   \everypar{}\%
```

```

1916 \else
1917 \addpenalty\@secpenalty
1918 % 次の行は削除
1919 % \addvspace\@tempskipa
1920 % 次の \noindent まで追加
1921 \ifdim \@tempskipa >\z@
1922 \if@slide\else
1923 \null
1924 \vspace*{-\baselineskip}%
1925 \fi
1926 \vskip\@tempskipa
1927 \fi
1928 \fi
1929 \noindent
1930 % 追加終わり
1931 \@ifstar
1932 {\@ssect{#3}{#4}{#5}{#6}}%
1933 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

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

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

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

```

1934 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1935 \let\bxjs@if@ceph\@firstofone
1936 \else \let\bxjs@if@ceph\@gobble
1937 \fi

```

```

1938 \def\@sect#1#2#3#4#5#6[#7]#8{%
1939 \ifnum #2>\c@secnumdepth
1940 \let\@svsec\@empty
1941 \else
1942 \refstepcounter{#1}%
1943 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1944 \fi
1945 % 見出し後の空きを \@tempskipa にセット
1946 \@tempskipa #5\relax
1947 % 条件判断の順序を入れ替えました
1948 \ifdim \@tempskipa<\z@
1949 \def\@svsechd{%
1950 #6{\hskip #3\relax
1951 \@svsec #8}%
1952 \csname #1mark\endcsname{#7}}%
1953 \addcontentsline{toc}{#1}{%
1954 \ifnum #2>\c@secnumdepth \else
1955 \protect\numberline{\bxjs@label@sect{#1}}%

```



```

1956         \fi
1957         #7}}% 目次にフルネームを載せるなら #8
1958     \else
1959         \begingroup
1960         \interlinepenalty \@M % 下から移動
1961         #6{%
1962             \@hangfrom{\hskip #3\relax\@svsec}%
1963 %         \interlinepenalty \@M % 上に移動
1964             #8\@@par}%
1965         \endgroup
1966         \csname #1mark\endcsname{#7}%
1967         \addcontentsline{toc}{#1}{%
1968             \ifnum #2>\c@secnumdepth \else
1969                 \protect\numberline{\bxjs@label@sect{#1}}%
1970             \fi
1971             #7}}% 目次にフルネームを載せるならここは #8
1972     \fi
1973     \@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。

```

1974 \def\@xsect#1{%
1975 % 見出しの後ろの空きを \@tempskipa にセット
1976     \@tempskipa #1\relax
1977 % 条件判断の順序を変えました
1978     \ifdim \@tempskipa<\z@
1979         \@nbreakfalse
1980         \global\@noskipsecttrue
1981         \everypar{%
1982             \if@noskipsec
1983                 \global\@noskipsecfalse
1984                 {\setbox\z@\lastbox}%
1985                 \clubpenalty\@M
1986                 \begingroup \@svsechd \endgroup
1987                 \unskip
1988                 \@tempskipa #1\relax
1989                 \hskip -\@tempskipa
1990             \else
1991                 \clubpenalty \@clubpenalty
1992                 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1993                 \fi\bxjs@if@ceph\everyparhook}%
1994     \else
1995         \par \nbreak
1996         \vskip \@tempskipa
1997         \@afterheading
1998     \fi

```

```

1999 \if@slide
2000   {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
2001     \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
2002     \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
2003 \fi
2004 \par % 2000-12-18
2005 \ignorespaces}
2006 \def\@ssect#1#2#3#4#5{%
2007   \@tempskipa #3\relax
2008   \ifdim \@tempskipa<\z@
2009     \def\@svsechd{#4{\hskip #1\relax #5}}%
2010   \else
2011     \begingroup
2012       #4{%
2013         \@hangfrom{\hskip #1}%
2014         \interlinepenalty \@M #5\@@par}%
2015     \endgroup
2016 \fi
2017 \@xsect{#3}}

```

■柱関係の命令

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

```

\subsectionmark 2018 \newcommand*\chaptermark[1]{}
\subsubsectionmark 2019 % \newcommand*\sectionmark[1]{}
2020 % \newcommand*\subsectionmark[1]{}
\paragraphmark 2021 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 2022 % \newcommand*\paragraphmark[1]{}
2023 % \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決めるカウンタです。

```

2024 %<!book&!report>\setcounter{secnumdepth}{3}
2025 %<book|report>\setcounter{secnumdepth}{2}

```

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

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

```

\c@subsection 2026 \newcounter{part}
2027 %<book|report>\newcounter{chapter}
\c@subsubsection 2028 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2029 %<!book&!report>\newcounter{section}
\c@subparagraph 2030 \newcounter{subsection}[section]
2031 \newcounter{subsubsection}[subsection]
2032 \newcounter{paragraph}[subsubsection]
2033 \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`

`\thesection`

`\thesubsection`

`\thesubsubsection`

`\theparagraph`

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>\arabic{COUNTER}</code>	1, 2, 3, ...
<code>\roman{COUNTER}</code>	i, ii, iii, ...
<code>\Roman{COUNTER}</code>	I, II, III, ...
<code>\alph{COUNTER}</code>	a, b, c, ...
<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。

```
2034 \renewcommand{\thepart}{\@Roman\c@part}
2035 %<*!book&!report>
2036 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2037 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2038 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2039 \else
2040 \renewcommand{\thesection}{\@arabic\c@section}
2041 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2042 \fi
2043 %</!book&!report>
2044 %<*book|report>
2045 \renewcommand{\thechapter}{\@arabic\c@chapter}
2046 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2047 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2048 %</book|report>
2049 \renewcommand{\thesubsubsection}{\%
2050   \thesubsection.\@arabic\c@subsubsection}
2051 \renewcommand{\theparagraph}{\%
2052   \thesubsubsection.\@arabic\c@paragraph}
2053 \renewcommand{\thesubparagraph}{\%
2054   \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。

```
2055 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2056 %<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した。 `openany` かどうかにかかわらず奇数ページまで繰るように

修正することで、問題を解消しました。実は、 \LaTeX の標準クラスでは 1998 年に修正されていた問題です (コミュニティ版 $\text{p}\text{\LaTeX}$ の標準クラス 2017/03/05 も参照)。

```
2057 %<*book|report>
2058 \newcommand\frontmatter{%
2059   \pltx@cleartooddpage
2060   \@mainmatterfalse
2061   \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。

```
2062 \newcommand\mainmatter{%
2063   \pltx@cleartooddpage
2064   \@mainmattertrue
2065   \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。

```
2066 \newcommand\backmatter{%
2067   \if@openleft
2068     \cleardoublepage
2069   \else\if@openright
2070     \cleardoublepage
2071   \else
2072     \clearpage
2073   \fi\fi
2074   \@mainmatterfalse}
2075 %</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。

```
2076 %<*&!book&!report>
2077 \newcommand\part{%
2078   \if@noskipsec \leavevmode \fi
2079   \par
2080   \addvspace{4ex}%
2081   \if@english \@afterindentfalse \else \@afterindenttrue \fi
```

```

2082 \secdef\@part\@spart}
2083 %</*!book&!report>

book および report クラスの場合は、少し複雑です。

2084 %<*book|report>
2085 \newcommand\part{%
2086 \if@openleft
2087 \cleardoublepage
2088 \else\if@openright
2089 \cleardoublepage
2090 \else
2091 \clearpage
2092 \fi\fi
2093 \thispagestyle{empty}% 欧文用標準スタイルでは plain
2094 \if@twocolumn
2095 \onecolumn
2096 \@restonecoltrue
2097 \else
2098 \@restonecolfalse
2099 \fi
2100 \null\vfil
2101 \secdef\@part\@spart}
2102 %</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。

```

2103 %<*!book&!report>
2104 \def\@part[#1]#2{%
2105 \ifnum \c@secnumdepth >\m@ne
2106 \refstepcounter{part}%
2107 \addcontentsline{toc}{part}{%
2108 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2109 \else
2110 \addcontentsline{toc}{part}{#1}%
2111 \fi
2112 \markboth{}{}%
2113 {\parindent\z@
2114 \raggedright
2115 \interlinepenalty \@M
2116 \normalfont
2117 \ifnum \c@secnumdepth >\m@ne
2118 \Large\headfont\prepartname\thepart\postpartname
2119 \par\nobreak
2120 \fi
2121 \huge \headfont #2%
2122 \markboth{}{}\par}%
2123 \nobreak
2124 \vskip 3ex

```

```

2125 \afterheading}
2126 %<*/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。

```

2127 %<*book|report>
2128 \def\@part[#1]#2{%
2129   \ifnum \c@secnumdepth >-2\relax
2130     \refstepcounter{part}%
2131     \addcontentsline{toc}{part}{%
2132       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2133   \else
2134     \addcontentsline{toc}{part}{#1}%
2135   \fi
2136   \markboth{}{}%
2137   {\centering
2138     \interlinepenalty \@M
2139     \normalfont
2140     \ifnum \c@secnumdepth >-2\relax
2141       \huge\headfont \prepartname\thepart\postpartname
2142       \par\vskip20\p@?
2143     \fi
2144     \Huge \headfont #2\par}%
2145   \@endpart}
2146 %</book|report>

```

\@spart 番号を付けない部です。

```

2147 %<*!book&!report>
2148 \def\@spart#1{%
2149   \parindent \z@ \raggedright
2150   \interlinepenalty \@M
2151   \normalfont
2152   \huge \headfont #1\par}%
2153   \nobreak
2154   \vskip 3ex
2155   \@afterheading}
2156 %<*/book&!report>
2157 %<*book|report>
2158 \def\@spart#1{%
2159   \centering
2160   \interlinepenalty \@M
2161   \normalfont
2162   \Huge \headfont #1\par}%
2163   \@endpart}
2164 %</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。

```
2165 %<*book|report>
2166 \def\@endpart{\vfil\newpage
2167   \if@twoside
2168     \if@openleft %% added (2017/02/24)
2169       \null\thispagestyle{empty}\newpage
2170     \else\if@openright %% added (2016/12/13)
2171       \null\thispagestyle{empty}\newpage
2172     \fi\fi %% added (2016/12/13, 2017/02/24)
2173   \fi
2174   \if@restonecol
2175     \twocolumn
2176   \fi}
2177 %</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。

```
2178 %<*book|report>
2179 \newcommand{\chapter}{%
2180   \if@openleft\cleardoublepage\else
2181     \if@openright\cleardoublepage\else\clearpage\fi\fi
2182   \plainifnotempty % 元: \thispagestyle{plain}
2183   \global\@topnum\z@
2184   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2185   \secdef
2186     {\@omit@numberfalse\@chapter}%
2187     {\@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。

```
2188 \def\@chapter[#1]#2{%
2189   \ifnum \c@secnumdepth >\m@ne
2190     \if@mainmatter
2191       \refstepcounter{chapter}%
2192       \typeout{\@chapapp\thechapter\@chappos}%
2193       \addcontentsline{toc}{chapter}%
2194         {\protect\numberline
2195 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2196         {\@chapapp\thechapter\@chappos}%
2197         #1}%
2198     \else\addcontentsline{toc}{chapter}{#1}\fi
2199   \else
2200     \addcontentsline{toc}{chapter}{#1}%
2201   \fi
2202   \chaptermark{#1}%
2203   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}}
```

```

2204 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}}%
2205 \if@twocolumn
2206   \@topnewpage[\@makechapterhead{#2}]%
2207 \else
2208   \@makechapterhead{#2}%
2209   \@afterheading
2210 \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。

```

2211 \def\@makechapterhead#1{%
2212   \vspace*{2\Cvs}% 欧文は 50pt
2213   {\parindent \z@ \raggedright \normalfont
2214     \ifnum \c@secnumdepth >\m@ne
2215       \if@mainmatter
2216         \huge\headfont \@chapapp\thechapter\@chappos
2217         \par\nobreak
2218         \vskip \Cvs % 欧文は 20pt
2219       \fi
2220     \fi
2221     \interlinepenalty\@M
2222     \Huge \headfont #1\par\nobreak
2223     \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。

```

2224 \def\@schapter#1{%
2225   \chaptermark{#1}%
2226   \if@twocolumn
2227     \@topnewpage[\@makeschapterhead{#1}]%
2228   \else
2229     \@makeschapterhead{#1}\@afterheading
2230   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2231 \def\@makeschapterhead#1{%
2232   \vspace*{2\Cvs}% 欧文は 50pt
2233   {\parindent \z@ \raggedright
2234     \normalfont
2235     \interlinepenalty\@M
2236     \Huge \headfont #1\par\nobreak
2237     \vskip 3\Cvs}} % 欧文は 40pt
2238 %</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。

```

2239 \if@twocolumn

```



```

2240 \newcommand{\section}{%
2241 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2242 \@startsection{section}{1}{\z@}%
2243 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2244 %<kiyou> {\Cvs}{0.5\Cvs}%
2245 % {\normalfont\large\headfont\@secapp}%
2246 {\normalfont\large\headfont\raggedright}}
2247 \else
2248 \newcommand{\section}{%
2249 \if@slide\clearpage\fi
2250 \@startsection{section}{1}{\z@}%
2251 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2252 {.5\Cvs \@plus.3\Cdp}% 後アキ
2253 % {\normalfont\Large\headfont\@secapp}%
2254 {\normalfont\Large\headfont\raggedright}}
2255 \fi

```

\subsection 同上です。

```

2256 \if@twocolumn
2257 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2258 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2259 {\normalfont\normalsize\headfont}}
2260 \else
2261 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2262 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2263 {.5\Cvs \@plus.3\Cdp}% 後アキ
2264 {\normalfont\large\headfont}}
2265 \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)。

```

2266 \if@twocolumn
2267 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2268 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2269 {\normalfont\normalsize\headfont}}
2270 \else
2271 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2272 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2273 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2274 {\normalfont\normalsize\headfont}}
2275 \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。

```

2276 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2277 \let\bxjs@org@paragraph@mark\jsParagraphMark
2278 \ifx\bxjs@paragraph@mark\empty
2279   \let\jsParagraphMark\empty
2280 \else\ifx\bxjs@paragraph@mark\undefined\else
2281   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2282 \fi\fi
2283 \let\jsJaChar\empty
2284 \if@twocolumn
2285   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2286     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2287     {\normalfont\normalsize\headfont}}
2288   \<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2289 \else
2290   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2291     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2292     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2293     {\normalfont\normalsize\headfont}}
2294   \<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2295 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2296 \if@twocolumn
2297   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2298     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2299     {\normalfont\normalsize\headfont}}
2300 \else
2301   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2302     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2303     {\normalfont\normalsize\headfont}}
2304 \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 ですが、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。

```
2305 \if@slide
2306   \setlength\leftmargini{1\jsZw}
2307 \else
2308   \if@twocolumn
2309     \setlength\leftmargini{2\jsZw}
2310   \else
2311     \setlength\leftmargini{3\jsZw}
2312   \fi
2313 \fi
```

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

```
\leftmarginiv 2314 \if@slide
\leftmarginv 2315   \setlength\leftmarginii {1\jsZw}
2316   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2317   \setlength\leftmarginiv {1\jsZw}
2318   \setlength\leftmarginv {1\jsZw}
2319   \setlength\leftmarginvi {1\jsZw}
2320 \else
2321   \setlength\leftmarginii {2\jsZw}
2322   \setlength\leftmarginiii{2\jsZw}
2323   \setlength\leftmarginiv {2\jsZw}
2324   \setlength\leftmarginv {1\jsZw}
2325   \setlength\leftmarginvi {1\jsZw}
2326 \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。

```
2327 \setlength \labelsep {0.5\jsZw} % .5em
2328 \setlength \labelwidth{\leftmargini}
2329 \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。

```
2330 \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 2331 \@beginparpenalty -\@lowpenalty
\@itempenalty 2332 \@endparpenalty -\@lowpenalty
2333 \@itempenalty -\@lowpenalty
```

`\@listi` `\@listi` は `\leftmargin`, `\parsep`, `\topsep`, `\itemsep` などのトップレベルの定義を
`\@listI` します。この定義は、フォントサイズコマンドによって変更されます（たとえば `\small` の
中では小さい値に設定されます）。このため、`\normalsize` がすべてのパラメータを戻せる

ように、`\@listI` で `\@listi` のコピーを保存します。元の値はかなり複雑ですが、ここでは簡素化してしまいました。特に最初と最後に行送りの半分の空きが入るようにしてあります。アスキーの標準スタイルではトップレベルの `itemize`, `enumerate` 環境でだけ最初と最後に行送りの半分の空きが入るようになっていました。

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

```
2334 \def\@listI{\leftmargin\leftmarginI
2335   \parsep \z@
2336   \topsep 0.5\baselineskip
2337   \itemsep \z@ \relax}
2338 \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）。

```
2339 \@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 2340 \def\@listii{\leftmargin\leftmarginii
\@listiv 2341   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2342   \topsep \z@
\@listv 2343   \parsep \z@
\@listvi 2344   \itemsep\parsep}
2345 \def\@listiii{\leftmargin\leftmarginiii
2346   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2347   \topsep \z@
2348   \parsep \z@
2349   \itemsep\parsep}
2350 \def\@listiv {\leftmargin\leftmarginiv
2351   \labelwidth\leftmarginiv
2352   \advance\labelwidth-\labelsep}
2353 \def\@listv {\leftmargin\leftmarginv
2354   \labelwidth\leftmarginv
2355   \advance\labelwidth-\labelsep}
2356 \def\@listvi {\leftmargin\leftmarginvi
2357   \labelwidth\leftmarginvi
2358   \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。

```
2359 \renewcommand{\theenumi}{\@arabic\c@enumi}
2360 \renewcommand{\theenumii}{\@alph\c@enumii}
2361 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2362 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

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

`\labelenumiv`

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

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

```

2363 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2364 \newcommand*\jsInJaParen}[1]{%
2365   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2366 \newcommand{\labelenumi}{\theenumi.}
2367 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2368 \newcommand{\labelenumiii}{\theenumiii.}
2369 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2370 \renewcommand{\p@enumii}{\theenumi}
2371 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2372 \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 2373 \newcommand\labelitemi{\textbullet}
\labelitemiii 2374 \newcommand\labelitemii{\normalfont\bfseries \textendash}
2375 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2376 \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。

```

2377 \newenvironment{description}{%
2378   \list{}{%
2379     \labelwidth=\leftmargin
2380     \labelsep=1\jsZw
2381     \advance \labelwidth by -\labelsep
2382     \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
(たとえば `\hspace{1\jsZw}`) を入れるのもいいと思います。

```

2383 \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。

```
2384 %<*book|report>
2385 \newenvironment{chapterabstract}{%
2386   \begin{list}{}{%
2387     \listparindent=1\jsZw
2388     \itemindent=\listparindent
2389     \rightmargin=0pt
2390     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2391 %</book|report>
```

“普通の” **abstract** 環境の定義。

```
2392 %<*article|report|slide>
2393 \newbox\@abstractbox
2394 \if@titlepage
2395   \newenvironment{abstract}{%
2396     \titlepage
2397     \null\vfil
2398     \@beginparpenalty\@lowpenalty
2399     \begin{center}%
2400       \headfont \abstractname
2401       \@endparpenalty\@M
2402     \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。

```
2403   \par}%
2404   {\par\vfil\null\endtitlepage}
2405 \else
2406   \newenvironment{abstract}{%
2407     \if@twocolumn
2408       \ifx\maketitle\relax
2409         \section*{\abstractname}%
```

```

2410 \else
2411 \global\setbox\@abstractbox\hbox\bgroup
2412 \begin{minipage}[b]{\textwidth}
2413 \small\parindent1\jsZw
2414 \begin{center}%
2415 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2416 \end{center}%
2417 \list{}{%
2418 \listparindent\parindent
2419 \itemindent \listparindent
2420 \rightmargin \leftmargin}%
2421 \item\relax
2422 \fi
2423 \else
2424 \small
2425 \begin{center}%
2426 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2427 \end{center}%
2428 \list{}{%
2429 \listparindent\parindent
2430 \itemindent \listparindent
2431 \rightmargin \leftmargin}%
2432 \item\relax
2433 \fi}{\if@twocolumn
2434 \ifx\maketitle\relax
2435 \else
2436 \endlist\end{minipage}\egroup
2437 \fi
2438 \else
2439 \endlist
2440 \fi}
2441 \fi
2442 %</article|report|slide>
2443 %<*jspf>
2444 \newbox\@abstractbox
2445 \newenvironment{abstract}{%
2446 \global\setbox\@abstractbox\hbox\bgroup
2447 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2448 \small
2449 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2450 {\end{minipage}\egroup}
2451 %</jspf>

```

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

```

2452 %<*book|report>
2453 \ifbxjs@force@chapterabstract
2454 \let\abstract\chapterabstract
2455 \let\endabstract\endchapterabstract

```

```

2456 \fi
2457 %</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。

```

2458 %<*jspf>
2459 %\newbox\@keywordsbox
2460 %\newenvironment{keywords}{%
2461 % \global\setbox\@keywordsbox\hbox\bgroup
2462 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2463 % \small\parindent0\jsZw}%
2464 % {\end{minipage}\egroup}
2465 %</jspf>

```

■verse 環境

verse 詩のための `verse` 環境です。

```

2466 \newenvironment{verse}{%
2467 \let \=\@centercr
2468 \list{}{%
2469 \itemsep \z@
2470 \itemindent -2\jsZw % 元: -1.5em
2471 \listparindent\itemindent
2472 \rightmargin \z@
2473 \advance\leftmargin 2\jsZw}% 元: 1.5em
2474 \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。

```

2475 \newenvironment{quotation}{%
2476 \list{}{%
2477 \listparindent\parindent
2478 \itemindent\listparindent
2479 \rightmargin \z@}%
2480 \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。

```

2481 \newenvironment{quote}%
2482 {\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。

```
2483 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2484   \item[\hskip \labelsep{\headfont #1\ #2}]}
2485 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2486   \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。

```
2487 \newenvironment{titlepage}{%
2488 %<book> \pltx@cleartooddpage %% 2017-02-24
2489   \if@twocolumn
2490     \@restonecoltrue\onecolumn
2491   \else
2492     \@restonecolfalse\newpage
2493   \fi
2494   \thispagestyle{empty}%
2495   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2496   24
2497 }%
2498 {\if@restonecol\twocolumn \else \newpage \fi
2499   \if@twoside\else
2500     \setcounter{page}\@ne
2501   \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2501 %<*&!book&!report>
2502 \newcommand{\appendix}{\par
2503   \setcounter{section}{0}%
2504   \setcounter{subsection}{0}%
2505   \ifnum\bxjs@label@section=\bxjs@label@section@compat
```

```

2506 \gdef\presectionname{\appendixname}%
2507 \gdef\postsectionname{}%
2508 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2509 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2510 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2511 \else
2512 \gdef\@secapp{\appendixname}%
2513 \gdef\@secpos{}%
2514 \gdef\thesection{\@Alph\c@section}%
2515 \fi}
2516 %</!book&!report>
2517 %<*book|report>
2518 \newcommand{\appendix}{\par
2519 \setcounter{chapter}{0}%
2520 \setcounter{section}{0}%
2521 \gdef\@chapapp{\appendixname}%
2522 \gdef\@chappos{}%
2523 \gdef\thechapter{\@Alph\c@chapter}}
2524 %</book|report>

```

8.4 パラメータの設定

■array と tabular 環境

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

```
2525 \setlength\arraycolsep{5\p@?}
```

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

```
2526 \setlength\tabcolsep{6\p@?}
```

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

```
2527 \setlength\arrayrulewidth{.4\p@}
```

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

```
2528 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2529 \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。

```
2530 \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。

```
2531 \setlength\fboxsep{3\p@?}
2532 \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。

```
2533 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2534 %<*book|report>
2535 \@addtoreset{equation}{chapter}
2536 \renewcommand\theequation
2537 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2538 %</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。

```
2539 % \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。

```
2540 % \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。

```
2541 % \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。

```
2542 %<!*book&!report>
2543 \newcounter{figure}
2544 \renewcommand \thefigure {\@arabic\c@figure}
2545 %</!*book&!report>
2546 %<*book|report>
2547 \newcounter{figure}[chapter]
2548 \renewcommand \thefigure
2549     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2550 %</book|report>
```

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

```
\ext@figure 2551 \def\fps@figure{tbp}
2552 \def\ftype@figure{1}
\fnun@figure 2553 \def\ext@figure{lof}
2554 \def\fnun@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2555 \newenvironment{figure}%
2556         {\@float{figure}}%
2557         {\end@float}
2558 \newenvironment{figure*}%
2559         {\@dblfloat{figure}}%
2560         {\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。

```
2561 %<!*book&!report>
2562 \newcounter{table}
2563 \renewcommand\thetable{\@arabic\c@table}
2564 %</!*book&!report>
2565 %<*book|report>
2566 \newcounter{table}[chapter]
2567 \renewcommand \thetable
2568     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2569 %</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 2570 \def\fps@table{tbp}
2571 \def\ftype@table{2}
\fnun@table 2572 \def\ext@table{lot}
2573 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` * は段抜きのフロートです。

`table*`

```

2574 \newenvironment{table}%
2575         {\@float{table}}%
2576         {\end@float}
2577 \newenvironment{table*}%
2578         {\@dblfloat{table}}%
2579         {\end@dblfloat}

```

8.6 キャプション

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

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

```

2580 \newlength\abovecaptionskip
2581 \newlength\belowcaptionskip
2582 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2583 \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。

```

2584 %<*\jpf>
2585 % \long\def\@makecaption#1#2{{\small
2586 %   \advance\leftskip10\jsc@mmm
2587 %   \advance\rightskip10\jsc@mmm
2588 %   \vskip\abovecaptionskip
2589 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2590 %   \ifdim \wd\@tempboxa >\hsize
2591 %     #1\hskip1\jsZw\relax #2\par
2592 %   \else
2593 %     \global \minipagefalse
2594 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2595 %   \fi
2596 %   \vskip\belowcaptionskip}}
2597 \long\def\@makecaption#1#2{{\small
2598   \advance\leftskip .0628\linewidth
2599   \advance\rightskip .0628\linewidth
2600   \vskip\abovecaptionskip
2601   \sbox\@tempboxa{#1\zwspace#2}%
2602   \ifdim \wd\@tempboxa <\hsize \centering \fi
2603   #1\zwspace#2\par
2604   \vskip\belowcaptionskip}}
2605 %</\jpf>
2606 %<*\jpf>
2607 \long\def\@makecaption#1#2{%

```

```

2608 \vskip\abovecaptionskip
2609 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2610 \ifdim \wd\@tempboxa >\hsize
2611     {\small\sffamily
2612         \list{#1}{%
2613             \renewcommand{\makelabel}[1]{##1\hfil}
2614             \itemsep \z@
2615             \itemindent \z@
2616             \labelsep \z@
2617             \labelwidth 11\jsc@mmm
2618             \listparindent\z@
2619             \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2620 \else
2621     \global \@minipagefalse
2622     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2623 \fi
2624 \vskip\belowcaptionskip}
2625 %</jspf>

```

9 フォントコマンド

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

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

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

```

\if@jsc@warnoldfontcmd


---


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


---


2626 \newif\if@jsc@warnoldfontcmd
2627 \@jsc@warnoldfontcmdtrue
2628 \newif\if@jsc@warnoldfontcmdexception
2629 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand
2630 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2631     \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2632     \DeclareOldFontCommand{#1}{%
2633         \bxjs@oldfontcmd{#1}#2%
2634     }{%

```

```

2635     \bxjs@oldfontcmd{#1}#3%
2636 }%
2637 }
2638 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2639   \ClassInfo\bxjs@clsname
2640   {Old font command '\string#1' is used!!\MessageBreak
2641     The first occurrence is}%
2642 }

```

`\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。

```

2643 \newcommand*{\allowoldfontcommands}{%
2644   \@jsc@warnoldfontcmdexceptiontrue}
2645 \newcommand*{\disallowoldfontcommands}{%
2646   \@jsc@warnoldfontcmdexceptionfalse}

2647 \let\bxjs@oldfontcmd@list\@empty
2648 \def\bxjs@oldfontcmd#1{%
2649   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2650 \def\bxjs@oldfontcmd@a#1#2{%
2651   \if@jsc@warnoldfontcmdexception\else
2652     \global\@jsc@warnoldfontcmdfalse
2653     \ifx#1\relax
2654       \global\let#1=t%
2655       \jsc@warnoldfontcmd{#2}%
2656     \fi
2657   \fi}
2658 \def\bxjs@warnoldfontcmd@final{%
2659 % \par
2660 \global\let\bxjs@warnoldfontcmd@final\@empty
2661 \let\@tempa\@empty
2662 \def\do##1{%
2663   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2664     \edef\@tempa{\@tempa \space\string##1}\fi}
2665 \bxjs@oldfontcmd@list
2666 \ifx\@tempa\@empty\else
2667   \ClassWarningNoLine\bxjs@clsname
2668   {Some old font commands were used in text:\MessageBreak
2669     \space\@tempa\MessageBreak
2670     You should note, that since 1994 LaTeX2e provides a\MessageBreak
2671     new font selection scheme called NFSS2 with several\MessageBreak
2672     new, combinable font commands. The
2673     class provides\MessageBreak
2674     the old font commands only for compatibility}
2675 \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。

```
2676 \def\bxjs@warnoldfontcmd@kick@final{%
2677   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2678 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}
```

`\mc` フォントファミリーを変更します。

```
\gt 2679 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2680 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2681 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2682 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2683 \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。

```
2684 \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。

```
2685 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2686 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2687 \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 2688 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2689 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

10 相互参照

10.1 目次の類

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

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

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

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

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

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


この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

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

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

という書式です。

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

インデント 左側の字下げ量です。

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

`\@pnumwidth` ページ番号の入る箱の幅です。

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

`\@dotsep` 点の間隔です (単位 mu)。

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

```
2690 \newcommand\@pnumwidth{1.55em}
2691 \newcommand\@tocrmarg{2.55em}
2692 \newcommand\@dotsep{4.5}
2693 %<!book&!report>\setcounter{tocdepth}{2}
2694 %<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)

```
2695 \newdimen\jsc@tocl@width
2696 \newcommand{\tableofcontents}{%
2697 %<*book|report>
2698 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2699 \settowidth\@tempdima{\headfont\appendixname}%
2700 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2701 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2702 \if@twocolumn
2703 \@restonecoltrue\onecolumn
2704 \else
2705 \@restonecolfalse
2706 \fi
2707 \chapter*{\contentsname}%
2708 \@mkboth{\contentsname}{%}
```

```

2709 %</book|report>
2710 %<*&book&!report>
2711 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2712 \settowidth\@tempdima{\headfont\appendixname}%
2713 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2714 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2715 \section*{\contentsname}%
2716 \@mkboth{\contentsname}{\contentsname}%
2717 %</!book&!report>
2718 \@starttoc{toc}%
2719 %<book|report> \if@restonecol\twocolumn\fi
2720 }

```

\l@part 部の目次です。

```

2721 \newcommand*{\l@part}[2]{%
2722 \ifnum \c@tocdepth >-2\relax
2723 %<!book&!report> \addpenalty\@secpenalty
2724 %<book|report> \addpenalty{-\@highpenalty}%
2725 \addvspace{2.25em \@plus\p@?}%
2726 \begingroup
2727 \parindent \z@
2728 % \@pnumwidth should be \@tocrmarg
2729 % \rightskip \@pnumwidth
2730 \rightskip \@tocrmarg
2731 \parfillskip -\rightskip
2732 {\leavevmode
2733 \large \headfont
2734 \setlength\@lnumwidth{4\jsZw}%
2735 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2736 \nobreak
2737 %<book|report> \global\@nobreaktrue
2738 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2739 \endgroup
2740 \fi}

```

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

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

```

2741 %<*&book|report>
2742 \newcommand*{\l@chapter}[2]{%
2743 \ifnum \c@tocdepth >\m@ne
2744 \addpenalty{-\@highpenalty}%
2745 \addvspace{1.0em \@plus\p@?}
2746 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2747 \begingroup
2748 \parindent\z@
2749 % \rightskip\@pnumwidth
2750 \rightskip\@tocrmarg
2751 \parfillskip-\rightskip

```

```

2752 \leavevmode\headfont
2753 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2754 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2755 \advance\leftskip\@lnumwidth \hskip-\leftskip
2756 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2757 \penalty\@highpenalty
2758 \endgroup
2759 \fi}
2760 %</book|report>

```

\l@section 節の目次です。

```

2761 %<*&!book&!report>
2762 \newcommand*{\l@section}[2]{%
2763 \ifnum \c@tocdepth >\z@
2764 \addpenalty{\@secpenalty}%
2765 \addvspace{1.0em \@plus\p@?}%
2766 \begingroup
2767 \parindent\z@
2768 % \rightskip\@pnumwidth
2769 \rightskip\@tocrmarg
2770 \parfillskip-\rightskip
2771 \leavevmode\headfont
2772 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2773 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2774 \advance\leftskip\@lnumwidth \hskip-\leftskip
2775 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2776 \endgroup
2777 \fi}
2778 %</!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。

```

2779 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

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

\l@subsubsection しれません。

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

```

\l@subparagraph 2780 %<*&!book&!report>
2781 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2782 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2783 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2784 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2785 %
2786 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2787 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2788 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2789 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2790 %

```

```

2791 \newcommand*{\l@section}{%
2792     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2793     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2794 \newcommand*{\l@subsubsection}{%
2795     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2796     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2797 \newcommand*{\l@paragraph}{%
2798     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2799     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2800 \newcommand*{\l@subparagraph}{%
2801     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2802     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2803 %<!/book&!report>
2804 %<*book|report>
2805 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2806 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2807 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2808 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2809 \newcommand*{\l@section}{%
2810     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2811     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2812 \newcommand*{\l@subsection}{%
2813     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2814     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2815 \newcommand*{\l@subsubsection}{%
2816     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2817     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2818 \newcommand*{\l@paragraph}{%
2819     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2820     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2821 \newcommand*{\l@subparagraph}{%
2822     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2823     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2824 %</book|report>

```

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

```

2825 \newdimen\@lnumwidth
2826 \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 · \hss}\hfill}
```

とします。

```
2827 \def\jsTocLine{\leaders\hbox{%
2828   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2829 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2830   \vskip \z@ \@plus.2\p@?
2831   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2832     \parindent #2\relax\@afterindenttrue
2833     \interlinepenalty\@M
2834     \leavevmode
2835     \@lnumwidth #3\relax
2836     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2837     {#4}\nobreak
2838     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2839       \hfil\normalfont \normalcolor #5}\par}\fi}
```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```
2840 \newcommand{\listoffigures}{%
2841   %<*book|report>
2842   \if@twocolumn\@restonecoltrue\onecolumn
2843   \else\@restonecolfalse\fi
2844   \chapter*{\listfigurename}%
2845   \@mkboth{\listfigurename}{}%
2846   %</book|report>
2847   %<!*book&!report>
2848   \section*{\listfigurename}%
2849   \@mkboth{\listfigurename}{\listfigurename}%
2850   %</!*book&!report>
2851   \@starttoc{lof}%
2852   %<book|report> \if@restonecol\twocolumn\fi
2853 }
```

`\l@figure` 図目次の項目を出力します。

```
2854 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```
2855 \newcommand{\listoftables}{%
2856   %<*book|report>
2857   \if@twocolumn\@restonecoltrue\onecolumn
2858   \else\@restonecolfalse\fi
2859   \chapter*{\listtablename}%
2860   \@mkboth{\listtablename}{}%
2861   %</book|report>
2862   %<!*book&!report>
2863   \section*{\listtablename}%
2864   \@mkboth{\listtablename}{\listtablename}%

```

```

2865 %</!book&!report>
2866 \starttoc{lot}%
2867 %<book|report> \if@restonecol\twocolumn\fi
2868 }

```

`\l@table` 表目次は図目次と同じです。

```

2869 \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。

```

2870 \newdimen\bibindent
2871 \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。

```

2872 \newenvironment{thebibliography}[1]{%
2873   \@jsc@warnoldfontcmdexceptiontrue
2874   \global\let\presectionname\relax
2875   \global\let\postsectionname\relax
2876 %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2877 %<*kiyou>
2878   \vspace{1.5\baselineskip}
2879   \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2880   \vspace{0.5\baselineskip}
2881 %</kiyou>
2882 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2883 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2884   \list{\@biblabel{\@arabic\c@enumiv}}%
2885     {\settowidth\labelwidth{\@biblabel{#1}}%
2886      \leftmargin\labelwidth
2887      \advance\leftmargin\labelsep
2888      \@openbib@code
2889      \usecounter{enumiv}%
2890      \let\p@enumiv\@empty
2891      \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2892 %<kiyou> \small
2893   \sloppy
2894   \clubpenalty4000
2895   \@clubpenalty\clubpenalty
2896   \widowpenalty4000%
2897   \sfcode`\.\@m}
2898 {\def\@noitemerr
2899   {\@latex@warning{Empty `thebibliography' environment}}}%
2900 \endlist}

```

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

```
2901 \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。

```
2902 \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。

```
2903 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

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

```
2904 % \def\@citex[#1]#2{\leavevmode
2905 %   \let\@citea\empty
2906 %   \@cite{\@for\@citeb:=#2\do
2907 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2908 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2909 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2910 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2911 %       \G@refundefinedtrue
2912 %       \@latex@warning
2913 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2914 %       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2915 % \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。

```
2916 % \DeclareRobustCommand\cite{\unskip
2917 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2918 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2919 %   , \jsInhibitGlue\ #2\fi}} }$}
```

10.3 索引

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

```
2920 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2921   \if@twocolumn
2922     \onecolumn\@restonecolfalse
2923   \else
2924     \clearpage\@restonecoltrue
2925   \fi
```

```

2926 \columnseprule.4pt \columnsep 2\jsZw
2927 \ifx\multicols\@undefined
2928 %<book|report> \twocolumn[\@makeschapterhead{\indexname}%
2929 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2930 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2931 %<!book&!report> \twocolumn[\section*{\indexname}]]%
2932 \else
2933 \ifdim\textwidth<\fullwidth
2934 \setlength{\evensidemargin}{\oddsidemargin}
2935 \setlength{\textwidth}{\fullwidth}
2936 \setlength{\linewidth}{\fullwidth}
2937 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}%
2938 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2939 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2940 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]]%
2941 \else
2942 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}%
2943 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2944 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2945 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]]%
2946 \fi
2947 \fi
2948 %<book|report> \mkboth{\indexname}{}%
2949 %<!book&!report> \mkboth{\indexname}{\indexname}%
2950 \plainifnotempty % \thispagestyle{plain}
2951 \parindent\z@
2952 \parskip\z@ \@plus .3\p@?\relax
2953 \let\item\@idxitem
2954 \raggedright
2955 \footnotesize\narrowbaselines
2956 }{
2957 \ifx\multicols\@undefined
2958 \if@restonecol\onecolumn\fi
2959 \else
2960 \end{multicols}
2961 \fi
2962 \clearpage
2963 }

```

`\@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 2964 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2965 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2966 \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。

```

2967 \newcommand{\indexspace}{\par \vskip 10\p@? \@plus5\p@? \@minus3\p@?\relax}

```

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

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

などでもいいでしょう。

```
2968 \newcommand\seename{\if@english see\else →\fi}  
2969 \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、`\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。

```
2970 \ifx\pltx@foot@penalty\@undefined  
2971   \let\footnotes@ve=\footnote  
2972   \def\footnote{\jsInhibitGlue\footnotes@ve}  
2973   \let\footnotemarks@ve=\footnotemark  
2974   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}  
2975 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

pT_EX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
2976 \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。

```
2977 % \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。

```
2978 \renewcommand{\footnoterule}{%
2979   \kern-2.6\p@? \kern-.4\p@
2980   \hrule width .4\columnwidth
2981   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2982 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

[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。

```
2983 \long\def\@footnotetext{%
2984   \insert\footins\bgroup
2985     \normalfont\footnotesize
2986     \interlinepenalty\interfootnotelinepenalty
2987     \splittopskip\footnotesep
2988     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2989     \hsize\columnwidth \@parboxrestore
2990     \protected@edef\@currentlabel{%
2991       \csname p@footnote\endcsname\@thefnmark
2992     }%
2993     \color@begingroup
2994     \makefnmark
2995     \rule\z@\footnotesep\ignorespaces}%
2996     \futurelet\jsc@next\jsc@fo@t}
2997 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@fo@t
2998               \else \let\jsc@next\jsc@fo@t\fi \jsc@next}
2999 \def\jsc@fo@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3000 \def\jsc@fo@t#1{#1\jsc@@foot}
3001 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
3002   \ifx\pltx@foot@penalty\@undefined\else
3003     \ifhmode\null\fi
3004     \ifnum\pltx@foot@penalty=\z@\else
3005       \penalty\pltx@foot@penalty
3006       \pltx@foot@penalty\z@
3007     \fi
3008   \fi}
```

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

```
3009 \newcommand\@makefntext[1]{%
3010   \advance\leftskip 3\jsZw
3011   \parindent 1\jsZw
3012   \noindent
3013   \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。

```
3014 % \def\@xfootnotenext[#1]{%
3015 %   \begingroup
3016 %     \ifnum#1>\z@
3017 %       \csname c@\@mpfn\endcsname #1\relax
3018 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
3019 %     \else
3020 %       \unrestored@protected@xdef\@thefnmark{}%
3021 %     \fi
3022 %   \endgroup
3023 %   \@footnotetext}
```

ここまでのコードは JS クラスを踏襲する。

11 段落の頭へのグルー挿入禁止

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

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

```
3024 \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。

```
3025 \def\everyparhook{\jsInhibitGlueAtParTop}
3026 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
3027 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3028 \fi
```

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

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

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

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

```
3029 \ifx j\jsEngine
3030 \def\@inhibitglue{%
3031   \futurelet\@let@token\@@inhibitglue}
3032 \begingroup
3033 \let\GDEF=\gdef
3034 \let\CATCODE=\catcode
3035 \let\ENDGROUP=\endgroup
3036 \CATCODE`k=12
3037 \CATCODE`a=12
3038 \CATCODE`n=12
3039 \CATCODE`j=12
3040 \CATCODE`i=12
3041 \CATCODE`c=12
3042 \CATCODE`h=12
3043 \CATCODE`r=12
3044 \CATCODE`t=12
3045 \CATCODE`e=12
3046 \GDEF\KANJI@CHARACTER{kanji character }
3047 \ENDGROUP
3048 \def\@@inhibitglue{%
3049   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3050 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
3051   \def\jsc@ig@temp{#1}%
3052   \ifx\jsc@ig@temp\@empty
3053     \ifnum\the\inhibitxspcode`#2=2\relax
3054       \inhibitglue
3055     \fi
3056   \fi}
3057 \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。

```
3058 \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。

```
3059 \def\@doendpe{%
3060   \@endpetrue
3061   \def\par{%
3062     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3063   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] `minipage` 環境にも対策します。

```
3064 \def\@setminipage{%
3065   \@minipagetrue
3066   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3067 }
```

`\item` 命令の直後です。

```
3068 \def\@item[#1]{%
3069   \if@noparitem
3070     \@donoparitem
3071   \else
3072     \if@inlabel
3073       \indent \par
3074     \fi
3075     \ifhmode
3076       \unskip\unskip \par
3077     \fi
3078     \if@newlist
3079       \if@nobreak
3080         \@nbitem
3081       \else
3082         \addpenalty\@beginparpenalty
3083         \addvspace\@topsep
3084         \addvspace{-\parskip}%
3085       \fi
3086     \else
3087       \addpenalty\@itempenalty
3088       \addvspace\itemsep
3089     \fi
3090     \global\@inlabeltrue
3091   \fi
3092   \everypar{%
3093     \@minipagefalse
3094     \global\@newlistfalse
3095     \if@inlabel
3096       \global\@inlabelfalse
3097     {\setbox\z@\lastbox
```

```

3098     \ifvoid\z@
3099     \kern-\itemindent
3100     \fi}%
3101     \box\@labels
3102     \penalty\z@
3103     \fi
3104     \if@nobreak
3105     \@nobreakfalse
3106     \clubpenalty \@M
3107     \else
3108     \clubpenalty \@clubpenalty
3109     \everypar{\everyparhook}%
3110     \fi
3111     \everyparhook}%
3112 \if@noitemarg
3113 \@noitemargfalse
3114 \if@nmbrlist
3115 \refstepcounter\@listctr
3116 \fi
3117 \fi
3118 \sbox\@tempboxa{\makelabel{#1}}}%
3119 \global\setbox\@labels\hbox{%
3120 \unhbox\@labels
3121 \hskip \itemindent
3122 \hskip -\labelwidth
3123 \hskip -\labelsep
3124 \ifdim \wd\@tempboxa >\labelwidth
3125 \box\@tempboxa
3126 \else
3127 \hbox to\labelwidth {\unhbox\@tempboxa}%
3128 \fi
3129 \hskip \labelsep}%
3130 \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。

```

3131 \def\@afterheading{%
3132 \@nobreaktrue
3133 \everypar{%
3134 \if@nobreak
3135 \@nobreakfalse
3136 \clubpenalty \@M
3137 \if@afterindent \else
3138 {\setbox\z@\lastbox}%
3139 \fi
3140 \else
3141 \clubpenalty \@clubpenalty
3142 \everypar{\everyparhook}%
3143 \fi\everyparhook}}

```

「\everyparhook 用のパッチ処理」はここまで。

3144 \fi

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

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

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

```
3145 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3146 \AtEndOfPackage{%
3147 \def\@gnewline #1{%
3148   \ifvmode
3149     \@nolnerr
3150   \else
3151     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3152     \jsInhibitGlue \ignorespaces
3153   \fi}
3154 }
3155 \fi
```

12 いろいろなロゴ

\LaTeX 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。

```
3156 \if@jslogo
3157   \IfFileExists{jslogo.sty}{%
3158     \RequirePackage{jslogo}%
3159   }{%
3160     \ClassWarningNoLine\bxjs@clsname
3161       {The package 'jslogo' is not installed.\MessageBreak
3162       It is included in the recent release of\MessageBreak
```

```

3163         the 'jsclasses' bundle}
3164     }
3165 \fi

```

13 amsmath との衝突の回避

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

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

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

```

3166 %\let\ltx@ifnextchar\@ifnextchar
3167 %\def\ProvidesFile#1{%
3168 %    \begingroup
3169 %        \catcode\ 10 %
3170 %        \ifnum \endlinechar<256 %
3171 %            \ifnum \endlinechar>\m@ne
3172 %                \catcode\endlinechar 10 %
3173 %            \fi
3174 %        \fi
3175 %        \@makeother\/%
3176 %        \@makeother\&%
3177 %        \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 3178 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3179 \newcommand{\postpartname}{\if@english\else 部\fi}
\postchaptername 3180 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3181 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3182 \newcommand{\presectionname}{}% 第
\postsectionname 3183 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3184 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3185 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3186 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname
\indexname

```



```

3187 \newcommand{\refname}{\if@english References\else 参考文献\fi}
3188 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3189 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3190 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3191 %<jspf>\newcommand{\figurename}{Fig.~}
3192 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3193 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3194 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3195 \newcommand{\appendixname}{\if@english \else 付録\fi}
3196 %<!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。

```

3197 \@onlypreamble\bxjs@decl@Seireki@cmds
3198 \@tempswafalse
3199 \if p\jsEngine \@tempswatruetrue \fi
3200 \if n\jsEngine \@tempswatruetrue \fi
3201 \bxjs@cond\if@tempswa\fi{%
3202 % 欧文 8bitTeX の場合
3203 \newif\ifjsSeireki \jsSeirekitruetrue
3204 \def\bxjs@decl@Seireki@cmds{%
3205   \def\西暦{\jsSeirekitruetrue}%
3206   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3207 \def\Seireki{\jsSeirekitruetrue}
3208 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3209 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3210 \def\bxjs@iai{\noexpand~}
3211 }{%
3212 \newif\if 西暦 \西暦 true
3213 \def\bxjs@decl@Seireki@cmds{%
3214   \def\西暦{\西暦 true}%
3215   \def\和暦{\西暦 false\bxjs@wareki@used}}
3216 \def\Seireki{\西暦 true}
3217 \def\Wareki{\西暦 false\bxjs@wareki@used}
3218 \def\bxjs@if@use@seireki{\bxjs@cond\if 西暦\fi}
3219 \let\bxjs@iai\@empty
3220 }

```

```

3221 \bxjs@decl@Seireki@cmds
3222 \let\bxjs@unxp@\firstofone \let\bxjs@onxp@\firstofone
3223 \bxjs@test@engine\unexpanded{%
3224   \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。

```

3225 \newif\ifbxjs@bxwareki@avail
3226 \IfFileExists{bxwareki.sty}{%
3227   \RequirePackage{bxwareki}[]%
3228   \bxjs@bxwareki@availtrue}{%

```

\bxjs@wareki@used 和暦が非対応の場合に警告を出す。

```

3229 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\@empty
3230 \else
3231   \bxjs@robust@def\bxjs@wareki@used{%
3232     \global\let\bxjs@wareki@used\@empty
3233     \ClassWarning\bxjs@clsname
3234       {Wareki mode is not supported, since\MessageBreak
3235         'bxwareki' is unavailable, found}}
3236   \g@addto@macro\bxjs@begin@document@hook{%
3237     \let\bxjs@wareki@used\@empty}
3238 \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。

```

3239 \ifbxjs@bxwareki@avail
3240   \let\jayear\warekiyear
3241   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3242     \newcount\heisei \heisei=\value{warekiyear}
3243   \fi

```

bxwareki が使えない場合は西暦表示にする。

```

3244 \else
3245   \edef\jayear{\the\year \bxjs@iai}
3246 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3247 \let\bxjs@next\relax
3248 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3249   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3250 \fi\fi
3251 \edef\bxjs@today{%
3252   \if@english
3253     \ifcase\month\or
3254       January\or February\or March\or April\or May\or June\or
3255       July\or August\or September\or October\or November\or December\fi
3256     \space\number\day, \number\year
3257   \else

```

```

3258 \ifx\bxjs@next\relax \expandafter\@firstoftwo
3259 \else \noexpand\bxjs@if@use@seireki
3260 \fi {%
3261 \number\year\bxjs@iai\bxjs@unxp{年}%
3262 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3263 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3264 }{\bxjs@next}%
3265 \fi}
3266 \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。

```

3267 \g@addto@macro\bxjs@begin@document@hook{%
3268 \ifx\bbl@jpn@maybekansuji\@undefined\else
3269 \bxjs@decl@Seireki@cmds
3270 \g@addto@macro\datejapanese{%
3271 \let\today\bxjs@today}%
3272 \fi}

```

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

```

3273 \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。

```

3274 %<slide>\pagestyle{empty}%
3275 %<article|report>\pagestyle{plain}%
3276 %<book>\pagestyle{headings}%
3277 \pagenumbering{arabic}
3278 \if@twocolumn
3279 \twocolumn
3280 \sloppy
3281 \flushbottom
3282 \else
3283 \onecolumn
3284 \raggedbottom
3285 \fi
3286 %<*slide>
3287 \renewcommand\familydefault{\sfdefault}
3288 \raggedright
3289 %</slide>

```

■BXJS 独自の追加処理 

フックを実行する。

3290 \bxjs@pre@jadriver@hook

和文ドライバのファイルを読み込む。

3291 \input{bxjsja-\bxjs@jadriver.def}

おしまい。

3292 %</cls>

以上です。

付録 A 和文ドライバの仕様

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

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

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

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

以下で標準で用意されている和文ドライバの実装を示す。

```
3293 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

ただし、(u)p_{TeX} 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

```
3294 %<*minimal>
```

```
3295 %% 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.
```

```
3296 \def\DeclareJaTextFontCommand#1#2{%
```

```
3297   \DeclareRobustCommand#1[1]{%
```

```
3298     \relax
```

```
3299     \ifmmode \expandafter\nfss@text \fi
```

```
3300     {#2##1}}%
```

```
3301 }
```

```
\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。
```

```
3302 \def\DeclareJaMathFontCommand#1#2{%
```

```
3303   \DeclareRobustCommand#1[1]{%
```

```
3304     \relax
```

```
3305     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3306     \nfss@text{\fontfamily\familydefault
```

```
3307       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3308       #2##1}}%
```

```

3309 }%
3310 }

```

`\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。

```

3311 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3312 \@onlypreamble\bxjs@if@sf@default
3313 \def\bxjs@if@sf@default#1{%
3314   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3315   \g@addto@macro\bxjs@begin@document@hook{%
3316     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3317 }

```

`\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。

```

3318 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3319 \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。

```

3320 \def\jsLetHeadChar#1#2{%
3321   \begingroup
3322     \escapechar=`\ %
3323     \let\bxjs@tmpa={% brace-match-hack
3324       \bxjs@let@hchar@exp#2}%
3325   \endgroup
3326   \let#1\bxjs@g@tmpa}
3327 \def\bxjs@let@hchar@exp{%
3328   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3329 \def\bxjs@let@hchar@exp@a{%
3330   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3331   \bxjs@let@hchar@out\let\relax
3332   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3333   \bxjs@let@hchar@out\let\space%
3334   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3335   \bxjs@let@hchar@out\let\@backslashchar
3336   }{\bxjs@let@hchar@exp@b}}}%
3337 \def\bxjs@let@hchar@exp@b#1{%
3338   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3339 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3340   %\message{<#1#2>}%
3341   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3342 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3343 \bxjs@let@hchar@out\let\relax
3344 }{%else
3345 \expandafter\bxjs@let@hchar@exp
3346 }%
3347 }{%else
3348 \bxjs@let@hchar@chr#1%
3349 }}
3350 \def\bxjs@let@hchar@chr#1{%
3351 \bxjs@let@hchar@out\def{{#1}}}}
3352 \def\bxjs@let@hchar@out#1#2{%
3353 \global#1\bxjs@g@tmpa#2\relax
3354 \toks@{\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3355 \chardef\bxjs@let@hchar@csta=128
3356 \chardef\bxjs@let@hchar@cstb=192
3357 \chardef\bxjs@let@hchar@cstc=224
3358 \chardef\bxjs@let@hchar@cstd=240
3359 \chardef\bxjs@let@hchar@cste=248
3360 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3361 \def\bxjs@let@hchar@chr@ue#1{%
3362 \@tempcnta=#1\relax
3363 %\message{\the\@tempcnta}%
3364 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3365 \bxjs@let@hchar@chr@ue@a#1%
3366 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3367 \bxjs@let@hchar@out\let\relax
3368 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3369 \bxjs@let@hchar@chr@ue@b
3370 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3371 \bxjs@let@hchar@chr@ue@c
3372 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3373 \bxjs@let@hchar@chr@ue@d
3374 }{%else
3375 \bxjs@let@hchar@out\let\relax
3376 }}}}}
3377 \def\bxjs@let@hchar@chr@ue@a#1{%
3378 \bxjs@let@hchar@out\def{{#1}}}}
3379 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3380 \bxjs@let@hchar@out\def{{#1#2}}}}
3381 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3382 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3383 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3384 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定


```
3385 \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。

```
3386 \def\bxjs@let@hchar@chr@pp#1#2{%
3387   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3388 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3389   \%message{(\meaning#3:\meaning#4)}%
3390   \bxjs@cond\if#1k\fi{%
3391     \bxjs@let@hchar@out\def{#{#4}}%
3392   }{%else
3393     \bxjs@let@hchar@chr@ue#3#4%
3394   }}
3395 \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。

```
3396 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3397 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3398 \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。

```
3399 \@onlypreamble\bxjs@declarefontshape
3400 \ifjsWithupTeX
3401 \def\bxjs@declarefontshape{%
3402   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3403   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3404   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3405   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3406 }
3407 \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。

```
3408 \else
3409 \def\bxjs@declarefontshape{%
3410   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3411   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3412   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3413 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3414 }
3415 \def\bxjs@sizereference{jis}
3416 \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。

```

3417 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3418   \def\bxjs@tmpb{#5}}
3419 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3420 \else
3421 \expandafter\expandafter\expandafter\bxjs@next
3422 \expandafter\string\the\jfont\relax
3423 \fi
3424 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3425                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3426   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3427    \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。

```

3428 \begingroup
3429 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3430 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3431 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3432 % 幅が丁度 10pt なら補正は不要
3433 \ifdim\wd\z@=10pt
3434   \global\let\bxjs@scale\jsScale
3435 \else
3436 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3437   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3438   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3439   \bxjs@invscale\@tempdima\bxjs@tmpa
3440   \xdef\bxjs@scale{\strip@pt\@tempdima}
3441 \fi
3442 \endgroup
3443 %\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。

```

3444 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3445 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3446 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3447 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3448 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3449 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3450 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3451 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3452 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{f}
3453 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{f}
3454 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{f}
3455 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{f}
3456 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{f}
3457 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{f}
3458 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{f}
3459 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{f}
3460 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3461 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3462 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3463 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3464 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3465 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3466 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3467 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{f}
3468 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{f}
3469 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{f}
3470 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{f}
3471 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{f}
3472 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{f}
3473 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{f}
3474 \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。

```

3475 \ifx\@rmfamilyhook\@undefined % old
3476 \DeclareRobustCommand\rmfamily
3477   {\not@math@alphabet\rmfamily\mathrm
3478     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3479 \DeclareRobustCommand\sffamily
3480   {\not@math@alphabet\sffamily\mathsf
3481     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3482 \DeclareRobustCommand\ttfamily
3483   {\not@math@alphabet\ttfamily\mathtt
3484     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3485 \g@addto@macro\bxjs@begin@document@hook{%
3486   \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3487     % my definitions above should have been overwritten, recover it!
3488     % \selectfont is executed twice but I don't care about speed...
3489     \expandafter\g@addto@macro\csname rmfamily \endcsname
3490       {\kanjifamily\mcdefault\selectfont}%
3491     \expandafter\g@addto@macro\csname sffamily \endcsname
3492       {\kanjifamily\gtdefault\selectfont}%

```

```

3493 \expandafter\g@addto@macro\csname ttfamily \endcsname
3494 {\kanjifamily\gtdefault\selectfont}%
3495 \fi}
3496 \else % 2020-02-02
3497 \g@addto@macro\@rmfamilyhook
3498 {\prepare@family@series@update@kanji{mc}\mcdefault}
3499 \g@addto@macro\@sffamilyhook
3500 {\prepare@family@series@update@kanji{gt}\gtdefault}
3501 \g@addto@macro\@ttfamilyhook
3502 {\prepare@family@series@update@kanji{gt}\gtdefault}
3503 \fi
3504 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3505 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3506 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3507 \fi
3508 \bxjs@if@sf@default{%
3509 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3510 \selectfont

```

`\bxjs@parse@qh` の処理は p_TE_X 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）。

```

3511 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3512 \let\bxjs@parse@qh@a\@undefined
3513 \let\bxjs@parse@qh@b\@undefined

```

■パラメタの設定

```

3514 \prebreakpenalty\jis"2147=10000
3515 \postbreakpenalty\jis"2148=10000
3516 \prebreakpenalty\jis"2149=10000
3517 \inhibitxspcode`!=1
3518 \inhibitxspcode`〒=2
3519 \xspcode`+=3
3520 \xspcode`%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3521 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3522 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3523 \let\jsInhibitGlueAtParTop\@inhibitglue

```

`\jsResetDimen` は空のままでよい。

■組方向依存の処理 組方向判定の `if-トークン` (`\if?dir`) は p_TE_X 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。

```

3524 \begingroup
3525 \catcode`\!=0

```

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

```
3526 \gdef\bxjs@ptex@dir{%
3527   !iftdir t%
3528   !else!ifydir y%
3529   !else ?%
3530   !fi!fi}
```

新版の p_TE_X 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 p_TE_X 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。

```
3531 % 古い \@makefnmark の定義
3532 \long\def\bxjs@tmpa{\hbox{%
3533   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3534   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3535 \ifx\@makefnmark\bxjs@tmpa
3536 \long\gdef\@makefnmark{%
3537   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3538   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3539 \fi
3540 \endgroup
```

B.3 pdf_TE_X 用の処理

```
3541 \else\ifx p\jsEngine
3542 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3543 \@onlypreamble\bxjs@cjk@loaded
3544 \def\bxjs@cjk@loaded{%
3545   \def\@footnotemark{%
3546     \leavevmode
3547     \ifhmode
3548       \edef\x@sf{\the\spacefactor}%
3549       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3550         \unkern\unkern
3551         \ifdim\lastskip>\z@ \unskip \fi
3552       \fi\fi
3553       \nobreak
3554     \fi
3555     \@makefnmark
3556     \ifhmode \spacefactor\x@sf \fi
3557     \relax}%
3558 \let\bxjs@cjk@loaded\relax
3559 }
3560 \g@addto@macro\bxjs@begin@document@hook{%
3561   \@ifpackageloaded{CJK}{%
3562     \bxjs@cjk@loaded
3563   }{}%
3564 }
```

B.4 XeTeX 用の処理

3565 \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。

```
3566 \def\bxjs@let@hchar@chr#1{%
3567   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3568   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3569     \bxjs@let@hchar@chr@xe
3570   }\bxjs@let@hchar@out\def{{#1}}}}
3571 \def\bxjs@let@hchar@chr@xe#1{%
3572   \lccode`0=`#1\relax
3573   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3574 \@onlypreamble\bxjs@do@precisetext
3575 \ifx\XeTeXgenerateactualtext\@undefined\else
3576   \def\bxjs@do@precisetext{%
3577     \XeTeXgenerateactualtext=\@ne}
3578 \fi
```

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

```
3579 \@onlypreamble\bxjs@do@simplejasetup
3580 \def\bxjs@do@simplejasetup{%
3581   \ifnum\XeTeXinterchartokenstate>\z@
3582   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3583     \jsSimpleJaSetup
3584     \ClassInfo\bxjs@clsname
3585     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3586   \fi\fi}
```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```
3587 \newcommand*{\jsSimpleJaSetup}{%
3588   \XeTeXlinebreaklocale "ja"\relax
3589   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3590   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

3591 \fi\fi\fi

simplejasetup オプションの処理。

```
3592 \ifx\bxjs@do@simplejasetup\@undefined\else
3593   \g@addto@macro\bxjs@begin@document@hook{%
3594     \ifbxjs@simplejasetup
3595       \bxjs@do@simplejasetup
3596     \fi}
3597 \fi
```

precisetext オプションの処理。

```

3598 \ifbxjs@precisetext
3599   \ifx\bxjs@do@precisetext\@undefined
3600     \ClassWarning\bxjs@clsname
3601       {The current engine does not support the\MessageBreak
3602         'precise-text' option\@gobble}
3603   \else
3604     \bxjs@do@precisetext
3605   \fi
3606 \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。

```

3607 \g@addto@macro\bxjs@begin@document@hook{%
3608   \ifx\jsInhibitGlueAtParTop\@empty
3609     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3610     \ifx\everyparhook\bxjs@tmpa
3611       \let\everyparhook\@empty
3612     \fi
3613 \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。

```

3614 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3615 \let\bxjs@everypar\everypar
3616 \newtoks\everypar
3617 \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。

```

3618 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3619 \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。

```

3620 \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。

```

3621 \@onlypreamble\bxjs@adjust@fancyhdr
3622 \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。

```
3623 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3624 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3625 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3626 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3627 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3628 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3629 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3630 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3631 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3632 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3633 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3634 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3635 \def\bxjs@tmpa{\rm\thepage\strut}%
3636 \def\bxjs@tmpb{\thepage\strut}%
3637 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3638 \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。

```
3639 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3640 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3641 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3642 }\bxjs@tmpa
3643 \fi\fi
3644 \PackageInfo\bxjs@clsname
3645 {Patch to fancyhdr is applied\@gobble}}
```

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

```
3646 \def\bxjs@pagestyle@hook{%
3647 \ifpackageloaded{fancyhdr}{%
3648 \bxjs@adjust@fancyhdr
3649 \global\let\bxjs@adjust@fancyhdr\relax
3650 }{}}
```

\pagestyle にフックを入れ込む。

```
3651 \let\bxjs@org@pagestyle\pagestyle
3652 \def\pagestyle{%
3653 \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。

```
3654 \g@addto@macro\bxjs@begin@document@hook{%
3655 \bxjs@pagestyle@hook
3656 \global\let\bxjs@pagestyle@hook\relax}
```

■和文空白命令

```
3657 \ifbxjs@jaspace@cmd
```


`\jaenspace` 半角幅の水平空き。

```
3658 \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。

```
3659 \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。

```
3660 \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3661 \DeclareRobustCommand*{\jaspace}[1]{%
3662   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3663   \ClassError\bxjs@clsname
3664   {Unknown jaspacespace: #1}{\@eha}%
3665   \else
3666   \csname bxjs@jaspace@@#1\endcsname
3667   \fi}
3668 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3669 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3670 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3671 \fi
```

終わり。

```
3672 \fi
```

以上で終わり。

```
3673 %</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。

```
3674 %<*standard>
3675 %% このファイルは日本語文字を含みます
3676 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3677 \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。

```
3678 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3679 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3680 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3681 \define@key{bxjsStd}{jis2004}[true]{%
3682   \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。

```
3683 \newif\ifbxjs@jp@units

units オプションの処理。
3684 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3685 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3686 \define@key{bxjsStd}{units}[true]{%
3687   \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。

```
3688 \let\bxjs@jp@font\@empty

font オプションの処理。
3689 \define@key{bxjsStd}{font}{%
3690   \edef\bxjs@jp@font{#1}}
```

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

```
3691 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue

strong-cmd オプションの処理。
3692 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
3693 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
3694 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}
```

実際の japaram の値を適用する。

```
3695 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3696 \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。

```
3697 \ifbxjs@jp@jismmiv
3698   \g@addto@macro\@classoptionslist{,jis2004}
3699 % \@ifpackagewith 判定への対策
3700   \PassOptionsToPackage{jis2004}{otf}
3701 \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。

```
3702 \if j\jsEngine
3703   \bxjs@jp@unitsfalse
3704 \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。

```
3705 \ifbxjs@jp@units
3706   \IfFileExists{bxcalc.sty}{%
3707     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3708     \ifx\usepTeXunits\@undefined
3709       \PackageWarningNoLine\bxjs@clsname
3710         {Cannot support pTeX units (zw etc.), since\MessageBreak
3711           the package 'bxcalc' is too old}%
3712       \bxjs@jp@unitsfalse
3713     \else \usepTeXunits
3714     \fi
3715   }{%else
3716     \PackageWarningNoLine\bxjs@clsname
3717       {Cannot support pTeX units (zw etc.), since\MessageBreak
3718         the package 'bxcalc' is unavailable}%
3719     \bxjs@jp@unitsfalse
3720   }
3721 \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。

```
3722 \ifbxjs@jp@units
3723 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3724 \let\bxjs@parse@qh@a\@undefined
3725 \let\bxjs@parse@qh@b\@undefined
3726 \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。

```
3727 \ifbxjs@jp@units
3728   \def\bxjs@let@lenexpr#1#2{%
3729     \edef#1{#2}%
3730     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3731 \else
3732   \def\bxjs@let@lenexpr{\edef}
3733 \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に `strongenv` にする。

※

```
3734 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3735   \ifx\strong\undefined\ifx\strongenv\undefined
3736     \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3737     \DeclareTextFontCommand{\strong}{\strongenv}%
```

`fontspec` と互換の `\strongfontdeclare` 命令も提供する。ただし、BXJS 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。

```
3738   \@ifdefinable{\strongfontdeclare}{\bxjs@protected\def\strongfontdeclare#1{%
3739     \bxjs@strong@font@declare@a#1,\@nil}}%
3740   \def\bxjs@strong@font@declare@a#1,#2\@nil{\def\bxjs@strong@font{#1}}%
3741   \def\bxjs@strong@font{\bfseries}%
3742   \fi\fi
3743 }\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。

```
3744 \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。

```
\jH3745   \@tempdima=0.25mm
3746   \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3747   \let\jH\jQ
```

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

```
\trueH3748   \ifjsc@mag
3749     \@tempdimb=\jsBaseFontSize\relax
3750     \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
```

```

3751 \tempdima=2.5mm
3752 \bxjs@invscale\tempdima\bxjs@tmpa
3753 \protected\edef\trueQ{\dimexpr\the\tempdima\relax}
3754 \tempdima=10pt
3755 \bxjs@invscale\tempdima\bxjs@tmpa
3756 \protected\edef\bxjs@truept{\dimexpr\the\tempdima\relax}
3757 \else \let\trueQ\jQ \let\bxjs@truept\p@
3758 \fi
3759 \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。

```

3760 \tempdima\trueQ \bxjs@invscale\tempdima\jsScale
3761 \protected\edef\ascQ{\dimexpr\the\tempdima\relax}
3762 \tempdima\bxjs@truept \bxjs@invscale\tempdima\jsScale
3763 \protected\edef\ascpt{\dimexpr\the\tempdima\relax}
3764 \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。

```

3765 \def\jafontsize#1#2{%
3766 \begingroup
3767 \bxjs@jafontsize@a{#1}%
3768 \tempdimb\jsInverseScale\tempdima
3769 \bxjs@jafontsize@a{#2}%
3770 \xdef\bxjs@g@tmpa{%
3771 \noexpand\fontsize{\the\tempdimb}{\the\tempdima}}%
3772 \endgroup\bxjs@g@tmpa}
3773 \def\bxjs@jafontsize@a#1{%
3774 \bxjs@parse@qh{#1}%
3775 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3776 \@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。

```

3777 \def\bxjs@kanjiskip{0pt}

```

\setkanjiskip 和文間空白の量を設定する。

```

3778 \newcommand*\setkanjiskip[1]{%
3779 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3780 \bxjs@reset@kanjiskip}

```

\getkanjiskip 和文間空白の量を表すテキストに展開する。

```

3781 \newcommand*\getkanjiskip{%
3782 \bxjs@kanjiskip}

```

`\ifbxjs@kanjiskip@enabled` 和文間空白の挿入が有効か。ただし p_TE_X 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。

```
3783 \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。(p_TE_X 以外)

```
\bxjs@disable@kanjiskip 3784 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3785   \bxjs@kanjiskip@enabledtrue
3786   \bxjs@reset@kanjiskip}
3787 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3788   \bxjs@kanjiskip@enabledfalse
3789   \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。

```
3790 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3791   \ifbxjs@kanjiskip@enabled
3792     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3793   \else \@tempskipa\z@
3794   \fi
3795   \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 3796 \def\bxjs@xkanjiskip{Opt}
```

```
\getxkanjiskip 3797 \newcommand*\setxkanjiskip[1]{%
```

```
3798   \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3799   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3800 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3801   \bxjs@xkanjiskip}
```

```
3802 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3803 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3804   \bxjs@xkanjiskip@enabledtrue
```

```
3805   \bxjs@reset@xkanjiskip}
```

```
3806 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3807   \bxjs@xkanjiskip@enabledfalse
```

```
3808   \bxjs@reset@xkanjiskip}
```

```
3809 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```
3810   \ifbxjs@xkanjiskip@enabled
```

```
3811     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
```

```
3812   \else \@tempskipa\z@
```

```
3813   \fi
```

```
3814   \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。

```
3815 \g@addto@macro\jsResetDimen{%
```

```
3816   \bxjs@reset@kanjiskip
```

```
3817   \bxjs@reset@xkanjiskip}
```

```
3818 \let\bxjs@apply@kanjiskip\relax
```

```
3819 \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、T_EX Live の kanji-config-updmap コマンドで使う“ファミリー”と同じにすることを想定する。特別な値として、auto は kanji-config-updmap で現在指定されているファミリーを表す。

\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。

```

3820 \@onlypreamble\bxjs@adjust@jafont
3821 \def\bxjs@adjust@jafont#1{%
3822   \ifx\jsJaFont\bxjs@@auto
3823     \bxjs@get@kanjiEmbed
3824     \ifx\bxjs@kanjiEmbed\relax
3825       \let\bxjs@tmpa\@empty
3826     \else
3827       \let\bxjs@tmpa\bxjs@kanjiEmbed
3828     \fi
3829   \else
3830     \let\bxjs@tmpa\jsJaFont
3831   \fi
3832   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3833     \ClassWarningNoLine\bxjs@clsname
3834     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3835      not available on the current situation}%
3836     \let\bxjs@tmpa\@empty
3837   \fi\fi
3838 }
3839 \def\bxjs@@auto{auto}
3840 \def\bxjs@@noEmbed{noEmbed}

```

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

```

3841 \let\bxjs@kanjiEmbed\relax

```

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

```

3842 \@onlypreamble\bxjs@get@kanjiEmbed
3843 \def\bxjs@get@kanjiEmbed{%
3844   \begingroup\setbox\z@=\hbox{%
3845     \global\let\bxjs@g@tmpa\relax
3846     \endlinechar\m@ne
3847     \let\do\@makeother\dospecials
3848     \catcode32=10 \catcode12=10 %form-feed
3849     \let\bxjs@tmpa\@empty
3850     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3851     \ifeof\@inputcheck\else
3852       \read\@inputcheck to\bxjs@tmpa
3853     \closein\@inputcheck

```

```

3854 \fi
3855 \ifx\bxjs@tmpa\@empty\else
3856 \openin\@inputcheck="\bxjs@tmpa"\relax
3857 \@tempwattrue
3858 \loop\if@tempwa
3859 \read\@inputcheck to\bxjs@tmpa
3860 \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3861 \ifx\bxjs@tmpb\relax\else
3862 \global\let\bxjs@g@tmpa\bxjs@tmpb
3863 \@tempwafalse
3864 \fi
3865 \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3866 \ifx\bxjs@tmpb\relax\else
3867 \global\let\bxjs@g@tmpa\bxjs@tmpb
3868 \@tempwafalse
3869 \fi
3870 \ifeof\@inputcheck \@tempwafalse \fi
3871 \repeat
3872 \fi
3873 }\endgroup
3874 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3875 }
3876 \@onlypreamble\bxjs@get@ke@a
3877 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3878 \ifx$#1$\def\bxjs@tmpb{#2}%
3879 \else \let\bxjs@tmpb\relax
3880 \fi}
3881 \@onlypreamble\bxjs@get@ke@b
3882 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3883 \ifx$#1$\def\bxjs@tmpb{#2}%
3884 \else \let\bxjs@tmpb\relax
3885 \fi}

```

\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。

```

3886 \@onlypreamble\bxjs@resolve@jafont@paren
3887 \def\bxjs@resolve@jafont@paren#1{%
3888 \def\bxjs@tmpb{\let#1}%
3889 \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3890 \@onlypreamble\bxjs@resolve@jafont@paren@a
3891 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3892 \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3893 \else
3894 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3895 \bxjs@tmpb\bxjs@tmpa
3896 \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。

```
3897 \newcommand*\jachar[1]{%
3898   \begingroup
      \jsLetHeadChar で先頭の“文字”を拾ってそれを \bxjs@jachar に渡す。
3899   \jsLetHeadChar\bxjs@tmpa{#1}%
3900   \ifx\bxjs@tmpa\relax
3901     \ClassWarningNoLine\bxjs@clsname
3902       {Illegal argument given to \string\jachar}%
3903   \else
3904     \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3905   \fi
3906 \endgroup}

\jsJaChar を \jachar と等価にする。
3907 \let\jsJaChar\jachar

下請けの \bxjs@jachar の実装はエンジンにより異なる。
3908 \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。

```
3909 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3910 \@onlypreamble\bxjs@fix@hyperref@unicode
3911 \def\bxjs@fix@hyperref@unicode#1{%
3912   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3913   \@namedef{KV@Hyp@bxjs/hook}##1{%
3914     \KV@Hyp@unicode{##1}%
3915     \def\KV@Hyp@unicode####1{%
3916       \expandafter\ifx\csname if##1\expandafter\endcsname
3917         \csname if####1\endcsname\else
3918         \ClassWarningNoLine\bxjs@clsname
3919           {Blcked hyperref option 'unicode=####1'}%
3920       \fi
3921     }%
3922   }%
3923 }
```

`\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。

```
3924 \@onlypreamble\jsCheckHyperrefUnicode
3925 \let\jsCheckHyperrefUnicode\@empty
3926 \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。

```
3927 \@onlypreamble\bxjs@check@hyperref@unicode
3928 \def\bxjs@check@hyperref@unicode#1{%
3929   \g@addto@macro\jsCheckHyperrefUnicode{%
```

```

3930 \@tempswafalse
3931 \beginingroup
3932 \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3933 \aftergroup\@tempswattrue \fi
3934 \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3935 \csname if#1\endcsname
3936 \aftergroup\@tempswattrue \fi
3937 \endgroup
3938 \if@tempswa\else
3939 \ClassError\bxjs@clsname
3940 {The value of hyperref 'unicode' key is not suitable\MessageBreak
3941 for the present engine (must be #1)}%
3942 {\@ehc}%
3943 \fi}}

```

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

```

3944 \@onlypreamble\bxjs@urgent@special
3945 \def\bxjs@urgent@special#1{%
3946 \AtBeginDvi{\special{#1}}%
3947 \g@addto@macro\bxjs@begin@document@hook{%
3948 \@ifpackageloaded{atbegshi}{%
3949 \beginingroup
3950 \toks\z@{\special{#1}}%
3951 \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3952 \xdef\AtBegShi@HookFirst{\the\toks@}\the\toks\tw@}%
3953 \endgroup
3954 }{}}%
3955 }%
3956 }

```

C.4 pTeX 用設定

```
3957 \if j\jsEngine
```

■ 共通命令の実装

```

3958 \def\bxjs@apply@kanjiskip{%
3959 \kanjiskip\@tempskipa}
3960 \def\bxjs@apply@xkanjiskip{%
3961 \xkanjiskip\@tempskipa}

```

`\jaJaChar` のサブマクロ。

```

3962 \def\bxjs@jachar#1{%
3963 \bxjs@jachar@a#1...\@nil}
3964 \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。

```
3965 \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。

```

3966 \else\ifx.#3%
3967 \@tempcnta`#1 \multiply\@tempcnta64
3968 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3969 \bxjs@jachar@b
3970 \else\ifx.#4%
3971 \@tempcnta`#1 \multiply\@tempcnta64
3972 \advance\@tempcnta`#2 \multiply\@tempcnta64
3973 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3974 \bxjs@jachar@b
3975 \else
3976 \@tempcnta`#1 \multiply\@tempcnta64
3977 \advance\@tempcnta`#2 \multiply\@tempcnta64
3978 \advance\@tempcnta`#3 \multiply\@tempcnta64
3979 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3980 \bxjs@jachar@b
3981 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3982 \ifjsWithupTeX
3983 \def\bxjs@jachar@b{\kchar\@tempcnta}
3984 \else
3985 \def\bxjs@jachar@b{%
3986 \ifx\bxUInt\@undefined\else
3987 \bxUInt{\@tempcnta}%
3988 \fi}
3989 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

3990 \ifbxjs@jaspace@cmd
3991 \def\jathinspace{\hskip\xkanjiskip}
3992 \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。

```

3993 \ifbxjs@jp@jismmiv
3994 \PassOptionsToPackage{prefer2004jis}{pxchfon}
3995 \PassOptionsToPackage{prefer2004jis}{pxbabel}
3996 \fi

```

■和文フォント指定の扱い pTeX は既定で kanji-config-updmap の設定に従うため、\jsJaFont が auto の場合は何もする必要がない。無指定でも auto でもない場合は、\jsJaFont をオプションにして pxchfon パッケージを読み込む。ここで、和文ドライバパラメタ font が指定されている場合は、その値を pxchfon のオプションに追加する。

```

3997 \let\bxjs@jafont@paren\@firstofone
3998 \let\bxjs@tmpa\jsJaFont
3999 \ifx\bxjs@tmpa\bxjs@@auto
4000 \let\bxjs@tmpa\@empty
4001 \else\ifx\bxjs@tmpa\bxjs@@noEmbed

```

```

4002 \def\bxjs@tmpa{noembed}
4003 \fi\fi
4004 \bxjs@resolve@jafont@paren\bxjs@tmpa
4005 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4006 \ifx\bxjs@tmpa\@empty\else
4007 \edef\bxjs@next{%
4008 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4009 }\bxjs@next
4010 \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……。)

```

4011 \begingroup
4012 \global\let\bxjs@g@tmpa\relax
4013 \catcode`\|=0 \catcode`\|=12
4014 |def|bxjs@tmpdo@a#1|@nil{%
4015 |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4016 |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4017 |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4018 |catcode`\|=0 \catcode`\|=12
4019 \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4020 \ifx$#2$\else
4021 \xdef\bxjs@g@tmpa{%
4022 \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4023 \fi}
4024 \@firstofone{%
4025 \catcode10=12 \endlinechar\m@ne
4026 \let\do@makeother \dospecials \catcode32=10
4027 \openin\@inputcheck=otf.sty\relax
4028 \@tempswatrue
4029 \loop\if@tempswa
4030 \ifeof\@inputcheck \@tempswafalse \fi
4031 \if@tempswa
4032 \read\@inputcheck to\bxjs@next
4033 \expandafter\bxjs@tmpdo\bxjs@next\@nil
4034 \fi
4035 \repeat
4036 \closein\@inputcheck
4037 \endgroup}
4038 \bxjs@g@tmpa

```

■hyperref 対策 unicode にしてはいけない。

```

4039 \ifbxjs@hyperref@enc
4040 \bxjs@check@hyperref@unicode{false}

```

```

4041 \fi

      tounicode special 命令を出力する。
4042 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4043   \else\ifjsWithpTeXng T\else F\fi\fi T%
4044   \ifnum\jis"2121="A1A1 %euc
4045     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4046   \else\ifnum\jis"2121="8140 %sjis
4047     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4048   \else\ifnum\jis"2121="3000 %uptex
4049     \ifbxjs@bigcode
4050       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4051       \PassOptionsToPackage{bigcode}{pxjahyper}
4052     \else
4053       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4054     \fi
4055   \fi\fi\fi
4056   \let\bxToUnicodeSpecialDone=t
4057 \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。

```

4058 \ifx f\bxjs@enablejfam\else
4059   \@enablejfamtrue
4060 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

4061 \if@enablejfam
4062   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4063   \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4064   \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4065   \jfam\symmincho
4066   \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4067   \g@addto@macro\bxjs@begin@document@hook{%
4068     \ifx\reDeclareMathAlphabet\@undefined\else
4069       \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4070       \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4071       \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4072     \fi}
4073 \fi

```

C.5 pdfTeX 用設定 : CJK + bxcjkatype

```

4074 \else\if p\jsEngine

```

■`bxcjkatype` パッケージの読み込 `\jsJaFont` が指定されている場合は、その値を `bxcjkatype` のオプション（プリセット指定）に渡す。（`auto` ならば `\bxjs@get@kanjiEmbed` を実行する。）スケール値（`\jsScale`）の反映は `bxcjkatype` の側で行われる。

※ Pandoc モードでは autotilde を指定しない。

```
4075 \bxjs@adjust@jafont{f}
4076 \let\bxjs@jafont@paren\@firstofone
4077 \bxjs@resolve@jafont@paren\bxjs@tmpa
4078 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4079 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4080 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4081   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4082 \fi
4083 \edef\bxjs@next{%
4084   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkjatype}[2013/10/15]% v0.2c
4085 }\bxjs@next
4086 \bxjs@cjk@loaded
```

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

※取りあえず固定はしない。

```
4087 \ifbxjs@hyperref@enc
4088   \PassOptionsToPackage{unicode}{hyperref}
4089 \fi
```

\hypersetup 命令で (CJK* 環境に入れなくても) 日本語文字を含む文書情報を設定できるようにするための細工。

※ bxcjkjatype を whole 付きで使っていることが前提。

※パッケージオプションでの指定に対応するのは、「アクティブな高位バイトトークンがその場で展開されてしまう」ため困難である。

```
4090 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
4091 \begingroup
4092   \CJK@input{UTF8.bdg}
4093 \endgroup
4094 \g@addto@macro\pdfstringdefPreHook{%
4095   \@nameuse{CJK@UTF8Binding}%
4096 }
4097 \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。

```
4098 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
4099 \g@addto@macro\pdfstringdefPreHook{%
4100   \ifx~\bxjs@@CJKtilde
4101     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4102     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4103     \let~\@empty
4104   \fi
4105 }
4106 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4107 \def\bxjs@@tildecmd{~}
4108 \def\bxjs@LetUnexpandableSpace#1{%
4109   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4110     \bxjs@org@LetUnexpandableSpace#1%
```

```
4111 \fi}
4112 \fi
```

■共通命令の実装

```
4113 \newskip\jsKanjiSkip
4114 \newskip\jsXKanjiSkip
4115 \ifx\CJKecglue\@undefined
4116 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4117 \fi
4118 \let\autospacing\bxjs@enable@kanjiskip
4119 \let\noautospacing\bxjs@disable@kanjiskip
4120 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4121 \def\bxjs@apply@kanjiskip{%
4122 \jsKanjiSkip\@tempskipa
4123 \let\CJKglue\bxjs@CJKglue}
4124 \let\autoxspacing\bxjs@enable@xkanjiskip
4125 \let\noautoxspacing\bxjs@disable@xkanjiskip
4126 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4127 \def\bxjs@apply@xkanjiskip{%
4128 \jsXKanjiSkip\@tempskipa
4129 \let\CJKecglue\bxjs@CJKecglue}
```

\jachar のサブマクロの実装。

```
4130 \def\bxjs@jachar#1{%
4131 \CJKforced{#1}}
```

和欧文間空白の命令 \jathinspace の実装。

```
4132 \ifbxjs@jaspace@cmd
4133 \protected\def\jathinspace{\CJKecglue}
4134 \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。

```
4135 \ifx t\bxjs@enablejfam
4136 \ClassWarningNoLine\bxjs@clsname
4137 {You cannot use 'enablejfam=true', since the\MessageBreak
4138 CJK package does not support Japanese math}
4139 \fi
```

C.6 X₃TEX 用設定 : xeCJK + zxjatype

```
4140 \else\if x\jsEngine
```

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

```
4141 \RequirePackage{zxjatype}
4142 \PassOptionsToPackage{no-math}{fontspec}%!
4143 \PassOptionsToPackage{xetex}{graphicx}%!
4144 \PassOptionsToPackage{xetex}{graphics}%!
4145 \ifx\zxJaFamilyName\@undefined
```

```

4146 \ClassError\bxjs@clsname
4147 {xeCJK or zxjatype is too old}\@ehc
4148 \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。

```

4149 \bxjs@adjust@jafont{f}
4150 \let\bxjs@jafont@paren\@gobble
4151 \bxjs@resolve@jafont@paren\bxjs@tmpa
4152 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4153 \ifx\bxjs@tmpa\@empty
4154 \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
Regular.otf}
4155 \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
Medium.otf}
4156 \else
4157 \edef\bxjs@next{%
4158 \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4159 }\bxjs@next
4160 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X_ƎTeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、X_ƎTeX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「X_ƎTeX 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。

```

4161 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4162 \ifbxjs@hyperref@enc
4163 \PassOptionsToPackage{unicode}{hyperref}
4164 \fi
4165 \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。


```

4166 \AtEndOfPackage{%
4167 \def\@gnewline #1{%
4168   \ifvmode \@nolnerr
4169   \else
4170     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4171     \nobreak \hskip-1sp\hskip1sp\relax
4172     \ignorespaces
4173   \fi}
4174 }

```

■ 共通命令の実装

```

4175 \newskip\jsKanjiSkip
4176 \newskip\jsXKanjiSkip
4177 \ifx\CJKecglue\@undefined
4178   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4179 \fi
4180 \let\autospacing\bxjs@enable@kanjiskip
4181 \let\noautospacing\bxjs@disable@kanjiskip
4182 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4183 \def\bxjs@apply@kanjiskip{%
4184   \jsKanjiSkip\@tempskipa
4185   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4186 \let\autoxspacing\bxjs@enable@xkanjiskip
4187 \let\noautoxspacing\bxjs@disable@xkanjiskip
4188 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4189 \def\bxjs@apply@xkanjiskip{%
4190   \jsXKanjiSkip\@tempskipa
4191   \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。

```

4192 \ifx\mcfamily\@undefined
4193   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4194   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4195 \fi

```

\jachar のサブマクロの実装。

```

4196 \def\bxjs@jachar#1{%
4197   \xeCJKDeclareCharClass{CJK}{`#1}\relax
4198   #1}

```

\jathinspace の実装。

```

4199 \ifbxjs@jaspace@cmd
4200   \protected\def\jathinspace{\CJKecglue}
4201 \fi

```

■ 和文数式ファミリ 和文数式ファミリは既定で無効とする。すなわち enablejfam=true の場合にのみ @enablejfam を真にする。

```

4202 \ifx t\bxjs@enablejfam

```

```
4203 \enablejfamtrue
4204 \fi
```

実際に和文用の数式ファミリの設定を行う。

※ FIXME: 要検討。

```
4205 \if@enablejfam
4206 \xeCJKsetup{CJKmath=true}
4207 \fi
```

C.7 Lua_TE_X 用設定 : Lua_TE_X-ja

```
4208 \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。

```
4209 \let\zw\@undefined
4210 \RequirePackage{luatexja}
4211 \edef\bxjs@next{%
4212 \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4213 }\bxjs@next
```

フォント代替の明示的定義。

```
4214 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4215 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4216 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4217 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4218 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4219 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4220 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4221 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4222 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4223 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4224 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4225 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4226 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4227 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4228 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4229 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4230 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4231 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4232 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4233 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4234 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
```

```

4235 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4236 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4237 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4238 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4239 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4240 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4241 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4242 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4243 \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。

```

4244 \bxjs@adjust@jafont{t}
4245 \ifx\bxjs@tmpa\bxjs@noEmbed
4246   \def\bxjs@tmpa{noembed}
4247 \fi
4248 \let\bxjs@jafont@paren\@gobble
4249 \bxjs@resolve@jafont@paren\bxjs@tmpa
4250 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4251 \ifx\bxjs@tmpa\@empty
4252   \defaultjfontfeatures{ Kerning=Off }
4253   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4254   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4255 \else
4256   \edef\bxjs@next{%
4257     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4258   }\bxjs@next
4259 \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。

```

4260 \@ifpackagelater{luatexja}{2016/03/31}{ }{\%else
4261 \DeclareRobustCommand\rmfamily
4262   {\not@math@alphabet\rmfamily\mathrm
4263     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4264 \DeclareRobustCommand\sffamily
4265   {\not@math@alphabet\sffamily\mathsf
4266     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4267 \DeclareRobustCommand\ttfamily
4268   {\not@math@alphabet\ttfamily\mathtt
4269     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4270 }
4271 \long\def\jttdefault{\gtdefault}
4272 \unless\ifx\@ltj@match@familytrue\@undefined

```

```

4273 \ltj@match@familytrue
4274 \fi
4275 \g@addto@macro\bxjs@begin@document@hook{%
4276 \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4277 \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4278 \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4279 \bxjs@if@sf@default{%
4280 \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

4281 % 次の 3 つは既定値の通り
4282 \ltjsetparameter{prebreakpenalty={`',10000}}
4283 \ltjsetparameter{postbreakpenalty={`",10000}}
4284 \ltjsetparameter{prebreakpenalty={`",10000}}
4285 \ltjsetparameter{jaxspmode={`!',1}}
4286 \ltjsetparameter{jaxspmode={`〒,2}}
4287 \ltjsetparameter{alxspmode={`+,3}}
4288 \ltjsetparameter{alxspmode={`%,3}}

```

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

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の `LuaTeX-jā` で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合に備えて同等のものを用意する。

```

4289 \ifx\ltjfakeparbegin\undefined
4290 \protected\def\ltjfakeparbegin{%
4291 \ifhmode
4292 \relax\directlua{%
4293 luatexja.jfmglue.create_beginpar_node()}}
4294 \fi}
4295 \fi

ltjs* クラスの定義と同等になるようにパッチを当てる。
4296 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4297 \begingroup
4298 \let\%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4299 \@gobble\if\def\bxjs@tmpa{\@{\everypar}\fi}}
4300 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4301 \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}\fi}}\fi
4302 \directlua{
4303 local function patchcmd(cs, code, from, to)
4304 tex.sprint(code:gsub(from:gsub("\%W", "\%\%\%0"), "\%0"..to)
4305 :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4306 end
4307 patchcmd(\@xsect, [[\meaning\@xsect]],
4308 \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4309 patchcmd(\@item, [[\meaning\@item]],
4310 \bxjs@tmpa, \@ltjfakeparbegin)}

```

```
4311 \endgroup
4312 \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。

```
4313 \ifbxjs@hyperref@enc
4314   \PassOptionsToPackage{unicode}{hyperref}
4315   \bxjs@check@hyperref@unicode{true}
4316 \fi
```

■共通命令の実装

```
4317 \protected\def\autospacing{%
4318   \ltjsetparameter{autospacing=true}}
4319 \protected\def\noautospacing{%
4320   \ltjsetparameter{autospacing=false}}
4321 \protected\def\autoxspacing{%
4322   \ltjsetparameter{autoxspacing=true}}
4323 \protected\def\noautoxspacing{%
4324   \ltjsetparameter{autoxspacing=false}}
4325 \def\bxjs@apply@kanjiskip{%
4326   \ltjsetparameter{kanjiskip={\@tempskipa}}}}
4327 \def\bxjs@apply@xkanjiskip{%
4328   \ltjsetparameter{xkanjiskip={\@tempskipa}}}}
```

\jachar のサブマクロの実装。

```
4329 \def\bxjs@jachar#1{%
4330   \ltjjachar`#1\relax}
```

\jathinspace の実装。

```
4331 \ifbxjs@jaspace@cmd
4332   \protected\def\jathinspace{%
4333     \hskip\ltjgetparameter{xkanjiskip}\relax}
4334 \fi
```

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

```
4335 \ifx f\bxjs@enablejfam
4336   \ClassWarningNoLine\bxjs@clsname
4337     {You cannot use 'enablejfam=false', since the\MessageBreak
4338      LuaTeX-jd always provides Japanese math families}
4339 \fi
```

C.8 共通処理 (2)

```
4340 \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

```

4341 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4342 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4343 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4344 \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バックの定義を行う。

```

4345 \ifx\mathmc\@undefined
4346   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4347   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4348 \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)。

```

4349 \ifbxjs@jaspace@cmd
4350   \bxjs@protected\def\bxjs@choice@jathinspace{%
4351     \relax\ifmmode \mskip\medmuskip
4352     \else \jathinspace\ignorespaces
4353   \fi}
4354   \jsAtEndOfClass{%
4355     \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4356     \else \def\>{\protect\bxjs@choice@jathinspace}%
4357   \fi}
4358 \fi

```

■和文・和欧文間空白の初期値

```

4359 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4360 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4361 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4362 \fi

```

以上で終わり。

```

4363 %</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。

```

4364 %<*modern>
4365 \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。

```
4366 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4367 \def\encodingdefault{T1}%
4368 \input{t1enc.def}%
4369 \fontencoding\encodingdefault\selectfont
4370 \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。

```
4371 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4372 \renewcommand{\rmdefault}{lmr}
4373 \renewcommand{\sfdefault}{lmss}
4374 \renewcommand{\ttdefault}{lmtt}
4375 \fi
```

大型演算子用の数式フォントの設定。

※`amsfonts` パッケージと同等にする。

```
4376 \DeclareFontShape{OMX}{cmex}{m}{n}{-}%
4377   <-7.5>cmex7<7.5-8.5>cmex8%
4378   <8.5-9.5>cmex9<9.5->cmex10}{-}%
4379 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
```

`amsmath` 読込時に上書きされるのを防ぐ。

```
4380 \def\cmex@opt{10}
```

D.2 fixltx2e 読込

※`fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```
4381 \ifx\@IncludeInRelease\@undefined
4382 \RequirePackage{fixltx2e}
4383 \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。

```
4384 \RequirePackage{bxjcsjkat}
```

D.4 完了

おしまい。

```
4385 %</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。

```
4386 %<*pandoc>
```

```
4387 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4388 \RequirePackage{bxjspandoc}
```

\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。)

```
4389 \@onlypreamble\bxjs@endpreamble@hook
```

```
4390 \let\bxjs@endpreamble@hook\@empty
```

```
4391 \ifjsWithTeX
```

```
4392 \RequirePackage{etoolbox}[2010/08/21]% v2.0
```

```
4393 \AtEndPreamble{\bxjs@endpreamble@hook}
```

```
4394 \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。

```
4395 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue
```

fix-strong オプションの処理。

```
4396 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
```

```
4397 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
```

```
4398 \define@key{bxjsPan}{fix-strong}[true]{%
```

```
4399 \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。

```
4400 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue
```

fix-code オプションの処理。

```
4401 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
```

```
4402 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
```

```
4403 \define@key{bxjsPan}{fix-code}[true]{%
```

```
4404 \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。

```
4405 \chardef\bxjs@jp@strong=0
```

`strong` オプションの処理。

```
4406 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
```

```
4407 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
```

```
4408 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }
```

```
4409 \define@key{bxjsPan}{strong}{%
```

```
4410 \bxjs@set@keyval{strong}{#1}{}}
```

実際の `japaram` の値を適用する。

```
4411 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}
```

```
4412 \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〉}` 特定のファイルの読み込みが `\@filewithoptions` で指示されて、しかもそのファイルが読み込み済である場合に、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。

```
4413 \@onlypreamble\bxjs@set@dupload@proc
```

```
4414 \def\bxjs@set@dupload@proc#1{%
```

```
4415 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
4416 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
4417 \def\bxjs@set@dupload@proc@a#1{%
```

```
4418 \@onlypreamble#1\def#1##1}
```

```
4419 \def\bxjs@unset@dupload@proc#1{%
```

```
4420 \expandafter\let\csname bxjs@dlp/#1\endcsname\@undefined}
```

`\@if@options` `\@if@options` の再定義。

```
4421 \@onlypreamble\bxjs@org@if@options
```

```
4422 \let\bxjs@org@if@options\@if@options
```

```
4423 \@onlypreamble\bxjs@org@reset@options
```

```
4424 \let\bxjs@org@reset@options\relax
```

```
4425 \def\@if@options#1#2#3{%
```

```
4426 \let\bxjs@next\@secondoftwo
```

```
4427 \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
```

```
4428 \ifx\bxjs@tmpa\bxjs@tmpb
```

```
4429 \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
```

```
4430 \let\bxjs@next\@firstoftwo \fi
```

```
4431 \fi
```

```
4432 \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
```

```
4433 \g@addto@macro\bxjs@begin@document@hook{%
```

```
4434 \let\@if@options\bxjs@org@if@options}
```

```
4435 \@onlypreamble\bxjs@do@dupload@proc
```

```

4436 \def\bxjs@do@dupload@proc#1#2#3{%
4437   \ifx\bxjs@org@reset@options\relax
4438     \let\bxjs@org@reset@options\@reset@options
4439   \fi
4440   \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4441   \def\@reset@options{%
4442     \let\@reset@options\bxjs@org@reset@options
4443     \@reset@options
4444     \bxjs@next{#3}}%
4445   \@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。

`\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。

```
4446 \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。

```
4447 \let\bxjs@babel@options\relax
```

■Polyglossia について つまり $\text{X}\text{\LaTeX}$ ・ $\text{Lua}\text{\TeX}$ の場合。

※この場合 `etoolbox` が使用可能になっている。

```
4448 \ifnum0\if x\jsEngine1\fi\if l\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。

```

4449 \pandocSkipLoadPackage{polyglossia}
4450 \bxjs@set@dupload@proc{polyglossia.sty}{%
4451   \bxjs@unset@dupload@proc{polyglossia.sty}%
4452   \ClassWarning\bxjs@clsname
4453     {Package polyglossia is required}%
4454   \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。

```
4455 \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。

```

4456   \ifblank{##2}{%
4457     \ClassWarning\bxjs@clsname
4458       {Main language is invalid, thus the fallback\MessageBreak
4459         definitions will be employed}%
4460     \bxjs@pandoc@polyglossia@ja

```

言語名が非空ならば、本来の polyglossia の処理を実行する。

```
4461   }{%else
4462     \ClassWarning{bxjs@clsname
4463       {Main language is valid (##2),\MessageBreak
4464         thus polyglossia will be loaded}}%
4465     \csundef{ver@polyglossia.sty}%
4466     \edef{bxjs@next}{%
4467       \noexpand\RequirePackage[ $\backslash$ bxjs@polyglossia@options]{polyglossia}[]%
4468     } $\backslash$ bxjs@next
4469     \setmainlanguage[##1]{##2}%
4470   }}}
```

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。

```
4471 \g@addto@macro{bxjs@endpreamble@hook}{%
4472   \ifx{bxjs@polyglossia@options}\relax
4473     \csundef{ver@polyglossia.sty}%
4474   \fi}
```

\backslash 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 \backslash setotherlanguage(s) を定義する。

```
4475 \onlypreamble{bxjs@pandoc@polyglossia@ja
4476 \def{bxjs@pandoc@polyglossia@ja}{%
4477   \renewcommand*{setmainlanguage}[2] [] {}%
4478   \newcommand*{setotherlanguage}[2] [] {}%
4479   \ifblank{##2}{-}{%else
4480     \cslet{##2}\@empty \cslet{end##2}\@empty
4481     \cslet{text##2}\@firstofone}}%
4482   \newcommand*{setotherlanguages}[2] [] {}%
4483   \@for{bxjs@tmpa:}={##2}\do{%
4484     \setotherlangauge{bxjs@tmpa}}}%

```

Polyglossia の読込済マークは外れるようにしておく。

```
4485   \let{bxjs@polyglossia@options}\relax}%
4486 \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。

```
4487 \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。

```
4488 \pandocSkipLoadPackage{babel}
4489 \bxjs@set@dupload@proc{babel.sty}{%
4490   \bxjs@unset@dupload@proc{babel.sty}%
4491   \ClassWarning{bxjs@clsname
4492     {Package babel is required}}%
```

パッケージオプションに言語名が空の `main=` があるかを調べる。ある場合は `lang=ja` 対策を実行する。

※`\bxjs@babel@options` には `main=` を除いたオプション列を格納する。

```

4493 \@tempswafalse \let\bxjs@babel@options\@empty
4494 \def\bxjs@tmpb{main=}%
4495 \@for\bxjs@tmpa:=#1\do{%
4496   \ifx\bxjs@tmpa\bxjs@tmpb \@tempswatrue
4497   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4498   \fi}%
4499 \if@tempswa
4500   \ClassWarning\bxjs@clsname
4501     {Main language is invalid, so fallback\MessageBreak
4502       definitions will be employed}%
4503   \bxjs@pandoc@polyglossia@ja

```

ない場合は、本来の `babel` の処理を実行する。

```

4504 \else
4505   \ClassWarning\bxjs@clsname
4506     {Main language is valid,\MessageBreak
4507       thus babel will be loaded}%
4508   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4509   \RequirePackage[#1]{babel}[]%
4510 \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。

```

4511 \g@addto@macro\bxjs@endpreamble@hook{%
4512   \ifx\bxjs@babel@options\relax
4513     \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4514   \fi}

```

`\bxjs@pandoc@babel@ja` Pandoc 側で `lang=ja` が指定されていた場合の処理。

```

4515 \@onlypreamble\bxjs@pandoc@babel@ja
4516 \def\bxjs@pandoc@babel@ja{%
4517   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4518   \edef\bxjs@next{%
4519     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4520   }\bxjs@next
4521   \if j\jsEngine
4522     \RequirePackage[main=japanese]{pxbabel}[]%
4523   \else
4524     \RequirePackage{bxorigcapt}[]%
4525   \fi}

```

`lang` 対策はこれで終わり。

```

4526 \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。

```
4527 \bxjs@set@dupload@proc{geometry.sty}{%
4528   \setpagelayout*{#1}}
```

E.6 CJKmainfont 変数

LuaTeX (+ LuaTeX-jā) の場合に CJKmainfont 変数が指定された場合は `\setmainfont` の指定にまわす。

```
4529 \if 1\jsEngine
4530   \pandocSkipLoadPackage{xeCJK}
4531   \providecommand*{\setCJKmainfont}{\setmainfont}
4532 \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。

```
4533 \if x\jsEngine
4534   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4535     ,space}
4536 \fi
```

E.8 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。

```
4537 \g@addto@macro\bxjs@begin@document@hook{%
4538   \@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。

```
4539   \ifx\oldparagraph\@undefined\else
4540     \@tempwatrue
```

```
4541 \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4542 \ifnum\c@secnumdepth>3
```

```
4543 \@tempwattrue
```

```
4544 \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。

```
4545 \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
```

```
4546 \let\jsParagraphMark\@empty
```

```
4547 \fi\fi}
```

E.9 全角空白文字

```
4548 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>\z@
```

```
4549 \catcode"3000=\active
```

```
4550 \begingroup \catcode`\!=7
```

```
4551 \protected\gdef!!!!3000{\zwspace}
```

```
4552 \endgroup
```

```
4553 \else\ifx\DeclareUnicodeCharacter\@undefined\else
```

```
4554 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
```

```
4555 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
```

```
4556 \fi\fi
```

E.10 hyperref 対策

`hyperref` の unicode オプションの固定を行う。

```
4557 \if j\jsEngine
```

```
4558 \bxjs@fix@hyperref@unicode{false}
```

```
4559 \else
```

```
4560 \bxjs@fix@hyperref@unicode{true}
```

```
4561 \fi
```

E.11 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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 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)。

```
4562 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
```

```
4563 \let\orgtextbf\textbf
```

```
4564 \DeclareRobustCommand\bxjstextbf[1]{%
```

```
4565 \begingroup
```

```
4566 \let\textbf\orgtextbf
```

```
4567 \strong{#1}%
```

```
4568 \endgroup}%
```

```
4569 \g@addto@macro\bxjs@begin@document@hook{%
```

```
4570 \let\textbf\bxjstextbf}
```

```
4571 \fi\fi
```

`\strong` の書体を設定する。

```
4572 \jsAtEndOfClass{%
4573   \ifx\strongfontdeclare\@undefined\else
4574     \ifcase\bxjs@jp@strong
4575       \or \strongfontdeclare{\sffamily}%
4576       \or \strongfontdeclare{\sffamily\bfseries}%
4577     \fi
4578   \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。

```
4579 \ifbxjs@jp@fix@code
4580   \chardef\bxjs@eghost@c=23
4581   \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4582   \else\ifx l\jsEngine \ltjsetparameter{alxspmode={\bxjs@eghost@c,3}}
4583   \else\ifx x\jsEngine %no-op
4584   \else \let\bxjs@eghost@c\@undefined
4585   \fi\fi\fi
4586   \ifx\bxjs@eghost@c\@undefined\else
4587     \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4588     \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4589     \let\orgtexttt\texttt
4590     \DeclareRobustCommand\bxjstexttt[1]{%
4591       \ifmmode \nfss@text{\ttfamily #1}%
4592       \else
4593         \ifvmode \leavevmode \fi
4594         \bxjs@pan@eghost\texttt
4595         \begingroup \ttfamily #1\endgroup
4596         \null\bxjs@pan@eghost
4597       \fi}
4598     \g@addto@macro\bxjs@begin@document@hook{%
4599       \let\texttt\bxjstexttt}
```

さらに `\verb` の外側にも欧文ゴーストが入るようにする。

※`bxjaghost` の実装を参考にした。

```
4600   \expandafter\def\expandafter\verb\expandafter{%
4601     \expandafter\bxjs@pan@eghost\verb}
4602   \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4603   \fi
4604 \fi
```

E.12 完了

おしまい。

```
4605 %</pandoc>
```

和文ドライバ実装はここまで。

```
4606 %</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。

```
4607 %<*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

```
4608 %<*compat>
```

```
4609 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
4610 \let\bxac@engine=n
```

```
4611 \def\bxac@do#1#2{%
```

```
4612   \edef\bxac@tmpa{\string#1}%
```

```
4613   \edef\bxac@tmpb{\meaning#1}%
```

```
4614   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
4615 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
4616 \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。

```
4617 \ifx\jsAtEndOfClass\undefined
```

```
4618   \let\bxac@delayed@if@bxjs\@firstofone
```

```
4619 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
4620 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4621 \newif\ifbxac@in@old@behavior
```

```
4622 \let\ImposeOldLuaTeXBehavior\relax
```

```
4623 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 X_YTEX 部分

```
4624 \ifx x\bxac@engine
```

X_YTEX 文字クラスのムニャムニャ。


```

4625 \@onlypreamble\bxac@adjust@charclass
4626 \bxac@delayed@if@bxjs{%
4627   \@ifpackageloaded{xeCJK}{-}{%else
4628     \ifx\XeTeXalloc@intercharclass\undefined\else
4629     \ifnum\XeTeXalloc@intercharclass=\z@
4630     \PackageInfo\bxac@pkgname
4631       {Setting up interchar class for CJK...\@gobble}%
4632     \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4633       \XeTeXalloc@intercharclass=3
4634     }{%else
4635       \PackageWarning\bxac@pkgname
4636         {Cannot find file 'load-unicode-xetex-classes.tex'%
4637         \@gobble}%
4638     }%
4639   \fi\fi
4640   \ifnum\XeTeXcharclass"3041=\z@
4641   \PackageInfo\bxac@pkgname
4642     {Adjusting interchar class for CJK...\@gobble}%
4643   \@for\bxac@x:={%
4644     3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4645     3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4646     30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4647     31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4648     31FF%
4649   }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4650   \fi
4651 }%
4652 }

```

以上。

```
4653 \fi
```

G.3 LuaTeX 部分

```

4654 \ifx l\bxac@engine
      ムニヤムニヤ。
4655 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4656   \chardef\pdfTeXversion=200
4657   \def\pdfTeXrevision{0}
4658   \let\pdfTeXbanner\luatexbanner
4659 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior
4660 \begingroup\expandafter\expandafter\expandafter\endgroup
4661 \expandafter\ifx\csname outputmode\endcsname\relax\else
4662 \def\bxac@ob@list{%
4663   \do{\let}\pdfoutput{\outputmode}%
4664   \do{\let}\pdfpagewidth{\pagewidth}%
4665   \do{\let}\pdfpageheight{\pageheight}%
4666   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%

```

```

4667 \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}}
4668 \def\bxac@ob@do#1#2{\begingroup
4669 \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4670 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4671 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4672 \else \let#3#1\relax \let#1\@undefined
4673 \fi}
4674 \protected\def\ImposeOldLuaTeXBehavior{%
4675 \unless\ifbxac@in@old@behavior
4676 \bxac@in@old@behaviortrue
4677 \let\do\bxac@ob@do \bxac@ob@list
4678 \fi}
4679 \protected\def\RevokeOldLuaTeXBehavior{%
4680 \ifbxac@in@old@behavior
4681 \bxac@in@old@behaviorfalse
4682 \let\do\bxac@ob@do \bxac@ob@list
4683 \fi}
4684 \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。

```

4685 \ifnum\luatexversion>64 \directlua{
4686   local function range(cs, ce, cc, ff)
4687     if ff or not tex.getcatcode(cs) == cc then
4688       local setcc = tex.setcatcode
4689       for c = cs, ce do setcc(c, cc) end
4690     end
4691   end
4692   range(0x3400, 0x4DB5, 11, false)
4693   range(0x4DB5, 0x4DBF, 11, true)
4694   range(0x4E00, 0x9FCC, 11, false)
4695   range(0x9FCD, 0x9FFF, 11, true)
4696   range(0xAC00, 0xD7A3, 11, false)
4697   range(0x20000, 0x2A6D6, 11, false)
4698   range(0x2A6D7, 0x2A6FF, 11, true)
4699   range(0x2A700, 0x2B734, 11, false)
4700   range(0x2B735, 0x2B73F, 11, true)
4701   range(0x2B740, 0x2B81D, 11, false)
4702   range(0x2B81E, 0x2B81F, 11, true)
4703   range(0x2B820, 0x2CEA1, 11, false)
4704   range(0x2CEA2, 0x2FFFD, 11, true)
4705 } \fi

```

以上。

```

4706 \fi

```

G.4 完了

おしまい。

```

4707 %</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

```
4708 %<*cjkcat>
4709 \def\bxjx@pkgname{bxjscjkat}
4710 \newcount\bxjx@cna
4711 \@onlypreamble\bxjx@tmpdo
4712 \@onlypreamble\bxjx@tmpdo@a
4713 \@onlypreamble\bxjx@tmpdo@b
```

\bxjx@engine エンジンの種別。

```
4714 \let\bxjx@engine=n
4715 \def\bxjx@tmpdo#1#2{%
4716   \edef\bxjx@tmpa{\string#1}%
4717   \edef\bxjx@tmpb{\meaning#1}%
4718   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4719 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4720 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4721 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4722 \bxjx@tmpdo\pdftexversion{\let\bxjx@engine=p}
4723 \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。

```
4724 \def\bxjx@tmpdo#1#2{%
4725   \if#1\bxjx@engine
4726     \@ifpackageloaded{#2}{\}%else
4727     \PackageError\bxjx@pkgname
4728       {Package '#2' must be loaded}%
4729       {Package loading is aborted.\MessageBreak\@ehc}%
4730     \endinput}
4731 \fi}
4732 \bxjx@tmpdo{p}{bxcjkatype}
4733 \bxjx@tmpdo{x}{xeCJK}
4734 \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。

```
4735 \ifx\TextOrMath\@undefined
4736   \RequirePackage{fixltx2e}
4737 \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-jā と（ほぼ）等価なものに変更する。

※ LuaTeX-jā との相違点：A830、A960、1B000。

```
4738 \if u\bxjx@engine
4739 \@for\bxjx@tmpa:={%
4740 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4741 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4742 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4743 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4744 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4745 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4746 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4747 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4748 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4749 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4750 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4751 FE70,FFF0,%
4752 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4753 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4754 10480,104B0,10500,10530,10600,10800,10840,10860,%
4755 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4756 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4757 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4758 11200,11280,112B0,11300,11400,11480,11580,11600,%
4759 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4760 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4761 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4762 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4763 1EE00,1F000,1F030,1FOA0,1F300,1F600,1F650,1F680,%
4764 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4765 00C0%
4766 }\do{%
4767 \@tempcnta="\bxjx@tmpa\relax
4768 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4769 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4770 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4771 \fi
```

H.3 ギリシャ・キリル文字の扱い

「特定 CJK 曖昧文字」について、和文・欧文扱いを制御できるようにする。ここで「特定 CJK 曖昧文字」とは以下に該当する文字の集合を指す：

- Unicode と JIS X 0213 に共通して含まれるギリシャ文字・キリル文字。

- Latin-1 の上位部分と JIS X 0208 に共通して含まれる文字 (LuaT_EX-j_a の定める“範囲 8”)。

`\bxjx@grkcyr@list` 「特定 CJK 曖昧文字」に関する情報をもつ `\do-` リスト。各項目の形式は以下の通り：

`\do{⟨Unicode 符号値⟩}{⟨対象 fontenc⟩}{⟨テキスト LICR⟩}{⟨数式 LICR⟩}`

※数式で使わない文字は `⟨数式 LICR⟩` を空にする。

```

4772 \@onlypreamble\bxjx@grkcyr@list
4773 \def\bxjx@grkcyr@list{%
4774 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4775 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4776 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4777 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4778 \do{0395}{LGR}{\textEpsilon}{E}%       % GR. C. L. EPSILON
4779 \do{0396}{LGR}{\textZeta}{Z}%          % GR. C. L. ZETA
4780 \do{0397}{LGR}{\textEta}{H}%           % GR. C. L. ETA
4781 \do{0398}{LGR}{\textTheta}{\Theta}%    % GR. C. L. THETA
4782 \do{0399}{LGR}{\textIota}{I}%          % GR. C. L. IOTA
4783 \do{039A}{LGR}{\textKappa}{K}%         % GR. C. L. KAPPA
4784 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4785 \do{039C}{LGR}{\textMu}{M}%           % GR. C. L. MU
4786 \do{039D}{LGR}{\textNu}{N}%          % GR. C. L. NU
4787 \do{039E}{LGR}{\textXi}{\Xi}%         % GR. C. L. XI
4788 \do{039F}{LGR}{\textOmicron}{O}%     % GR. C. L. OMICRON
4789 \do{03A0}{LGR}{\textPi}{\Pi}%         % GR. C. L. PI
4790 \do{03A1}{LGR}{\textRho}{P}%          % GR. C. L. RHO
4791 \do{03A3}{LGR}{\textSigma}{\Sigma}%    % GR. C. L. SIGMA
4792 \do{03A4}{LGR}{\textTau}{T}%          % GR. C. L. TAU
4793 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4794 \do{03A6}{LGR}{\textPhi}{\Phi}%       % GR. C. L. PHI
4795 \do{03A7}{LGR}{\textChi}{X}%         % GR. C. L. CHI
4796 \do{03A8}{LGR}{\textPsi}{\Psi}%       % GR. C. L. PSI
4797 \do{03A9}{LGR}{\textOmega}{\Omega}%   % GR. C. L. OMEGA
4798 \do{03B1}{LGR}{\textalpha}{\alpha}%   % GR. S. L. ALPHA
4799 \do{03B2}{LGR}{\textbeta}{\beta}%     % GR. S. L. BETA
4800 \do{03B3}{LGR}{\textgamma}{\gamma}%   % GR. S. L. GAMMA
4801 \do{03B4}{LGR}{\textdelta}{\delta}%   % GR. S. L. DELTA
4802 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
4803 \do{03B6}{LGR}{\textzeta}{\zeta}%     % GR. S. L. ZETA
4804 \do{03B7}{LGR}{\texteta}{\eta}%       % GR. S. L. ETA
4805 \do{03B8}{LGR}{\texttheta}{\theta}%   % GR. S. L. THETA
4806 \do{03B9}{LGR}{\textiota}{\iota}%    % GR. S. L. IOTA
4807 \do{03BA}{LGR}{\textkappa}{\kappa}%   % GR. S. L. KAPPA
4808 \do{03BB}{LGR}{\textlambda}{\lambda}% % GR. S. L. LAMDA
4809 \do{03BC}{LGR}{\textmu}{\mu}%        % GR. S. L. MU
4810 \do{03BD}{LGR}{\textnu}{\nu}%        % GR. S. L. NU
4811 \do{03BE}{LGR}{\textxi}{\xi}%        % GR. S. L. XI
4812 \do{03BF}{LGR}{\textomicron}{o}%       % GR. S. L. OMICRON
4813 \do{03C0}{LGR}{\textpi}{\pi}%        % GR. S. L. PI

```

4814 \do{03C1}{LGR}{\textrho}{\rho}% % GR. S. L. RHO
 4815 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
 4816 \do{03C3}{LGR}{\textsigma}{\sigma}% % GR. S. L. SIGMA
 4817 \do{03C4}{LGR}{\texttau}{\tau}% % GR. S. L. TAU
 4818 \do{03C5}{LGR}{\textupsilon}{\upsilon}% % GR. S. L. UPSILON
 4819 \do{03C6}{LGR}{\textphi}{\phi}% % GR. S. L. PHI
 4820 \do{03C7}{LGR}{\textchi}{\chi}% % GR. S. L. CHI
 4821 \do{03C8}{LGR}{\textpsi}{\psi}% % GR. S. L. PSI
 4822 \do{03C9}{LGR}{\textomega}{\omega}% % GR. S. L. OMEGA
 4823 \do{0401}{T2A}{\CYRYO}{}% % CY. C. L. IO
 4824 \do{0410}{T2A}{\CYRA}{}% % CY. C. L. A
 4825 \do{0411}{T2A}{\CYRB}{}% % CY. C. L. BE
 4826 \do{0412}{T2A}{\CYRV}{}% % CY. C. L. VE
 4827 \do{0413}{T2A}{\CYRG}{}% % CY. C. L. GHE
 4828 \do{0414}{T2A}{\CYRD}{}% % CY. C. L. DE
 4829 \do{0415}{T2A}{\CYRE}{}% % CY. C. L. IE
 4830 \do{0416}{T2A}{\CYRZH}{}% % CY. C. L. ZHE
 4831 \do{0417}{T2A}{\CYRZ}{}% % CY. C. L. ZE
 4832 \do{0418}{T2A}{\CYRI}{}% % CY. C. L. I
 4833 \do{0419}{T2A}{\CYRISHRT}{}% % CY. C. L. SHORT I
 4834 \do{041A}{T2A}{\CYRK}{}% % CY. C. L. KA
 4835 \do{041B}{T2A}{\CYRL}{}% % CY. C. L. EL
 4836 \do{041C}{T2A}{\CYRM}{}% % CY. C. L. EM
 4837 \do{041D}{T2A}{\CYRN}{}% % CY. C. L. EN
 4838 \do{041E}{T2A}{\CYRO}{}% % CY. C. L. O
 4839 \do{041F}{T2A}{\CYRP}{}% % CY. C. L. PE
 4840 \do{0420}{T2A}{\CYRR}{}% % CY. C. L. ER
 4841 \do{0421}{T2A}{\CYRS}{}% % CY. C. L. ES
 4842 \do{0422}{T2A}{\CYRT}{}% % CY. C. L. TE
 4843 \do{0423}{T2A}{\CYRU}{}% % CY. C. L. U
 4844 \do{0424}{T2A}{\CYRF}{}% % CY. C. L. EF
 4845 \do{0425}{T2A}{\CYRH}{}% % CY. C. L. HA
 4846 \do{0426}{T2A}{\CYRC}{}% % CY. C. L. TSE
 4847 \do{0427}{T2A}{\CYRCH}{}% % CY. C. L. CHE
 4848 \do{0428}{T2A}{\CYRSH}{}% % CY. C. L. SHA
 4849 \do{0429}{T2A}{\CYRSHCH}{}% % CY. C. L. SHCHA
 4850 \do{042A}{T2A}{\CYRHRDSN}{}% % CY. C. L. HARD SIGN
 4851 \do{042B}{T2A}{\CYRERY}{}% % CY. C. L. YERU
 4852 \do{042C}{T2A}{\CYRSFTSN}{}% % CY. C. L. SOFT SIGN
 4853 \do{042D}{T2A}{\CYREREV}{}% % CY. C. L. E
 4854 \do{042E}{T2A}{\CYRYU}{}% % CY. C. L. YU
 4855 \do{042F}{T2A}{\CYRYA}{}% % CY. C. L. YA
 4856 \do{0430}{T2A}{\cyra}{}% % CY. S. L. A
 4857 \do{0431}{T2A}{\cyrb}{}% % CY. S. L. BE
 4858 \do{0432}{T2A}{\cyrv}{}% % CY. S. L. VE
 4859 \do{0433}{T2A}{\cyrg}{}% % CY. S. L. GHE
 4860 \do{0434}{T2A}{\cyrd}{}% % CY. S. L. DE
 4861 \do{0435}{T2A}{\cyre}{}% % CY. S. L. IE
 4862 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE

```

4863 \do{0437}{T2A}{\cyrz}{}% % CY. S. L. ZE
4864 \do{0438}{T2A}{\cyri}{}% % CY. S. L. I
4865 \do{0439}{T2A}{\cyrishrt}{}% % CY. S. L. SHORT I
4866 \do{043A}{T2A}{\cyrk}{}% % CY. S. L. KA
4867 \do{043B}{T2A}{\cyr1}{}% % CY. S. L. EL
4868 \do{043C}{T2A}{\cyrn}{}% % CY. S. L. EM
4869 \do{043D}{T2A}{\cyrn}{}% % CY. S. L. EN
4870 \do{043E}{T2A}{\cyro}{}% % CY. S. L. O
4871 \do{043F}{T2A}{\cyrp}{}% % CY. S. L. PE
4872 \do{0440}{T2A}{\cyrr}{}% % CY. S. L. ER
4873 \do{0441}{T2A}{\cyrs}{}% % CY. S. L. ES
4874 \do{0442}{T2A}{\cyr1}{}% % CY. S. L. TE
4875 \do{0443}{T2A}{\cyr1}{}% % CY. S. L. U
4876 \do{0444}{T2A}{\cyr1}{}% % CY. S. L. EF
4877 \do{0445}{T2A}{\cyrh}{}% % CY. S. L. HA
4878 \do{0446}{T2A}{\cyr1}{}% % CY. S. L. TSE
4879 \do{0447}{T2A}{\cyrch}{}% % CY. S. L. CHE
4880 \do{0448}{T2A}{\cyrsh}{}% % CY. S. L. SHA
4881 \do{0449}{T2A}{\cyrshch}{}% % CY. S. L. SHCHA
4882 \do{044A}{T2A}{\cyrhrdsn}{}% % CY. S. L. HARD SIGN
4883 \do{044B}{T2A}{\cyrery}{}% % CY. S. L. YERU
4884 \do{044C}{T2A}{\cyr1}{}% % CY. S. L. SOFT SIGN
4885 \do{044D}{T2A}{\cyrerev}{}% % CY. S. L. E
4886 \do{044E}{T2A}{\cyr1}{}% % CY. S. L. YU
4887 \do{044F}{T2A}{\cyr1}{}% % CY. S. L. YA
4888 \do{0451}{T2A}{\cyr1}{}% % CY. S. L. IO
4889 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4890 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4891 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4892 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN
4893 \do{00B4}{TS1}{\textasciicircum}{}% % ACUTE ACCENT
4894 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4895 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4896 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN
4897 }

```

`\mathdegree` 面倒なので補っておく。

```
4898 \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。

```
4899 \newif\ifbxjx@gcc@CJK
```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```
4900 \newcommand*\greekasCJK{%
```

```
4901 \bxjx@gcc@CJKtrue}
```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```
4902 \newcommand*\nogreekasCJK{%
```

```
4903 \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。

```

4904 \def\bxjx@tmpdo#1\relax{%
4905   \def\bxjx@fake@grk##1##2{%
4906     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4907   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4908     \ifx\\##1\\%
4909       \bxjx@ccta##4\divide\bxjx@ccta\@cclvi
4910       \multiply\bxjx@ccta\@cclvi \advance\bxjx@ccta`##3\relax
4911       \mathchar\bxjx@ccta
4912     \else ##3\fi}
4913 }\expandafter\bxjx@tmpdo\string\mathchar\relax

```

■pdfTeX・upTeX の場合

```

4914 \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。

```

4915 \@ifpackageloaded{inputenc}{\fi}{%else
4916   \RequirePackage[utf8]{inputenc}}
4917 \def\bxjx@tmpa{utf8}
4918 \ifx\bxjx@tmpa\inputencdoingname
4919   \PackageWarningNoLine\bxjx@pkgname
4920     {Input encoding changed to utf8}%
4921   \inputencoding{utf8}%
4922 \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。

```

4923 \if u\bxjx@engine
4924   \kcatcode"0370=15
4925   \kcatcode"0400=15
4926   \kcatcode"0500=15
4927 \fi

```

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

```

4928 \def\bxjx@tmpdo#1{%
4929   \@tempccta="#1\relax

```



```

4930 \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4931 \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。

```

4932 \ifx\#5\%
4933 \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。

```

4934 \else\ifcat A\noexpand#5%
4935 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4936 {\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。

```

4937 \else \def\bxjx@tmpa{#5}%
4938 \fi\fi
4939 \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4940 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upT_EX の場合。

```

4941 \if u\bxjx@engine
4942 \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。

```

4943 \kchardef#1=\@tempcnta
4944 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4945 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdfT_EX の場合も処理はほとんど同じ。ただし、和文用の定義として \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。

```

4946 \else\if p\bxjx@engine
4947 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4948 \mathchardef#1=\@tempcnta
4949 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4950 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4951 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

4952 \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。

```

4953 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter

```

```

4954 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4955 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4956 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4957   \count@=#1\relax
4958   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4959     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4960   \else
4961     \wlog{ \space\space skipped defining Unicode char U+#1}%
4962   \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。

```

4963 \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。

```

4964   \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。

```

4965   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4966   \else \UseTextSymbol{#2}{#3}%
4967   \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。

```

4968 \begingroup
4969 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4970 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4971   \noexpand\bxjx@swap@DUC@cmd
4972   \the\toks@
4973   \noexpand\bxjx@swap@DUC@cmd}}
4974 \endgroup\next
4975 \def\bxjx@swap@DUC@cmd{%
4976   \let\bxjx@tmpa\DeclareUnicodeCharacter
4977   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4978   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
4979   \let\bxjx@tmpa\relax}

```

以上。

■ $\mathrm{X\TeX}$ ・ $\mathrm{Lua\TeX}$ の場合

```

4980 \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。

```

4981 \def\bxjx@tmpdo#1{%
4982   \bxjx@cmta=#1\relax
4983   \begingroup
4984     \lccode`~=\bxjx@cmta
4985   \lowercase{\endgroup

```

```

4986 \bxjx@tmpdo@a{~}{#1}}
4987 \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。

```

4988 \ifx\#5\\\let\bxjx@tmpa\relax
4989 \else\ifcat A\noexpand#5%
4990 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4991 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4992 \else \def\bxjx@tmpa{#5}%
4993 \fi\fi
4994 \ifx\bxjx@tmpa\relax\else
4995 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4996 \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。

```

4997 \mathchardef\bxjx@tmpa="119
4998 \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。

```

4999 \if 1\bxjx@engine
5000 \protected\def\greekasCJK{%
5001 \bxjx@gcc@cjctrue
5002 \ltjsetparameter{jacharrange={+2, +8}}
5003 \protected\def\nogreekasCJK{%
5004 \bxjx@gcc@cjcfalse
5005 \ltjsetparameter{jacharrange={-2, -8}}
5006 \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。

```

5007 \if x\bxjx@engine
5008 \let\bxjx@gcc@cjkl@list\@empty
5009 \def\do#1#2#3#4{%
5010 \edef\bxjx@gcc@cjkl@list{\bxjx@gcc@cjkl@list
5011 \noexpand\XeTeXcharclass"#1\bxjx@cmta}}
5012 \bxjx@grkcyr@list
5013 \protected\def\greekasCJK{%
5014 \bxjx@gcc@cjctrue
5015 \bxjx@cmta=\@ne \bxjx@gcc@cjkl@list}
5016 \protected\def\nogreekasCJK{%
5017 \bxjx@gcc@cjcfalse
5018 \bxjx@cmta=\z@ \bxjx@gcc@cjkl@list}
5019 \fi

```

以上。

```
5020 \fi\fi
```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5021 \nogreekasCJK
```

H.5 完了

おしまい。

```
5022 %</cjkat>
```

付録 I 補助パッケージ : bxjspanc

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

```
5023 %<*ancpandoc>
```

```
5024 %% このファイルは日本語文字を含みます。
```

```
5025 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```
5026 \let\bxjsp@engine=n
```

```
5027 \@onlypreamble\bxjsp@do
```

```
5028 \def\bxjsp@do#1#2{%
```

```
5029   \edef\bxjsp@tmpa{\string#1}%
```

```
5030   \edef\bxjsp@tmpb{\meaning#1}%
```

```
5031   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
```

```
5032 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
```

```
5033 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
```

```
5034 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
```

```
5035 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

I.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。BXJS クラスの英語モードを想定しているが、それ以外の場合でも、一般的な L^AT_EX の

習慣として、グローバルの `english` は「その文書の基底言語が英語である」ことを示す。

```
5036 \newif\ifbxjsp@english
5037 \DeclareOption{english}{\bxjsp@englishttrue}

    オプション定義はおしまい。
5038 \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。

```
5039 \@onlypreamble\pandocSkipLoadFile
5040 \newcommand*\pandocSkipLoadFile[1]{%
5041   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5042 \def\bxjsp@skip@load@file@a#1#2{%
5043   \ifx#1\relax
5044     \def#1{2001/01/01}%
5045     \PackageInfo\bxjsp@pkgname
5046       {File '#2' marked as loaded\@gobble}%
5047   \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。

```
5048 \@onlypreamble\pandocSkipLoadPackage
5049 \newcommand*\pandocSkipLoadPackage[1]{%
5050   \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 \LaTeX ではこれで警告が出る。これを抑止する。

\LaTeX カーネルが新しい場合は `fixltx2e` を読込済にする。

```
5051 \ifx\@IncludeInRelease\@undefined\else
5052   \pandocSkipLoadPackage{fixltx2e}
5053 \fi
```

1.5 cmap パッケージ

エンジンが (u)p \LaTeX のときに `cmap` パッケージが読み込まれるのを阻止する。(実際は警告が出るだけで無害であるが。)

```
5054 \if j\bxjsp@engine
5055   \pandocSkipLoadPackage{cmap}
5056 \fi
```

I.6 microtype パッケージ

警告が多すぎなので消す。

```
5057 \if j\bxjsp@engine \else
5058   \PassOptionsToPackage{verbose=silent}{microtype}
5059 \fi
```

エンジンが (u)p \LaTeX 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。

```
5060 \if j\bxjsp@engine
5061   \pandocSkipLoadPackage{microtype}
5062   \newcommand*\UseMicrotypeSet[2][]{\fi}
5063 \fi
```

I.7 Unicode 文字変換対策

Pandoc で \LaTeX 形式に書き出す場合は、元データ中の一部の Unicode 文字を「 \LaTeX の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

…→ \backslash ldots{ } ‘→` ’→' “→` ”→''

日本語 \LaTeX では「 \LaTeX の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「 \backslash ldots を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の \backslash ldots の実装。非数式である場合は代わりに … を実行する。

※以前は「Pandoc が必ず \backslash 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 \backslash ldots を全て … に戻す動作に変更した。

```
5064 \DeclareRobustCommand{\pandocLdots}{%
5065   \relax\ifmmode \expandafter\bxjsp@org@ldots
5066   \else \expandafter\bxjsp@ja@ellipsis
5067   \fi}
5068 \def\bxjsp@ja@ellipsis{…}
5069 \let\bxjsp@org@ldots\ldots
```

\backslash ldots の実装を置き換える。

※ Benglish オプション指定時は置き換えない。

```
5070 \AtBeginDocument{%
5071   \let\bxjsp@org@ldots\ldots
5072   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}
```

`\ldots` の直後の文字が非英字の場合、Pandoc は「`\ldots。`」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが $\text{Xe}\text{L}\text{a}\text{T}\text{E}\text{X}$ ・ $\text{L}\text{u}\text{a}\text{T}\text{E}\text{X}$ 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。

```

5073 \chardef\bxjsp@cc@other=12
5074 \@onlypreamble\bxjsp@makeother@range
5075 \def\bxjsp@makeother@range#1#2{%
5076   \@tempcnta"#1\relax \@tempcntb"#2\relax
5077   \loop\ifnum\@tempcnta<\@tempcntb
5078     \catcode\@tempcnta\bxjsp@cc@other
5079     \advance\@tempcnta\@ne
5080   \repeat}
5081 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
5082   \catcode"1F23B=\bxjsp@cc@other
5083   \bxjsp@makeother@range{9FCD}{A000}
5084   \bxjsp@makeother@range{1B002}{1B170}
5085   \bxjsp@makeother@range{2B820}{2EBF0}
5086 \fi

```

1.8 PandoLa モジュール

インストール済であれば読み込む。

```

5087 \IfFileExists{bxpandola.sty}{%
5088   \RequirePackage{bxpandola}\relax
5089   \PackageInfo\bxjsp@pkgname
5090   {PandoLa module is loaded\@gobble}
5091 }{}

```

1.9 完了

おしまい。

```
5092 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
5093 %</anc>
```