

# 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”)

v2.8 [2023/06/14]

---

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

---

## 目次

|     |                            |    |
|-----|----------------------------|----|
| 1   | はじめに                       | 3  |
| 2   | オプション                      | 11 |
| 3   | 和文フォントの変更                  | 41 |
| 4   | フォントサイズ                    | 42 |
| 5   | レイアウト                      | 48 |
| 5.1 | ページレイアウト                   | 49 |
| 6   | 改ページ (日本語 TeX 開発コミュニティ版のみ) | 64 |
| 7   | ページスタイル                    | 65 |
| 8   | 文書のマークアップ                  | 68 |
| 8.1 | 表題                         | 68 |
| 8.2 | 章・節                        | 74 |
| 8.3 | リスト環境                      | 86 |
| 8.4 | パラメータの設定                   | 93 |
| 8.5 | フロート                       | 95 |
| 8.6 | キャプション                     | 96 |
| 9   | フォントコマンド                   | 97 |

|             |   |            |
|-------------|---|------------|
| <b>10</b>   | <b>相互参照</b>                             | <b>100</b> |
| 10.1        | 目次の類                                    | 100        |
| 10.2        | 参考文献                                    | 105        |
| 10.3        | 索引                                      | 107        |
| 10.4        | 脚注                                      | 108        |
| <b>11</b>   | <b>段落の頭へのグルー挿入禁止</b>                    | <b>111</b> |
| <b>12</b>   | <b>いろいろなロゴ</b>                          | <b>115</b> |
| <b>13</b>   | <b>amsmath との衝突の回避</b>                  | <b>115</b> |
| <b>14</b>   | <b>初期設定</b>                             | <b>116</b> |
| <b>付録 A</b> | <b>和文ドライバの仕様</b> 🍱                      | <b>120</b> |
| <b>付録 B</b> | <b>和文ドライバ：minimal</b> 🍱                 | <b>121</b> |
| B.1         | 補助マクロ                                   | 121        |
| B.2         | (u)pTeX 用の設定                            | 124        |
| B.3         | pdfTeX 用の処理                             | 128        |
| B.4         | X <sub>Y</sub> TeX 用の処理                 | 129        |
| B.5         | 後処理（エンジン共通）                             | 130        |
| <b>付録 C</b> | <b>和文ドライバ：standard</b> 🍱                | <b>133</b> |
| C.1         | 準備                                      | 133        |
| C.2         | 和文ドライバパラメタ                              | 133        |
| C.3         | 共通処理 (1)                                | 134        |
| C.4         | pTeX 用設定                                | 142        |
| C.5         | pdfTeX 用設定：CJK + bxcjkatype             | 147        |
| C.6         | X <sub>Y</sub> TeX 用設定：xeCJK + zxjatype | 149        |
| C.7         | LuaTeX 用設定：LuaTeX-ja                    | 151        |
| C.8         | 共通処理 (2)                                | 155        |
| <b>付録 D</b> | <b>和文ドライバ：modern</b> 🍱                  | <b>156</b> |
| D.1         | フォント設定                                  | 156        |
| D.2         | fixltx2e 読込                             | 157        |
| D.3         | 和文カテゴリコード                               | 157        |
| D.4         | 完了                                      | 157        |
| <b>付録 E</b> | <b>和文ドライバ：pandoc</b> 🍱                  | <b>157</b> |
| E.1         | 準備                                      | 157        |
| E.2         | 和文ドライバパラメタ                              | 158        |
| E.3         | dupload システム                            | 159        |

|             |                             |            |
|-------------|-----------------------------|------------|
| E.4         | lang 変数                     | 160        |
| E.5         | geometry 変数                 | 164        |
| E.6         | CJKmainfont 変数              | 164        |
| E.7         | Option clash 対策             | 164        |
| E.8         | レイアウト上書き禁止                  | 164        |
| E.9         | paragraph のマーク              | 165        |
| E.10        | 全角空白文字                      | 166        |
| E.11        | hyperref 対策                 | 167        |
| E.12        | Pandoc 要素に対する和文用の補正         | 167        |
| E.13        | ifPDFTeX スイッチ               | 168        |
| E.14        | 完了                          | 169        |
| <b>付録 F</b> | <b>補助パッケージ一覧</b> 📦          | <b>169</b> |
| <b>付録 G</b> | <b>補助パッケージ：bxjscompat</b> 📦 | <b>170</b> |
| G.1         | 準備                          | 170        |
| G.2         | X <sub>Y</sub> TeX 部分       | 170        |
| G.3         | LuaTeX 部分                   | 172        |
| G.4         | 完了                          | 173        |
| <b>付録 H</b> | <b>補助パッケージ：bxjscjkat</b> 📦  | <b>173</b> |
| H.1         | 準備                          | 173        |
| H.2         | 和文カテゴリコードの設定                | 174        |
| H.3         | ギリシャ・キリル文字の扱い               | 175        |
| H.4         | 初期設定                        | 182        |
| H.5         | 完了                          | 182        |
| <b>付録 I</b> | <b>補助パッケージ：bxjspandoc</b> 📦 | <b>182</b> |
| I.1         | 準備                          | 182        |
| I.2         | パッケージオプション                  | 183        |
| I.3         | パッケージ読込の阻止                  | 183        |
| I.4         | fixltx2e パッケージ              | 184        |
| I.5         | cmap パッケージ                  | 184        |
| I.6         | microtype パッケージ             | 184        |
| I.7         | Unicode 文字変換対策              | 185        |
| I.8         | PandoLa モジュール               | 186        |
| I.9         | 完了                          | 186        |

## 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>&lt;article&gt;</code> | <code>bxjsarticle.cls</code> | 短いレポート（章なし） |
| <code>&lt;report&gt;</code>  | <code>bxjsreport.cls</code>  | 長いレポート（章あり） |
| <code>&lt;book&gt;</code>    | <code>bxjsbook.cls</code>    | 書籍用         |
| <code>&lt;slide&gt;</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<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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<sup>A</sup>T<sub>E</sub>X3 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<sub>E</sub>X 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<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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>&lt;article&gt;</code> | <code>jsarticle.cls</code> | 論文・レポート用 |
| <code>&lt;book&gt;</code>    | <code>jsbook.cls</code>    | 書籍用      |
| <code>&lt;report&gt;</code>  | <code>jsreport.cls</code>  | レポート用    |
| <code>&lt;jspf&gt;</code>    | <code>jspf.cls</code>      | 某学会誌用    |
| <code>&lt;kiyou&gt;</code>   | <code>kiyou.cls</code>     | 某紀要用     |

以下では実際のコードに即して説明します。

minijs は、jsclasses に似た設定を行うパッケージです。

```
1 %<*minijs>
2 %% if jsclasses loaded, abort loading this package
3 \ifx\@jsc@uplatrixtrue\undefined\else
4   \PackageInfo{minijs}{jsclasses does not need minijs, exiting}
5   \expandafter\endinput
6 \fi
7 %% "fake" jsarticle
8 \expandafter\def\csname ver@jsarticle.cls\endcsname{}
9 %</minijs>
```

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

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

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

```
16 %</class>
17 %<*class|minijs>
18 \newif\ifjsc@needsp@tch
19 \jsc@needsp@tchfalse
20 %</class|minijs>
21 %<*class>
```

## ■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。

```
22 \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。

```
23 \RequirePackage{keyval}
```

クラスの本体ではこの他に以下のパッケージが読み込まれる。

```
geometry、iftex
```

また状況によっては以下のパッケージが読み込まれる可能性がある。

```
bxwareki、jslogo、plautopatch、type1cm
```

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

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

互換性のための補助パッケージを読み込む。

26 \IfFileExists{bxjscompat.sty}{%
27   \RequirePackage{bxjscompat}%
28 }{}
```

`\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`。

```

29 \let\jsArticle=a
30 \let\jsBook=b
31 \let\jsReport=r
32 \let\jsSlide=s
33 %<article>\let\jsDocClass\jsArticle
34 %<book>\let\jsDocClass\jsBook
35 %<report>\let\jsDocClass\jsReport
36 %<slide>\let\jsDocClass\jsSlide

```

`\jsEngine` [暗黙文字トークン] エンジン (T<sub>E</sub>X の種類) の種別: `j` = pT<sub>E</sub>X 系、`x` = X<sub>Ǝ</sub>T<sub>E</sub>X、`p` = pdfT<sub>E</sub>X (含 DVI モード)、`l` = LuaT<sub>E</sub>X、`J` = NTT jT<sub>E</sub>X、`0` = Omega 系、`n` = 以上の何れでもない。

```

37 \let\jsEngine=n
38 \def\bxjs@test@engine#1#2{%
39   \edef\bxjs@tmpa{\string#1}%
40   \edef\bxjs@tmpb{\meaning#1}%
41   \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
42 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
43 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
44 \bxjs@test@engine\Omegaversion{\let\jsEngine=0}
45 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
46 \bxjs@test@engine\pdfTeXversion{\let\jsEngine=p}
47 \bxjs@test@engine\luatexversion{\let\jsEngine=l}

```

`\ifjsWithupTeX` [スイッチ] エンジンが (内部漢字コードが Unicode の) upT<sub>E</sub>X であるか。

```

48 \newif\ifjsWithupTeX
49 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
50   \jsWithupTeXtrue
51 \fi\fi
52 \let\if@jsc@uplatex\ifjsWithupTeX

```

`\ifjsWithpTeXng` [スイッチ] エンジンが pT<sub>E</sub>X-ng であるか。

```

53 \newif\ifjsWithpTeXng
54 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}

```

`\ifjsWitheTeX` [スイッチ] エンジンが  $\epsilon$ -T<sub>E</sub>X 拡張をもつか。

```

55 \newif\ifjsWitheTeX
56 \bxjs@test@engine\epsilonTeXversion{\jsWitheTeXtrue}

```

非サポートのエンジンの場合は強制終了させる。

※ NTT jT<sub>E</sub>X と Omega 系。

```

57 \let\bxjs@tmpa\relax
58 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
59 \ifx 0\jsEngine \def\bxjs@tmpa{Omega}\fi
60 \ifx\bxjs@tmpa\relax \expandafter\@gobble
61 \else
62   \ClassError\bxjs@clsname
63   {The engine in use (\bxjs@tmpa) is not supported}

```

```

64   {It's a fatal error. I'll quit right now.}
65   \expandafter\@firstofone
66 \fi{\endinput\@@end}

```

LuaTeX の場合、本クラス用の Lua モジュールを用意する。

```

67 \ifx 1\jsEngine
68   \directlua{ bxjs = {} }
69 \fi

```

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

```

70 \ifjsWithTeX \let\bxjs@protected\protected
71 \else \let\bxjs@protected\@empty
72 \fi

```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

```

73 \ifjsWithTeX
74   \def\bxjs@robust@def{\protected\def}
75 \else
76   \def\bxjs@robust@def{\DeclareRobustCommand*}
77 \fi

```

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

```

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

```

`\ifbxjs@TUenc` [スイッチ] L<sup>A</sup>T<sub>E</sub>X の既定のフォントエンコーディングが TU であるか。

※ 2017 年 1 月以降の L<sup>A</sup>T<sub>E</sub>X カーネルにおいて「Unicode を表す L<sup>A</sup>T<sub>E</sub>X 公式のフォントエンコーディング」である “TU” が導入され、これ以降の L<sup>A</sup>T<sub>E</sub>X を Xe<sub>L</sub>TeX 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。

```

80 \newif\ifbxjs@TUenc
81 \def\bxjs@tmpa{TU}\edef\bxjs@tmpb{\f@encoding}
82 \ifx\bxjs@tmpa\bxjs@tmpb
83   \bxjs@TUenctrue
84 \fi

```

`\ifbxjs@old@hook@system` [スイッチ] L<sup>A</sup>T<sub>E</sub>X の新しいフック管理システムが未導入であるか。

※カーネルの 2020/10/01 版で導入された。

```

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

```

`\bxjs@CGHN` L<sup>A</sup>T<sub>E</sub>X カーネルの 2021/11/15 版の改修で「要素の順が変わった」フック名について、“新仕様において正しい名前”を“使用中の L<sup>A</sup>T<sub>E</sub>X において正しい名前”に変換する。例えば、`\bxjs@CGHN{package/PKG/after}` は旧仕様の L<sup>A</sup>T<sub>E</sub>X では “package/after/PKG” に展開される。

```

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

```

```

89 }{%else
90   \def\bxjs@CGHN#1{\bxjs@CGHN@a#1//}%
91   \def\bxjs@CGHN@a#1/#2/#3/{#1/#3/#2}}

\bxjs@cond \bxjs@cond\ifXXX……\fi{\langle真\rangle}\fi{\langle偽\rangle}
  TeX の if-文 (\ifXXX……\else\fi) を末尾呼出形式に変換するためのマクロ。
92 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
93   #1\expandafter\@firstoftwo
94   \else\expandafter\@secondoftwo
95   \fi}

\bxjs@cslet \bxjs@cslet{\langle名前 1\rangle}\制御綴：
96 \def\bxjs@cslet#1{%
97   \expandafter\let\csname#1\endcsname}

\bxjs@csletcs \bxjs@csletcs{\langle名前 1\rangle}\langle名前 2\rangle：
98 \def\bxjs@csletcs#1#2{%
99   \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}

\ifjsInPdfMode [スイッチ] pdfTeX / LuaTeX が PDF モードで動作しているか。
100 \RequirePackage{iftex}
101 \newif\ifjsInPdfMode
102 \@nameuse{jsInPdfMode\ifnum0%
103   \ifx\pdfoutput\undefined\else\the\pdfoutput\fi
104   \ifx\outputmode\undefined\else\the\outputmode\fi
105   >0 true\else false\fi}
106 \ifx\pdffalse\undefined\else \bxjs@csletcs{ifjsInPdfMode}{ifpdf}\fi

\bxjs@catopt \bxjs@catopt{\langle文字列 1\rangle}\langle文字列 2\rangle： 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。
107 \def\bxjs@catopt#1#2{%
108   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

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

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

```



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

※各要素について、先頭・末尾の空白トークン群は除去される。

```
119 \def\bxjs@set@array@from@clist#1#2{%
120   \@tempcnta\z@
121   \@for\bxjs@tmpa:=\@empty#2\do{%
122     \bxjs@trim\bxjs@tmpa \bxjs@cslet{#1/\the\@tempcnta}\bxjs@tmpa
123     \advance\@tempcnta\@ne}
124   \bxjs@cslet{#1/\the\@tempcnta}\relax}
```

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

```
125 \let\c@bxjs@tempcnta\@tempcnta
126 \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）。

```
127 \def\jsSetQHLlength#1#2{%
128   \begingroup
129   \bxjs@parse@qh{#2}%
130   \ifx\bxjs@tmpb\relax
131     \setlength\@tempdima{#2}%
132     \xdef\bxjs@g@tmpa{\the\@tempdima}%
133   \else \global\let\bxjs@g@tmpa\bxjs@tmpb
134   \fi
135   \endgroup
136   #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<sup>A</sup>T<sub>E</sub>X 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。

```
137 \if j\jsEngine \def\bxjs@parse@qh@units{zw,zh}
138 \else \def\bxjs@parse@qh@units{trueQ,trueH,Q,H,zw,zh}
139 \fi
140 \def\bxjs@parse@qh#1{%
141   \let\bxjs@tmpb\relax
142   \@for\bxjs@tmpa:=\bxjs@parse@qh@units\do{%
143     \ifx\bxjs@tmpb\relax
144       \edef\bxjs@next{{\bxjs@tmpa}{#1}}%
145       \expandafter\bxjs@parse@qh@a\cename bxjs@unit@\bxjs@tmpa\expandafter
146       \endcsname\bxjs@next
147     \fi}}
148 \def\bxjs@parse@qh@a#1#2#3{%
149   \def\bxjs@next##1#2\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}#1}%
150   \bxjs@next#3\@nil#2\@nil\@nnil}
```

```

151 \def\bxjs@parse@qh@b#1#2#3{%
152   \ifx\@nnil#2\@nnil\else
153     \ifx#3\relax
154       \ClassError\bxjs@clsname
155       {You cannot use '\bxjs@tmpa' here}{\@ehc}%
156       \def\bxjs@tmpb{0pt}%
157     \else
158       \@tempdimb#3\relax \@tempdimb#1\@tempdimb
159       \edef\bxjs@tmpb{\the\@tempdimb}%
160     \fi
161   \fi}

```

今の段階では Q/H だけが使用可能。

```

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

```

\ifbxjs@after@preamble [スイッチ] 文書本体が開始しているか。

```

163 \newif\ifbxjs@after@preamble

```

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

```

164 \@onlypreamble\bxjs@begin@document@hook
165 \def\bxjs@begin@document@hook{\bxjs@after@preambletrue}
166 \AtBeginDocument{\bxjs@begin@document@hook}

```

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

```

167 \@onlypreamble\bxjs@post@option@hook
168 \let\bxjs@post@option@hook\empty

```

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

```

169 \@onlypreamble\bxjs@pre@jadriver@hook
170 \let\bxjs@pre@jadriver@hook\empty

```

一時的な手続き用の制御綴。

```

171 \@onlypreamble\bxjs@tmpdo
172 \@onlypreamble\bxjs@tmpdo@a
173 \@onlypreamble\bxjs@tmpdo@b
174 \@onlypreamble\bxjs@tmpdo@c
175 \@onlypreamble\bxjs@tmpdo@d

```

\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。

```

176 \bxjs@robust@def\jsInhibitGlue{%
177   \ifx\inhibitglue\undefined\else \inhibitglue \fi}

```

■環境検査 🐛 T<sub>E</sub>X 処理系のバージョンがサポート対象であるかを検査する。

※現状での処理系バージョン要件は「X<sub>Ǝ</sub>T<sub>E</sub>X は 0.997 版（2007 年頃）以上」という現実離れたものになっている。

**TODO:** 3.0 版において、もっと現実的なバージョン要件を定める予定。多分「本体」と「標準和文ドライバ」で条件を分けることになる。

```

178 \@tempswatruue
179 \if x\jsEngine
180 \ifdim\the\XeTeXversion\XeTeXrevision\p@<0.997\p@
181 \@tempswafalse \fi
182 \fi

```

非サポートのバージョン場合は強制終了させる。

```

183 \if@tempswa \expandafter\@gobble
184 \else
185 \ClassError\bxjs@clsname
186 {The engine in use is all too old}
187 {It's a fatal error. I'll quit right now.}
188 \expandafter\@firstofone
189 \fi{\endinput\@@end}

```

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。

```

190 \if@compatibility
191 \ClassError\bxjs@clsname
192 {Something went chaotic!\MessageBreak
193 (How come '\string\documentstyle' is there?)\MessageBreak
194 I cannot go a single step further...}
195 {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
196 then there'll still be hope....}
197 \expandafter\@firstofone
198 \else \expandafter\@gobble
199 \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。

```
200 \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。

```
201 \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。

```
202 %<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。

```
203 %<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。

---

```
204 %<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。

---

```
205 \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  $1\text{ m}^2$ 、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  $1.5\text{ m}^2$  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、 $\text{\LaTeX} 2_{\epsilon}$  の `b5paper` は  $250\text{ mm} \times 176\text{ mm}$  ですが、 $\text{p}\text{\LaTeX} 2_{\epsilon}$  の `b5paper` は  $257\text{ mm} \times 182\text{ mm}$  になっています。ここでは  $\text{p}\text{\LaTeX} 2_{\epsilon}$  にならって JIS に従いました。

デフォルトは `a4paper` です。

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

---

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

```
206 \@onlypreamble\bxjs@setpaper
207 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
208 \newif\ifbxjs@iso@bsize
209 \DeclareOption{iso-bsize}{\bxjs@iso@bsizetrue}
210 \@onlypreamble\bxjs@setpaper@bsize
211 \def\bxjs@setpaper@bsize#1{\edef\bxjs@param@paper{%
212   b#1\ifbxjs@iso@bsize paper\else j\fi}}
213 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
214 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
215 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
216 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
217 \DeclareOption{b4paper}{\bxjs@setpaper@bsize{4}}
218 \DeclareOption{b5paper}{\bxjs@setpaper@bsize{5}}
219 \DeclareOption{b6paper}{\bxjs@setpaper@bsize{6}}
220 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
221 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
222 \DeclareOption{b4j}{\bxjs@setpaper{b4j}}
```

```

223 \DeclareOption{b5j}{\bxjs@setpaper{b5j}}
224 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
225 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
226 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
227 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
228 \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。

```

229 \@for\bxjs@tmpa:={%
230   a0,a1,a2,c0,c1,c2,c3,c4,c5,c6,ansia,ansib,ansic,ansid,ansie%
231 }\do{\edef\bxjs@next{%
232   \noexpand\DeclareOption{\bxjs@tmpa paper}%
233   {\noexpand\bxjs@setpaper{\bxjs@tmpa paper}}}%
234 }\bxjs@next}
235 \DeclareOption{screen}{\bxjs@setpaper{screen}}

```

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

```

236 \@for\bxjs@tmpa:={0,1,2,3}\do{\edef\bxjs@next{%
237   \noexpand\DeclareOption{b\bxjs@tmpa paper}%
238   {\noexpand\bxjs@setpaper@bsize{\bxjs@tmpa}}}%
239 }\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。

```

240 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
241 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
242 \DeclareOption{screenpaper}{\bxjs@setpaper{screen}}
243 \DeclareOption{custompaper}{\}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

244 \newif\if@landscape
245 \@landscapefalse
246 \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もので、簡単のため article のみで使えるオプションとしました。

```

247 \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。

```
248 %<!slide>\@slidefalse
249 %<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` の定義が変わったのでご迷惑をおかけしましたが, 標準的なドキュメントクラスと同様にポイント数から 10 を引いたものに直しました。

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

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

```
250 \newcommand{\@ptsize}{0}
251 \def\bxjs@param@basefontsize{10pt}
252 \def\jsUnusualPtSize{-20}
```

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

```
253 \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、 $\text{\LaTeX}$  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。

```
254 \jsSetQHLength\@tempdima{#1}%
255 \edef\bxjs@param@basefontsize{\the\@tempdima}%
256 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
257 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
258 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
259 \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。

```
260 \def\bxjs@setjbasefontsize#1{%
261 \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。

```
262 \newif\ifjsc@mag
263 \newif\ifjsc@mag@xreal
264 %\let\jsc@magscale\undefined
```

```

265 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
266 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
267 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
268 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
269 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
270 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
271 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
272 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
273 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
274 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
275 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
276 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
277 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
278 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
279 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
280 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
281 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
282 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
283 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

284 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@@usemag}
285 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@@nomag}
286 \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<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 2<sub>ε</sub> 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<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 2<sub>ε</sub> 本体で宣言されています。

---

取りあえず、p<sub>T</sub>E<sub>X</sub> 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。

---

```

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

```

```

300 \setlength{\@tombowwidth}{.1\p@}%
301 \maketombowbox}
302 \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。

```

303 \if j\jsEngine
304 \DeclareOption{mentuke}{%
305 \tombowtrue \tombowdatefalse
306 \setlength{\@tombowwidth}{\z@}%
307 \maketombowbox}
308 \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。

```

309 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
310 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
311 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 `twocolumn` で二段組になります。

```

312 \DeclareOption{onecolumn}{\@twocolumnfalse}
313 \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。

```

314 \DeclareOption{titlepage}{\@titlepagetrue}
315 \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  $\text{\LaTeX}$  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。

```

316 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
317 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
318 %<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` (*env.*)  $\text{\LaTeX}$  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。

[2022-09-13]  $\text{\LaTeX}$  2<sub>ε</sub> 2021-11-15 (`ltmath.dtx` 2021/10/14 v1.2j) で `\@currentcounter` が追加されましたので、追従します。



```

319 \def\eqnarray{%
320   \stepcounter{equation}%
321   \def\@currentlabel{\p@equation\theequation}%
322   \def\@currentcounter{equation}%
323   \global\@eqnswtrue
324   \m@th
325   \global\@eqcnt\z@
326   \tabskip\@centering
327   \let\\\@eqncr
328   $$\everycr{}\halign to\displaywidth\bgroup
329     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse
330     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
331     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
332     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
333     \tabskip\z@skip
334   \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。

[2022-09-13] L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 2021-11-15 (l<sup>A</sup>T<sub>E</sub>Math.dtx 2021/10/14 v1.2j) で`\@currentcounter` が追加されましたので、追従します。

```

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

```

362     }}

■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。

```
363 % \DeclareOption{openbib}{%
364 %   \AtEndOfPackage{%
365 %     \renewcommand\@openbib@code{%
366 %       \advance\leftmargin\bibindent
367 %       \itemindent -\bibindent
368 %       \listparindent \itemindent
369 %       \parsep \z@}%
370 %     \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：

```
371 %\let\bxjs@enablejfam\@undefined

enablejfam オプションの処理。

372 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
373 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
374 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\@undefined}
375 \define@key{bxjs}{enablejfam}[true]{%
376   \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。

```
377 \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。

```
378 \newif\ifjsDraft
379 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
380 \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` などでも出力出来るようになる) が使えます。

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

```
381 \newif\ifbxjs@papersize
382 \bxjs@papersizetrue
383 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
384 \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。

---

```
385 \newif\if@english
386 \@englishfalse
387 \DeclareOption{english}{\@englishtrue}
```

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

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

---

BXJS では当初から `bxjsreport` クラスが用意されている。

---

■jslogo パッケージの読み込み IAT<sub>E</sub>X 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。

---

```
388 \newif\if@jslogo \@jslogofalse
389 \DeclareOption{jslogo}{\@jslogotrue}
390 \DeclareOption{nojslogo}{\@jslogofalse}
```

#### ■複合設定オプション

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

`\bxjs@invscale` `\bxjs@invscale` は T<sub>E</sub>X における「長さのスケール」の逆関数を求めるもの。例えば `\bxjs@invscale\dimX{1.3}` は `\dimX=1.3\dimX` の逆の演算を行う。  
※局所化の `\begingroup~\endgroup` について、以前は `\group~\egroup` を使っていたが、これだと数式モード中では空のサブ数式を生み出してしまうため修正した。  
※元の長さが 128 pt 以上の場合でも動作するように修正した。

```
391 \mathchardef\bxjs@isc@ll=128
392 \mathchardef\bxjs@isc@sl=259
393 \def\bxjs@isc@sl@h{65539 }
394 \def\bxjs@invscale#1#2{%
395   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
396   \ifdim\@tempdima<\bxjs@isc@ll\p@
397     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
398     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
399   \else
400     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
401     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
402   \fi
403   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
404   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
405   \@tempdimb\@tempcnta\@ne
406   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
407   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
408   \@whiledim\@tempdimb<\@tempdimc\do{%
409     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
410     \advance\@tempcntb\@ne \divide\@tempcntb\tw@
411   \ifdim #2\@tempcntb>\@tempdima
412     \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
```

```

413 \else \@tempdimb=\@tempcntb\@ne \fi}%
414 \xdef\bxjs@gtmpa{\the\@tempdimb}%
415 \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性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  $\text{\LaTeX}$  用の既定テンプレートを用いて他形式から  $\text{\LaTeX}$  (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。

```

416 \DeclareOption{pandoc}{%
417 \bxjs@apply@pandoc@opt}
418 \@onlypreamble\bxjs@apply@pandoc@opt
419 \def\bxjs@apply@pandoc@opt{%

```

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

※実際の和文ドライバ・エンジン設定より優先される。

```

420 \g@addto@macro\bxjs@post@option@hook{%
421 \bxjs@oldfontcommandstrue
422 \setkeys{bxjs}{ja=pandoc}%
423 \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)。

```

424 \ifx\bxjs@driver@opt\@undefined
425 \def\bxjs@driver@opt{dvipdfmx}%
426 \bxjs@dvi@opttrue
427 \fi
428 \global\let\bxjs@apply@pandoc@opt\relax}

```

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

```

429 \DeclareOption{pandoc+}{%
430 \g@addto@macro\bxjs@post@option@hook{%
431 \edef\jsJaParam{\bxjs@catopt{_plus}\jsJaParam}}%
432 \ExecuteOptions{pandoc}}

```

---

## ■エンジン・ドライバオプション

---

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

```
433 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
434 %\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<sup>A</sup>T<sub>E</sub>X の慣習に従って「ドライバの指定」とみなすべきだから。

```

435 \DeclareOption{autodetect-engine}{%
436   \let\bxjs@engine@given=*}
437 \DeclareOption{latex}{%
438   \def\bxjs@engine@opt{latex}%
439   \let\bxjs@engine@given=n}
440 \DeclareOption{platex}{%
441   \def\bxjs@engine@opt{platex}%
442   \let\bxjs@engine@given=j}
443 \DeclareOption{uplatex}{%
444   \def\bxjs@engine@opt{uplatex}%
445   \let\bxjs@engine@given=u}
446 \DeclareOption{xelatex}{%
447   \def\bxjs@engine@opt{xelatex}%
448   \let\bxjs@engine@given=x}
449 \DeclareOption{pdflatex}{%
450   \def\bxjs@engine@opt{pdflatex}%
451   \let\bxjs@engine@given=p}
452 \DeclareOption{lualatex}{%
453   \def\bxjs@engine@opt{lualatex}%
454   \let\bxjs@engine@given=l}
455 \DeclareOption{platex-ng}{%
456   \def\bxjs@engine@opt{platex-ng}%
457   \let\bxjs@engine@given=g}
458 \DeclareOption{platex-ng*}{%
459   \def\bxjs@engine@opt{platex-ng*}%
460   \let\bxjs@platexng@nodrv=t%
461   \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。

```

462 %\let\bxjs@driver@given\@undefined
463 \let\bxjs@driver@@dvimode=0
464 \let\bxjs@driver@@dvipdfmx=1
465 \let\bxjs@driver@@pdfmode=2
466 \let\bxjs@driver@@xetex=3
467 \let\bxjs@driver@@dvips=4
468 \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。

```

469 %\let\bxjs@driver@opt\@undefined

```

※ `nodvidriver*` は BXJS クラスの仕様上は `nodvidriver` と完全に等価であるが、「グローバルオプションに何があるか」の点で異なる。

```

470 \DeclareOption{dvips}{%
471   \def\bxjs@driver@opt{dvips}%
472   \let\bxjs@driver@given\bxjs@driver@@dvips}
473 \DeclareOption{dviout}{%

```

```

474 \def\bxjs@driver@opt{dviout}%
475 \let\bxjs@driver@given\bxjs@driver@@dvimode}
476 \DeclareOption{xdvi}{%
477 \def\bxjs@driver@opt{xdvi}%
478 \let\bxjs@driver@given\bxjs@driver@@dvimode}
479 \DeclareOption{dvipdfmx}{%
480 \def\bxjs@driver@opt{dvipdfmx}%
481 \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
482 \DeclareOption{nodvidriver}{%
483 \def\bxjs@driver@opt{nodvidriver}%
484 \let\bxjs@driver@given\bxjs@driver@@none}
485 \DeclareOption{nodvidriver*}{%
486 \def\bxjs@driver@opt{nodvidriver*}%
487 \let\bxjs@driver@given\bxjs@driver@@none}
488 \DeclareOption{pdftex}{%
489 \def\bxjs@driver@opt{pdftex}%
490 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
491 \DeclareOption{luatex}{%
492 \def\bxjs@driver@opt{luatex}%
493 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
494 \DeclareOption{xetex}{%
495 \def\bxjs@driver@opt{xetex}%
496 \let\bxjs@driver@given\bxjs@driver@@xetex}

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

```

---

## ■その他の BXJS 独自オプション

---

**TODO:** 互換用オプションを分離する (2.0 版で?)。

`\bxjs@depre@opt` 非推奨のオプションについて警告を出す。

```

\bxjs@depre@opt@do 498 \@onlypreamble\bxjs@depre@opt
499 \def\bxjs@depre@opt#1#2{%
500 \ClassWarningNoLine\bxjs@clsname
501 {The old option '#1' is DEPRECATED\MessageBreak
502 and may be abolished in future!\MessageBreak
503 You should instead write:\MessageBreak
504 \space\space #2}}
505 \@onlypreamble\bxjs@depre@opt@do
506 \def\bxjs@depre@opt@do#1#2{%
507 \bxjs@depre@opt{#1}{#2}%
508 \setkeys{bxjs}{#2}}

```

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

指定することとする。

※ 2.0 版より、既定値を常に真とする。

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

nobigcode / bigcode オプションの定義。

```
510 \DeclareOption{nobigcode}{%
```

```
511   \bxjs@bigcodefalse}
```

```
512 \DeclareOption{bigcode}{%
```

```
513   \bxjs@bigcodetrue}
```

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

```
514 \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。

```
515 \DeclareOption{nooldfontcommands}{%
```

```
516   \bxjs@oldfontcommandsfalse}
```

```
517 \DeclareOption{oldfontcommands}{%
```

```
518   \bxjs@oldfontcommandstrue}
```

---

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

```
519 \DeclareOption{winjis}{%
```

```
520   \ClassWarningNoLine\bxjs@clsname
```

```
521   {This class does not support `winjis' option}%
```

```
522 }
```

```
523 \DeclareOption{mingoth}{%
```

```
524   \ClassWarningNoLine\bxjs@clsname
```

```
525   {This class does not support `mingoth' option}%
```

```
526 }
```

```
527 \DeclareOption{jis}{%
```

```
528   \ClassWarningNoLine\bxjs@clsname
```

```
529   {This class does not support `jis' option}%
```

```
530 }
```

■ 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。

```
531 \DeclareOption*{%
```

```
532   \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
```

```
533   \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。

```
534 \def\bxjs@safe@setkeys#1#2{%
```

```
535   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
```



```

536 \setkeys{#1}{#2}%
537 \let\KV@errx\bxjs@save@KV@errx}

```

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

“<オプション名>=<値>” のオプション指定に対して、`\[bxjs@<enum 名>]` 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。

```

538 \@onlypreamble\bxjs@declare@enum@option
539 \def\bxjs@declare@enum@option#1#2#3{%
540 \bxjs@csletcs{bxjs@#2}{bxjs@#2@#3}%
541 \define@key{bxjs}{#1}{%
542 \expandafter\ifx\csname bxjs@#2@#1\endcsname\relax
543 \bxjs@error@keyval{#1}{##1}%
544 \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@#1}%
545 \fi}}

```

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

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

```

546 \@onlypreamble\bxjs@declare@bool@option
547 \def\bxjs@declare@bool@option#1#2#3{%
548 \csname newif\expandafter\endcsname\csname ifbxjs@#2\endcsname
549 \@nameuse{bxjs@#2#3}%
550 \define@key{bxjs}{#1}[true]{%
551 \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
552 \bxjs@error@keyval{#1}{##1}%
553 \else \@nameuse{bxjs@#2##1}%
554 \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。

```

555 \def\bxjs@set@keyval#1#2#3{%
556 \bxjs@csletcs{bxjs@next}{bxjs@kv@#1@#2}%
557 \ifx\bxjs@next\relax
558 \bxjs@error@keyval{#1}{#2}%
559 #3%
560 \else \bxjs@next
561 \fi}
562 \@onlypreamble\bxjs@error@keyval
563 \def\bxjs@error@keyval#1#2{%
564 \ClassError\bxjs@clsname
565 {Invalid value '#2' for option #1}\@ehc}

```

`\jsScale` [実数値マクロ] 和文スケール値。

```

566 \def\jsScale{0.924715}

```

`\bxjs@base@opt` 明示された base オプションの値。

```

567 %\let\bxjs@base@opt\@undefined

```

base オプションの処理。

```
568 \define@key{bxjs}{base}{%
569   \edef\bxjs@base@opt{#1}%
570   \bxjs@setbasefontsize{#1}}
571 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}
```

\bxjs@jbase@opt 明示された jbase オプションの値。

```
572 %\let\bxjs@jbase@opt\@undefined
```

jbase オプションの処理。

```
573 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
574 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}
```

\bxjs@scale@opt 明示された scale オプションの値。

```
575 %\let\bxjs@scale@opt\@undefined
```

scale オプションの処理。

```
576 \define@key{bxjs}{scale}{%
577   \edef\bxjs@scale@opt{#1}%
578   \let\jsScale\bxjs@scale@opt}
579 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}
```

noscale オプションの処理。

**TODO:** noscale は 3.0 版で廃止の予定。

```
580 \DeclareOption{noscale}{\bxjs@depre@opt@do{noscale}{scale=1}}
```

\bxjs@param@mag mag オプションの値。

```
581 \let\bxjs@param@mag\relax
```

mag オプションの処理。

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

paper オプションの処理。

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

\bxjs@jadriver 和文ドライバの名前。

```
584 \let\bxjs@jadriver\relax
585 %\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。

**TODO:** jadriver は 3.0 版で廃止の予定。

※単なる ja という指定は無視される (Pandoc 対策)。

```
586 \define@key{bxjs}{jadriver}{%
587   \bxjs@depre@opt{jadriver}{ja=#1}\edef\bxjs@jadriver@opt{#1}}
588 \define@key{bxjs}{ja}{\relax}{%
589   \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}
```

`\jsJaFont` 和文フォント設定の名前。

```
590 \let\jsJaFont\@empty
```

`jafont` オプションの処理。

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

`\jsJaParam` 和文ドライバパラメタの文字列。

```
592 \let\jsJaParam\@empty
```

`japaram` オプションの処理。

```
593 \define@key{bxjs}{japaram}{%
```

```
594 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}
```

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

```
595 \define@key{bxjs}{pandoc}[]{%
```

```
596 \ExecuteOptions{pandoc}%
```

```
597 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}
```

```
598 \define@key{bxjs}{pandoc+}[]{%
```

```
599 \ExecuteOptions{pandoc+}%
```

```
600 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}
```

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

```
601 \let\bxjs@magstyle@@mag=m
```

```
602 \let\bxjs@magstyle@@real=r
```

```
603 \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。

```
604 \let\bxjs@magstyle@@usemag\bxjs@magstyle@@mag
```

```
605 \let\bxjs@magstyle@@nomag\bxjs@magstyle@@real
```

```
606 \bxjs@cslet{bxjs@magstyle@@nomag*}\bxjs@magstyle@@xreal
```

`\bxjs@magstyle@@default` は既定の値を表す。

```
607 \let\bxjs@magstyle@@default\bxjs@magstyle@@usemag
```

```
608 \ifx l\jsEngine \ifnum\luatexversion>86
```

```
609 \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
```

```
610 \fi\fi
```

```
611 \ifjsWithpTeXng
```

```
612 \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
```

```
613 \fi
```

```
614 \let\bxjs@magstyle\bxjs@magstyle@@default
```

`magstyle` オプションの処理。

```
615 \define@key{bxjs}{magstyle}{%
```

```
616 \bxjs@csletcs{bxjs@magstyle}{bxjs@magstyle@@#1}%
```

```
617 \ifx\bxjs@magstyle\relax
```

```
618 \bxjs@error@keyval{magstyle}{#1}%
```

```
619 \let\bxjs@magstyle\bxjs@magstyle@@default
```

```
620 \fi}
```

`\bxjs@geometry geometry` オプションの指定値。

```
621 \let\bxjs@geometry@@class=c
622 \let\bxjs@geometry@@user=u
623 \bxjs@declare@enum@option{geometry}{geometry}{class}
```

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

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

`\ifbxjs@dvi@opt` `dvi` オプションが指定されたか。

```
625 \newif\ifbxjs@dvi@opt
```

DVI モードのドライバとドライバ種別との対応。

```
626 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
627 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
628 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
629 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
630 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none
631 \bxjs@cslet{bxjs@dvidriver@@nodvidriver*}\bxjs@driver@@none
```

`dvi` オプションの処理。

```
632 \define@key{bxjs}{dvi}{%
633   \bxjs@csletcs{bxjs@tmpa}{bxjs@dvidriver@@#1}%
634   \ifx\bxjs@tmpa\relax
635     \bxjs@error@keyval{dvi}{#1}%
636   \else
```

`\bxjs@driver@given` を未定義にしていることに注意。

```
637   \def\bxjs@driver@opt{#1}%
638   \let\bxjs@driver@given\undefined
639   \bxjs@dvi@opttrue
640 \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。

```
641 \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。

```
642 \newif\ifbxjs@force@chapterabstract
643 %<book>\bxjs@force@chapterabstracttrue
```

`layout` オプションの処理。

```
644 \@namedef{bxjs@kv@layout@v1}{%
645 %<book>\bxjs@layout@buggyhmargintrue
646 %<report>\bxjs@force@chapterabstracttrue
647 }
648 \@namedef{bxjs@kv@layout@v2}{%
649 %<book>\bxjs@layout@buggyhmarginfalse
650 %<report>\bxjs@force@chapterabstractfalse
```

```

651 }
652 \define@key{bxjs}{layout}{%
653   \bxjs@set@keyval{layout}{#1}{}}

```

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

```

654 %\let\bxjs@textwidth@limit@opt\@undefined
655 \define@key{bxjs}{textwidth-limit}{%
656   \bxjs@depre@opt{textwidth-limit}{textwidth=#1zw}%
657   \edef\bxjs@textwidth@limit@opt{#1}}

```

`\bxjs@textwidth@opt` textwidth の指定値。

```

658 %\let\bxjs@textwidth@opt\@undefined
659 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
660 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

```

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

```

661 %\let\bxjs@number@of@lines@opt\@undefined
662 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
663 \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。

```

664 %\let\bxjs@paragraph@mark\@undefined
665 \define@key{bxjs}{paragraph-mark}{%
666   \edef\bxjs@paragraph@mark{#1}}

```

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

```

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

```

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

```

668 \bxjs@declare@bool@option{jaspacercmd}{jaspacercmd}{true}
669 \define@key{bxjs}{xkanjiskip-cmd}[true]{\setkeys{bxjs}{jaspacercmd=#1}}

```

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

```

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

```

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

```

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

```

`\bxjs@everyparhook` everyparhook の指定値。

```

672 \chardef\bxjs@everyparhook@none=0
673 \chardef\bxjs@everyparhook@compat=1
674 \chardef\bxjs@everyparhook@modern=2
675 \bxjs@declare@enum@option{everyparhook}{everyparhook}{%
676   \if j\jsEngine compat\else modern\fi}

```

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

```

677 \chardef\bxjs@label@section@none=0
678 \chardef\bxjs@label@section@compat=1
679 \chardef\bxjs@label@section@modern=2
680 \bxjs@declare@enum@option{label-section}{label@section}{compat}

```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

**TODO:** `zw/nozw` は 3.0 版で廃止の予定。

```
681 \bxjs@declare@bool@option{use-zw}{usezw}{true}
682 \DeclareOption{nozw}{\bxjs@depre@opt@do{nozw}{use-zw=false}}
683 \DeclareOption{zw}{\bxjs@depre@opt@do{zw}{use-zw=true}}
```

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

**TODO:** `js/nojs` は 3.0 版で廃止の予定。

```
684 \bxjs@declare@bool@option{disguise-js}{disguise@js}{true}
685 \DeclareOption{nojs}{\bxjs@depre@opt@do{nojs}{disguise-js=false}}
686 \DeclareOption{js}{\bxjs@depre@opt@do{js}{disguise-js=true}}
```

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

```
687 \bxjs@declare@bool@option{precise-text}{precisetext}{false}
688 \DeclareOption{noprecisetext}{\bxjs@depre@opt@do{noprecisetext}{precise-
  text=false}}
689 \DeclareOption{precisetext}{\bxjs@depre@opt@do{precisetext}{precise-
  text=true}}
```

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

```
690 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}{true}
691 \DeclareOption{nosimplejasetup}{\bxjs@depre@opt@do{nosimplejasetup}{simple-
  ja-setup=false}}
692 \DeclareOption{simplejasetup}{\bxjs@depre@opt@do{simplejasetup}{simple-ja-
  setup=true}}
```

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

```
693 \bxjs@declare@bool@option{plautopatch}{plautopatch}{false}
694 \g@addto@macro\bxjs@plautopatchtrue{\let\bxjs@plautopatch@given\@undefined}
695 \g@addto@macro\bxjs@plautopatchfalse{\def\bxjs@plautopatch@given{false}}
```

---

## ■オプションの実行

---

L<sup>A</sup>T<sub>E</sub>X 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。

```
696 \let\bxjs@org@removeelement\@removeelement
697 \def\@removeelement#1#2#3{%
698   \def\reserved@a{#2}%
699   \ifx\reserved@a\@empty \let#3\@empty
700   \else \bxjs@org@removeelement{#1}{#2}{#3}%
701   \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。

```
702 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
703 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
704 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
705 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
706 \ProcessOptions\relax
707 \bxjs@post@option@hook
```

後処理

```
708 \if@slide
709   \def\maybeblue{\@ifundefined{ver@color.sty}{{\color{blue}}}}
710 \fi
711 \if@landscape
712   \setlength\@tempdima {\paperheight}
713   \setlength\paperheight{\paperwidth}
714   \setlength\paperwidth {\@tempdima}
715 \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。

```
716 \def\bxjs@tmpdo{%
717   \def\bxjs@tmpa{\@gobble}%
718   \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
719   \let\@classoptionslist\bxjs@tmpa}
720 \def\bxjs@tmpdo@a#1,{%
721   \ifx\@nil#1\relax\else
722     \bxjs@tmpdo@b#1{}\@nil
723     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
724     \expandafter\bxjs@tmpdo@a
725   \fi}
726 \def\bxjs@tmpdo@b#1#\bxjs@tmpdo@c}
727 \def\bxjs@tmpdo@c#1\@nil{%
728   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
729 \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。

```
730 \@expandtwoargs\@removeelement
731 {papersize}\@classoptionslist\@classoptionslist
732 \@expandtwoargs\@removeelement
733 {10pt}\@classoptionslist\@classoptionslist
734 \@expandtwoargs\@removeelement
```

■使用エンジンの検査・自動判定 デフォルトで現在使われているエンジンが pL<sup>A</sup>T<sub>E</sub>X か upL<sup>A</sup>T<sub>E</sub>X かを判定します。ユーザによって platex オプションまたは uplatex オプションが明示的に指定されている場合は、実際に使われているエンジン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<sup>A</sup>T<sub>E</sub>X/ upL<sup>A</sup>T<sub>E</sub>X を自動判別するオプション autodetect-engine を新設しました。upL<sup>A</sup>T<sub>E</sub>X 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。

[2023-02-12] autodetect-engine 指定時の挙動を規定化しました。また platex を新設しました。オプション autodetect-engine, platex, uplatex のうち最後に指定されたものが有効になります。

---

正規化前の和文ドライバの値を \bxjs@jadriver に設定する。

```
736 \ifx\bxjs@jadriver@opt\undefined\else
737   \let\bxjs@jadriver\bxjs@jadriver@opt
738 \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。

```
739 \let\bxjs@tmpb\jsEngine
740 \ifx j\bxjs@tmpb\ifjsWithpTeXng
741   \let\bxjs@tmpb=g
742 \fi\fi
743 \ifx j\bxjs@tmpb\ifjsWithupTeX
744   \let\bxjs@tmpb=u
745 \fi\fi
746 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
747   \let\bxjs@tmpb=n
748 \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。)

```
749 \ifx *\bxjs@engine@given
750   \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<sup>A</sup>T<sub>E</sub>X 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。

```
751 \ifx j\bxjs@engine@given
752   \g@addto@macro\@classoptionslist{,platex}
753 \else\ifx u\bxjs@engine@given
754   \g@addto@macro\@classoptionslist{,uplatex}
755 \fi\fi
756 \fi
757 \ifx\bxjs@engine@given\undefined\else
```



```

758 \ifx\bxjs@engine@given\bxjs@tmpb\else
759   \ClassError\bxjs@clsname
760   {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
761 \fi
762 \fi

```

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

```

763 \ifjsWithpTeXng
764   \g@addto@macro\@classoptionslist{,uplatex}
765 \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。

```

766 \@tempwattrue
767 \ifx \bxjs@driver@given\@undefined\else
768   \ifjsInPdfMode
769     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
770       \@tempwafalse
771     \fi
772   \else\ifx x\jsEngine
773     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
774       \@tempwafalse
775     \fi
776   \else
777     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
778       \@tempwafalse
779     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
780       \@tempwafalse
781     \fi\fi
782     \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
783       \@tempwafalse
784     \fi\fi
785   \fi\fi
786 \fi
787 \if@tempswa\else
788   \ClassError\bxjs@clsname
789   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
790 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

791 \ifjsInPdfMode \@tempwafalse
792 \else\ifx x\jsEngine \@tempwafalse
793 \else\ifjsWithpTeXng \@tempwafalse
794 \else \@tempwattrue
795 \fi\fi\fi
796 \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)。

```

797 \ifx\bxjs@driver@opt\@undefined
798 \if \ifbxjs@explIII T\else\ifx\bxjs@jadriver@opt\@undefined F\else T\fi\fi T%
799 \ClassWarningNoLine\bxjs@clsname
800 {A driver option is MISSING!!\MessageBreak
801 You should properly specify one of the valid\MessageBreak
802 driver options according to the DVI driver\MessageBreak
803 that is in use:\MessageBreak
804 \@spaces dvips, dvipdfmx, dviout, xdvi,\MessageBreak
805 \@spaces nodvidriver}
806 \fi
807 \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。)

```

808 \ifbxjs@dvi@opt
809 \edef\bxjs@next{%
810 \let\noexpand\bxjs@driver@given
811 \csname bxjs@dvidriver@\bxjs@driver@opt\endcsname
812 \noexpand\g@addto@macro\noexpand\@classoptionslist
813 {,\bxjs@driver@opt}%
814 }\bxjs@next
815 \fi
816 \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。

```

817 \ifjsWithpTeXng
818 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
819 \let\bxjs@platexng@nodrv\@undefined
820 \else\ifx t\bxjs@platexng@nodrv\else
821 \g@addto@macro\@classoptionslist{,dvipdfmx}
822 \fi\fi
823 \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。

```

824 \ifx\bxjs@driver@given\bxjs@driver@@none
825 \bxjs@papersizefalse
826 \fi

```

■その他の BXJS 特有の後処理 🐞 \documentclass より前に plautopatch パッケージが読み込まれている場合は bxjs@plautopatch を真にする。

```

827 \@ifpackageloaded{plautopatch}{%
828 \bxjs@plautopatchtrue
829 }{}

```

標準の和文ドライバの名前の定数。

```

830 \def\bxjs@@minimal{minimal}
831 \def\bxjs@@standard{standard}

```

```

832 \def\bxjs@@pandoc{pandoc}
833 \def\bxjs@@modern{modern}

```

\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。

```

834 \ifx\bxjs@jadriver\relax
835   \ifx j\jsEngine
836     \let\bxjs@jadriver\bxjs@@standard
837   \else
838     \ClassWarningNoLine\bxjs@clsname
839     {The option 'ja' is MISSING!!\MessageBreak
840     So 'ja=minimal' is assumed as fallback, but\MessageBreak
841     such implicit setting is now DEPRECATED!\MessageBreak
842     You should write 'ja=minimal' explicitly,\MessageBreak
843     if it is intended}
844     \let\bxjs@jadriver\bxjs@@minimal
845   \fi
846 \fi

```

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

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

※ Pandoc モードでは plautopatch の既定値を真とする。

```

847 \ifx\bxjs@jadriver\bxjs@@pandoc \ifx\bxjs@plautopatch@given\@undefined
848   \ifjsWithTeX
849   \bxjs@plautopatchtrue
850 \fi\fi\fi
851 \ifx j\jsEngine \ifbxjs@plautopatch
852   \RequirePackage{plautopatch}[2018/08/22]\%v0.3
853 \fi\fi

```

エンジンオプションがない場合はエラーを出す。

※ただし ja 非指定の場合はスキップする。

```

854 \ifx\bxjs@jadriver@opt\@undefined\else
855   \ifx\bxjs@engine@given\@undefined
856     \ClassError\bxjs@clsname
857     {An engine option must be explicitly given}%
858     {When you use a Japanese-driver you must specify a correct\MessageBreak
859     engine option.\MessageBreak\@ehc}
860 \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。)

```

861 \ifx\bxjs@magstyle@@default\bxjs@magstyle@@mag\else
862   \ifx\bxjs@magstyle\bxjs@magstyle@@mag
863     \let\bxjs@magstyle\bxjs@magstyle@@default
864     \ClassError\bxjs@clsname
865     {The engine does not support 'magstyle=usemag'}%

```

```

866      {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
867      The default value 'nomag*' is used instead.\MessageBreak \@ehc}
868    \fi
869  \fi

```

base、jbase、scale の値を用いて和文スケール値を解決する。

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

```

870 \ifx\bxjs@jbase@opt\@undefined\else
871   \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。

```

872     \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
873     \bxjs@invscale\@tempdima\jsScale
874     \bxjs@setbasefontsize{\@tempdima}%
875   \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。

```

876     \ifx\bxjs@scale@opt\@undefined\else
877       \ClassWarningNoLine\bxjs@clsname
878       {Redundant 'scale' option is ignored}%
879     \fi
880     \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
881     \@tempdimb=\bxjs@param@basefontsize\relax
882     \edef\jsScale{\strip@pt\@tempdimb}%
883     \bxjs@invscale\@tempdima\jsScale
884     \edef\jsScale{\strip@pt\@tempdima}%
885   \fi
886 \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。

```

887 \let\Cjascale\jsScale

```

---



---

8bit 欧文 T<sub>E</sub>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。）

```

888 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
889   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
890     \catcode\@tempcnta\active
891     \advance\@tempcnta\@ne
892   \repeat
893 \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。

```

894 \ifbxjs@disguise@js
895 %<book|report>\def\bxjs@js@clsname{jsbook}
896 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
897 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
898 \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。

```

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

```

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

```

900 \ifbxjs@oldfontcommands
901 \AtEndOfClass{\allowoldfontcommands}
902 \fi

```

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

トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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

```

-O -0.5in,-0.5in

```

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

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

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

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

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

[2022-09-12] 次期 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> カーネルに \stockwidth, \stockheight が追加されるようですので、クラスファイル側では未定義のときのみこれらの長さ変数を定義します。h20y6m さん、ありがとうございます。

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。

```

903 \ifx\stockwidth\@undefined\newdimen\stockwidth\fi
904 \ifx\stockheight\@undefined\newdimen\stockheight\fi
905 \begingroup\expandafter\expandafter\expandafter\endgroup
906 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
907 % \newdimen\stockwidth \newdimen\stockheight
908 \setlength{\stockwidth}{\paperwidth}
909 \setlength{\stockheight}{\paperheight}
910 \advance \stockwidth 2in
911 \advance \stockheight 2in
912 \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。

```

913 %<slide>\def\n@baseline{13}%
914 %<!\slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
915 %<!\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。

```

916 \ifx\bxjs@magstyle\bxjs@magstyle@@mag
917 \jsc@magtrue
918 \else\ifx\bxjs@magstyle\bxjs@magstyle@@xreal
919 \jsc@mag@xrealtrue
920 \fi\fi

```

---

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

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

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

---

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

```

921 \ifx\bxjs@param@mag\relax
922 \@tempdima=\bxjs@param@basefontsize
923 \advance\@tempdima.001pt \multiply\@tempdima25
924 \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
925 \edef\bxjs@param@mag{\the\@tempcnta}
926 \else
927 % mag 値が直接指定された場合

```

```

928 \bxjs@gset@tempcnta{\bxjs@param@mag}
929 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
930 % 有効な mag 値の範囲は 1--32768
931 \edef\bxjs@param@mag{\the\@tempcnta}
932 \advance\@tempcnta100000
933 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
934 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
935 \edef\bxjs@param@basefontsize{\the\@tempdima}
936 \fi
937 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
938 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
939 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
940 \edef\jsc@magscale{\strip@pt\@tempdima}
941 \let\jsBaseFontSize\bxjs@param@basefontsize

```

---

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

---

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

```

942 \newdimen\jsc@mpt
943 \newdimen\jsc@mmm
944 \ifjsc@mag
945   \jsc@mpt=1\p@
946   \jsc@mmm=1mm
947   \catcode`\?=9 % \p@? read as \p@
948 \else
949   \jsc@mpt=\jsc@magscale\p@
950   \jsc@mmm=\jsc@magscale mm
951   \catcode`\?=11 \let\p@?\jsc@mpt
952 \fi
953 \chardef\bxjs@qmcc=\catcode`\?\relax
954 \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。

use-zw が真の時は \zw を \jsZw と同義にする。

```

955 \newdimen\jsZw
956 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
957 \ifbxjs@usezw
958   \providecommand*\zw{\jsZw}
959 \fi

```

\zwspace 全角幅の水平空き。

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

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

```
961 \ifjsc@mag@xreal
962   \RequirePackage{type1cm}
963   \let\jsc@invscale\bxjs@invscale
```

ムニャムニャムニャ……。

---

```
964   \ifbxjs@TUenc
965     \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
966   \else
967     \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
968   \fi
969   \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
970   \let\jsc@get@external@font\get@external@font
971   \def\get@external@font{%
972     \jsc@preadjust@extract@font
973     \jsc@get@external@font}
974   \def\jsc@fstrunc#1{%
975     \edef\jsc@tmpa{\strip@pt#1}%
976     \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
977   \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
978     \if#5*\else
979       \edef\jsc@tmpa{#1%
980         \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
981     \fi}
982   \def\jsc@preadjust@extract@font{%
983     \let\jsc@req@size\f@size
984     \dimen@f@sizep@ \jsc@invscale\dimen@\jsc@magscale
985     \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
986     \let\jsc@ref@size\jsc@tmpa
987     \let\f@size\jsc@ref@size}
988   \def\execute@size@function#1{%
989     \let\jsc@cref@size\f@size
990     \let\f@size\jsc@req@size
991     \csname s@fct@#1\endcsname}
992   \let\jsc@DeclareErrorFont\DeclareErrorFont
993   \def\DeclareErrorFont#1#2#3#4#5{%
994     \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
995     \edef\jsc@tmpa{{#1}-{#2}-{#3}-{#4}}{\strip@pt\@tempdimc}}
996     \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
997   \def\gen@sfcnt{%
998     \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
999     \empty@sfcnt}
1000   \def\genb@sfcnt{%
1001     \edef\mandatory@arg{%
1002       \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@@}%
1003     \empty@sfcnt}
```



```

1004 \ifbxjs@TUenc\else
1005 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
1006 \fi
1007 \fi

```

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

```

\jsc@smallskip
\jsc@medskip 1008 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 1009 %\def\jsc@medskip{\vspace\jsc@medskipamount}
1010 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 1011 \newskip\jsc@smallskipamount
\jsc@bigskipamount 1012 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
1013 %\newskip\jsc@medskipamount
1014 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
1015 %\newskip\jsc@bigskipamount
1016 %\jsc@bigskipamount =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

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

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

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

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

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

```

1017 % \ifpapersize
1018 % \setlength{\@tempdima}{\paperwidth}
1019 % \setlength{\@tempdimb}{\paperheight}
1020 % \iftombow
1021 % \advance \@tempdima 2truein
1022 % \advance \@tempdimb 2truein
1023 % \fi
1024 % \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
1025 % \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、 $\text{\LaTeX}$  の `\def\@{\spacefactor\@m}` という定義 ( $\@m$  は 1000) では `I watch TV\@.` と書くと `V` とピリオドのペアカーニングが効かなくなります。そこで、次のような定義に直し、`I watch TV.\@` と書くことにします。

[2016-07-14] 2015-01-01 の  $\text{\LaTeX}$  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。

---

```
1026 \chardef\bxjs@periodchar=`\.  
1027 \bxjs@protected\def\bxjs@SE{%  
1028   \ifnum\spacefactor<\@m \spacefactor\@m  
1029   \else \spacefactor\sfcode\bxjs@periodchar  
1030   \fi}  
1031 \ifbxjs@fix@at@cmd  
1032   \def\@{\bxjs@SE{}}  
1033 \fi
```

## 4 フォントサイズ

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

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

のようにして行います。これは

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

という意味です。ただし、処理を速くするため、以下では 10 と同義の  $\text{\LaTeX}$  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、 $\text{\LaTeX}$  本体で定義されています。

|                      |       |                      |    |                      |      |
|----------------------|-------|----------------------|----|----------------------|------|
| <code>\@vpt</code>   | 5     | <code>\@vipt</code>  | 6  | <code>\@vipt</code>  | 7    |
| <code>\@viipt</code> | 8     | <code>\@ixpt</code>  | 9  | <code>\@xpt</code>   | 10   |
| <code>\@xipt</code>  | 10.95 | <code>\@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  $\text{\LaTeX 2}_{\epsilon}$  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は、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

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

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

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

---

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

```
1034 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
1035 \expandafter\bxjs@tmpa\expandafter{%
1036   \set@fontsize{#1}{#2}{#3}%
1037 % 末尾にコードを追加
1038   \expandafter\def\expandafter\size@update\expandafter{%
1039     \size@update
1040     \jsFontSizeChanged}%
1041 }
```

`\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。

```
1042 \newcommand*\jsFontSizeChanged{%
1043   \jsZw=\f@size\p@
1044   \jsZw=\jsScale \jsZw
1045   \ifdim\parindent>\z@
1046     \if@english \parindent=1em
1047     \else       \parindent=1\jsZw
1048     \fi
1049   \fi\relax
1050   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
1051 \newcommand*\jsResetDimen{}
```

---

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

```
1052 \ifjsc@mag
1053   \let\jsc@setfontsize\@setfontsize
1054 \else
1055   \def\jsc@setfontsize#1#2#3{%
1056     \@setfontsize#1{#2\jsc@mpt}{#3\jsc@mpt}}
```

```

1057 % microtype 対策
1058 \ifjsWithTeX\if j\jsEngine\else
1059   \def\jsc@setfontsize#1#2#3{%
1060     \edef\bxjs@sfs@next{%
1061       \unexpanded{\@setfontsize#1}%
1062       {\the\dimexpr#2\jsc@mpt\relax}{\the\dimexpr#3\jsc@mpt\relax}%
1063     }\bxjs@sfs@next}
1064 \fi\fi
1065 \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。

---

```

1066 \emergencystretch 3\jsZw

```

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

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

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

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

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

---

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

---

```

1067 \newif\ifnarrowbaselines
1068 \if@english
1069   \narrowbaselinestrue
1070 \fi
1071 \def\narrowbaselines{%
1072   \narrowbaselinestrue
1073   \skip0=\abovedisplayskip
1074   \skip2=\abovedisplayshortskip
1075   \skip4=\belowdisplayskip
1076   \skip6=\belowdisplayshortskip
1077 % 一時的に警告を無効化する
1078 \let\bxjs@save@nomath\@nomath
1079 \let\@nomath\@gobble
1080 \@currsize\selectfont

```

```

1081 \let\@nomath\bxjs@save@nomath
1082 \abovedisplayskip=\skip0
1083 \abovedisplayshortskip=\skip2
1084 \belowdisplayskip=\skip4
1085 \belowdisplayshortskip=\skip6\relax}
1086 \def\widebaselines{\narrowbaselinesfalse\@currsizel\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。

```

1087 \def\bxjs@if@narrowbaselines{%
1088   \ifnarrowbaselines\expandafter\@firstoftwo
1089   \else \expandafter\@secondoftwo
1090   \fi
1091 }

```

---

`\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。

---

```

1092 \renewcommand{\normalsize}{%
1093   \bxjs@if@narrowbaselines{%
1094     \jsc@setfontsize\normalsize\@xpt\@xipt
1095   }{%else
1096     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1097   }%

```

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] `TEX 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、微調節してことにしました。

```

1098 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1099 \abovedisplayshortskip \z@ \@plus3\p@?
1100 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1101 \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。

```
1102 \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。

```
1103 %</class>
1104 %<*class|minijs>
1105 %% initialize
1106 \normalsize
1107 %</class|minijs>
1108 %<*class>
```

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

---

`\Chs`

`\Cwd` 等の変数は p<sub>T</sub>E<sub>X</sub> 系以外では未定義なのでここで定義する。

```
1109 \ifx\Cht\undefined \newdimen\Cht \fi
1110 \ifx\Cdp\undefined \newdimen\Cdp \fi
1111 \ifx\Cwd\undefined \newdimen\Cwd \fi
1112 \ifx\Cvs\undefined \newdimen\Cvs \fi
1113 \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。

---

```
1114 \setlength\Cht{0.88\jsZw}
1115 \setlength\Cdp{0.12\jsZw}
1116 \setlength\Cwd{1\jsZw}
1117 \setlength\Cvs{\baselineskip}
1118 \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  $4 \pm 2$ ,  $2 \pm 1$  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。

---

```
1119 \newcommand{\small}{%
1120 \bxjs@if@narrowbaselines{%
1121 %<!kiyou> \jsc@setfontsize\small\@ixpt{11}%
1122 %<kiyou> \jsc@setfontsize\small{8.8888}{11}%
1123 }{%else
1124 %<!kiyou> \jsc@setfontsize\small\@ixpt{13}%
```

```

1125 %<kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
1126 }%
1127 \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1128 \abovedisplayshortskip \z@ \@plus3\p@?
1129 \belowdisplayskip \abovedisplayskip
1130 \belowdisplayshortskip \belowdisplayskip
1131 \def\@listi{\leftmargin\leftmargini
1132 \topsep \z@
1133 \parsep \z@
1134 \itemsep \parsep}}

```

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

```

1135 \newcommand{\footnotesize}{%
1136 \bxjs@if@narrowbaselines{%
1137 %<kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
1138 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
1139 }{%else
1140 %<kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
1141 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1142 }%
1143 \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1144 \abovedisplayshortskip \z@ \@plus2\p@?
1145 \belowdisplayskip \abovedisplayskip
1146 \belowdisplayshortskip \belowdisplayskip
1147 \def\@listi{\leftmargin\leftmargini
1148 \topsep \z@
1149 \parsep \z@
1150 \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 1151 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1152 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 1153 \if@twocolumn
\LARGE 1154 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1155 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1156 \else
1157 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
1158 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1159 \fi
1160 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1161 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1162 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1163 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}

```

```
1164 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1165 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}
```

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

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

```
1166 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}
```

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

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

```
1167 % \newcommand{\headfont}{\bfseries}
1168 \newcommand{\headfont}{\sffamily}
1169 % \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。

```
1170 %<!kiyou>\setlength\columnsep{2\Cwd}
1171 %<kiyou>\setlength\columnsep{28truebp}
1172 \setlength\columnseprule{\z@}
```

### ■段落

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

```
\lineskiplimit 1173 \setlength\lineskip{1\jsc@mpt}
1174 \setlength\normallineskip{1\jsc@mpt}
\normallineskiplimit 1175 \setlength\lineskiplimit{1\jsc@mpt}
1176 \setlength\normallineskiplimit{1\jsc@mpt}
```

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

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



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

```
1177 \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。

```
1178 \setlength\parskip{\z@}
1179 \if@slide
1180   \setlength\parindent{0\p@}
1181 \else
1182   \setlength\parindent{1\Cwd}
1183 \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 1184 \@lowpenalty 51
1185 \@medpenalty 151
1186 \@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。

```
1187 % \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。

```
1188 % \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。

```
1189 \@onlypreamble\bxjs@bd@pre@geometry@hook
1190 \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。

```
1191 \ifjsc@mag
1192 \mag=\bxjs@param@mag
1193 \fi
1194 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLlength` のための和文単位の定義。

```
1195 \def\bxjs@unit@trueQ{0.25trueem}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1196 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw
```

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

```
1197 \@tempwafalse
1198 \def\bxjs@tmpdo{\@ifnextchar\bgroup\bxjs@tmpdo@a\remove@to@nnil}
1199 \def\bxjs@tmpdo@a#1{\edef\bxjs@tmpa{#1}%
1200   \@ifnextchar\bgroup\bxjs@tmpdo@b\remove@to@nnil}
1201 \def\bxjs@tmpdo@b#1{\edef\bxjs@tmpa{\bxjs@tmpa,#1}%
1202   \@ifnextchar\@nnil\bxjs@tmpdo@c\remove@to@nnil}
1203 \def\bxjs@tmpdo@c\@nnil{\@tempwattrue
1204   \edef\bxjs@param@paper{papersize={\bxjs@tmpa}}}
1205 \expandafter\bxjs@tmpdo\bxjs@param@paper\@nnil
```

`W,H` の形式について。

```
1206 \if@tempswa\else
1207   \def\bxjs@tmpa{\@nil,\@nil}
1208   \def\bxjs@tmpdo#1,#2,#3\@nnil{%
1209     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1210       \@tempwattrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1211   \expandafter\bxjs@tmpdo\bxjs@param@paper,\@nil,\@nil\@nnil
1212 \fi
```

`W*H` の形式について。

```
1213 \if@tempswa\else
1214   \def\bxjs@tmpa{\@nil*\@nil}
1215   \def\bxjs@tmpdo#1*#2*#3\@nnil{%
1216     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1217       \@tempwattrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1218   \expandafter\bxjs@tmpdo\bxjs@param@paper*\@nil*\@nil\@nnil
1219 \fi
```

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

```
1220 \edef\bxjs@layout@paper{%
1221   \ifjsc@mag truedimen,\fi
1222   \if@landscape landscape,\fi
1223   \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。

```
1224 %<*article|report>
1225 \def\bxjs@layout@base{%
1226   headheight=\topskip,footskip=0.03367\paperheight,%
1227   headsep=\footskip-\topskip,includeheadfoot,%
1228 }
1229 \edef\bxjs@layout{\bxjs@layout@base
1230   hscale=0.76,hmarginratio=1:1,%
1231   vscale=0.83,vmarginratio=1:1,%
```

```

1232 }
1233 %</article|report>
1234 %<*book>
1235 \def\bxjs@layout@base{%
1236   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1237 }
1238 \ifbxjs@layout@buggyhmargin      %---
1239 % アレ
1240 \edef\bxjs@layout{\bxjs@layout@base
1241   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1242   vscale=0.83,vmarginratio=1:1,%
1243 }
1244 \else                                %---
1245 % 非アレ
1246 \edef\bxjs@layout{\bxjs@layout@base
1247   hmargin=18\jsc@mmm,%
1248   vscale=0.83,vmarginratio=1:1,%
1249 }
1250 \fi                                %---
1251 %</book>
1252 %<*slide>
1253 \def\bxjs@layout@base{%
1254   noheadfoot,%
1255 }
1256 \edef\bxjs@layout{\bxjs@layout@base
1257   hscale=0.9,hmarginratio=1:1,%
1258   vscale=0.944,vmarginratio=1:1,%
1259 }
1260 %</slide>

```

textwidth オプションの設定を反映する。

```

1261 %<!*book>
1262 \ifx\bxjs@textwidth@opt\@undefined\else
1263   \jsSetQHLLength\@tempdima{\bxjs@textwidth@opt}
1264   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1265 \fi
1266 %</!*book>
1267 \ifx\bxjs@number@of@lines@opt\@undefined\else
1268   \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1269   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1270 \fi

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1271 \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。

```
1272 %<*book>
```

```

1273 \newcommand\jsTextWidthLimit{40}
1274 \@tempdima=\jsTextWidthLimit\Cwd
1275 \ifx\bxjs@textwidth@limit@opt\@undefined\else
1276   \bxjs@gset@tempcnta{\bxjs@textwidth@limit@opt}
1277   \@tempdima=\@tempcnta\Cwd
1278 \fi
1279 \ifx\bxjs@textwidth@opt\@undefined\else
1280   \jsSetQHLLength\@tempdima{\bxjs@textwidth@opt}
1281 \fi
1282 \edef\bxjs@textwidth@limit{\the\@tempdima}
1283 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1284   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1285   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1286 \fi
1287 %</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。

```

1288 \def\bxjs@preproc@layout{%
1289   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@empt}

```

\bxjs@postproc@layout geometry の後処理。

```

1290 \def\bxjs@postproc@layout{%

```

geometry のドライバを再設定する。

```

1291   \ifx\bxjs@geometry@driver\relax\else
1292     \let\Gm@driver\bxjs@geometry@driver
1293   \fi

```

\ht\strutbox の値を元に戻す。

```

1294   \ht\strutbox=\bxjs@save@ht@strutbox\relax

```

\textwidth の値を補正する。

```

1295   \ifbxjs@whole@zw@lines
1296     \@tempdimb=\textwidth
1297     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1298     \advance\textwidth.005pt\relax
1299     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1300     \advance\@tempdimb-\textwidth
1301     \advance\oddsidemargin 0.5\@tempdimb
1302     \advance\evensidemargin 0.5\@tempdimb
1303   \fi
1304   \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。

```

1305 %<*book>

```

```

1306 \@tempdima=\bxjs@textwidth@limit\relax
1307 \ifbxjs@whole@zw@lines
1308   \advance\@tempdima.005pt\relax
1309   \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1310 \fi
1311 \ifdim\textwidth>\@tempdima
1312   \textwidth=\@tempdima
1313   \addtolength\evensidemargin{\fullwidth-\textwidth}
1314 \fi
1315 %</book>

```

\textheight 関連の調整。

```

1316 \@tempdimb=\textheight
1317 \advance\textheight-\topskip
1318 \advance\textheight.005pt\relax
1319 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1320 \advance\textheight\topskip
1321 \advance\@tempdimb-\textheight
1322 \advance\topmargin0.5\@tempdimb

```

\headheight 関連の調整。

```

1323 \@tempdima=\topskip
1324 \advance\headheight\@tempdima
1325 \advance\topmargin-\@tempdima

```

marginpar 関連の調整。

```

1326 \setlength\marginparsep{\columnsep}
1327 \setlength\marginparpush{\baselineskip}
1328 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1329   -\textwidth-10\jsc@mmm-\marginparsep}
1330 \ifbxjs@whole@zw@lines
1331   \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1332 \fi

```

連動する変数。

```

1333 \maxdepth=.5\topskip
1334 \stockwidth=\paperwidth
1335 \stockheight=\paperheight
1336 }

```

\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。

```

1337 \edef\jsGeometryOptions{%
1338   \bxjs@layout@paper,\bxjs@layout}

```

---

■ geometry パッケージを読み込む 

---

ムニャムニャ。

```

1339 \def\bxjs@geometry@name{geometry}
1340 \ifbxjs@old@hook@system
1341   \let\bxjs@apply@bd@pre@geometry@hook\AtBeginDocument
1342 \else
1343   \def\bxjs@apply@bd@pre@geometry@hook{%
1344     \AddToHook{begindocument}{\bxjs@geometry@name}}
1345 \fi

```

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

```

1346 \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。

```

1347 \ifbxjs@papersize
1348   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1349     \PassOptionsToPackage{dvipdfm}{geometry}
1350   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1351     \PassOptionsToPackage{dvipdfm}{geometry}
1352   \fi\fi
1353   \let\bxPapersizeSpecialDone=t
1354 \else
1355   \PassOptionsToPackage{driver=none}{geometry}
1356 \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。

```

1357 \bxjs@apply@bd@pre@geometry@hook{\bxjs@bd@pre@geometry@hook}
1358 \bxjs@apply@bd@pre@geometry@hook{\ImposeOldLuaTeXBehavior}
1359 \bxjs@preproc@layout
1360 \edef\bxjs@next{%
1361   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1362 }\bxjs@next
1363 \bxjs@apply@bd@pre@geometry@hook{\RevokeOldLuaTeXBehavior}

```

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

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

```

1364 \let\bxjs@geometry@driver\Gm@driver
1365 \bxjs@postproc@layout

```

geometry のドライバ自動判別に対する前処理。

```

1366 \g@addto@macro\bxjs@bd@pre@geometry@hook{%

```

BXJS の 2.0 版より、geometry の 4.x 版のサポートは廃止された。

```

1367   \ifpackageafter{geometry}{2010/02/12}{-}{%else
1368     \PackageError\bxjs@clsname
1369     {Your 'geometry' package is too old (< v5.0)}%

```

```

1370      {\@ehc}%
1371      \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。

```

1372      \ifjsWithpTeXng
1373      \ifx\Gm@driver\@empty
1374      \def\Gm@driver{pdftex}%
1375      \fi
1376      \fi}

```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```

1377 \def\setpagelayout{%
1378   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{%else
1379     \@ifstar{\bxjs@setpagelayout@a\@ne}{\bxjs@setpagelayout@a\z@}}
1380 \def\bxjs@setpagelayout@a#1#2{%
1381   \ifcase#1% modify
1382     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1383   \or% reset(*)
1384     \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1385   \or% semireset(+)
1386     \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1387   \fi
1388   \bxjs@preproc@layout
1389   \edef\bxjs@next{%
1390     \noexpand\geometry{\bxjs@next}%
1391   }\bxjs@next
1392   \bxjs@postproc@layout}

```

## ■geometry パッケージを読み込まない 🐼

`geometry=user` の場合の処理。

```

1393 \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。

```

1394 \g@addto@macro\bxjs@begin@document@hook{%
1395   \ifdim\textwidth=.5\maxdimen
1396     \ClassError\bxjs@clsname
1397     {Page layout is not properly set}%
1398     {\@ehd}%
1399   \fi}
1400 \def\jsUseMinimalPageLayout{%
1401   \setlength{\textwidth}{6.5in}%
1402   \setlength{\textheight}{8in}}

```

`\setpagelayout` はとりあえず無効にしておく。

```

1403 \let\bxjs@geometry@driver\relax
1404 \def\setpagelayout{%
1405   \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1406     \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1407 \def\bxjs@pagelayout@a#1{%
1408   \ClassError\bxjs@clsname
1409     {Command '\string\setpagelayout' is not supported,\MessageBreak
1410       because 'geometry' value is not 'class'}}\@eha}
1411 %
1412 \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。

---

```

1413 %<*\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 `\topskip` 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。

```

1414 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1415 \if@slide
1416   \setlength\headheight{0\jsc@mpt}
1417 \else
1418   \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1419 \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。

```

1420 %<*article|kiyou>
1421 \if@slide
1422   \setlength\footskip{0pt}
1423 \else
1424   \setlength\footskip{0.03367\paperheight}
1425   \ifdim\footskip<\baselineskip
1426     \setlength\footskip{\baselineskip}
1427   \fi
1428 \fi
1429 %</article|kiyou>
1430 %<jspf>\setlength\footskip{9\jsc@mmm}
1431 %<*book>
1432 \if@report
1433   \setlength\footskip{0.03367\paperheight}
1434   \ifdim\footskip<\baselineskip
1435     \setlength\footskip{\baselineskip}
1436   \fi
1437 \else
1438   \setlength\footskip{0pt}
1439 \fi
1440 %</book>
1441 %<*report>
1442 \setlength\footskip{0.03367\paperheight}
1443 \ifdim\footskip<\baselineskip
1444   \setlength\footskip{\baselineskip}
1445 \fi
1446 %</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)。

```

1447 %<*article>
1448 \if@slide
1449   \setlength\headsep{0\jsc@mpt}
1450   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1451   \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1452 \else
1453   \setlength\headsep{\footskip}
1454   \addtolength\headsep{-\topskip}
1455 \fi
1456 %</article>
1457 %<*book>
1458 \if@report

```

```

1459 \setlength\headsep{\footskip}
1460 \addtolength\headsep{-\topskip}
1461 \else
1462 \setlength\headsep{6\jsc@mmm}
1463 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1464 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1465 \fi
1466 %</book>
1467 %<*report>
1468 \setlength\headsep{\footskip}
1469 \addtolength\headsep{-\topskip}
1470 %</report>
1471 %<*jspf>
1472 \setlength\headsep{9\jsc@mmm}
1473 \addtolength\headsep{-\topskip}
1474 %</jspf>
1475 %<*kiyou>
1476 \setlength\headheight{0\jsc@mpt}
1477 \setlength\headsep{0\jsc@mpt}
1478 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1479 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1480 %</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<sup>3</sup>、`\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。

```

1481 \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。

```

1482 \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。

```

1483 %<*article>
1484 \if@slide

```

```

1485 \setlength\fullwidth{0.9\paperwidth}
1486 \else
1487 \setlength\fullwidth{0.76\paperwidth}
1488 \fi
1489 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1490 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1491 \setlength\textwidth{\fullwidth}
1492 %</article>
1493 %<*book>
1494 \if@report
1495 \setlength\fullwidth{0.76\paperwidth}
1496 \else
1497 \setlength\fullwidth{\paperwidth}
1498 \addtolength\fullwidth{-36\jsc@mmm}
1499 \fi
1500 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1501 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1502 \setlength\textwidth{\fullwidth}
1503 \if@report \else
1504 \if@twocolumn \else
1505 \ifdim \fullwidth>40zw
1506 \setlength\textwidth{40zw}
1507 \fi
1508 \fi
1509 \fi
1510 %</book>
1511 %<*report>
1512 \setlength\fullwidth{0.76\paperwidth}
1513 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1514 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1515 \setlength\textwidth{\fullwidth}
1516 %</report>
1517 %<*jspf>
1518 \setlength\fullwidth{50zw}
1519 \addtolength\fullwidth{8\jsc@mmm}
1520 \setlength\textwidth{\fullwidth}
1521 %</jspf>
1522 %<*kiyou>
1523 \setlength\fullwidth{48zw}
1524 \addtolength\fullwidth{\columnsep}
1525 \setlength\textwidth{\fullwidth}
1526 %</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)。

```
1527 %<*article|book|report>
1528 \if@slide
1529   \setlength{\textheight}{0.95\paperheight}
1530 \else
1531   \setlength{\textheight}{0.83\paperheight}
1532 \fi
1533 \addtolength{\textheight}{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1534 \addtolength{\textheight}{-\headsep}
1535 \addtolength{\textheight}{-\footskip}
1536 \addtolength{\textheight}{-\topskip}
1537 \divide\textheight\baselineskip
1538 \multiply\textheight\baselineskip
1539 %</article|book|report>
1540 %<jspf>\setlength{\textheight}{51\baselineskip}
1541 %<kiyou>\setlength{\textheight}{47\baselineskip}
1542 \addtolength{\textheight}{\topskip}
1543 \addtolength{\textheight}{0.1\jsc@empt}
1544 %<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  $\text{IAT}_{\text{E}}\text{X} 2_{\epsilon}$  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。

```
1545 \def\flushbottom{%
1546   \def\@textbottom{\vskip \z@ \@plus.1\jsc@empt}%
1547   \let\@texttop\relax}
```

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

```
1548 \setlength\marginparsep{\columnsep}
1549 \setlength\marginparpush{\baselineskip}
```

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

`\oddsidemargin` が使われます。TEX は上・左マージンに `1truein` を挿入しますが、トンボ関係のオプションが指定されると pL<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (`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。

```

1550 \setlength{\oddsidemargin}{\paperwidth}
1551 \addtolength{\oddsidemargin}{-\fullwidth}
1552 \setlength{\oddsidemargin}{.5\oddsidemargin}
1553 \iftombow
1554   \addtolength{\oddsidemargin}{-1in}
1555 \else
1556   \addtolength{\oddsidemargin}{-\inv@mag in}
1557 \fi
1558 \setlength{\evensidemargin}{\oddsidemargin}
1559 \if@mparswitch
1560   \addtolength{\evensidemargin}{\fullwidth}
1561   \addtolength{\evensidemargin}{-\textwidth}
1562 \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。

```

1563 \setlength\marginparwidth{\paperwidth}
1564 \addtolength\marginparwidth{-\oddsidemargin}
1565 \addtolength\marginparwidth{-\inv@mag in}
1566 \addtolength\marginparwidth{-\textwidth}
1567 \addtolength\marginparwidth{-10\jsc@mmm}
1568 \addtolength\marginparwidth{-\marginparsep}
1569 \@tempdima=1zw
1570 \divide\marginparwidth\@tempdima
1571 \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)。

```

1572 \setlength\topmargin{\paperheight}
1573 \addtolength\topmargin{-\textheight}
1574 \if@slide
1575   \addtolength\topmargin{-\headheight}
1576 \else
1577   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1578 \fi
1579 \addtolength\topmargin{-\headsep}
1580 \addtolength\topmargin{-\footskip}
1581 \setlength\topmargin{0.5\topmargin}

```

```

1582 %<kiyou>\setlength\topmargin{81truebp}
1583 \iftombow
1584 \addtolength\topmargin{-1in}
1585 \else
1586 \addtolength\topmargin{-\inv@mag in}
1587 \fi
1588 %</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用ことにする。

---

```
1589 \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。

```
1590 \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<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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。

```
1591 \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。

```
1592 \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。

```
1593 \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。

```
1594 \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。

1595 `\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。

1596 `\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。

1597 `\renewcommand{\floatpagefraction}{.8}`

`\c@dbltopnumber` 二段組のとき本文ページ上部に出力できる段抜きフロートの最大数です。

[2003-08-23] ちょっと増やしました。

1598 `\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。

1599 `\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。

1600 `\renewcommand{\dblfloatpagefraction}{.8}`

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

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

`\intextsep` 文との距離です。

1601 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

1602 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`

1603 `\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` 1604 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

1605 `\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` 1606 `\setlength\@fptop{0\p@? \@plus 1fil}`

1607 `\setlength\@fpsep{8\p@? \@plus 2fil}`

1608 `\setlength\@fpbot{0\p@? \@plus 1fil}`

`\@dblfpsep` 段抜きフロートについての値です。

`\@dblfpsep` 1609 `\setlength\@dblfpsep{0\p@? \@plus 1fil}`

1610 `\setlength\@dblfpsep{8\p@? \@plus 2fil}`

`\@dblfpbot` 1611 `\setlength\@dblfpbot{0\p@? \@plus 1fil}`

## 6 改ページ（日本語 T<sub>E</sub>X 開発コミュニティ版のみ）

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

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

となっています。

```
1612 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1613 %  \ifodd\c@page
1614 %    \iftdir
1615 %      \hbox{}\thispagestyle{empty}\newpage
1616 %      \if@twocolumn\hbox{}\newpage\fi
1617 %    \fi
1618 %  \else
1619 %    \ifydir
1620 %      \hbox{}\thispagestyle{empty}\newpage
1621 %      \if@twocolumn\hbox{}\newpage\fi
1622 %    \fi
1623 %  \fi\fi}
1624 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1625 %  \ifodd\c@page
1626 %    \ifydir
1627 %      \hbox{}\thispagestyle{empty}\newpage
1628 %      \if@twocolumn\hbox{}\newpage\fi
1629 %    \fi
1630 %  \else
1631 %    \iftdir
1632 %      \hbox{}\thispagestyle{empty}\newpage
1633 %      \if@twocolumn\hbox{}\newpage\fi
1634 %    \fi
1635 %  \fi\fi}
1636 \def\pltx@cleartooddpage{\clearpage\if@twoside
1637   \ifodd\c@page\else
1638     \hbox{}\thispagestyle{empty}\newpage
1639     \if@twocolumn\hbox{}\newpage\fi
1640   \fi\fi}
1641 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1642   \ifodd\c@page
1643     \hbox{}\thispagestyle{empty}\newpage
1644     \if@twocolumn\hbox{}\newpage\fi
1645   \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。

```
1646 \let\pltx@cleartorightpage\pltx@cleartooddpage
1647 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

\vsiz の値がアレな場合は本体開始まで \clearpage を無効にする。
1648 \ifdim\vsiz=\z@
1649 \begingroup
1650 \toks@\expandafter{\clearpage}
1651 \xdef\clearpage{\noexpand\ifbxjs@after@preamble\the\toks@\noexpand\fi}
1652 \endgroup
1653 \fi
```

---

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

```
1654 %<*book|report>
1655 \if@openleft
1656 \let\cleardoublepage\pltx@cleartoleftpage
1657 \else\if@openright
1658 \let\cleardoublepage\pltx@cleartorightpage
1659 \fi\fi
1660 %</book|report>
```

## 7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

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

柱を設定する命令は、右の柱が左の柱の下位にある場合は十分まともに動作します。たと

えば左マークを `\chapter`, 右マークを `\section` で変更する場合がこれにあたります。しかし, 同一ページに複数の `\markboth` があると, おかしな結果になることがあります。

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

`\ps@empty` `empty` ページスタイルの定義です。L<sup>A</sup>T<sub>E</sub>X 本体で定義されているものをコメントアウトした形で載せておきます。

```
1661 % \def\ps@empty{%
1662 %   \let\mkboth\gobbletwo
1663 %   \let\oddhead\empty
1664 %   \let\oddfoot\empty
1665 %   \let\evenhead\empty
1666 %   \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。

```
1667 \def\ps@plainfoot{%
1668   \let\mkboth\gobbletwo
1669   \let\oddhead\empty
1670   \def\oddfoot{\normalfont\hfil\thepage\hfil}%
1671   \let\evenhead\empty
1672   \let\evenfoot\oddfoot}
1673 \def\ps@plainhead{%
1674   \let\mkboth\gobbletwo
1675   \let\oddfoot\empty
1676   \let\evenfoot\empty
1677   \def\evenhead{%
1678     \ifmparswitch \hss \fi
1679     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1680     \ifmparswitch\else \hss \fi}%
1681   \def\oddhead{%
1682     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1683 %<book>\let\ps@plain\ps@plainhead
1684 %<!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。

```
1685 %<*article|slide>
1686 \if@twoside
1687   \def\ps@headings{%
1688     \let\oddfoot\empty
1689     \let\evenfoot\empty
1690     \def\evenhead{\ifmparswitch \hss \fi
1691       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1692     \ifmparswitch\else \hss \fi}%
1693   \def\oddhead{\hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}
```

```

1693 \def\@oddhead{%
1694 \underline{%
1695 \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1696 \let\@mkboth\markboth
1697 \def\sectionmark##1{\markboth{%
1698 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1699 ##1}{}}%
1700 \def\subsectionmark##1{\markright{%
1701 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1702 ##1}}%
1703 }
1704 \else % if not twoside
1705 \def\ps@headings{%
1706 \let\@oddfoot\@empty
1707 \def\@oddhead{%
1708 \underline{%
1709 \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1710 \let\@mkboth\markboth
1711 \def\sectionmark##1{\markright{%
1712 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1713 ##1}}%
1714 \fi
1715 %</article|slide>

```

次は book および report の場合です。[2011-05-10] しっぱ愛好家さん [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。

---

```

1716 %<*book|report>
1717 \def\bxjs@maybe@autoxspacing{%
1718 \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1719 \newif\if@omit@number
1720 \def\ps@headings{%
1721 \let\@oddfoot\@empty
1722 \let\@evenfoot\@empty
1723 \def\@evenhead{%
1724 \if@mparswitch \hss \fi
1725 \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1726 \textbf{\thepage}\hfil\leftmark}}}%
1727 \if@mparswitch\else \hss \fi}%
1728 \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1729 {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1730 \let\@mkboth\markboth
1731 \def\chaptermark##1{\markboth{%
1732 \ifnum \c@secnumdepth >\m@ne
1733 \if@mainmatter

```

```

1734         \if@omit@number\else
1735         \@chapapp\thechapter\@chappos\hskip1\jsZw
1736         \fi
1737     \fi
1738 \fi
1739 ##1}{}}%
1740 \def\sectionmark##1{\markright{%
1741     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1742     ##1}}}%
1743 %</book|report>

```

最後は学会誌の場合です。

```

1744 %<*jspf>
1745 \def\ps@headings{%
1746     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1747     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1748     \def\@oddhead{\normalfont\hfil \@title \hfil}
1749     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1750 %</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。

```

1751 \def\ps@myheadings{%
1752     \let\@oddfoot\@empty\let\@evenfoot\@empty
1753     \def\@evenhead{%
1754         \if@mparswitch \hss \fi%
1755         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1756         \if@mparswitch\else \hss \fi}%
1757     \def\@oddhead{%
1758         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1759     \let\@mkboth\@gobbletwo
1760 %<book|report> \let\chaptermark\@gobble
1761     \let\sectionmark\@gobble
1762 %<!\book&!report> \let\subsectionmark\@gobble
1763 }

```

## 8 文書のマークアップ

### 8.1 表題

\title これらは L<sup>A</sup>T<sub>E</sub>X 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 1764 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1765 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1766 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1767 % \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。

**TODO:** `\subtitle` の遅延処理は Pandoc モードに移す。

本体を `\jsSubtitle` として定義する。

```
1768 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1769 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。
1770 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1771 \g@addto@macro\bxjs@begin@document@hook{\bxjs@decl@subtitle}
1772 \def\bxjs@decl@subtitle{%
1773   \global\let\bxjs@decl@subtitle\relax
1774   \ifx\subtitle\@undefined
1775     \global\let\subtitle\jsSubtitle
1776   \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。

```
1777 \def\bxjs@annihilate@subtitle{%
1778   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1779   \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` 1780 %<\*jspf>

```
\keywords 1781 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1782 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
1783 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1784 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1785 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1786 %</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。

```
1787 \def\plainifnotempty{%
1788   \ifx \@oddhead \@empty
1789     \ifx \@oddfoot \@empty
1790       \else
1791         \thispagestyle{plainfoot}%
```

```

1792 \fi
1793 \else
1794 \thispagestyle{plainhead}%
1795 \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。

```

1796 %<article|book|report|slide>
1797 \if@titlepage
1798 \newcommand{\maketitle}{%
1799 \begin{titlepage}%
1800 \let\footnotesize\small
1801 \let\footnoterule\relax
1802 \let\footnote\thanks
1803 \null\vfil
1804 \if@slide
1805 {\footnotesize \@date}%
1806 \begin{center}
1807 \mbox{} \[\[1\jsZw]
1808 \large
1809 {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1810 \jsc@smallskip
1811 \@title
1812 \ifx\bxjs@subtitle\@undefined\else
1813 \par\vskip\z@
1814 {\small \bxjs@subtitle\par}
1815 \fi
1816 \jsc@smallskip
1817 {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1818 \vfill
1819 {\small \@author}%
1820 \end{center}
1821 \else
1822 \vskip 60\p@?
1823 \begin{center}%
1824 {\LARGE \@title \par}%
1825 \ifx\bxjs@subtitle\@undefined\else
1826 \vskip5\p@?
1827 {\normalsize \bxjs@subtitle\par}
1828 \fi
1829 \vskip 3em%
1830 {\large
1831 \lineskip .75em
1832 \begin{tabular}[t]{c}%
1833 \@author

```

```

1834         \end{tabular}\par}%
1835         \vskip 1.5em
1836         {\large \@date \par}%
1837     \end{center}%
1838     \fi
1839     \par
1840     \@thanks\vfil\null
1841 \end{titlepage}%
1842 \setcounter{footnote}{0}%
1843 \global\let\thanks\relax
1844 \global\let\maketitle\relax
1845 \global\let\@thanks\@empty
1846 \global\let\@author\@empty
1847 \global\let\@date\@empty
1848 \global\let\@title\@empty
1849 \global\let\title\relax
1850 \global\let\author\relax
1851 \global\let\date\relax
1852 \global\let\and\relax
1853 \bxjs@annihilate@subtitle
1854 }%
1855 \else
1856 \newcommand{\maketitle}{\par
1857     \begingroup
1858         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1859         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1860         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1861             \parindent 1\jsZw\noindent
1862             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1863         \if@twocolumn
1864             \ifnum \col@number=\@ne
1865                 \@maketitle
1866             \else
1867                 \twocolumn[\@maketitle]%
1868             \fi
1869         \else
1870             \newpage
1871             \global\@topnum\z@ % Prevents figures from going at top of page.
1872             \@maketitle
1873         \fi
1874         \plainifnotempty
1875         \@thanks
1876     \endgroup
1877     \setcounter{footnote}{0}%
1878     \global\let\thanks\relax
1879     \global\let\maketitle\relax
1880     \global\let\@thanks\@empty
1881     \global\let\@author\@empty
1882     \global\let\@date\@empty

```

```

1883 \global\let\@title\@empty
1884 \global\let\title\relax
1885 \global\let\author\relax
1886 \global\let\date\relax
1887 \global\let\and\relax
1888 \bxjs@annihilate@subtitle
1889 }

```

\@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。

```

1890 \def\@maketitle{%
1891   \newpage\null
1892   \vskip 2em
1893   \begin{center}%
1894     \let\footnote\thanks
1895     {\LARGE \@title \par}%
1896     \ifx\bxjs@subtitle\@undefined\else
1897       \vskip3\p@?
1898       {\normalsize \bxjs@subtitle\par}
1899     \fi
1900     \vskip 1.5em
1901     {\large
1902       \lineskip .5em
1903       \begin{tabular}[t]{c}%
1904         \@author
1905       \end{tabular}\par}%
1906     \vskip 1em
1907     {\large \@date}%
1908   \end{center}%
1909   \par\vskip 1.5em
1910 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1911 }
1912 \fi
1913 %</article|book|report|slide>
1914 %<*jspf>
1915 \newcommand{\maketitle}{\par
1916   \begingroup
1917     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1918     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1919     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1920       \parindent 1\jsZw\noindent
1921       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1922     \twocolumn[\@maketitle]%
1923     \plainifnotempty
1924     \@thanks
1925   \endgroup
1926   \setcounter{footnote}{0}%
1927   \global\let\thanks\relax
1928   \global\let\maketitle\relax
1929   \global\let\@thanks\@empty

```



```

1930 \global\let\@author\@empty
1931 \global\let\@date\@empty
1932 % \global\let\@title\@empty % \@title は柱に使う
1933 \global\let\title\relax
1934 \global\let\author\relax
1935 \global\let\date\relax
1936 \global\let\and\relax
1937 \ifx\authors@mail\@undefined\else{%
1938     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1939     \footnotetext[0]{\itshape\authors@mail}%
1940 }\fi
1941 \global\let\authors@mail\@undefined}
1942 \def\@maketitle{%
1943     \newpage\null
1944     \vskip 6em % used to be 2em
1945     \begin{center}
1946         \let\footnote\thanks
1947         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1948         \lineskip .5em
1949         \ifx\@author\@undefined\else
1950             \vskip 1em
1951             \begin{tabular}[t]{c}%
1952                 \@author
1953             \end{tabular}\par
1954         \fi
1955         \ifx\@etitle\@undefined\else
1956             \vskip 1em
1957             {\large \@etitle \par}%
1958         \fi
1959         \ifx\@eauthor\@undefined\else
1960             \vskip 1em
1961             \begin{tabular}[t]{c}%
1962                 \@eauthor
1963             \end{tabular}\par
1964         \fi
1965         \vskip 1em
1966         \@date
1967     \end{center}
1968     \vskip 1.5em
1969     \centerline{\box\@abstractbox}
1970     \ifx\@keywords\@undefined\else
1971         \vskip 1.5em
1972         \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1973     \fi
1974     \vskip 1.5em}
1975 %</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。

```
1976 \def\bxjs@label@sect#1{%
1977   \expandafter\ifx\csname label#1\endcsname\relax
1978     \csname the#1\endcsname
1979   \else \csname label#1\endcsname
1980 \fi}
1981 \def\@secntformat#1{\bxjs@label@sect{#1}\quad}
```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```
1982 \ifnum\bxjs@label@section=\bxjs@label@section@@compat\else
1983 \def\@secapp{\presectionname}
1984 \def\@secpos{\postsectionname}
1985 \fi
```

`\labelsection` 節番号の出力書式。

```
1986 \ifnum\bxjs@label@section=\bxjs@label@section@@modern
1987 \def\labelsection{\@secapp\thesection\@secpos}
1988 \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。

```
1989 \def\@startsection#1#2#3#4#5#6{%
1990   \if@noskipsec \leavevmode \fi
1991   \par
1992 % 見出し上の空きを \@tempskipa にセットする
1993   \@tempskipa #4\relax
1994 % \@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
1995   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1996 % 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
1997   \ifdim \@tempskipa <\z@
1998     \@tempskipa -\@tempskipa \@afterindentfalse
1999   \fi
2000   \if@nobreak
2001 %   \everypar{\everyparhook}% これは間違い
2002     \everypar{}%
2003   \else
2004     \addpenalty\@secpenalty
2005 % 次の行は削除
2006 %   \addvspace\@tempskipa
2007 % 次の \noindent まで追加
2008     \ifdim \@tempskipa >\z@
2009       \if@slide\else
2010         \null
2011         \vspace*{-\baselineskip}%
2012       \fi
2013       \vskip\@tempskipa
2014     \fi
2015   \fi
2016   \noindent
2017 % 追加終わり
2018   \@ifstar
2019     {\@ssect{#3}{#4}{#5}{#6}}%
2020     {\@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。

```

2021 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
2022 \let\bxjs@if@ceph\@firstofone
2023 \else \let\bxjs@if@ceph\@gobble
2024 \fi

```

---

```

2025 \def\@sect#1#2#3#4#5#6[#7]#8{%
2026 \ifnum #2>\c@secnumdepth
2027 \let\@svsec\@empty
2028 \else
2029 \refstepcounter{#1}%
2030 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2031 \fi
2032 % 見出し後の空きを \@tempskipa にセット
2033 \@tempskipa #5\relax
2034 % 条件判断の順序を入れ換えました
2035 \ifdim \@tempskipa<\z@
2036 \def\@svsechd{%
2037 #6{\hskip #3\relax
2038 \@svsec #8}%
2039 \csname #1mark\endcsname{#7}%
2040 \addcontentsline{toc}{#1}{%
2041 \ifnum #2>\c@secnumdepth \else
2042 \protect\numberline{\bxjs@label@sect{#1}}%
2043 \fi
2044 #7}}% 目次にフルネームを載せるなら #8
2045 \else
2046 \begingroup
2047 \interlinepenalty \@M % 下から移動
2048 #6{%
2049 \@hangfrom{\hskip #3\relax\@svsec}%
2050 % \interlinepenalty \@M % 上に移動
2051 #8\@@par}%
2052 \endgroup
2053 \csname #1mark\endcsname{#7}%
2054 \addcontentsline{toc}{#1}{%
2055 \ifnum #2>\c@secnumdepth \else
2056 \protect\numberline{\bxjs@label@sect{#1}}%
2057 \fi
2058 #7}}% 目次にフルネームを載せるならここは #8
2059 \fi
2060 \@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。

```

2061 \def\@xsect#1{%
2062 % 見出しの後ろの空きを \@tempskipa にセット

```

```

2063 \@tempskipa #1\relax
2064 % 条件判断の順序を変えました
2065 \ifdim \@tempskipa<\z@
2066 \nobreakfalse
2067 \global\@noskipsectrue
2068 \everypar{%
2069 \if@noskipsec
2070 \global\@noskipsecfalse
2071 {\setbox\z@\lastbox}%
2072 \clubpenalty\@M
2073 \begingroup \@svsechd \endgroup
2074 \unskip
2075 \@tempskipa #1\relax
2076 \hskip -\@tempskipa
2077 \else
2078 \clubpenalty \@clubpenalty
2079 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
2080 \fi\bxjs@if@ceph\everyparhook}%
2081 \else
2082 \par \nobreak
2083 \vskip \@tempskipa
2084 \@afterheading
2085 \fi
2086 \if@slide
2087 {\vskip\if@twocolumn-5\jsc@mpt\else-6\jsc@mpt\fi
2088 \maybeblue\hrule height0\jsc@mpt depth1\jsc@mpt
2089 \vskip\if@twocolumn 4\jsc@mpt\else 7\jsc@mpt\fi\relax}%
2090 \fi
2091 \par % 2000-12-18
2092 \ignorespaces}
2093 \def\@ssect#1#2#3#4#5{%
2094 \@tempskipa #3\relax
2095 \ifdim \@tempskipa<\z@
2096 \def\@svsechd{#4{\hskip #1\relax #5}}%
2097 \else
2098 \begingroup
2099 #4{%
2100 \@hangfrom{\hskip #1}%
2101 \interlinepenalty \@M #5\@@par}%
2102 \endgroup
2103 \fi
2104 \@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<sup>A</sup>T<sub>E</sub>X 本体で定義済みです。

```

\subsectionmark 2105 \newcommand*\chaptermark[1]{ }
\subsubsectionmark 2106 % \newcommand*\sectionmark[1]{ }

\paragraphmark
\subparagraphmark

```

```

2107 % \newcommand*{\subsectionmark}[1]{ }
2108 % \newcommand*{\subsubsectionmark}[1]{ }
2109 % \newcommand*{\paragraphmark}[1]{ }
2110 % \newcommand*{\subparagraphmark}[1]{ }

```

## ■カウンタの定義

`\c@secnumdepth` `secnumdepth` は第何レベルの見出しまで番号を付けるかを定めるカウンタです。

```

2111 %< !book&!report>\setcounter{secnumdepth}{3}
2112 %<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 2113 \newcounter{part}
2114 %<book|report>\newcounter{chapter}
\c@subsubsection 2115 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2116 %< !book&!report>\newcounter{section}
\c@subparagraph 2117 \newcounter{subsection}[section]
2118 \newcounter{subsubsection}[subsection]
2119 \newcounter{paragraph}[subsubsection]
2120 \newcounter{subparagraph}[paragraph]

```

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

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

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

2121 \renewcommand{\thepart}{\@Roman\c@part}
2122 %< * !book&!report>
2123 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2124 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2125 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2126 \else
2127 \renewcommand{\thesection}{\@arabic\c@section}
2128 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2129 \fi
2130 %< / !book&!report>
2131 %< *book|report>
2132 \renewcommand{\thechapter}{\@arabic\c@chapter}
2133 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2134 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}

```

```

2135 %</book|report>
2136 \renewcommand{\thesubsubsection}{%
2137   \thesubsection.\@arabic\c@subsubsection}
2138 \renewcommand{\theparagraph}{%
2139   \thesubsubsection.\@arabic\c@paragraph}
2140 \renewcommand{\thesubparagraph}{%
2141   \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。

```

2142 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2143 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

[2017-03-05] `\frontmatter` と `\mainmatter` の2つの命令は, 改丁または改ページした後で `\pagenumbering{...}` でノンブルを1にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが, これではノンブルをリセットする際に偶奇逆転が起こる場合があります。 `openany` かどうかに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, L<sup>A</sup>T<sub>E</sub>X の標準クラスでは1998年に修正されていた問題です (コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X の標準クラス 2017/03/05 も参照)。

```

2144 %<*book|report>
2145 \newcommand\frontmatter{%
2146   \pltx@cleartooddpage
2147   \@mainmatterfalse
2148   \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。

```

2149 \newcommand\mainmatter{%
2150   \pltx@cleartooddpage
2151   \@mainmattertrue
2152   \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。

```

2153 \newcommand\backmatter{%
2154   \if@openleft
2155     \cleardoublepage
2156   \else\if@openright
2157     \cleardoublepage
2158   \else
2159     \clearpage
2160   \fi\fi
2161   \@mainmatterfalse}
2162 %</book|report>

```

## ■部

`\part` 新しい部を始めます。

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

```
\secdef{星なし}{星あり}
```

**星なし** \* のない形の定義です。

**星あり** \* のある形の定義です。

`\secdef` は次のようにして使います。

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

まず `book` と `report` のクラス以外です。

```
2163 %<!*book&!report>
2164 \newcommand\part{%
2165   \if@noskipsec \leavevmode \fi
2166   \par
2167   \addvspace{4ex}%
2168   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2169   \secdef\@part\@spart}
2170 %</!*book&!report>
```

`book` および `report` クラスの場合は、少し複雑です。

```
2171 %<*book|report>
2172 \newcommand\part{%
2173   \if@openleft
2174     \cleardoublepage
2175   \else\if@openright
2176     \cleardoublepage
2177   \else
2178     \clearpage
2179   \fi\fi
2180   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2181   \if@twocolumn
2182     \onecolumn
2183     \@restonecoltrue
2184   \else
2185     \@restonecolfalse
2186   \fi
2187   \null\vfil
2188   \secdef\@part\@spart}
2189 %</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。

```
2190 %<*!book&!report>
2191 \def\@part[#1]#2{%
2192   \ifnum \c@secnumdepth >\m@ne
2193     \refstepcounter{part}%
2194     \addcontentsline{toc}{part}{%
2195       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2196   \else
2197     \addcontentsline{toc}{part}{#1}%
2198   \fi
2199   \markboth{}{}%
2200   {\parindent\z@
2201     \raggedright
2202     \interlinepenalty \@M
2203     \normalfont
2204     \ifnum \c@secnumdepth >\m@ne
2205       \Large\headfont\prepartname\thepart\postpartname
2206       \par\nobreak
2207     \fi
2208     \huge \headfont #2%
2209     \markboth{}{}\par}%
2210   \nobreak
2211   \vskip 3ex
2212   \@afterheading}
2213 %</!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。

```
2214 %<*book|report>
2215 \def\@part[#1]#2{%
2216   \ifnum \c@secnumdepth >-2\relax
2217     \refstepcounter{part}%
2218     \addcontentsline{toc}{part}{%
2219       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2220   \else
2221     \addcontentsline{toc}{part}{#1}%
2222   \fi
2223   \markboth{}{}%
2224   {\centering
2225     \interlinepenalty \@M
2226     \normalfont
2227     \ifnum \c@secnumdepth >-2\relax
2228       \huge\headfont \prepartname\thepart\postpartname
2229       \par\vskip20\p@?
2230     \fi
2231     \Huge \headfont #2\par}%
2232   \@endpart}
2233 %</book|report>
```

\@spart 番号を付けない部です。

```

2234 %<!*book&!report>
2235 \def\@spart#1{%
2236     \parindent \z@ \raggedright
2237     \interlinepenalty \@M
2238     \normalfont
2239     \huge \headfont #1\par}%
2240 \nobreak
2241 \vskip 3ex
2242 \@afterheading}
2243 %</!book&!report>
2244 %<*book|report>
2245 \def\@spart#1{%
2246     \centering
2247     \interlinepenalty \@M
2248     \normalfont
2249     \Huge \headfont #1\par}%
2250 \@endpart}
2251 %</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<sup>A</sup>T<sub>E</sub>X では `classes.dtx` v1.4b (2000/05/19) で修正されています。

```

2252 %<*book|report>
2253 \def\@endpart{\vfil\newpage
2254     \if@twoside
2255         \if@openleft %% added (2017/02/24)
2256             \null\thispagestyle{empty}\newpage
2257         \else\if@openright %% added (2016/12/13)
2258             \null\thispagestyle{empty}\newpage
2259         \fi\fi %% added (2016/12/13, 2017/02/24)
2260     \fi
2261     \if@restonecol
2262         \twocolumn
2263     \fi}
2264 %</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。

```

2265 %<*book|report>
2266 \newcommand{\chapter}{%
2267     \if@openleft\cleardoublepage\else
2268     \if@openright\cleardoublepage\else\clearpage\fi\fi
2269     \plainifnotempty % 元: \thispagestyle{plain}

```

```

2270 \global\@topnum\z@
2271 \if@english \@afterindentfalse \else \@afterindenttrue \fi
2272 \secdef
2273   {\@omit@numberfalse\@chapter}%
2274   {\@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。

```

2275 \def\@chapter[#1]#2{%
2276   \ifnum \c@secnumdepth >\m@ne
2277     \if@mainmatter
2278       \refstepcounter{chapter}%
2279       \typeout{\@chapapp\thechapter\@chappos}%
2280       \addcontentsline{toc}{chapter}%
2281         {\protect\numberline
2282 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2283         {\@chapapp\thechapter\@chappos}%
2284         #1}%
2285     \else\addcontentsline{toc}{chapter}{#1}\fi
2286   \else
2287     \addcontentsline{toc}{chapter}{#1}%
2288   \fi
2289   \chaptermark{#1}%
2290   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}%
2291   \addtocontents{lot}{\protect\addvspace{10\jsc@empt}}%
2292   \if@twocolumn
2293     \@topnewpage[\@makechapterhead{#2}]%
2294   \else
2295     \@makechapterhead{#2}%
2296     \@afterheading
2297   \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。

```

2298 \def\@makechapterhead#1{%
2299   \vspace*{2\Cvs}% 欧文は 50pt
2300   {\parindent \z@ \raggedright \normalfont
2301     \ifnum \c@secnumdepth >\m@ne
2302       \if@mainmatter
2303         \huge\headfont \@chapapp\thechapter\@chappos
2304         \par\nobreak
2305         \vskip \Cvs % 欧文は 20pt
2306       \fi
2307     \fi
2308     \interlinepenalty\@M
2309     \Huge \headfont #1\par\nobreak
2310     \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。

```

2311 \def\@schapter#1{%

```

```

2312 \chaptermark{#1}%
2313 \if@twocolumn
2314 \topnewpage[\@makeschapterhead{#1}]%
2315 \else
2316 \makeschapterhead{#1}\@afterheading
2317 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2318 \def\@makeschapterhead#1{%
2319 \vspace*{2\Cvs}% 欧文は 50pt
2320 {\parindent \z@ \raggedright
2321 \normalfont
2322 \interlinepenalty\@M
2323 \Huge \headfont #1\par\nobreak
2324 \vskip 3\Cvs}} % 欧文は 40pt
2325 %</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。

```

2326 \if@twocolumn
2327 \newcommand{\section}{%
2328 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2329 \@startsection{section}{1}{\z@}%
2330 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2331 %<kiyou> {\Cvs}{0.5\Cvs}%
2332 % {\normalfont\large\headfont\@secapp}}
2333 {\normalfont\large\headfont\raggedright}}
2334 \else
2335 \newcommand{\section}{%
2336 \if@slide\clearpage\fi
2337 \@startsection{section}{1}{\z@}%
2338 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2339 {.5\Cvs \@plus.3\Cdp}% 後アキ
2340 % {\normalfont\Large\headfont\@secapp}}
2341 {\normalfont\Large\headfont\raggedright}}
2342 \fi

```

`\subsection` 同上です。

```

2343 \if@twocolumn
2344 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2345 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2346 {\normalfont\normalsize\headfont}}
2347 \else
2348 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2349 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ

```

```

2350     {.5\Cvs \@plus.3\Cdp}% 後アキ
2351     {\normalfont\large\headfont}}
2352 \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)。

```

2353 \if@twocolumn
2354   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2355     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2356     {\normalfont\normalsize\headfont}}
2357 \else
2358   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2359     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2360     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2361     {\normalfont\normalsize\headfont}}
2362 \fi

```

`\paragraph` 見出しの後ろで改行されません。

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

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

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

※ BXJS クラスでは、1.1 版 [2016-02-14] から `\jsParagraphMark` をサポートしている。

段落のマーク (■) が必ず和文フォントで出力されるようにする。

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

```

2363 \newcommand{\jsParagraphMark}{\relax\jsJaChar{■}}
2364 \let\bxjs@org@paragraph@mark\jsParagraphMark
2365 \ifx\bxjs@paragraph@mark\@empty
2366   \let\jsParagraphMark\@empty
2367 \else\ifx\bxjs@paragraph@mark\@undefined\else
2368   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2369 \fi\fi
2370 \let\jsJaChar\@empty
2371 \if@twocolumn
2372   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2373     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2374     %<jspf> {\normalfont\normalsize\headfont}}
2375     %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2376 \else
2377   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%

```

```

2378      {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2379      {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2380 %<jspf>      {\normalfont\normalsize\headfont}}
2381 %<!jspf>      {\normalfont\normalsize\headfont\jsParagraphMark}}
2382 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2383 \if@twocolumn
2384   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2385     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2386     {\normalfont\normalsize\headfont}}
2387 \else
2388   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2389     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2390     {\normalfont\normalsize\headfont}}
2391 \fi

```

### 8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2392 \if@slide
2393   \setlength\leftmargini{1\jsZw}
2394 \else
2395   \if@twocolumn
2396     \setlength\leftmargini{2\jsZw}
2397   \else
2398     \setlength\leftmargini{3\jsZw}
2399   \fi
2400 \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 2401 \if@slide
\leftmarginv 2402   \setlength\leftmarginii {1\jsZw}
\leftmarginvi 2403   \setlength\leftmarginiii{1\jsZw}
2404   \setlength\leftmarginiv {1\jsZw}
2405   \setlength\leftmarginv  {1\jsZw}
2406   \setlength\leftmarginvi {1\jsZw}
2407 \else
2408   \setlength\leftmarginii {2\jsZw}
2409   \setlength\leftmarginiii{2\jsZw}

```

```

2410 \setlength\leftmarginiv {2\jsZw}
2411 \setlength\leftmarginv {1\jsZw}
2412 \setlength\leftmarginvi {1\jsZw}
2413 \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。

```

2414 \setlength \labelsep {0.5\jsZw} % .5em
2415 \setlength \labelwidth{\leftmargini}
2416 \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。

```

2417 \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 2418 \@beginparpenalty -\@lowpenalty
\@itempenalty 2419 \@endparpenalty -\@lowpenalty
2420 \@itempenalty -\@lowpenalty

```

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

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

```

2421 \def\@listi{\leftmargin\leftmargini
2422 \parsep \z@
2423 \topsep 0.5\baselineskip
2424 \itemsep \z@ \relax}
2425 \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）。

```

2426 \@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 2427 \def\@listii{\leftmargin\leftmarginii
\@listiv 2428 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2429 \topsep \z@
\@listv 2430 \parsep \z@
\@listvi 2431 \itemsep\parsep}
2432 \def\@listiii{\leftmargin\leftmarginiii
2433 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2434 \topsep \z@
2435 \parsep \z@

```

```

2436 \itemsep\parsep}
2437 \def\@listiv {\leftmargin\leftmarginiv
2438             \labelwidth\leftmarginiv
2439             \advance\labelwidth-\labelsep}
2440 \def\@listv {\leftmargin\leftmarginv
2441             \labelwidth\leftmarginv
2442             \advance\labelwidth-\labelsep}
2443 \def\@listvi {\leftmargin\leftmarginvi
2444             \labelwidth\leftmarginvi
2445             \advance\labelwidth-\labelsep}

```

■**enumerate 環境** enumerate 環境はカウンタ enumi, enumii, enumiii, enumiv を使います。enum*n* 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<sup>A</sup>T<sub>E</sub>X 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。

```

2446 \renewcommand{\theenumi}{\@arabic\c@enumi}
2447 \renewcommand{\theenumii}{\@alph\c@enumii}
2448 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2449 \renewcommand{\theenumiv}{\@Alph\c@enumiv}

```

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

---

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

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

---

```

2450 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2451 \newcommand*{\jsInJaParen}[1]{%
2452   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2453 \newcommand{\labelenumi}{\theenumi.}
2454 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2455 \newcommand{\labelenumiii}{\theenumiii.}
2456 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2457 \renewcommand{\p@enumii}{\theenumi}
2458 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2459 \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。

```
\labelitemi 2460 \newcommand\labelitemi{\textbullet}
\labelitemii 2461 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 2462 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2463 \newcommand\labelitemiv{\textperiodcentered}
```

## ■description 環境

`description` (*env.*) 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。

```
2464 \newenvironment{description}{%
2465   \list{}{%
2466     \labelwidth=\leftmargin
2467     \labelsep=1\jsZw
2468     \advance \labelwidth by -\labelsep
2469     \let \makelabel=\descriptionlabel}}{\endlist}
```

`\descriptionlabel` `description` 環境のラベルを出力するコマンドです。好みに応じて #1 の前に適当な空き（たとえば `\hspace{1\jsZw}`）を入れるのもいいと思います。

```
2470 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}
```

## ■概要

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

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

---

bxjsreport クラスの abstract 環境は：

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

`chapterabstract` (*env.*) 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。

```
2471 %<*book|report>
2472 \newenvironment{chapterabstract}{%
2473   \begin{list}{}{%
2474     \listparindent=1\jsZw
2475     \itemindent=\listparindent
```

```

2476 \rightmargin=0pt
2477 \leftmargin=5\jsZw\item[]\end{list}\vspace{\baselineskip}}
2478 %</book|report>

```

“普通の” abstract 環境の定義。

```

2479 %<*article|report|slide>
2480 \newbox\@abstractbox
2481 \if@titlepage
2482 \newenvironment{abstract}{%
2483 \titlepage
2484 \null\vfil
2485 \@beginparpenalty\@lowpenalty
2486 \begin{center}%
2487 \headfont \abstractname
2488 \@endparpenalty\@M
2489 \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。

```

2490 \par}%
2491 {\par\vfil\null\endtitlepage}
2492 \else
2493 \newenvironment{abstract}{%
2494 \if@twocolumn
2495 \ifx\maketitle\relax
2496 \section*{\abstractname}%
2497 \else
2498 \global\setbox\@abstractbox\hbox\bgroup
2499 \begin{minipage}[b]{\textwidth}
2500 \small\parindent1\jsZw
2501 \begin{center}%
2502 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2503 \end{center}%
2504 \list{}{%
2505 \listparindent\parindent
2506 \itemindent \listparindent
2507 \rightmargin \leftmargin}%
2508 \item\relax
2509 \fi
2510 \else
2511 \small
2512 \begin{center}%
2513 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2514 \end{center}%
2515 \list{}{%
2516 \listparindent\parindent
2517 \itemindent \listparindent
2518 \rightmargin \leftmargin}%
2519 \item\relax
2520 \fi}{\if@twocolumn

```

```

2521      \ifx\maketitle\relax
2522      \else
2523      \endlist\end{minipage}\egroup
2524      \fi
2525      \else
2526      \endlist
2527      \fi}
2528 \fi
2529 %</article|report|slide>
2530 %<*jspf>
2531 \newbox\@abstractbox
2532 \newenvironment{abstract}{%
2533   \global\setbox\@abstractbox\hbox\bgroup
2534   \begin{minipage}[b]{157\jsc@mmm}\sffamily Abstract}\par
2535   \small
2536   \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2537   {\end{minipage}\egroup}
2538 %</jspf>

```

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

```

2539 %<*book|report>
2540 \ifbxjs@force@chapterabstract
2541   \let\abstract\chapterabstract
2542   \let\endabstract\endchapterabstract
2543 \fi
2544 %</book|report>

```

## ■キーワード

**keywords** (*env.*) 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。

```

2545 %<*jspf>
2546 %\newbox\@keywordsbox
2547 %\newenvironment{keywords}{%
2548 %   \global\setbox\@keywordsbox\hbox\bgroup
2549 %   \begin{minipage}[b]{1570\jsc@mmm}\sffamily Keywords:}\par
2550 %   \small\parindent0\jsZw}%
2551 %   {\end{minipage}\egroup}
2552 %</jspf>

```

## ■verse 環境

**verse** (*env.*) 詩のための verse 環境です。

```

2553 \newenvironment{verse}{%
2554   \let \\\=\@centercr
2555   \list{}{%
2556     \itemsep \z@

```

```

2557 \itemindent -2\jsZw % 元: -1.5em
2558 \listparindent\itemindent
2559 \rightmargin \z@
2560 \advance\leftmargin 2\jsZw}% 元: 1.5em
2561 \item\relax}{\endlist}

```

## ■quotation 環境

`quotation (env.)` 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。

```

2562 \newenvironment{quotation}{%
2563 \list{}{%
2564 \listparindent\parindent
2565 \itemindent\listparindent
2566 \rightmargin \z@}%
2567 \item\relax}{\endlist}

```

## ■quote 環境

`quote (env.)` `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。

```

2568 \newenvironment{quote}%
2569 {\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。

```

2570 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2571 \item[\hskip \labelsep{\headfont #1\ #2}]}
2572 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2573 \item[\hskip \labelsep{\headfont #1\ #2 (#3)}]}

```

`titlepage (env.)` タイトルを独立のページに出力するのに使われます。

[2017-02-24] コミュニティ版 p<sub>La</sub>T<sub>E</sub>X の標準クラス 2017/02/15 に合わせて、book クラスでタイトルを必ず奇数ページに送るようにしました。といっても、横組クラスしかありませんでしたので、従来の挙動は何も変わっていません。また、book 以外の場合のページ番号のリセットもコミュニティ版 p<sub>La</sub>T<sub>E</sub>X の標準クラス 2017/02/15 に合わせましたが、こちらで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。

```

2574 \newenvironment{titlepage}{%

```

```

2575 %<book>      \pltx@cleartooddpage %% 2017-02-24
2576      \if@twocolumn
2577          \@restonecoltrue\onecolumn
2578      \else
2579          \@restonecolfalse\newpage
2580      \fi
2581      \thispagestyle{empty}%
2582      \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-
02-24
2583  }%
2584  {\if@restonecol\twocolumn \else \newpage \fi
2585      \if@twoside\else
2586          \setcounter{page}\@ne
2587      \fi}

```

## ■付録

\appendix 本文と付録を分離するコマンドです。

```

2588 %<!*book&!report>
2589 \newcommand{\appendix}{\par
2590     \setcounter{section}{0}%
2591     \setcounter{subsection}{0}%
2592     \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2593         \gdef\presectionname{\appendixname}%
2594         \gdef\postsectionname{}%
2595     % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2596     \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2597     \gdef\thesubsection{\@Alph\c@section.\@Arabic\c@subsection}%
2598     \else
2599         \gdef\@secapp{\appendixname}%
2600         \gdef\@secpos{}%
2601         \gdef\thesection{\@Alph\c@section}%
2602     \fi}
2603 %</!*book&!report>
2604 %<*book|report>
2605 \newcommand{\appendix}{\par
2606     \setcounter{chapter}{0}%
2607     \setcounter{section}{0}%
2608     \gdef\@chapapp{\appendixname}%
2609     \gdef\@chappos{}%
2610     \gdef\thechapter{\@Alph\c@chapter}}
2611 %</book|report>

```

## 8.4 パラメータの設定

### ■array と tabular 環境

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

2612 `\setlength\arraycolsep{5\p@?}`

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

2613 `\setlength\tabcolsep{6\p@?}`

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

2614 `\setlength\arrayrulewidth{.4\p@}`

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

2615 `\setlength\doublerulesep{2\p@}`

### ■tabbing 環境

`\tabbingsep` \ ' コマンドで入るアキです。

2616 `\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。

2617 `\skip\@mpfootins = \skip\footins`

### ■framebox 環境

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

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

2618 `\setlength\fbxsep{3\p@?}`

2619 `\setlength\fbxrule{.4\p@}`

### ■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

2620 `%<!book&!report>\renewcommand \theequation {\@arabic\c@equation}`

2621 `%<*book|report>`

2622 `\@addtoreset{equation}{chapter}`

2623 `\renewcommand\theequation`

2624 `{\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}`

2625 `%</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。

2626 `% \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。

2627 `% \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。

```
2628 % \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。

```
2629 %<!*book&!report>
2630 \newcounter{figure}
2631 \renewcommand \thefigure {\@arabic\c@figure}
2632 %</!*book&!report>
2633 %<*book|report>
2634 \newcounter{figure}[chapter]
2635 \renewcommand \thefigure
2636     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2637 %</book|report>
```

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

```
\ext@figure 2638 \def\fps@figure{tbp}
2639 \def\ftype@figure{1}
\fnum@figure 2640 \def\ext@figure{lof}
2641 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure (env.)` \* 形式は段抜きのフロートです。

```
figure* (env.) 2642 \newenvironment{figure}%
2643                 {\@float{figure}}%
2644                 {\end@float}
2645 \newenvironment{figure*}%
2646                 {\@dblfloat{figure}}%
2647                 {\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。

```
2648 %<*!book&!report>
2649 \newcounter{table}
2650 \renewcommand\thetable{\@arabic\c@table}
2651 %</!book&!report>
2652 %<*book|report>
2653 \newcounter{table}[chapter]
2654 \renewcommand \thetable
2655     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2656 %</book|report>
```

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

```
\ext@table 2657 \def\fps@table{tbp}
2658 \def\ftype@table{2}
\fnun@table 2659 \def\ext@table{lot}
2660 \def\fnun@table{\tablename\nobreak\thetable}
```

`table (env.) *` は段抜きのフロートです。

```
table* (env.) 2661 \newenvironment{table}%
2662             {\@float{table}}%
2663             {\end@float}
2664 \newenvironment{table*}%
2665             {\@dblfloat{table}}%
2666             {\end@dblfloat}
```

## 8.6 キャプション

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

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

```
2667 \newlength\abovecaptionskip
2668 \newlength\belowcaptionskip
2669 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2670 \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。

```
2671 %<*!jspf>
2672 % \long\def\@makecaption#1#2{{\small
2673 %     \advance\leftskip10\jsc@mmm
```



```

2674 % \advance\rightskip10\jsc@mmm
2675 % \vskip\abovcaptionskip
2676 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2677 % \ifdim \wd\@tempboxa >\hsize
2678 % #1\hskip1\jsZw\relax #2\par
2679 % \else
2680 % \global \@minipagefalse
2681 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2682 % \fi
2683 % \vskip\belowcaptionskip}}
2684 \long\def\@makecaption#1#2{{\small
2685 \advance\leftskip .0628\linewidth
2686 \advance\rightskip .0628\linewidth
2687 \vskip\abovcaptionskip
2688 \sbox\@tempboxa{#1\zwspace#2}%
2689 \ifdim \wd\@tempboxa <\hsize \centering \fi
2690 #1\zwspace#2\par
2691 \vskip\belowcaptionskip}}
2692 %<*/!jspf>
2693 %<*jspf>
2694 \long\def\@makecaption#1#2{%
2695 \vskip\abovcaptionskip
2696 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2697 \ifdim \wd\@tempboxa >\hsize
2698 {\small\sffamily
2699 \list{#1}{%
2700 \renewcommand{\makelabel}[1]{##1\hfil}
2701 \itemsep \z@
2702 \itemindent \z@
2703 \labelsep \z@
2704 \labelwidth 11\jsc@mmm
2705 \listparindent\z@
2706 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2707 \else
2708 \global \@minipagefalse
2709 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2710 \fi
2711 \vskip\belowcaptionskip}
2712 %</jspf>

```

## 9 フォントコマンド

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

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

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

---

```

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

```

---

```

2713 \newif\if@jsc@warnoldfontcmd
2714 \@jsc@warnoldfontcmdtrue
2715 \newif\if@jsc@warnoldfontcmdexception
2716 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand

2717 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2718   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2719   \DeclareOldFontCommand{#1}{%
2720     \bxjs@oldfontcmd{#1}#2%
2721   }{%
2722     \bxjs@oldfontcmd{#1}#3%
2723   }%
2724 }
2725 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2726   \ClassInfo\bxjs@clsname
2727   {Old font command '\string#1' is used!!\MessageBreak
2728     The first occurrence is}%
2729 }

```

---

```

\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。

2730 \newcommand*{\allowoldfontcommands}{%
2731   \@jsc@warnoldfontcmdexceptiontrue}
2732 \newcommand*{\disallowoldfontcommands}{%
2733   \@jsc@warnoldfontcmdexceptionfalse}

2734 \let\bxjs@oldfontcmd@list\@empty
2735 \def\bxjs@oldfontcmd#1{%
2736   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2737 \def\bxjs@oldfontcmd@a#1#2{%
2738   \if@jsc@warnoldfontcmdexception\else
2739     \global\@jsc@warnoldfontcmdfalse
2740     \ifx#1\relax
2741       \global\let#1=t%
2742       \jsc@warnoldfontcmd{#2}%
2743     \fi
2744     \fi}

```

```

2745 \def\bxjs@warnoldfontcmd@final{%
2746 % \par
2747 \global\let\bxjs@warnoldfontcmd@final\@empty
2748 \let\@tempa\@empty
2749 \def\do##1{%
2750   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2751     \edef\@tempa{\@tempa \space\string##1}\fi}
2752 \bxjs@oldfontcmd@list
2753 \ifx\@tempa\@empty\else
2754   \ClassWarningNoLine\bxjs@clsname
2755     {Some old font commands were used in text:\MessageBreak
2756       \space\@tempa\MessageBreak
2757       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2758       new font selection scheme called NFSS2 with several\MessageBreak
2759       new, combinable font commands. The
2760       class provides\MessageBreak
2761       the old font commands only for compatibility}
2762 \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。

```

2763 \def\bxjs@warnoldfontcmd@kick@final{%
2764   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2765 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2766 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2767 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2768 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf 2769 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2770 \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。

```

2771 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2772 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2773 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2774 \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`

```
2775 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2776 \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。

```
2777 \newcommand\@pnumwidth{1.55em}
2778 \newcommand\@tocrmarg{2.55em}
2779 \newcommand\@dotsep{4.5}
2780 %<!book&!report>\setcounter{tocdepth}{2}
2781 %<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)

```

2782 \newdimen\jsc@tocl@width
2783 \newcommand{\tableofcontents}{%
2784 %<*book|report>
2785   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2786   \settowidth\@tempdima{\headfont\appendixname}%
2787   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2788   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2789   \if@twocolumn
2790     \@restonecoltrue\onecolumn
2791   \else
2792     \@restonecolfalse
2793   \fi
2794   \chapter*{\contentsname}%
2795   \@mkboth{\contentsname}{}%
2796 %</book|report>
2797 %<!*book&!report>
2798   \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2799   \settowidth\@tempdima{\headfont\appendixname}%
2800   \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2801   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2802   \section*{\contentsname}%
2803   \@mkboth{\contentsname}{\contentsname}%
2804 %</!book&!report>
2805   \@starttoc{toc}%
2806 %<book|report> \if@restonecol\twocolumn\fi
2807 }
```

`\l@part` 部の目次です。

```

2808 \newcommand*{\l@part}[2]{%
2809   \ifnum \c@tocdepth >-2\relax
2810 %<!book&!report> \addpenalty\@secpenalty
2811 %<book|report> \addpenalty{-\@highpenalty}%
2812   \addvspace{2.25em \@plus\p@}%
2813   \begingroup
2814     \parindent \z@
2815 % \@pnumwidth should be \@tocrmarg
2816 % \rightskip \@pnumwidth
2817 \rightskip \@tocrmarg
2818 \parfillskip -\rightskip
2819 {\leavevmode
2820   \large \headfont
2821   \setlength\@lnumwidth{4\jsZw}%
2822   #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2823   \nobreak
```

```

2824 %<book|report> \global\@nobreaktrue
2825 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2826 \endgroup
2827 \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)

```

2828 %<*book|report>
2829 \newcommand*{\l@chapter}[2]{%
2830 \ifnum \c@tocdepth >\m@ne
2831 \addpenalty{-\@highpenalty}%
2832 \addvspace{1.0em \@plus\p@}%
2833 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2834 \begingroup
2835 \parindent\z@
2836 % \rightskip\@pnumwidth
2837 \rightskip\@tocrmarg
2838 \parfillskip-\rightskip
2839 \leavevmode\headfont
2840 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2841 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2842 \advance\leftskip\@lnumwidth \hskip-\leftskip
2843 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2844 \penalty\@highpenalty
2845 \endgroup
2846 \fi}
2847 %</book|report>

```

\l@section 節の目次です。

```

2848 %<*!book&!report>
2849 \newcommand*{\l@section}[2]{%
2850 \ifnum \c@tocdepth >\z@
2851 \addpenalty{\@secpenalty}%
2852 \addvspace{1.0em \@plus\p@}%
2853 \begingroup
2854 \parindent\z@
2855 % \rightskip\@pnumwidth
2856 \rightskip\@tocrmarg
2857 \parfillskip-\rightskip
2858 \leavevmode\headfont
2859 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2860 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2861 \advance\leftskip\@lnumwidth \hskip-\leftskip
2862 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2863 \endgroup
2864 \fi}
2865 %</!book&!report>

```

インデントと幅はそれぞれ 1.5em, 2.3em ですが, 1zw, 3.683zw に変えました。

```
2866 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

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

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

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

```
\l@subparagraph 2867 %<!*book&!report>  
2868 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}  
2869 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}  
2870 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}  
2871 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}  
2872 %  
2873 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}  
2874 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}  
2875 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}  
2876 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}  
2877 %  
2878 \newcommand*{\l@subsection}{%  
2879 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw  
2880 \@dottedtocline{2}{\@tempdima}{3\jsZw}}  
2881 \newcommand*{\l@subsubsection}{%  
2882 \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw  
2883 \@dottedtocline{3}{\@tempdima}{4\jsZw}}  
2884 \newcommand*{\l@paragraph}{%  
2885 \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw  
2886 \@dottedtocline{4}{\@tempdima}{5\jsZw}}  
2887 \newcommand*{\l@subparagraph}{%  
2888 \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw  
2889 \@dottedtocline{5}{\@tempdima}{6\jsZw}}  
2890 %</!*book&!report>  
2891 %<*book|report>  
2892 % \newcommand*{\l@subsection} {\@dottedtocline{2}{3.8em}{3.2em}}  
2893 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}  
2894 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}  
2895 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}  
2896 \newcommand*{\l@section}{%  
2897 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw  
2898 \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}  
2899 \newcommand*{\l@subsection}{%  
2900 \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw  
2901 \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}  
2902 \newcommand*{\l@subsubsection}{%  
2903 \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw  
2904 \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}  
2905 \newcommand*{\l@paragraph}{%  
2906 \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw  
2907 \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
```

```

2908 \newcommand*{\l@subparagraph}{%
2909         \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2910         \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2911 %</book|report>

```

`\numberline` 欧文版 L<sup>A</sup>T<sub>E</sub>X 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。

```

2912 \newdimen\@lnumwidth
2913 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

`\@dottedtocline` L<sup>A</sup>T<sub>E</sub>X 本体 (ltsect.dtx 参照) での定義と同じですが、`\@tempdima` を `\@lnumwidth` に  
`\jsTocLine` 変えています。

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

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

とします。

```

2914 \def\jsTocLine{\leaders\hbox{%
2915     $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$\hfill}
2916 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2917     \vskip \z@ \@plus.2\p@?
2918     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2919     \parindent #2\relax\@afterindenttrue
2920     \interlinepenalty\@M
2921     \leavevmode
2922     \@lnumwidth #3\relax
2923     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2924     {#4}\nobreak
2925     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2926         \hfil\normalfont \normalcolor #5}\par}\fi}

```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2927 \newcommand{\listoffigures}{%
2928 %<*book|report>
2929     \if@twocolumn\@restonecoltrue\onecolumn
2930     \else\@restonecolfalse\fi
2931     \chapter*{\listfigurename}%
2932     \@mkboth{\listfigurename}{}%
2933 %</book|report>
2934 %<!*book&!report>
2935     \section*{\listfigurename}%
2936     \@mkboth{\listfigurename}{\listfigurename}%

```



```

2937 %</!book&!report>
2938 \@starttoc{lof}%
2939 %<book|report> \if@restonecol\twocolumn\fi
2940 }

```

`\l@figure` 図目次の項目を出力します。

```

2941 \newcommand{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2942 \newcommand{\listoftables}{%
2943 %<*book|report>
2944 \if@twocolumn\@restonecoltrue\onecolumn
2945 \else\@restonecolfalse\fi
2946 \chapter*{\listtablename}%
2947 \@mkboth{\listtablename}{}%
2948 %</book|report>
2949 %<*!book&!report>
2950 \section*{\listtablename}%
2951 \@mkboth{\listtablename}{\listtablename}%
2952 %</!book&!report>
2953 \@starttoc{lot}%
2954 %<book|report> \if@restonecol\twocolumn\fi
2955 }

```

`\l@table` 表目次は図目次と同じです。

```

2956 \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。

```

2957 \newdimen\bibindent
2958 \setlength\bibindent{2\jsZw}

```

`thebibliography` (*env.*) 参考文献リストを出力します。

[2016-07-16] L<sup>A</sup>T<sub>E</sub>X 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。

```

2959 \newenvironment{thebibliography}[1]{%
2960 \@jsc@warnoldfontcmdexceptiontrue
2961 \global\let\presectionname\relax
2962 \global\let\postsectionname\relax
2963 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2964 %<*kiyou>
2965 \vspace{1.5\baselineskip}
2966 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2967 \vspace{0.5\baselineskip}
2968 %</kiyou>

```

```

2969 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2970 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2971 \list{\@biblabel{\@arabic{c@enumiv}}}%
2972     {\settowidth\labelwidth{\@biblabel{#1}}%
2973     \leftmargin\labelwidth
2974     \advance\leftmargin\labelsep
2975     \@openbib@code
2976     \usecounter{enumiv}%
2977     \let\p@enumiv\empty
2978     \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2979 %<kiyou> \small
2980 \sloppy
2981 \clubpenalty4000
2982 \@clubpenalty\clubpenalty
2983 \widowpenalty4000%
2984 \sfcode`.\@m}
2985 {\def\@noitemerr
2986     {\@latex@warning{Empty `thebibliography' environment}}}%
2987 \endlist}

```

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

```

2988 \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。

```

2989 \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。

```

2990 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

`\cite` 文献の番号を出力する部分は ltbib1.dtx で定義されていますが、コンマとカッコを和文 `\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必要に応じて生かしてください。かつこの前後に入るグルーを `\jsInhibitGlue` で取りますので、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。

```

2991 % \def\@citex[#1]#2{\leavevmode
2992 %     \let\@citea\empty
2993 %     \@cite{\@for\@citeb:=#2\do
2994 %         {\@citea\def\@citea{, \inhibitglue\penalty\@m }}%
2995 %         \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2996 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2997 %     \ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2998 %         \G@refundefinedtrue
2999 %         \@latex@warning
3000 %             {Citation `'\@citeb' on page \thepage \space undefined}}%
3001 %         {\@cite\ofmt{\csname b@\@citeb\endcsname}}}{#1}}
3002 % \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。

```
3003 % \DeclareRobustCommand\cite{\unskip
3004 %   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]
3005 % \def\@cite#1#2{$\hbox{\scriptsize{#1}\if@tempswa
3006 %   , \jsInhibitGlue\ #2\fi)} }$}
```

## 10.3 索引

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

```
3007 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
3008   \if@twocolumn
3009     \onecolumn\@restonecolfalse
3010   \else
3011     \clearpage\@restonecoltrue
3012   \fi
3013   \columnseprule.4pt \columnsep 2\jsZw
3014   \ifx\multicols\@undefined
3015 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
3016 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]]%
3017 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
3018 %<!book&!report>    \twocolumn[\section*{\indexname}]]%
3019   \else
3020     \ifdim\textwidth<\fullwidth
3021       \setlength{\evensidemargin}{\oddsidemargin}
3022       \setlength{\textwidth}{\fullwidth}
3023       \setlength{\linewidth}{\fullwidth}
3024 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
3025 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]]%
3026 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
3027 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]]%
3028   \else
3029 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}%
3030 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]]%
3031 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
3032 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]]%
3033   \fi
3034   \fi
3035 %<book|report>      \@mkboth{\indexname}{}%
3036 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
3037   \plainifnotempty % \thispagestyle{plain}
3038   \parindent\z@
3039   \parskip\z@ \@plus .3\p@?\relax
3040   \let\item\@idxitem
3041   \raggedright
3042   \footnotesize\narrowbaselines
```

```

3043 }{
3044   \ifx\multicols\@undefined
3045     \if@restonecol\onecolumn\fi
3046   \else
3047     \end{multicols}
3048   \fi
3049   \clearpage
3050 }

```

`\@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 3051 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 3052 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
3053 \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。

```

3054 \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。

```

3055 \newcommand\seename{\if@english see\else →\fi}
3056 \newcommand\alsoname{\if@english see also\else →\fi}

```

## 10.4 脚注

`\footnote` 和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため、`\footnotemark` `\inhibitglue` を入れることにします。pL<sup>A</sup>T<sub>E</sub>X 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。

---

```

3057 \ifx\pltx@foot@penalty\@undefined
3058   \let\footnotes@ve=\footnote
3059   \def\footnote{\jsInhibitGlue\footnotes@ve}
3060   \let\footnotemarks@ve=\footnotemark
3061   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
3062 \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 `\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<sub>E</sub>X では脚注番号のまわりにスペースが入りすぎることを防ぐため、北川さんのパッチ [qa:57090] を取り込みました。

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

[2016-07-11] コミュニティ版 p $\text{\LaTeX}$  の変更に従いました (Thanks: 角藤さん)。p $\text{\LaTeX}$  の日付が 2016/04/17 より新しい場合は、このパッチが不要なのであてません。

---

p $\text{\TeX}$  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 `newp $\text{\TeX}$ text` 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 `defaultsup $\text{\TeX}$ s` オプションを付けてください (qa:57284, qa:57287)。

```
3063 \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。

```
3064 % \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。

```
3065 \renewcommand{\footnoterule}{%
3066   \kern-2.6\p@? \kern-.4\p@
3067   \hrule width .4\columnwidth
3068   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
3069 %<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 $\text{\E}$ X and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9)

[2016-08-25] コミュニティ版 p $\text{\LaTeX}$  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 $\text{\LaTeX}$  のバグ修正に従いました。

[2016-11-29] 古い p $\text{\LaTeX}$  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。

[2022-09-13] L $\text{\A}$ T $\text{\E}$ X 2 $\epsilon$  2021-11-15 (l $\text{\f}$ loat.d $\text{\t}$ x 2021/10/14 v1.2g) で `\@currentcounter` が追加されましたので、追従します。なお、L $\text{\A}$ T $\text{\E}$ X 2 $\epsilon$  2021-06-01 (l $\text{\f}$ loat.d $\text{\t}$ x 2021/02/10 v1.2e) で `parhook` 対応として `\par` が追加されていますが、実は同時に `\color@endgroup` も `\endgraf` するように変更されていますので、不要だと思います。というわけで追加しません。

```
3070 \long\def\@footnotetext{%
3071   \insert\footins\bgroup
```

```

3072 \normalfont\footnotesize
3073 \interlinepenalty\interfootnotelinepenalty
3074 \splittopskip\footnotesep
3075 \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
3076 \hsize\columnwidth \@parboxrestore
3077 \def\@currentcounter{footnote}%
3078 \protected@edef\@currentlabel{%
3079     \csname p@footnote\endcsname\@thefnmark
3080 }%
3081 \color@begingroup
3082 \@makefnmark{%
3083     \rule{z@\footnotesep\ignorespaces}%
3084     \futurelet\jsc@next\jsc@fo@t}
3085 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@fo@t
3086     \else \let\jsc@next\jsc@f@t\fi \jsc@next}
3087 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3088 \def\jsc@f@t#1{\#1\jsc@@foot}
3089 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
3090     \ifx\pltx@foot@penalty\@undefined\else
3091         \ifhmode\null\fi
3092         \ifnum\pltx@foot@penalty=z@\else
3093             \penalty\pltx@foot@penalty
3094             \pltx@foot@penaltyz@
3095         \fi
3096     \fi}

```

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

```

3097 \newcommand\@makefnmark[1]{%
3098     \advance\leftskip 3\jsZw
3099     \parindent 1\jsZw
3100     \noindent
3101     \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。

```

3102 % \def\@xfootnotenext[#1]{%
3103 %     \begingroup
3104 %         \ifnum#1>z@
3105 %             \csname c@\mpfn\endcsname #1\relax
3106 %             \unrestored@protected@xdef\@thefnmark{\thempfn}%
3107 %         \else
3108 %             \unrestored@protected@xdef\@thefnmark{}%
3109 %         \fi

```

```

3110 % \endgroup
3111 % \@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。

```

3112 \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。

```

3113 \def\everyparhook{\jsInhibitGlueAtParTop}
3114 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
3115 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3116 \fi

```

---

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

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

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

---

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

---

```

3117 \ifx j\jsEngine
3118 \def\@inhibitglue{%
3119   \futurelet\@let@token\@@inhibitglue}
3120 \begingroup
3121 \let\GDEF=\gdef
3122 \let\CATCODE=\catcode
3123 \let\ENDGROUP=\endgroup
3124 \CATCODE`k=12
3125 \CATCODE`a=12
3126 \CATCODE`n=12

```

```

3127 \CATCODE`j=12
3128 \CATCODE`i=12
3129 \CATCODE`c=12
3130 \CATCODE`h=12
3131 \CATCODE`r=12
3132 \CATCODE`t=12
3133 \CATCODE`e=12
3134 \GDEF\KANJI@CHARACTER{kanji character }
3135 \ENDGROUP
3136 \def\@@inhibitglue{%
3137   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@
3138   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\j
3139   \def\jsc@ig@temp{#1}%
3140   \ifx\jsc@ig@temp\@empty
3141     \ifnum\the\inhibitxspcode`#2=2\relax
3142       \inhibitglue
3143     \fi
3144   \fi}
3145 \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。

---

```

3146 \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。

```

3147 \def\@doendpe{%
3148   \@endpetrue
3149   \def\par{%
3150     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3151   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3152 \def\@setminipage{%
3153   \@minipagetrue
3154   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3155 }

```

`\item` 命令の直後です。

```

3156 \def\@item[#1]{%
3157   \if@noparitem
3158     \@donoparitem
3159   \else
3160     \if@inlabel

```



```

3161     \indent \par
3162 \fi
3163 \ifhmode
3164     \unskip\unskip \par
3165 \fi
3166 \if@newlist
3167     \if@nobreak
3168         \@nbitem
3169     \else
3170         \addpenalty\@beginparpenalty
3171         \addvspace\@topsep
3172         \addvspace{-\parskip}%
3173     \fi
3174 \else
3175     \addpenalty\@itempenalty
3176     \addvspace\itemsep
3177 \fi
3178 \global\@inlabeltrue
3179 \fi
3180 \everypar{%
3181     \@minipagefalse
3182     \global\@newlistfalse
3183     \if@inlabel
3184         \global\@inlabelfalse
3185         {\setbox\z@\lastbox
3186         \ifvoid\z@
3187             \kern-\itemindent
3188         \fi}%
3189     \box\@labels
3190     \penalty\z@
3191 \fi
3192 \if@nobreak
3193     \@nobreakfalse
3194     \clubpenalty \@M
3195 \else
3196     \clubpenalty \@clubpenalty
3197     \everypar{\everyparhook}%
3198 \fi
3199 \everyparhook}%
3200 \if@noitemarg
3201     \@noitemargfalse
3202     \if@nmbrlist
3203         \refstepcounter\@listctr
3204     \fi
3205 \fi
3206 \sbox\@tempboxa{\makelabel{#1}}%
3207 \global\setbox\@labels\hbox{%
3208     \unhbox\@labels
3209     \hskip \itemindent

```

```

3210 \hskip -\labelwidth
3211 \hskip -\labelsep
3212 \ifdim \wd\@tempboxa >\labelwidth
3213 \box\@tempboxa
3214 \else
3215 \hbox to\labelwidth {\unhbox\@tempboxa}%
3216 \fi
3217 \hskip \labelsep}%
3218 \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。

```

3219 \def\@afterheading{%
3220 \@nobraektrue
3221 \everypar{%
3222 \if@nobraek
3223 \@nobraekfalse
3224 \clubpenalty \@M
3225 \if@afterindent \else
3226 {\setbox\z@\lastbox}%
3227 \fi
3228 \else
3229 \clubpenalty \@clubpenalty
3230 \everypar{\everyparhook}%
3231 \fi\everyparhook}}

```

---

「`\everyparhook` 用のパッチ処理」はここまで。

---

```
3232 \fi
```

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

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

---

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

---

```

3233 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none\else
3234 \AtEndOfPackage{%
3235 \def\@gnewline #1{%
3236 \ifvmode
3237 \@nolnerr
3238 \else
3239 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null

```

```

3240 \jsInhibitGlue \ignorespaces
3241 \fi}
3242 }
3243 \fi

```

## 12 いろいろなロゴ

L<sup>A</sup>T<sub>E</sub>X 関連のロゴを作り直します。

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

---

BXJS クラスでも jslogo オプション指定の場合に jslogo パッケージを読み込むようにした。ただし JS クラスと異なり、既定では読み込まない。

※\小、\上小 の制御綴は定義しない。

---

```

3244 \if@jslogo
3245 \IfFileExists{jslogo.sty}{%
3246 \RequirePackage{jslogo}%
3247 }{%
3248 \ClassWarningNoLine{bxjs}{clsname
3249 {The package 'jslogo' is not installed.\MessageBreak
3250 It is included in the recent release of\MessageBreak
3251 the 'jsclasses' bundle}
3252 }
3253 \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<sup>A</sup>T<sub>E</sub>X の \ProvidesFile \ProvidesFile で悪さをする例が F<sub>T</sub>E<sub>X</sub> で報告されています。これを避けるための tDB さんのフィックスを挿入しておきます。副作用がありましたらお知らせください。

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

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

```

3254 %\let\ltx@ifnextchar\@ifnextchar
3255 %\def\ProvidesFile#1{%
3256 % \begingroup
3257 % \catcode\ 10 %
3258 % \ifnum \endlinechar<256 %
3259 % \ifnum \endlinechar>\m@ne

```

```

3260 %      \catcode\endlinechar 10 %
3261 %      \fi
3262 %      \fi
3263 %      \@makeother\/%
3264 %      \@makeother\&%
3265 %      \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 3266 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3267 \newcommand{\postpartname}{\if@english\else 部\fi}
\prechaptername 3268 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3269 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3270 \newcommand{\presectionname}{}% 第
\postsectionname 3271 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 3272 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3273 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3274 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3275 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3276 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3277 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3278 %<!\jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3279 %<jspf>\newcommand{\figurename}{Fig.~}
3280 %<!\jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3281 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3282 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3283 \newcommand{\appendixname}{\if@english \else 付録\fi}
3284 %<!\book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

■今日の日付  $\text{\LaTeX}$  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。

---

```
3285 \@onlypreamble\bxjs@decl@Seireki@cmds
3286 \@tempswafalse
3287 \if p\jsEngine \@tempwattrue \fi
3288 \if n\jsEngine \@tempwattrue \fi
3289 \bxjs@cond\if@tempswa\fi{%
3290 % 欧文 8bitTeX の場合
3291 \newif\ifjsSeireki \jsSeirekitrue
3292 \def\bxjs@decl@Seireki@cmds{%
3293   \def\西暦{\jsSeirekitrue}%
3294   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3295 \def\Seireki{\jsSeirekitrue}
3296 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3297 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3298 \def\bxjs@iai{\noexpand~}
3299 }{%
3300 \newif\if 西暦 \西暦 true
3301 \def\bxjs@decl@Seireki@cmds{%
3302   \def\西暦{\西暦 true}%
3303   \def\和暦{\西暦 false\bxjs@wareki@used}}
3304 \def\Seireki{\西暦 true}
3305 \def\Wareki{\西暦 false\bxjs@wareki@used}
3306 \def\bxjs@if@use@seireki{\bxjs@cond\if 西暦\fi}
3307 \let\bxjs@iai\@empty
3308 }
3309 \bxjs@decl@Seireki@cmds
3310 \let\bxjs@unxp\@firstofone \let\bxjs@onxp\@firstofone
3311 \bxjs@test@engine\unexpanded{%
3312   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}
```

\ifbxjs@bxwareki@avail bxwareki パッケージが利用できるか。

```
3313 \newif\ifbxjs@bxwareki@avail
3314 \IfFileExists{bxwareki.sty}{%
3315   \RequirePackage{bxwareki}[]%
3316   \bxjs@bxwareki@availtrue}{}
```

\bxjs@wareki@used 和暦が非対応の場合に警告を出す。

```
3317 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\@empty
3318 \else
3319   \bxjs@robust@def\bxjs@wareki@used{%
3320     \global\let\bxjs@wareki@used\@empty
3321     \ClassWarning\bxjs@clsname
3322       {Wareki mode is not supported, since\MessageBreak
3323        'bxwareki' is unavailable, reported}}
3324   \g@addto@macro\bxjs@begin@document@hook{%
3325     \let\bxjs@wareki@used\@empty}
3326 \fi
```

`\jayear` 和暦における年の表記の「年」以前の部分（元号 + 年数）。

※`\heisei` の代替となる機能（だから常に和暦を扱う）。

`\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。

※ JS クラスと互換の機能。

```
3327 \ifbxjs@bxwareki@avail
3328   \let\jayear\warekiyear
3329   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3330     \newcount\heisei \heisei=\value{warekiyear}
3331   \fi
```

ただし `bxwareki` が使えない場合は西暦表示にフォールバックする。

```
3332 \else
3333   \edef\jayear{\the\year \bxjs@iai}
3334 \fi
```

`\today` 英語、西暦、和暦で場合分けをする。

```
3335 \let\bxjs@next\relax
3336 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3337   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3338 \fi\fi
3339 \edef\bxjs@today{%
3340   \if@english
3341     \ifcase\month\or
3342       January\or February\or March\or April\or May\or June\or
3343       July\or August\or September\or October\or November\or December\fi
3344     \space\number\day, \number\year
3345   \else
3346     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3347     \else \noexpand\bxjs@if@use@seireki
3348     \fi {%
3349       \number\year\bxjs@iai\bxjs@unxp{年}%
3350       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3351       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3352     }\bxjs@next}%
3353   \fi}
3354 \let\today\bxjs@today
```

---

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

```
3355 \g@addto@macro\bxjs@begin@document@hook{%
3356   \ifx\bb1@jpn@maybekansuji\undefined\else
3357     \bxjs@decl@Seireki@cmds
3358     \g@addto@macro\datejapanese{%
3359       \let\today\bxjs@today}%
3360   \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）

```
3361 \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。

```
3362 %<slide>\pagestyle{empty}%  
3363 %<article|report>\pagestyle{plain}%  
3364 %<book>\pagestyle{headings}%  
3365 \pagenumbering{arabic}  
3366 \if@twocolumn  
3367   \twocolumn  
3368   \sloppy  
3369   \flushbottom  
3370 \else  
3371   \onecolumn  
3372   \raggedbottom  
3373 \fi  
3374 %<*slide>  
3375 \renewcommand\familydefault{\sfdefault}  
3376 \raggedright  
3377 %</slide>
```

■BXJS 独自の追加処理 

---

フックを実行する。

```
3378 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3379 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

---

```
3380 %</class>
```

以上です。

## 付録 A 和文ドライバの仕様

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

- `\jsDocClass` [文字トークンの `let`] 文書クラスの種類を示し、次のいずれかと一致する (`\if` で判定可能)。
  - `\jsArticle` `bxjsarticle` クラス
  - `\jsBook` `bxjsbook` クラス
  - `\jsReport` `bxjsreport` クラス
  - `\jsSlide` `bxjsslide` クラス
- `\jsEngine` [文字トークンの `let`] 使用されているエンジンの種別。(`\if` で判定可能)。
  - `p` `pdfTeX` (DVI モードも含む)
  - `l` `LuaTeX` (〃)
  - `x` `XYTeX`
  - `j` `pTeX` または `upTeX`
  - `n` 以上の何れでもない
- `\ifjsWithupTeX` [スイッチ] 使用されているエンジンが `upTeX` であるか。
- `\ifjsWitheTeX` [スイッチ] 使用されているエンジンが  $\epsilon$ -`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 (呼び出すべき) 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。

```
3381 %<*drv>
```

## 付録 B 和文ドライバ：minimal

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

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

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

**TODO:** 本来は「minimal にすら依存しない」はずのものが minimal のコード中に書かれているような気がする……。

### B.1 補助マクロ

```
3382 %<*minimal>
```

```
3383 %% 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`。

```
3384 \def\DeclareJaTextFontCommand#1#2{%
3385   \DeclareRobustCommand#1[1]{%
3386     \relax
3387     \ifmmode \expandafter\nfss@text \fi
3388     {#2##1}}%
3389 }
```

`\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。

```
3390 \def\DeclareJaMathFontCommand#1#2{%
3391   \DeclareRobustCommand#1[1]{%
3392     \relax
3393     \ifmmode\else \non@alpherr{#1\space}\fi
```

```

3394 \nfss@text{\fontfamily\familydefault
3395 \fontseries{m}\fontshape{n}\selectfont\relax
3396 #2##1}%
3397 }%
3398 }

```

`\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。

```

3399 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3400 \@onlypreamble\bxjs@if@sf@default
3401 \def\bxjs@if@sf@default#1{%
3402 \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3403 \g@addto@macro\bxjs@begin@document@hook{%
3404 \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3405 }

```

`\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。

```

3406 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3407 \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。

```

3408 \def\jsLetHeadChar#1#2{%
3409 \begingroup
3410 \escapechar=`\ %
3411 \let\bxjs@tmpa={% brace-match-hack
3412 \bxjs@let@hchar@exp#2}%
3413 \endgroup
3414 \let#1\bxjs@g@tmpa}
3415 \def\bxjs@let@hchar@exp{%
3416 \futurelet\@let@token\bxjs@let@hchar@exp@a}
3417 \def\bxjs@let@hchar@exp@a{%
3418 \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
3419 \bxjs@let@hchar@out\let\relax
3420 }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi{% 空白
3421 \bxjs@let@hchar@out\let\space%
3422 }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
3423 \bxjs@let@hchar@out\let\@backslashchar
3424 }{\bxjs@let@hchar@exp@b}}}}
3425 \def\bxjs@let@hchar@exp@b#1{%
3426 \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}

```

```

3427 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3428 %\message{<#1#2>}%
3429 \bxjs@cond\if#1\@backslashchar\fi{% 制御綴
3430 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3431 \bxjs@let@hchar@out\let\relax
3432 }{%else
3433 \expandafter\bxjs@let@hchar@exp
3434 }%
3435 }{%else
3436 \bxjs@let@hchar@chr#1%
3437 }}
3438 \def\bxjs@let@hchar@chr#1{%
3439 \bxjs@let@hchar@out\def{{#1}}}
3440 \def\bxjs@let@hchar@out#1#2{%
3441 \global#1\bxjs@g@tmpa#2\relax
3442 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3443 \chardef\bxjs@let@hchar@csta=128
3444 \chardef\bxjs@let@hchar@cstb=192
3445 \chardef\bxjs@let@hchar@cstc=224
3446 \chardef\bxjs@let@hchar@cstd=240
3447 \chardef\bxjs@let@hchar@cste=248
3448 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3449 \def\bxjs@let@hchar@chr@ue#1{%
3450 \@tempcnta=`#1\relax
3451 %\message{\the\@tempcnta}%
3452 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3453 \bxjs@let@hchar@chr@ue@a#1%
3454 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3455 \bxjs@let@hchar@out\let\relax
3456 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3457 \bxjs@let@hchar@chr@ue@b
3458 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3459 \bxjs@let@hchar@chr@ue@c
3460 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3461 \bxjs@let@hchar@chr@ue@d
3462 }{%else
3463 \bxjs@let@hchar@out\let\relax
3464 }}}}}}
3465 \def\bxjs@let@hchar@chr@ue@a#1{%
3466 \bxjs@let@hchar@out\def{{#1}}}
3467 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3468 \bxjs@let@hchar@out\def{{#1#2}}}
3469 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3470 \bxjs@let@hchar@out\def{{#1#2#3}}}
3471 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3472 \bxjs@let@hchar@out\def{{#1#2#3#4}}}

```

## B.2 (u)pTeX 用の設定

```
3473 \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 UTF-8 バイト列と和文文字トークンに対応させる。

```
3474 \def\bxjs@let@hchar@chr@pp#1#2{%
3475   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3476 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3477   %\message{(\meaning#3:\meaning#4)}%
3478   \bxjs@cond@if#1k\fi{%
3479     \bxjs@let@hchar@out\def{{#4}}%
3480   }{%else
3481     \bxjs@let@hchar@chr@ue#3#4%
3482   }}
3483 \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。

```
3484 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3485 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3486 \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。

```
3487 \@onlypreamble\bxjs@declarefontshape
3488 \ifjsWithupTeX
3489 \def\bxjs@declarefontshape{%
3490   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3491   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3492   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3493   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3494 }
3495 \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。

```
3496 \else
3497 \def\bxjs@declarefontshape{%
```

```

3498 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3499 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3500 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
3501 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3502 }
3503 \def\bxjs@sizereference{jis}
3504 \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。

```

3505 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3506   \def\bxjs@tmpb{#5}}
3507 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3508 \else
3509 \expandafter\expandafter\expandafter\bxjs@next
3510 \expandafter\string\the\jfont\relax
3511 \fi
3512 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3513                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3514   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3515    \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。

```

3516 \begingroup
3517 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3518 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3519 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3520 % 幅が丁度 10pt なら補正は不要
3521 \ifdim\wd\z@=10pt
3522   \global\let\bxjs@scale\jsScale
3523 \else
3524 %  $(10*s)/(10*f)$  として計算、\bxjs@invscale は BXJS で定義
3525   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3526   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3527   \bxjs@invscale\@tempdima\bxjs@tmpa
3528   \xdef\bxjs@scale{\strip@pt\@tempdima}
3529 \fi
3530 \endgroup
3531 %\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。

```

3532 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3533 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{ }
3534 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
3535 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
3536 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
3537 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
3538 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
3539 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
3540 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
3541 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
3542 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
3543 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
3544 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
3545 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
3546 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
3547 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
3548 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{ }
3549 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
3550 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
3551 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
3552 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
3553 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
3554 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
3555 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
3556 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
3557 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
3558 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
3559 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
3560 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
3561 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
3562 \DeclareFontShape{\jsc@JTn}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }

```

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

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

```

3563 \@ifl@t@r\fmtversion{2020/10/01}
3564   {\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
3565 \ifjsc@needspace@tch % --- for 2020-02-02 or older BEGIN
3566 \ifx\@rmfamilyhook\@undefined % old
3567 \DeclareRobustCommand\rmfamily
3568   {\not@math@alphabet\rmfamily\mathrm
3569    \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3570 \DeclareRobustCommand\sffamily
3571   {\not@math@alphabet\sffamily\mathsf
3572    \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
3573 \DeclareRobustCommand\ttfamily
3574   {\not@math@alphabet\ttfamily\mathtt
3575    \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3576 \g@addto@macro\bxjs@begin@document@hook{%
3577   \ifx\mweights@init\@undefined\else % mweights.sty is loaded

```

```

3578 % my definitions above should have been overwritten, recover it!
3579 % \selectfont is executed twice but I don't care about speed...
3580 \expandafter\g@addto@macro\csname rmfamily \endcsname
3581   {\kanjifamily\mcdefault\selectfont}%
3582 \expandafter\g@addto@macro\csname sffamily \endcsname
3583   {\kanjifamily\gtdefault\selectfont}%
3584 \expandafter\g@addto@macro\csname ttfamily \endcsname
3585   {\kanjifamily\gtdefault\selectfont}%
3586 \fi}
3587 \else % 2020-02-02
3588 \g@addto@macro\@rmfamilyhook
3589   {\prepare@family@series@update@kanji{mc}\mcdefault}
3590 \g@addto@macro\@sffamilyhook
3591   {\prepare@family@series@update@kanji{gt}\gtdefault}
3592 \g@addto@macro\@ttfamilyhook
3593   {\prepare@family@series@update@kanji{gt}\gtdefault}
3594 \fi
3595 \else % --- for 2020-02-02 or older END & for 2020-10-01 BEGIN
3596 \AddToHook{rmfamily}%
3597   {\prepare@family@series@update@kanji{mc}\mcdefault}
3598 \AddToHook{sffamily}%
3599   {\prepare@family@series@update@kanji{gt}\gtdefault}
3600 \AddToHook{ttfamily}%
3601   {\prepare@family@series@update@kanji{gt}\gtdefault}
3602 \fi % --- for 2020-10-01 END
3603 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3604 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3605 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3606 \fi
3607 \bxjs@if@sf@default{%
3608   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3609 \selectfont

```

これ以降では、`\bxjs@parse@qh` の処理は p<sub>T</sub>E<sub>X</sub> 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）。

```

3610 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3611 \let\bxjs@parse@qh@a\@undefined
3612 \let\bxjs@parse@qh@b\@undefined

```

## ■パラメタの設定

```

3613 \prebreakpenalty\jis"2147=10000
3614 \postbreakpenalty\jis"2148=10000
3615 \prebreakpenalty\jis"2149=10000
3616 \inhibitxspcode`!=1
3617 \inhibitxspcode`〒=2
3618 \xspcode`+=3
3619 \xspcode`%=3

```

"80～"FF の範囲の \spcode を 3 に変更。

```
3620 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%  
3621 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
```

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

```
3622 \let\jsInhibitGlueAtParTop\@inhibitglue
```

\jsResetDimen は空のままでよい。

■組方向依存の処理 組方向判定の if-トークン (\if?dir) は pTeX 以外では未定義であるため、そのまま if 文に入れることができない。これを回避するため部分的に!をエスケープ文字に使う。

```
3623 \begingroup  
3624 \catcode`\!=0
```

\bxjs@ptex@dir 現在の組方向：t=縦、y=横、?=その他。

```
3625 \gdef\bxjs@ptex@dir{%  
3626 !iftdir t%  
3627 !else!ifydir y%  
3628 !else ?%  
3629 !fi!fi}
```

新版の pTeX で脚注番号の周囲の空きが過大になる現象への対処。

※現在の pLaTeX カーネルでは対処が既に行われている。ここでは、\@makefnmark の定義が古いものであった場合に、新しいものに置き換える。

```
3630 % 古い \@makefnmark の定義  
3631 \long\def\bxjs@tmpa{\hbox{%  
3632 !ifydir \@textsuperscript{\normalfont\@thefnmark}}%  
3633 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}  
3634 \ifx\@makefnmark\bxjs@tmpa  
3635 \long\gdef\@makefnmark{%  
3636 !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%  
3637 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}  
3638 \fi
```

エスケープ文字の変更はここまで。

```
3639 \endgroup
```

■minijs パッケージのブロック やっておく。

```
3640 \@namedef{ver@minijs.sty}{}
```

## B.3 pdfTeX 用の処理

```
3641 \else\if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
```

\jsLetHeadChar を UTF-8 バイト列に対応させる。

```
3642 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

ムニャムニャ。



```

3643 \@onlypreamble\bxjs@cjk@loaded
3644 \def\bxjs@cjk@loaded{%
3645   \def\@footnotemark{%
3646     \leavevmode
3647     \ifhmode
3648       \edef\@x@sf{\the\spacefactor}%
3649       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3650         \unkern\unkern
3651         \ifdim\lastskip>\z@ \unskip \fi
3652       \fi\fi
3653       \nobreak
3654     \fi
3655     \@makefnmark
3656     \ifhmode \spacefactor\@x@sf \fi
3657   \relax}%
3658 \let\bxjs@cjk@loaded\relax
3659 }
3660 \g@addto@macro\bxjs@begin@document@hook{%
3661   \@ifpackageloaded{CJK}{%
3662     \bxjs@cjk@loaded
3663   }{}%
3664 }

```

## B.4 X<sub>3</sub>TeX 用の処理

```

3665 \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。

```

3666 \def\bxjs@let@hchar@chr#1{%
3667   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3668   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3669     \bxjs@let@hchar@chr@xe
3670   }\bxjs@let@hchar@out\def{#{1}}}}
3671 \def\bxjs@let@hchar@chr@xe#1{%
3672   \lccode`0=`#1\relax
3673   \lowercase{\bxjs@let@hchar@out\def{#{0}}}}

```

\bxjs@do@precisetext precisetext オプションの実際の処理内容。

```

3674 \@onlypreamble\bxjs@do@precisetext
3675 \ifx\XeTeXgenerateactualtext\undefined\else
3676   \def\bxjs@do@precisetext{%
3677     \XeTeXgenerateactualtext=\@one}
3678 \fi

```

\bxjs@do@simplejasetup simplejasetup オプションの実際の処理内容。

**TODO:** バージョン要件を見直して暫定措置を解除する。

```

3679 \@onlypreamble\bxjs@do@simplejasetup
3680 \def\bxjs@do@simplejasetup{%
3681   \@namedef{bxjs@zeroglue/0.0pt}{T}%

```

```

3682 \ifnum\XeTeXinterchartokenstate>\z@
3683 \else\expandafter\ifx\csname bxjs@zeroglue/\the\XeTeXlinebreakskip\endcsname\relax\else
3684 \jsSimpleJaSetup
3685 \ClassInfo\bxjs@clsname
3686 {\string\jsSimpleJaSetup' is applied\@gobble}%
3687 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```

3688 \newcommand*{\jsSimpleJaSetup}{%
3689 \XeTeXlinebreaklocale "ja"\relax
3690 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3691 \XeTeXlinebreakpenalty=0\relax}

```

## B.5 後処理（エンジン共通）

```

3692 \fi\fi\fi

```

`simplejasetup` オプションの処理。

```

3693 \ifx\bxjs@do@simplejasetup\@undefined\else
3694 \g@addto@macro\bxjs@begin@document@hook{%
3695 \ifbxjs@simplejasetup
3696 \bxjs@do@simplejasetup
3697 \fi}
3698 \fi

```

`precisetext` オプションの処理。

```

3699 \ifbxjs@precisetext
3700 \ifx\bxjs@do@precisetext\@undefined
3701 \ClassWarning\bxjs@clsname
3702 {The current engine does not support the\MessageBreak
3703 'precise-text' option\@gobble}
3704 \else
3705 \bxjs@do@precisetext
3706 \fi
3707 \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。

```

3708 \g@addto@macro\bxjs@begin@document@hook{%
3709 \ifx\jsInhibitGlueAtParTop\@empty
3710 \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3711 \ifx\everyparhook\bxjs@tmpa
3712 \let\everyparhook\@empty
3713 \fi
3714 \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。

```

3715 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@modern

```

まず `\everypar` を“乗っ取る”処理を行う。

```
3716 \let\bxjs@everypar\everypar
3717 \newtoks\everypar
3718 \everypar\bxjs@everypar
```

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

```
3719 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3720 \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。

```
3721 \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。

```
3722 \@onlypreamble\bxjs@adjust@fancyhdr
3723 \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。

```
3724 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3725 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3726 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3727 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3728 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3729 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3730 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3731 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3732 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3733 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3734 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3735 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3736 \def\bxjs@tmpa{\rm\thepage\strut}%
3737 \def\bxjs@tmpb{\thepage\strut}%
3738 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3739 \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。

```
3740 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth
3741 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3742 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3743 }\bxjs@tmpa
3744 \fi\fi
```

```

3745 \PackageInfo\bxjs@clsname
3746 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3747 \def\bxjs@pagestyle@hook{%
3748   \@ifpackageloaded{fancyhdr}{%
3749     \bxjs@adjust@fancyhdr
3750     \global\let\bxjs@adjust@fancyhdr\relax
3751   }{}}

```

`\pagestyle` にフックを入れ込む。

```

3752 \let\bxjs@org@pagestyle\pagestyle
3753 \def\pagestyle{%
3754   \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。

```

3755 \g@addto@macro\bxjs@begin@document@hook{%
3756   \bxjs@pagestyle@hook
3757   \global\let\bxjs@pagestyle@hook\relax}

```

終わり。

```

3758 \fi

```

## ■和文空白命令

```

3759 \ifbxjs@jaspace@cmd

```

`\jaenspace` 半角幅の水平空き。

```

3760 \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。

```

3761 \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。

```

3762 \def\_{\zwspace}

```

`\jaspace` `jlreq` クラスと互換の命令。

```

3763 \DeclareRobustCommand*{\jaspace}[1]{%
3764   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3765     \ClassError\bxjs@clsname
3766       {Unknown jaspac: #1}{\@eha}%
3767   \else
3768     \csname bxjs@jaspace@@#1\endcsname
3769   \fi}
3770 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3771 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3772 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}

```

終わり。

```
3773 \fi
```

以上で終わり。

```
3774 %</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。

```
3775 %<*standard>
```

```
3776 %% このファイルは日本語文字を含みます
```

```
3777 \input{bxjsja-minimal.def}
```

simplejasetup は standard では無効になる。

```
3778 \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。

```
3779 \newif\ifbxjs@jp@jismmiv
```

jis2004 オプションの処理。

```
3780 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
```

```
3781 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
```

```
3782 \define@key{bxjsStd}{jis2004}[true]{%
```

```
3783 \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。

```
3784 \newif\ifbxjs@jp@units
```

units オプションの処理。

```
3785 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3786 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3787 \define@key{bxjsStd}{units}[true]{%
3788   \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。

```
3789 \let\bxjs@jp@font\@empty
```

font オプションの処理。

```
3790 \define@key{bxjsStd}{font}{%
3791   \edef\bxjs@jp@font{#1}}
```

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

```
3792 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue
```

strong-cmd オプションの処理。

```
3793 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
3794 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
3795 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}
```

実際の jparam の値を適用する。

```
3796 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3797 \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。

```
3798 \@onlypreamble\bxjs@apply@mmiv
3799 \def\bxjs@apply@mmiv{%
3800   \g@addto@macro\@classoptionslist{,jis2004}
3801   % \ifpackagewith 判定への対策
3802   \PassOptionsToPackage{jis2004}{otf}
3803   \global\let\bxjs@apply@mmiv\relax}
3804 \ifbxjs@jp@jismmiv \bxjs@apply@mmiv \fi
```

■和文用単位のサポート エンジンが (u)pTeX の場合は units を無効にする。

```
3805 \if j\jsEngine
3806   \bxjs@jp@unitsfalse
3807 \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。

```

3808 \ifbxjs@jp@units
3809   \IfFileExists{bxcalc.sty}{%
3810     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3811     \ifx\usepTeXunits\@undefined
3812       \PackageWarningNoLine{bxjs@clsname}
3813         {Cannot support pTeX units (zw etc.), since\MessageBreak
3814           the package 'bxcalc' is too old}%
3815       \bxjs@jp@unitsfalse
3816     \else \usepTeXunits
3817     \fi
3818   }{%else
3819     \PackageWarningNoLine{bxjs@clsname}
3820       {Cannot support pTeX units (zw etc.), since\MessageBreak
3821         the package 'bxcalc' is unavailable}%
3822     \bxjs@jp@unitsfalse
3823   }
3824 \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。

```

3825 \ifbxjs@jp@units
3826 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3827 \let\bxjs@parse@qh@a\@undefined
3828 \let\bxjs@parse@qh@b\@undefined
3829 \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。

```

3830 \ifbxjs@jp@units
3831   \def\bxjs@let@lenexpr#1#2{%
3832     \edef#1{#2}%
3833     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3834 \else
3835   \def\bxjs@let@lenexpr{\edef}
3836 \fi

```

## ■\strong 命令の補填

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

※既に利用可能である場合は何もしない。

```

3837 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3838   \ifx\strong\@undefined\ifx\strongenv\@undefined
3839     \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3840     \DeclareTextFontCommand{\strong}{\strongenv}%

```

fontspec と互換の \strongfontdeclare 命令も提供する。既定の設定は \bfseries (太字) である。

※`\strongfontdeclare` は試験的機能とする。

```
3841 \newcommand*{\strongfontdeclare}{\bxjs@strongfontdeclare}%
3842 \newcount\bxjs@strong@level
3843 \bxjs@protected\def\bxjs@strongfontdeclare#1{%
3844   \bxjs@set@array@from@clist{\bxjs@strong}{#1}%
3845   \bxjs@strong@level\z@}%
3846 \bxjs@strongfontdeclare{\bfseries}%
3847 \def\bxjs@strong@font{%
3848   \bxjs@csletcs{\bxjs@tmpa}{\bxjs@strong/\the\bxjs@strong@level}%
3849   \ifx\bxjs@tmpa\relax
3850     \advance\bxjs@strong@level\m@ne \bxjs@strong@font
3851   \else \advance\bxjs@strong@level\@ne \bxjs@tmpa
3852   \fi}%
3853 \fi\fi
3854 }\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  $\varepsilon$ -TeX 拡張が使えるか検査する。

```
3855 \ifjsWithTeX
```

使える場合は、「`\dimexpr` 外部寸法表記`\relax`」の形式（これは内部値なので単位として使える）で各命令定義する。

`\jQ` `\jQ` と `\jH` はともに 0.25 mm に等しい。

```
\jH 3856 \@tempdima=0.25mm
3857 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3858 \let\jH\jQ
```

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

```
\trueH 3859 \ifjsc@mag
3860   \@tempdimb=\jsBaseFontSize\relax
3861   \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3862   \@tempdima=2.5mm
3863   \bxjs@invscale\@tempdima\bxjs@tmpa
3864   \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3865   \@tempdima=10pt
3866   \bxjs@invscale\@tempdima\bxjs@tmpa
3867   \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3868 \else \let\trueQ\jQ \let\bxjs@truept\p@
3869 \fi
3870 \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。

```
3871 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3872 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3873 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
```



```

3874 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3875 \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。

```

3876 \def\jafontsize#1#2{%
3877   \begingroup
3878     \bxjs@jafontsize@a{#1}%
3879     \@tempdimb\jsInverseScale\@tempdima
3880     \bxjs@jafontsize@a{#2}%
3881     \xdef\bxjs@g@tmpa{%
3882       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3883   \endgroup\bxjs@g@tmpa}
3884 \def\bxjs@jafontsize@a#1{%
3885   \bxjs@parse@qh{#1}%
3886   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3887   \@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。

```

3888 \def\bxjs@kanjiskip{0pt}

```

`\setkanjiskip` 和文間空白の量を設定する。

```

3889 \newcommand*\setkanjiskip[1]{%
3890   \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3891   \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3892 \newcommand*\getkanjiskip{%
3893   \bxjs@kanjiskip}

```

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

```

3894 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue

```

`\bxjs@enable@kanjiskip` 和文間空白の挿入を有効／無効にする。(pTeX 以外)

```

\bxjs@disable@kanjiskip 3895 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3896   \bxjs@kanjiskip@enabledtrue
3897   \bxjs@reset@kanjiskip}
3898 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3899   \bxjs@kanjiskip@enabledfalse
3900   \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。

```

3901 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3902   \ifbxjs@kanjiskip@enabled
3903     \setlength{\@tempskipa}{\bxjs@kanjiskip}%

```

```

3904 \else \@tempskipa\z@
3905 \fi
3906 \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 3907 \def\bxjs@xkanjiskip{0pt}
\getxkanjiskip 3908 \newcommand*\setxkanjiskip[1]{%
\ifbxjs@xkanjiskip@enabled 3909 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
3910 \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3911 \newcommand*\getxkanjiskip{%
3912 \bxjs@xkanjiskip}
\bxjs@disable@xkanjiskip 3913 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3914 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3915 \bxjs@xkanjiskip@enabledtrue
3916 \bxjs@reset@xkanjiskip}
3917 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3918 \bxjs@xkanjiskip@enabledfalse
3919 \bxjs@reset@xkanjiskip}
3920 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3921 \ifbxjs@xkanjiskip@enabled
3922 \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3923 \else \@tempskipa\z@
3924 \fi
3925 \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。

```

3926 \g@addto@macro\jsResetDimen{%
3927 \bxjs@reset@kanjiskip
3928 \bxjs@reset@xkanjiskip}
3929 \let\bxjs@apply@kanjiskip\relax
3930 \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、 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  Live の `kanji-config-updmap` コマンドで使う“ファミリ”と同じにすることを想定する。特別な値として、`auto` は `kanji-config-updmap` で現在指定されているファミリを表す。

`\bxjs@adjust@jafont` `\jsJaFont` に入っている和文フォント設定の値を“調整”して、その結果を `\bxjs@tmpa` に返す。`#1` が `f` の場合は“非埋込 (noEmbed)”の設定が禁止される。この禁止の場合も含め、何か異常がある場合は `\bxjs@tmpa` は空になる。

```

3931 \@onlypreamble\bxjs@adjust@jafont
3932 \def\bxjs@adjust@jafont#1{%
3933 \ifx\jsJaFont\bxjs@@auto
3934 \bxjs@get@kanjiEmbed
3935 \ifx\bxjs@jaEmbed\relax
3936 \let\bxjs@tmpa\@empty
3937 \else

```

```

3938 \let\bxjs@tmpa\bxjs@jaEmbed
3939 \ifx\bxjs@jaVariant\bxjs@@hziv
3940 \bxjs@apply@mmiv
3941 \fi
3942 \fi
3943 \else
3944 \let\bxjs@tmpa\jsJaFont
3945 \fi
3946 \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3947 \ClassWarningNoLine\bxjs@clsname
3948 {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3949 not available on the current situation}%
3950 \let\bxjs@tmpa@empty
3951 \fi\fi
3952 }
3953 \def\bxjs@@auto{auto}
3954 \def\bxjs@@noEmbed{noEmbed}
3955 \def\bxjs@@hziv{-04}

```

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

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

```

3956 \let\bxjs@jaEmbed\relax
3957 \let\bxjs@jaVariant\relax

```

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

```

3958 \@onlypreamble\bxjs@get@kanjiEmbed
3959 \def\bxjs@get@kanjiEmbed{%
3960 \begingroup\setbox\z@=\hbox{%
3961 \global\let\bxjs@tmpdo@empty
3962 \def\bxjs@next##1##2##3{%
3963 \def##1####1##3 ####2\@nil####3\@nnil{%
3964 \ifx$####1$\gdef##2{####2}\fi}%
3965 \g@addto@macro\bxjs@tmpdo{%
3966 \expandafter##1\bxjs@tmpa\@nil##3 \@nil\@nnil}}%
3967 \bxjs@next\bxjs@tmpdo@a\bxjs@g@tmpa{kanjiEmbed}%
3968 \bxjs@next\bxjs@tmpdo@b\bxjs@g@tmpa{jaEmbed}%
3969 \bxjs@next\bxjs@tmpdo@c\bxjs@g@tmpb{kanjiVariant}%
3970 \bxjs@next\bxjs@tmpdo@d\bxjs@g@tmpb{jaVariant}%
3971 %
3972 \global\let\bxjs@g@tmpa\relax
3973 \global\let\bxjs@g@tmpb\relax
3974 \endlinechar@m@ne
3975 \let\do@makeoother\dospecials
3976 \catcode32=10 \catcode12=10 %form-feed
3977 \let\bxjs@tmpa@empty
3978 \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3979 \ifeof\@inputcheck\else
3980 \read\@inputcheck to\bxjs@tmpa

```

```

3981      \closein\@inputcheck
3982      \fi
3983      \ifx\bxjs@tmpa\@empty\else
3984          \openin\@inputcheck="\bxjs@tmpa"\relax
3985          \@tempswatrue
3986          \loop\if@tempswa
3987              \read\@inputcheck to\bxjs@tmpa
3988              \bxjs@tmpdo
3989              \ifeof\@inputcheck \@tempswafalse \fi
3990          \repeat
3991      \fi
3992  }\endgroup
3993  \let\bxjs@jaEmbed\bxjs@g@tmpa
3994  \let\bxjs@jaVariant\bxjs@g@tmpb
3995  }

```

`\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。

```

3996 \@onlypreamble\bxjs@resolve@jafont@paren
3997 \def\bxjs@resolve@jafont@paren#1{%
3998     \def\bxjs@tmpb{\let#1}%
3999     \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\@nnil#1}
4000 \@onlypreamble\bxjs@resolve@jafont@paren@a
4001 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\@nnil#5{%
4002     \ifx\relax#4\relax \bxjs@tmpb#5%
4003     \else
4004         \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
4005         \bxjs@tmpb\bxjs@tmpa
4006     \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。

```

4007 \newcommand*\jachar[1]{%
4008     \begingroup

```

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

```

4009     \jsLetHeadChar\bxjs@tmpa{#1}%
4010     \ifx\bxjs@tmpa\relax
4011         \ClassWarningNoLine\bxjs@clsname
4012             {Illegal argument given to \string\jachar}%
4013     \else
4014         \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
4015     \fi
4016 \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

4017 \let\jsJaChar\jachar

```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```
4018 \let\bxjs@jachar\@firstofone
```

■hyperref 対策 出力ページサイズに飽する処理は geometry パッケージが行うので、hyperref 側の処理は無効にしておく。

```
4019 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
4020 \@onlypreamble\bxjs@fix@hyperref@unicode
4021 \def\bxjs@fix@hyperref@unicode#1{%
4022   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
4023   \@namedef{KV@Hyp@bxjs/hook}##1{%
4024     \KV@Hyp@unicode{##1}%
4025     \def\KV@Hyp@unicode####1{%
4026       \expandafter\ifx\csname if##1\expandafter\endcsname
4027       \csname if####1\endcsname\else
4028       \ClassWarningNoLine{bxjs@clsname
4029       {Blcked hyperref option 'unicode=####1'}}%
4030     \fi
4031   }%
4032 }%
4033 }
```

\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。

※ pxjahyper-uni.def はこのフックを \relax に上書きすることで検証を無効化している。

```
4034 \@onlypreamble\jsCheckHyperrefUnicode
4035 \let\jsCheckHyperrefUnicode\@empty
4036 \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。

```
4037 \@onlypreamble\bxjs@check@hyperref@unicode
4038 \def\bxjs@check@hyperref@unicode#1{%
4039   \g@addto@macro\jsCheckHyperrefUnicode{%
4040     \@tempwafalse
4041     \begingroup
4042     \expandafter\ifx\csname ifHy@unicode\endcsname\relax
4043     \aftergroup\@tempwattrue \fi
4044     \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
4045     \csname if#1\endcsname
4046     \aftergroup\@tempwattrue \fi
4047   \endgroup
4048   \if@tempwa\else
4049     \ClassError{bxjs@clsname
4050     {The value of hyperref 'unicode' key is not suitable\MessageBreak
4051     for the present engine (must be #1)}}%
4052     {\@ehc}%
4053   \fi}}
```

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

```
4054 \@onlypreamble\bxjs@urgent@special
4055 \@onlypreamble\bxjs@urgent@special@a
```

L<sup>A</sup>T<sub>E</sub>X カーネルの新フック管理が導入済かを調べる。未導入の古い版である場合。

```
4056 \ifbxjs@old@hook@system
4057 \def\bxjs@urgent@special#1{%
4058   \AtBeginDvi{\special{#1}}%
4059   \g@addto@macro\bxjs@begin@document@hook{%
4060     \ifpackageloaded{atbegshi}{%
4061       \begingroup
4062         \toks\z@{\special{#1}}%
4063         \toks\tw@\expandafter{\AtBegShi@HookFirst}%
4064         \xdef\AtBegShi@HookFirst{\the\toks@the\toks\tw@}%
4065       \endgroup
4066     }{}%
4067   }%
4068 }
```

導入済の場合。

※自分が先行する必要がある対象のパッケージを適宜追加する。

※ `pxjahyper` パッケージの処理と合わせる。

```
4069 \else
4070   \def\bxjs@urgent@special#1{%
4071     \bxjs@urgent@special@a
4072     \AddToHook{shipout/firstpage}[pxjahyper/enc]{\special{#1}}
4073   \def\bxjs@urgent@special@a{%
4074     \DeclareHookRule{shipout/firstpage}{pxjahyper/enc}{<}{hyperref}%
4075     \global\let\bxjs@urgent@special@a\relax
4076 \fi
```

## C.4 pT<sub>E</sub>X 用設定

```
4077 \if j\jsEngine
```

### ■ 共通命令の実装

```
4078 \def\bxjs@apply@kanjiskip{%
4079   \kanjiskip\@tempskipa}
4080 \def\bxjs@apply@xkanjiskip{%
4081   \xkanjiskip\@tempskipa}
```

`\jaJaChar` のサブマクロ。

```
4082 \def\bxjs@jachar#1{%
4083   \bxjs@jachar@a#1...\@nil}
4084 \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。

```
4085   \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。

```

4086 \else\ifx.#3%
4087   \@tempcnta`#1 \multiply\@tempcnta64
4088   \advance\@tempcnta`#2 \advance\@tempcnta-"3080
4089   \bxjs@jachar@b
4090 \else\ifx.#4%
4091   \@tempcnta`#1 \multiply\@tempcnta64
4092   \advance\@tempcnta`#2 \multiply\@tempcnta64
4093   \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
4094   \bxjs@jachar@b
4095 \else
4096   \@tempcnta`#1 \multiply\@tempcnta64
4097   \advance\@tempcnta`#2 \multiply\@tempcnta64
4098   \advance\@tempcnta`#3 \multiply\@tempcnta64
4099   \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
4100   \bxjs@jachar@b
4101 \fi\fi\fi}

```

符号値が `\@tempcnta` の和文文字を出力する処理。

```

4102 \ifjsWithupTeX
4103   \def\bxjs@jachar@b{\kchar\@tempcnta}
4104 \else
4105   \def\bxjs@jachar@b{%
4106     \ifx\bxUInt\@undefined\else
4107       \bxUInt{\@tempcnta}%
4108     \fi}
4109 \fi

```

和欧文間空白の命令 `\jathinspace` の実装。

```

4110 \ifbxjs@jaspace@cmd
4111   \def\jathinspace{\hskip\xkanjiskip}
4112 \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。

```

4113 \ifbxjs@jp@jismmiv
4114   \PassOptionsToPackage{prefer2004jis}{pxchfon}
4115   \PassOptionsToPackage{prefer2004jis}{pxbabel}
4116 \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。

```

4117 \let\bxjs@jafont@paren\@firstofone
4118 \let\bxjs@tmpa\jsJaFont
4119 \ifx\bxjs@tmpa\bxjs@@auto

```

```

4120 \let\bxjs@tmpa\@empty
4121 \else\ifx\bxjs@tmpa\bxjs@noEmbed
4122 \def\bxjs@tmpa{noembed}
4123 \fi\fi
4124 \bxjs@resolve@jafont@paren\bxjs@tmpa
4125 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4126 \ifx\bxjs@tmpa\@empty\else
4127 \edef\bxjs@next{%
4128 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4129 }\bxjs@next
4130 \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……。)

```

4131 \begingroup
4132 \global\let\bxjs@g@tmpa\relax
4133 \catcode`\|=0 \catcode`\|=12
4134 |def|bxjs@tmpdo#a#1|@nil{%
4135 |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4136 |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4137 |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4138 |catcode`\|=0 \catcode`\|=12
4139 \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4140 \ifx$#2$\else
4141 \xdef\bxjs@g@tmpa{%
4142 \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4143 \fi}
4144 \@firstofone{%
4145 \catcode10=12 \endlinechar\m@ne
4146 \let\do\@makeother \dospecials \catcode32=10
4147 \openin\@inputcheck=otf.sty\relax
4148 \@tempswatrue
4149 \loop\if@tempswa
4150 \ifeof\@inputcheck \@tempswafalse \fi
4151 \if@tempswa
4152 \read\@inputcheck to\bxjs@next
4153 \expandafter\bxjs@tmpdo\bxjs@next\@nil
4154 \fi
4155 \repeat
4156 \closein\@inputcheck
4157 \endgroup}
4158 \bxjs@g@tmpa

```

■**hyperref 対策** hyperref の unicode オプションに対する調整を行う。



※ pxjahyper パッケージの「unicode 対応」サポートの履歴：

- 0.7 版 [2021-02-13]：upL<sup>A</sup>T<sub>E</sub>X 上に限り unicode 対応。
- 0.9c 版 [2021-06-06]：pxjahyper-uni.def ファイルを追加。
- 1.0 版 [2022-04-01]：pL<sup>A</sup>T<sub>E</sub>X 上の unicode 対応を試験的サポート。
- 1.3 版 [2023-03-01]：pL<sup>A</sup>T<sub>E</sub>X 上の unicode 対応を正式サポート。

```
4159 \ifbxjs@hyperref@enc
```

unicode オプションが偽であることを検証する。ただし、pxjahyper パッケージまたは pxjahyper-uni.def が読み込まれて（前提条件を満たして）「unicode 対応」が行われた場合は検証は無効化される。

```
4160 \bxjs@check@hyperref@unicode{false}
```

\bxjs@plautopatch@new は「pxjahyper の自動読込に対応した版の plautopatch が読み込まれているか」のフラグ。

```
4161 \ifpackagelater{plautopatch}{2020/05/25}{% v0.9g
```

```
4162 \let\bxjs@plautopatch@new=t}{}
```

「unicode を有効にできるか」を判定する。まず必要条件として「pxjahyper-uni.def が存在すること」「\bxjs@plautopatch@new が真、または、ファイルフックが利用可能であること」を検査する。

※ pxjahyper-uni.def をもつ pxjahyper の版であれば、upL<sup>A</sup>T<sub>E</sub>X 上の unicode には対応していることに注意。

```
4163 \let\bxjs@avail@hy@unicode=f
```

```
4164 \if \ifx t\bxjs@plautopatch@new T%
```

```
4165 \else\ifbxjs@old@hook@system F\else T\fi\fi T%
```

```
4166 \IfFileExists{pxjahyper-uni.def}{\let\bxjs@avail@hy@unicode=t}{}
```

```
4167 \fi
```

```
4168 \if t\bxjs@avail@hy@unicode
```

```
4169 \ifjsWithupTeX
```

必要条件が満たされていて、かつ upL<sup>A</sup>T<sub>E</sub>X である場合の処理。もしファイルフックが利用可能ならば、hyperref が読み込まれた場合にその直後に pxjahyper-uni.def が読まれるようにする。

※そうでないなら、前提条件より pxjahyper が読み込まれるはずなので何もしなくてよい。

```
4170 \ifbxjs@old@hook@system\else
```

```
4171 \AddToHook{\bxjs@CGHN{package/hyperref/after}}{%
```

```
4172 \input{pxjahyper-uni.def}}
```

```
4173 \fi
```

```
4174 \else
```

必要条件が満たされていて、かつ pL<sup>A</sup>T<sub>E</sub>X である場合の処理。pxjahyper が「pL<sup>A</sup>T<sub>E</sub>X 上の unicode 対応をもつほど新しい版（1.3 版以降）」であるかを判定する方法はない。しかし、新しい L<sup>A</sup>T<sub>E</sub>X システムで unicode を無効にするのは避けたいので、L<sup>A</sup>T<sub>E</sub>X カーネルが 2023-06-01 以降である場合に pxjahyper も十分に新しいと推定することにする。すなわち「pxjahyper が読み込まれるはず」かつ「L<sup>A</sup>T<sub>E</sub>X がカーネルが新しい」かを判定する。

```

4175 \let\bxjs@avail@hy@unicode=f
4176 \ifx t\bxjs@plautopatch@new
4177 \ifl@t@r\fmtversion{2023/06/01}{\let\bxjs@avail@hy@unicode=t}{\fi}
4178 \fi
4179 \fi
4180 \fi

```

この時点で「unicode を有効にできるか」の判定結果がフラグ \bxjs@avail@hy@unicode に入っている。unicode を有効にできない場合は unicode の既定値を偽に設定する。

```

4181 \if f\bxjs@avail@hy@unicode
4182 \PassOptionsToPackage{unicode=false}{hyperref}
4183 \fi
4184 \fi

```

tounicode special 命令を出力する。

```

4185 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4186 \else\ifjsWithpTeXng T\else F\fi\fi T%
4187 \IfFileExists{pxjahyper-enc.sty}{\@tempwattrue}{\@tempwafalse}
4188 \if@tempwa
4189 \RequirePackage{pxjahyper-enc}[2020/10/05]%v0.6
4190 \ifbxjs@bigcode\else \suppressbigcode \fi
4191 \else
4192 \ifnum\jis"2121="A1A1 %euc
4193 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4194 \else\ifnum\jis"2121="8140 %sjis
4195 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4196 \else\ifnum\jis"2121="3000 %uptex
4197 \ifbxjs@bigcode
4198 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4199 \PassOptionsToPackage{bigcode}{pxjahyper}
4200 \else
4201 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4202 \PassOptionsToPackage{nobigcode}{pxjahyper}
4203 \fi
4204 \fi\fi\fi
4205 \let\bxToUnicodeSpecialDone=t
4206 \fi
4207 \fi

```

■和文数式ファミリ 和文数式ファミリは既定で有効とする。すなわち enablejfam=false 以外の場合は @enablejfam を真にする。

```

4208 \ifx f\bxjs@enablejfam\else
4209 \@enablejfamtrue
4210 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

4211 \if@enablejfam
4212 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4213 \DeclareSymbolFontAlphabet{\mathmc}{mincho}

```

```

4214 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4215 \jfam\symmincho
4216 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4217 \g@addto@macro\bxjs@begin@document@hook{%
4218   \ifx\reDeclareMathAlphabet\@undefined\else
4219     \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4220     \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4221     \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4222   \fi}
4223 \fi

```

## C.5 pdfTeX 用設定：CJK + bxcjkatype

```

4224 \else\if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T

```

■**bxcjkatype** パッケージの読込 `\jsJaFont` が指定されている場合は、その値を `bxcjkatype` のオプション（プリセット指定）に渡す。ここで値が `auto` である場合は `\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。

```

4225 \bxjs@adjust@jafont{f}
4226 \let\bxjs@jafont@paren\@firstofone
4227 \bxjs@resolve@jafont@paren\bxjs@tmpa
4228 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4229 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4230 \ifx\bxjs@jadriver\bxjs@@pandoc\else
4231   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4232 \fi
4233 \edef\bxjs@next{%
4234   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4235 }\bxjs@next
4236 \bxjs@cjk@loaded

```

■**hyperref 対策** `bxcjkatype` 使用時は `unicode` にするべき。

```

4237 \ifbxjs@hyperref@enc
4238   \PassOptionsToPackage{unicode}{hyperref}
4239 \fi

```

`\hypersetup` 命令で（CJK\* 環境に入れなくても）日本語文字を含む文書情報を設定できるようにするための細工。

※ `bxcjkatype` を `whole` 付きで使っていることが前提。

※パッケージオプションでの指定に対応するのは、「アクティブな高位バイトトークンがその場で展開されてしまう」ため困難である。

```

4240 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
4241 \begingroup
4242   \CJK@input{UTF8.bdg}
4243 \endgroup

```

```

4244 \g@addto@macro\pdfstringdefPreHook{%
4245   \@nameuse{CJK@UTF8Binding}%
4246 }
4247 \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。

```

4248 \ifx\bxckjatypeHyperrefPatchDone\@undefined
4249 \g@addto@macro\pdfstringdefPreHook{%
4250   \ifx~\bxjs@CJKtilde
4251     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4252     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4253     \let~\@empty
4254   \fi
4255 }
4256 \def\bxjs@CJKtilde{\CJKecglue\ignorespaces}
4257 \def\bxjs@tildecmd{~}
4258 \def\bxjs@LetUnexpandableSpace#1{%
4259   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@tildecmd\else
4260     \bxjs@org@LetUnexpandableSpace#1%
4261   \fi}
4262 \fi

```

## ■ 共通命令の実装

```

4263 \newskip\jsKanjiSkip
4264 \newskip\jsXKanjiSkip
4265 \ifx\CJKecglue\@undefined
4266   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4267 \fi
4268 \let\autospacing\bxjs@enable@kanjiskip
4269 \let\noautospacing\bxjs@disable@kanjiskip
4270 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4271 \def\bxjs@apply@kanjiskip{%
4272   \jsKanjiSkip\@tempkipa
4273   \let\CJKglue\bxjs@CJKglue}
4274 \let\autoxspacing\bxjs@enable@xkanjiskip
4275 \let\noautoxspacing\bxjs@disable@xkanjiskip
4276 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4277 \def\bxjs@apply@xkanjiskip{%
4278   \jsXKanjiSkip\@tempkipa
4279   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4280 \def\bxjs@jachar#1{%
4281   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4282 \ifbxjs@jaspace@cmd
4283   \protected\def\jathinspace{\CJKecglue}
4284 \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。

```
4285 \ifx t\bxjs@enablejfam
4286   \ClassWarningNoLine\bxjs@clsname
4287   {You cannot use 'enablejfam=true', since the\MessageBreak
4288     CJK package does not support Japanese math}
4289 \fi
```

## C.6 X<sub>Y</sub>T<sub>E</sub>X 用設定：xeCJK + zxjatype

```
4290 \else\if x\jsEngine
```

■zxjatype パッケージの読込 スケール値 (\jsScale) の反映は zxjatype の側で行われる。

```
4291 \RequirePackage{zxjatype}
4292 \PassOptionsToPackage{no-math}{fontspec}%!
4293 \PassOptionsToPackage{xetex}{graphicx}%!
4294 \PassOptionsToPackage{xetex}{graphics}%!
4295 \ifx\zxJaFamilyName\@undefined
4296   \ClassError\bxjs@clsname
4297   {xeCJK or zxjatype is too old}\@ehc
4298 \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。

```
4299 \bxjs@adjust@jafont{f}
4300 \let\bxjs@jafont@paren@gobble
4301 \bxjs@resolve@jafont@paren\bxjs@tmpa
4302 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4303 \ifx\bxjs@tmpa\@empty
4304   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4305   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4306 \else
4307   \edef\bxjs@next{%
4308     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4309   }\bxjs@next
4310 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X<sub>Y</sub>T<sub>E</sub>X 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<sub>Y</sub>T<sub>E</sub>X 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

と見なして警告が発生する。

これを踏まえて、ここでは、「Xe<sub>q</sub>TeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※ TeX の小数の精度は十進で 4 桁までしか保証されないので、`\strcmp` を利用して文字列で比較している。(整数部が多桁になっても大丈夫。) しかし実は、`\strcmp` プリミティブが追加されたのは 0.9994 版 (2009 年 6 月) かららしい。

**TODO:** バージョン要件を見直して暫定措置を解除する。

```
4311 \ifx\strcmp@\undefined\else % 未定義なら条件を満たさない
4312 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4313   \ifbxjs@hyperref@enc
4314     \PassOptionsToPackage{unicode}{hyperref}
4315   \fi
4316 \fi
4317 \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。

```
4318 \AtEndOfPackage{%
4319 \def\@newline #1{%
4320   \ifvmode \@nolnerr
4321   \else
4322     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4323     \nobreak \hskip-1sp\hskip1sp\relax
4324     \ignorespaces
4325   \fi}
4326 }
```

## ■共通命令の実装

```
4327 \newskip\jsKanjiSkip
4328 \newskip\jsXKanjiSkip
4329 \ifx\CJKecglue@\undefined
4330   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4331 \fi
4332 \let\autospacing\bxjs@enable@kanjiskip
4333 \let\noautospacing\bxjs@disable@kanjiskip
4334 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4335 \def\bxjs@apply@kanjiskip{%
4336   \jsKanjiSkip\@tempskipa
4337   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4338 \let\autoxspacing\bxjs@enable@xkanjiskip
4339 \let\noautoxspacing\bxjs@disable@xkanjiskip
4340 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
```

```

4341 \def\bxjs@apply@xkanjiskip{%
4342   \jsXKanjiSkip\@tempskipa
4343   \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。

```

4344 \ifx\mcfamily\@undefined
4345   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4346   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4347 \fi
```

`\jachar` のサブマクロの実装。

```

4348 \def\bxjs@jachar#1{%
4349   \xeCJKDeclareCharClass{CJK}{`#1}\relax
4350   #1}
```

`\jathinspace` の実装。

```

4351 \ifbxjs@jaspace@cmd
4352   \protected\def\jathinspace{\CJKecglue}
4353 \fi
```

■和文数式ファミリー 和文数式ファミリーは既定で無効とする。すなわち `enablejfam=true` の場合にのみ `@enablejfam` を真にする。

```

4354 \ifx t\bxjs@enablejfam
4355   \@enablejfamtrue
4356 \fi
```

実際に和文用の数式ファミリーの設定を行う。

※ FIXME: 要検討。

```

4357 \if@enablejfam
4358   \xeCJKsetup{CJKmath=true}
4359 \fi
```

## C.7 LuaTeX 用設定：LuaTeX-ja

```

4360 \else\if l\jsEngine
```

■LuaTeX-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。

```

4361 \let\zw\@undefined
4362 \RequirePackage{luatexja}
4363 \edef\bxjs@next{%
4364   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
```

4365 }\bxjs@next

フォント代替の明示的定義。

```
4366 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{}
4367 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{}
4368 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{}
4369 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{}
4370 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{}
4371 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{}
4372 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{}
4373 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{}
4374 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{}
4375 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{}
4376 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{}
4377 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{}
4378 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{}
4379 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{}
4380 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{}
4381 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{}
4382 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{}
4383 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{}
4384 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{}
4385 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{}
4386 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{}
4387 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{}
4388 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{}
4389 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{}
4390 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{}
4391 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{}
4392 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{}
4393 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{}
4394 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{}
4395 \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。

```
4396 \bxjs@adjust@jafont{t}
4397 \ifx\bxjs@tmpa\bxjs@noEmbed
4398   \def\bxjs@tmpa{noembed}
4399 \fi
4400 \let\bxjs@jafont@paren\gobble
4401 \bxjs@resolve@jafont@paren\bxjs@tmpa
4402 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4403 \ifx\bxjs@tmpa@empty
4404   \defaultjfontfeatures{ Kerning=Off }
4405   \setmainjfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-

```



```

Regular.otf}
4406 \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
Medium.otf}
4407 \else
4408 \edef\bxjs@next{%
4409 \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4410 }\bxjs@next
4411 \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。

```

4412 \@ifpackagelater{luatexja}{2016/03/31}{\else
4413 \DeclareRobustCommand\rmfamily
4414 {\not@math@alphabet\rmfamily\mathrm
4415 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4416 \DeclareRobustCommand\sffamily
4417 {\not@math@alphabet\sffamily\mathsf
4418 \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4419 \DeclareRobustCommand\ttfamily
4420 {\not@math@alphabet\ttfamily\mathtt
4421 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4422 }
4423 \long\def\jttdefault{\gtdefault}
4424 \unless\ifx\@ltj@match@familytrue\@undefined
4425 \@ltj@match@familytrue
4426 \fi
4427 \g@addto@macro\bxjs@begin@document@hook{%
4428 \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4429 \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4430 \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4431 \bxjs@if@sf@default{%
4432 \renewcommand\kanjifamilydefault{\gtdefault}}

```

## ■和文パラメタの設定

```

4433 % 次の3つは既定値の通り
4434 %\ltjsetparameter{prebreakpenalty={` ,10000}}
4435 %\ltjsetparameter{postbreakpenalty={` ",10000}}
4436 %\ltjsetparameter{prebreakpenalty={` ”,10000}}
4437 \ltjsetparameter{jaxspmode={` !,1}}
4438 \ltjsetparameter{jaxspmode={` 〒,2}}
4439 \ltjsetparameter{alxspmode={` +,3}}
4440 \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-ja で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合に備えて同等のものを用意する。

```

4441 \ifx\ltjfakeparbegin\@undefined
4442 \protected\def\ltjfakeparbegin{%
4443   \ifhmode
4444     \relax\directlua{%
4445       luatexja.jfmglue.create_beginpar_node()}}
4446 \fi}
4447 \fi

    ltjs* クラスの定義と同等になるようにパッチを当てる。
4448 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4449 \begingroup
4450 \let\%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4451 \@gobble\if\def\bxjs@tmpa{\@{\everypar}{\fi}}
4452 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4453 \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}\fi}}\fi
4454 \directlua{
4455   local function patchcmd(cs, code, from, to)
4456     tex.sprint(code:gsub(from:gsub("\%W", "\%\%\%0"), "\%0"..to)
4457       :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1)..")
4458   end
4459   patchcmd(\@{\@xsect}, [[\meaning\@xsect]],
4460     \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4461   patchcmd(\@{\@item}, [[\meaning\@item]],
4462     \bxjs@tmpa, \@{\ltjfakeparbegin})}
4463 \endgroup
4464 \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。

```

4465 \ifbxjs@hyperref@enc
4466 \PassOptionsToPackage{unicode}{hyperref}
4467 \bxjs@check@hyperref@unicode{true}
4468 \fi

```

## ■共通命令の実装

```

4469 \protected\def\autospaceing{%
4470 \ltjsetparameter{autospaceing=true}}
4471 \protected\def\noautospaceing{%
4472 \ltjsetparameter{autospaceing=false}}
4473 \protected\def\autoxspaceing{%
4474 \ltjsetparameter{autoxspaceing=true}}
4475 \protected\def\noautoxspaceing{%
4476 \ltjsetparameter{autoxspaceing=false}}
4477 \def\bxjs@apply@kanjiskip{%
4478 \ltjsetparameter{kanjiskip={\@tempskipa}}}
4479 \def\bxjs@apply@xkanjiskip{%
4480 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4481 \def\bxjs@jachar#1{%
4482   \ltjjachar`#1\relax}

\jathinspace の実装。
4483 \ifbxjs@jaspace@cmd
4484   \protected\def\jathinspace{%
4485     \hskip\ltjgetparameter{xkanjiskip}\relax}
4486 \fi

```

■和文数式ファミリー Lua<sub>T</sub><sub>E</sub>X-j<sub>a</sub> では和文数式ファミリーは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablej<sub>f</sub>am は常に真になる。

```

4487 \ifx f\bxjs@enablejfam
4488   \ClassWarningNoLine\bxjs@clsname
4489     {You cannot use 'enablejfam=false', since the\MessageBreak
4490       LuaTEX-ja always provides Japanese math families}
4491 \fi

```

## C.8 共通処理 (2)

```

4492 \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 4493 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4494   \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4495   \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4496 \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。

```

4497 \ifx\mathmc\@undefined
4498   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4499   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4500 \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)。

```

4501 \ifbxjs@jaspace@cmd
4502   \bxjs@protected\def\bxjs@choice@jathinspace{%
4503     \relax\ifmmode \mskip\medmuskip
4504     \else \jathinspace\ignorespaces
4505     \fi}
4506 \jsAtEndOfClass{%
4507   \ifjsWitheTeX \let\>\bxjs@choice@jathinspace
4508   \else \def\>{\protect\bxjs@choice@jathinspace}%

```

```
4509 \fi}
4510 \fi
```

## ■和文・和欧文間空白の初期値

```
4511 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4512 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4513 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4514 \fi
```

以上で終わり。

```
4515 %</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。

```
4516 %<*modern>
4517 \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。

```
4518 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4519 \def\encodingdefault{T1}%
4520 \input{t1enc.def}%
4521 \fontencoding\encodingdefault\selectfont
4522 \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。

```
4523 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4524 \renewcommand{\rmdefault}{lrm}
4525 \renewcommand{\sfdefault}{lmss}
4526 \renewcommand{\ttdefault}{lmtt}
4527 \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。

```
4528 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4529 <-7.5>cmex7<7.5-8.5>cmex8%
4530 <8.5-9.5>cmex9<9.5->cmex10}{}%
4531 \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。

```
4532 \def\cmex@opt{10}
```

## D.2 fixltx2e 読込

※ fixltx2e 廃止前の L<sup>A</sup>T<sub>E</sub>X カーネルの場合。

```
4533 \ifx\@IncludeInRelease\@undefined
4534 \RequirePackage{fixltx2e}
4535 \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。

```
4536 \RequirePackage{bxjscjkcat}
```

## D.4 完了

おしまい。

```
4537 %</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。

```
4538 %<*pandoc>
```

```
4539 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4540 \RequirePackage{bxjspandoc}
```

ε-<sub>T</sub>E<sub>X</sub> ではない場合に警告を出す。

※近い将来に ε-<sub>T</sub>E<sub>X</sub> 拡張を必須にする予定。

```
4541 \ifjsWitheTeX\else
```

```
4542 \ClassWarningNoLine{bxjs}{clsname}
```

```
4543 {!!!!!!! WARNING !!!!!!!\MessageBreak
```

```
4544 This engine does not support e-TeX extension!\MessageBreak
```

```
4545 Some feature might not work properly}
```

```
4546 \fi
```

```
\ifbxjs@bxghost@available [スイッチ] bxghost パッケージが利用できるか。
```

```
4547 \newif\ifbxjs@bxghost@available
```

```
4548 \ifjsWitheTeX
```

```

4549 \RequirePackage{pdftexcmds}[2009/09/22]% v0.5
4550 \IfFileExists{bxghost.sty}{%
4551   \bxjs@bxghost@availabletrue
4552   \@namedef{bxjs@bgbv/79E70A0991967E27981832C84DB5DF99}{1}%v0.2.0
4553   \ifx\pdf@filemdfivesum\undefined\else
4554     \expandafter\ifx\csname bxjs@bgbv/\pdf@filemdfivesum{bxghost.sty}%
4555       \endcsname\relax\else \bxjs@bxghost@availablefalse \fi
4556   \fi
4557 }{}
4558 \fi

```

`\bxjs@endpreamble@hook` etoolbox の `\AtEndPreamble` で実行される BXJS クラス用のフック。

※  $\epsilon$ -TeX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```

4559 \@onlypreamble\bxjs@endpreamble@hook
4560 \let\bxjs@endpreamble@hook\@empty

```

パッケージ読込。

```

4561 \RequirePackage{iftex}[2013/04/04]% v0.2
4562 \ifjsWitheTeX
4563   \RequirePackage{etoolbox}[2010/08/21]% v2.0
4564   \AtEndPreamble{\bxjs@endpreamble@hook}
4565   \RequirePackage{filehook}[2011/10/12]% v0.5d
4566 \fi

```

## E.2 和文ドライバパラメタ

`keyval` のファミリは `bxjsPan` とする。

`\ifbxjs@jp@fix@strong` 重要要素を補正するか。

```

4567 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue

```

`fix-strong` オプションの処理。

```

4568 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
4569 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
4570 \define@key{bxjsPan}{fix-strong}[true]{%
4571   \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。

```

4572 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue

```

`fix-code` オプションの処理。

```

4573 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
4574 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
4575 \define@key{bxjsPan}{fix-code}[true]{%
4576   \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。

```

4577 \chardef\bxjs@jp@strong=0

```

`strong` オプションの処理。

```
4578 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
4579 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
4580 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }
4581 \define@key{bxjsPan}{strong}{%
4582   \bxjs@set@keyval{strong}{#1}{}}
```

`\ifbxjs@jp@or@indent` プレアンブルでのレイアウト上書きを許可するか。既定値は真。

```
\ifbxjs@jp@or@secnumdepth 4583 \newif\ifbxjs@jp@or@indent \bxjs@jp@or@indenttrue
\ifbxjs@jp@or@block@heading 4584 \newif\ifbxjs@jp@or@secnumdepth \bxjs@jp@or@secnumdepthtrue
4585 \newif\ifbxjs@jp@or@block@heading \bxjs@jp@or@block@headingtrue
```

クラスで `pandoc+` が指定された場合、内部和文パラメタ `_plus` が和文ドライバに渡される。この場合、レイアウト上書きを禁止する。

※ `_plus` は必ずパラメタ列の先頭にあるので、個別のパラメタ設定の方が常に優先される。

```
4586 \define@key{bxjsPan}{_plus}[]{%
4587   \bxjs@jp@or@indentfalse
4588   \bxjs@jp@or@secnumdepthfalse
4589   \bxjs@jp@or@block@headingfalse}
```

レイアウト上書き許可オプション (`or-indent`・`or-secnumdepth`・`or-block-heading`) の処理。

```
4590 \let\bxjs@kv@orindent@true\bxjs@jp@or@indenttrue
4591 \let\bxjs@kv@orindent@false\bxjs@jp@or@indentfalse
4592 \define@key{bxjsPan}{or-indent}[true]{%
4593   \bxjs@set@keyval{orindent}{#1}{}}
4594 \let\bxjs@kv@orsecnumdepth@true\bxjs@jp@or@secnumdepthtrue
4595 \let\bxjs@kv@orsecnumdepth@false\bxjs@jp@or@secnumdepthfalse
4596 \define@key{bxjsPan}{or-secnumdepth}[true]{%
4597   \bxjs@set@keyval{orsecnumdepth}{#1}{}}
4598 \let\bxjs@kv@orblockheading@true\bxjs@jp@or@block@headingtrue
4599 \let\bxjs@kv@orblockheading@false\bxjs@jp@or@block@headingfalse
4600 \define@key{bxjsPan}{or-block-heading}[true]{%
4601   \bxjs@set@keyval{orblockheading}{#1}{}}
```

実際の `japaram` の値を適用する。

```
4602 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}
4603 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

### E.3 dupload システム

パッケージが重複して読み込まれたときに“option clash”の検査をスキップする。この時に何らかのコードを実行させることができる。

`\bxjs@set@dupload@proc` `\bxjs@set@dupload@proc{〈ファイル名〉}{〈定義本体〉}`： 指定の名前の特定のファイルの読込が `\@filewithoptions` で指示されて、しかもそのファイルが読込済である場合に、オプション重複検査をスキップして、代わりに〈定義本体〉のコードを実行する。このコード中で `#1` は渡されたオプション列のテキストに置換される。

```

4604 \@onlypreamble\bxjs@set@dupload@proc
4605 \def\bxjs@set@dupload@proc#1{%
4606   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
4607 \@onlypreamble\bxjs@set@dupload@proc@a
4608 \def\bxjs@set@dupload@proc@a#1{%
4609   \@onlypreamble#1\def#1##1}
4610 \def\bxjs@unset@dupload@proc#1{%
4611   \bxjs@cslet{bxjs@dlp/#1}\@undefined}

```

\@if@options \@if@options の再定義。

```

4612 \@onlypreamble\bxjs@org@if@options
4613 \let\bxjs@org@if@options\@if@options
4614 \@onlypreamble\bxjs@org@reset@options
4615 \let\bxjs@org@reset@options\relax
4616 \def\@if@options#1#2#3{%
4617   \let\bxjs@next\@secondoftwo
4618   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4619   \ifx\bxjs@tmpa\bxjs@tmpb
4620     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4621       \let\bxjs@next\@firstoftwo \fi
4622   \fi
4623   \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4624 \g@addto@macro\bxjs@begin@document@hook{%
4625   \let\@if@options\bxjs@org@if@options}
4626 \@onlypreamble\bxjs@do@dupload@proc
4627 \def\bxjs@do@dupload@proc#1#2#3{%
4628   \ifx\bxjs@org@reset@options\relax
4629     \let\bxjs@org@reset@options\@reset@options
4630   \fi
4631   \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4632   \def\@reset@options{%
4633     \let\@reset@options\bxjs@org@reset@options
4634     \@reset@options
4635     \bxjs@next{#3}}%
4636   \@firstoftwo}

```

## E.4 lang 変数

lang=ja という言語指定が行われると、2.12 版より前の Pandoc はこれに対応していなかったため不完全な Babel や Polyglossia の設定を出力してしまっていた。現在では lang=ja 指定について正しく L<sup>A</sup>T<sub>E</sub>X 側の言語名 `japanese` に変換されるようになっているが、それでも日本語指定の場合は相変わらず調整処理が必要である。

※そもそも BXJS クラスは日本語用の文書クラスであるため、もし言語設定が行われているのであれば「メイン言語は日本語である」であるはずなので、「サブ言語が日本語である」ことは考慮しない。



■Polyglossia について 現在 CTAN に登録されている日本語用の gloss ファイルは超絶アレでかつ有害な設定を行うため、これの読込を避ける必要がある。そのため、メイン言語が `japanese` である場合（古い Pandoc ではこの場合に引数が空の `\setmainlanguage{}` が実行されるがこのパターンも同様に扱う）には、Polyglossia の処理を無効化してしまうことにする。つまり、Polyglossia が提供する命令について、何もしないダミーの定義を与える。※ Polyglossia は古い Pandoc のテンプレートにおいて、エンジンが  $\text{\TeX}$  か  $\text{\LaTeX}$  の場合に利用されていた。

`\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。

```
4637 \let\bxjs@polyglossia@options\relax
```

エンジンが  $\text{\TeX}$  か  $\text{\LaTeX}$  の場合が対象になる。

※この場合 `etoolbox` が使用可能になっている。

```
4638 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0
```

パッケージの読込を検知するため読込済のマークを付けて `dupload` の処理を仕込む。

```
4639 \pandocSkipLoadPackage{polyglossia}
4640 \bxjs@set@dupload@proc{polyglossia.sty}{%
4641   \bxjs@unset@dupload@proc{polyglossia.sty}%
4642   \ClassWarning\bxjs@clsname
4643     {Package polyglossia is requested}%
4644   \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。

※最初に `\setmainlanguage` 以外が実行された場合はエラーになる。

```
4645 \newcommand*\setmainlanguage[2][]{%
```

もし、`\setmainlanguage` の引数が空または `japanese` だった場合はメインが日本語である (`lang=ja` 指定) と見なす。

```
4646   \ifboolexpr{test{\ifblank{##2}}or test{\ifstrequal{##2}{japanese}}}{%
4647     \ClassWarning\bxjs@clsname
4648       {Main language is 'japanese', thus fallback\MessageBreak
4649         definitions will be employed}%
4650     \bxjs@pandoc@polyglossia@ja
```

それ以外は、改めて `polyglossia` を読み込んで、本来の処理を実行する。

```
4651   }{%else
4652     \ClassWarning\bxjs@clsname
4653       {Main language is '##2',\MessageBreak
4654         thus polyglossia will be loaded}%
4655     \csundef{ver@polyglossia.sty}%
4656     \edef\bxjs@next{%
4657       \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4658     }\bxjs@next
4659     \setmainlanguage[##1]{##2}%
```

```
4660   }}
```

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、(\AtEndPreamble において) 読込済マークを外す。

```
4661 \g@addto@macro\bxjs@endpreamble@hook{%
4662   \ifx\bxjs@polyglossia@options\relax
4663     \csundef{ver@polyglossia.sty}%
4664   \fi}
```

\bxjs@pandoc@polyglossia@ja Pandoc 側で lang=ja が指定されていた場合の処理。この場合は Polyglossia の処理を無効化するためにダミーの定義を行う。すなわち、サブ言語 xxx の各々について、xxx 環境と \textxxx 命令を（特に何も加工しないものとして）定義する。この目的のため、\setotherlanguage(s) をダミーを定義する命令として定義する。

```
4665 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4666 \def\bxjs@pandoc@polyglossia@ja{%
4667   \renewcommand*\setmainlanguage[2][]{}%
4668   \newcommand*\setotherlanguage[2][]{%
4669     \ifblank{##2}{\else
4670       \cslet{##2}\@empty \cslet{end##2}\@empty
4671       \cslet{text##2}\@firstofone}}%
4672   \newcommand*\setotherlanguages[2][]{%
4673     \@for\bxjs@tmpa:=##2\do{%
4674       \setotherlangauge{\bxjs@tmpa}}}%
4675 }
```

Polyglossia の読込済マークは外れるようにしておく。

```
4675 \let\bxjs@polyglossia@options\relax%
4676 \fi
```

■Babel について 現在の Pandoc では、テンプレートで用いられる多言語パッケージとしてエンジンの種別によらずに Babel が使われる。

※ Xe<sub>La</sub>TeX では 2.15 版で、Lua<sub>La</sub>TeX は 2.6 版で Polyglossia から Babel に変更されている。

\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。

```
4677 \let\bxjs@babel@options\relax
```

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。

```
4678 \pandocSkipLoadPackage{babel}
4679 \bxjs@set@dupload@proc{babel.sty}{%
4680   \bxjs@unset@dupload@proc{babel.sty}%
4681   \ClassWarning\bxjs@clsname
4682     {Package babel is requested}}%
```

パッケージオプションに言語名が空の main= がある場合は、main=japanese に置き換える。

```
4683 \@tempswafalse \let\bxjs@babel@options\@empty
4684 \def\bxjs@tmpb{main=}
4685 \@for\bxjs@tmpa:=#1\do{%
4686   \ifx\bxjs@tmpa\bxjs@tmpb \def\bxjs@tmpa{main=japanese}\fi
4687   \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}}%
```

```

4688 \bxjs@cslet{ver@babel.sty}\@undefined
4689 \edef\bxjs@next{%
4690   \noexpand\RequirePackage[\bxjs@babel@options]{babel}\relax
4691 } \bxjs@next
4692 \RequirePackage{bxorigcapt}\relax}

```

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。

```

4693 \g@addto@macro\bxjs@endpreamble@hook{%
4694   \ifx\bxjs@babel@options\relax
4695     \bxjs@cslet{ver@babel.sty}\@undefined
4696   \fi}

```

3.0 版より前の `japanese.ldf` はサポート対象エンジンが限られていた。ここでは、エンジンの種類を問わず、「`japanese.ldf` が古い場合は読込を回避してダミー定義で代替する」という対策を入れる。実は `japanese.ldf` で行う定義は `bxorigcapt` の機能等により実質的に全て無効化されている。最新の環境においては「`japanese` 指定の `Babel + bxorigrcapt` パッケージ」の状態にしておきたい。

```

4697 \ifjsWitheTeX

```

`filehook` の機能を用いて `japanese.ldf` の読込にフックを仕込む。

```

4698 \AtBeginOfFile{japanese.ldf}{\bxjs@begin@japanese@ldf@hook}
4699 \def\bxjs@begin@japanese@ldf@hook{%
4700   \let\bxjs@begin@japanese@ldf@hook\relax
4701   \let\bxjs@save@ProvidesLanguage\ProvidesLanguage
4702   \let\bxjs@save@LdfInit\LdfInit
4703   \def\ProvidesLanguage##1[##2]{\bxjs@do@japanese@ldf{##2}}%
4704   \def\LdfInit##1##2{\bxjs@do@japanese@ldf{0000/00/00}}

```

バージョンを判定する部分。

※`\LdfInit` にも細工を入れている理由は、初期の `japanese.ldf` には `\ProvidesLanguage` が記述されていないため。

```

4705 \def\bxjs@do@japanese@ldf#1{\bxjs@do@japanese@ldf@a#1\@nil}
4706 \def\bxjs@do@japanese@ldf@a#1/#2/#3#4#5\@nil{%
4707   \let\LdfInit\bxjs@save@LdfInit
4708   \ClassInfo\bxjs@clsname
4709   {Release date of japanese.ldf is:\MessageBreak
4710     \@spaces #1/#2/#3#4\@gobble}%
4711   \ifnum#1#2#3#4<20201206 % v3.0
4712     \let\bxjs@japanese@ldf@skipped=t\csuse{endingput}%
4713   \fi}
4714 \AtEndOfFile{japanese.ldf}{\bxjs@end@japanese@ldf@hook}
4715 \def\bxjs@end@japanese@ldf@hook{%
4716   \let\bxjs@end@japanese@ldf@hook\relax
4717   \let\ProvidesLanguage\bxjs@save@ProvidesLanguage
4718   \let\LdfInit\bxjs@save@LdfInit
4719   \ifx t\bxjs@japanese@ldf@skipped
4720     \ClassWarningNoLine\bxjs@clsname
4721     {Loading japanese.ldf is skipped}%

```

ダミーの言語定義。

```
4722 \ifundef\l@japanese{\chardef\l@japanese\z@}{}%
4723 \let\datejapanese\@empty\let\captionjapanese\@empty
4724 \let\extrajapanese\@empty\let\noextrajapanese\@empty
4725 \main@language{japanese}%
4726 \fi}
4727 \g@addto@macro\bxjs@begin@document@hook{%
4728 \let\bxjs@begin@japanese\ldf@hook\relax
4729 \let\bxjs@end@japanese\ldf@hook\relax}
4730 \fi
```

lang 対策はこれで終わり。

## 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。

```
4731 \bxjs@set@dupload@proc{geometry.sty}{%
4732 \setpagelayout*{#1}}
```

## E.6 CJKmainfont 変数

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

```
4733 \if 1\jsEngine
4734 \pandocSkipLoadPackage{xeCJK}
4735 \providecommand*\setCJKmainfont{}\setmainfont}
4736 \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。

```
4737 \if x\jsEngine
4738 \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4739 ,space}
4740 \fi
```

## E.8 レイアウト上書き禁止

レイアウト上書き禁止の実装は etoolbox の機能を使う。

```
4741 \ifjsWithTeX
4742 \@onlypreamble\bxjs@info@or@ban
4743 \def\bxjs@info@or@ban#1{%
```

```

4744 \PackageInfo{bxjs@clsname
4745   {Freeze layout on '#1',\MessageBreak reported}}

```

■**indent** について indent 変数を指定しない場合に「段落表現形式をインデント方式に変更する」動作を抑止する。

```

4746 \unless\ifbxjs@jp@or@indent
4747   \bxjs@info@or@ban{indent}

```

parskip がある場合はそれを読み込もうとするため、parskip の読込をブロックする。

```

4748 \IfFileExists{parskip.sty}{%
4749   \pandocSkipLoadPackage{parskip}%

```

parskip がない場合はパラメタを変更しようとするため、該当のパラメタを復帰させる。

```

4750 }{%else
4751   \eappto\bxjs@endpreamble@hook{%
4752     \parindent=\the\parindent\relax
4753     \parskip=\the\parskip\relax}}
4754 \fi

```

■**secnumdepth** について secnumdepth の値を決めるのは numbersections 変数 (-N/--number-sections オプションに連動する) や secnumdepth 変数であるが、何れにしても secnumdepth の値は書き換えられる。そのため、secnumdepth を復帰させる。

```

4755 \ifbxjs@jp@or@secnumdepth\else
4756   \bxjs@info@or@ban{secnumdepth}
4757   \eappto\bxjs@endpreamble@hook{%
4758     \c@secnumdepth=\the\c@secnumdepth\relax}
4759 \fi

```

■**block-heading** について \paragraph、\subparagraph を別行見出しに変える処理を抑止する。

※ 2.7.1 版以前では別行見出し変更が既定で有効であった。

```

4760 \ifbxjs@jp@or@block@heading\else
4761   \let\bxjs@frozen@paragraph\paragraph
4762   \let\bxjs@frozen@subparagraph\subparagraph
4763   \bxjs@info@or@ban{block-heading}
4764   \appto\bxjs@endpreamble@hook{%
4765     \let\oldparagraph\@undefined
4766     \let\paragraph\bxjs@frozen@paragraph
4767     \let\subparagraph\bxjs@frozen@subparagraph}
4768 \fi

```

以上。

```

4769 \fi

```

## E.9 paragraph のマーク

BXJS クラスでは \paragraph の見出しの前に \jsParagraphMark で指定したマークが付加され、既定ではこれは “■” である。しかし、この規定は \paragraph が本来のレイア

ウトを保っている、すなわち「行内見出しである」「節番号が付かない」ことが前提になっていると考えられる。Pandocはこの規定を変更することがある（特に既定で `\paragraph` を別行見出しに再定義する）ため、変更された場合は `\jsParagraphMark` の既定値を空にする。

Pandoc がプレアンブルで行う再定義の結果を調べるため、`begin-document` フックを利用する。

```
4770 \g@addto@macro\bxjs@begin@document@hook{%
4771   \@tempswafalse
```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```
4772   \ifx\oldparagraph\@undefined\else
4773     \@tempwattrue
4774   \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4775   \ifnum\c@secnumdepth>3
4776     \@tempwattrue
4777   \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。

```
4778   \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4779     \let\jsParagraphMark\@empty
4780   \fi\fi}
```

## E.10 全角空白文字

LaTeX でない入力では、全角空きを入れるために全角空白文字（U+3000）が使われる可能性があるので、全角空白文字を和文文字でなく空きとして扱うようにしておく。

※ (u)pLaTeX では対応できないので対象外。

`\pandocZWSpace` 全角空白文字の入力で実行されるコード。

```
4781 \def\pandocZWSpace{\zwspace}
```

全角空白文字の入力で `\pandocZWSpace` が実行されるようにする。

```
4782 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4783   \catcode"3000=\active
4784   \begingroup \catcode`\!=7
4785   \protected\gdef!!!!3000{\pandocZWSpace}
4786   \endgroup
4787 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4788   \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4789   \bxjs@protected\def\bxjs@zsp@char{\pandocZWSpace}
4790 \fi\fi
```

## E.11 hyperref 対策

hyperref の unicode オプションの固定を行う。

**TODO:** hyperref の開発状況を鑑みる限り、unicode オプションの固定処理は“危険”だと思われるので、可能ならば廃止したい。

```
4791 \if j\jsEngine
4792   \bxjs@fix@hyperref@unicode{false}
4793 \else
4794   \bxjs@fix@hyperref@unicode{true}
4795 \fi
```

## E.12 Pandoc 要素に対する和文用の補正

■**重要要素** 重要 (Strong) 要素に対する L<sup>A</sup>T<sub>E</sub>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)。

```
4796 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
4797   \let\orgtextbf\textbf
4798   \DeclareRobustCommand\pandocTextbf[1]{%
4799     \begingroup
4800       \let\textbf\orgtextbf
4801       \strong{#1}%
4802     \endgroup}%
4803   \g@addto@macro\bxjs@begin@document@hook{%
4804     \let\textbf\pandocTextbf}
4805 \fi\fi
```

`\strong` の書体を設定する。

```
4806 \jsAtEndOfClass{%
4807   \ifx\strongfontdeclare\@undefined\else
4808     \ifcase\bxjs@jp@strong
4809     \or \strongfontdeclare{\sffamily}%
4810     \or \strongfontdeclare{\sffamily\bfseries}%
4811   \fi
4812 \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<sup>A</sup>T<sub>E</sub>X 出力は `\texttt` となる。`\texttt` の両端に欧文ゴーストが入るようにする。さらに `\verb` の外側にも欧文ゴーストが入るようにする。

```
4813 \ifbxjs@jp@fix@code
```

`bxghost` パッケージが利用できる場合はその機能を利用する。使えない場合は自前実装を用いる。

```
4814   \ifbxjs@bxghost@available
4815     \RequirePackage[verb]{bxghost}[2020/01/31]% v0.3.0
4816     \let\bxjs@eghostguarded\eghostguarded
4817   \else
```

```

4818 \chardef\bxjs@eghost@c=23
4819 \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4820 \else\ifx l\jsEngine \ltjsetparameter{alxspmode={\bxjs@eghost@c,3}}
4821 \else\ifx x\jsEngine %no-op
4822 \else \let\bxjs@eghost@c\@undefined
4823 \fi\fi\fi
4824 \ifx\bxjs@eghost@c\@undefined\else
4825 \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4826 \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4827 \def\bxjs@eghostguarded#1{%
4828 \bxjs@pan@eghost\null#1\null\bxjs@pan@eghost}
4829 \fi
4830 \fi
4831 \ifx\bxjs@eghostguarded\@undefined\else
4832 \let\orgtexttt\texttt
4833 \DeclareRobustCommand\pandocTexttt[1]{%
4834 \ifmmode \nfss@text{\ttfamily #1}%
4835 \else
4836 \ifvmode \leavevmode \fi
4837 \bxjs@eghostguarded{\begingroup\ttfamily#1\endgroup}%
4838 \fi}
4839 \g@addto@macro\bxjs@begin@document@hook{%
4840 \let\texttt\pandocTexttt}

```

bxghost を使わない場合の \verb の処理。

※ bxghost の実装を参考にした。

```

4841 \ifbxjs@bxghost@available\else
4842 \expandafter\def\expandafter\verb\expandafter{%
4843 \expandafter\bxjs@pan@eghost\verb}
4844 \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4845 \fi
4846 \fi
4847 \fi

```

## E.13 ifPDFTeX スイッチ

Pandoc モードでは Pandoc の既定テンプレートを（無理やり）(u)pTeX に対応させることを目的にしている。

旧版のテンプレートでは ifxetex と ifluatex パッケージを読み込んだ上で「XeTeX でも LuaTeX でもないものは pdfTeX」という前提の動作をしていた。よって、(u)pTeX に対応させる際には「pdfTeX 用の処理が実行される」ことを前提にすればよかった。

ところが、Pandoc の 2.12 版では iftex パッケージが導入されて「pdfTeX の判定を直接 \ifPDFTeX で行う」ように改修された。このため、(u)pTeX での実行でどのコードが実行されるかを予測することが困難になってしまった。

これに対処するため、「文書ファイルのプレアンブル実行中に限って \ifPDFTeX が（実際とは異なり）真になるようにする」という細工を施すことで、従来通り「pdfTeX 用の処理



が実行される」前提が維持されるようにする。

```
4848 \if j\jsEngine
```

```
\bxjs@check@frontier \bxjs@check@frontier\CS は現在のパッケージ読込ネストレベルが丁度 1 であるときにのみ \CS を実行する。
```

```
4849 \def\bxjs@check@frontier{%
4850   \expandafter\bxjs@check@frontier@a\@currnamestack\noindent...\@nil}
4851 \def\bxjs@check@frontier@a#1#2#3#4#5\@nil#6{%
4852   \ifx\noindent#4#6\fi}
```

```
\bxjs@unforge@ifPDFTeX \ifPDFTeX を偽（正しい値）にする。
```

```
4853 \@onlypreamble\bxjs@unforge@ifPDFTeX
4854 \def\bxjs@unforge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iffalse}}
```

```
\bxjs@forge@ifPDFTeX \ifPDFTeX を真（偽装した値）にする。
```

```
4855 \@onlypreamble\bxjs@forge@ifPDFTeX
4856 \def\bxjs@forge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iftrue}}
```

```
\bxjs@unload@forge@ifPDFTeX \ifPDFTeX に対する細工を無効化する。
```

```
4857 \def\bxjs@unload@forge@ifPDFTeX{%
4858   \bxjs@unforge@ifPDFTeX
4859   \global\let\bxjs@check@frontier\@gobble}
```

プレアンブル開始時は \ifPDFTeX は真で、終了時に偽装を無効化する。filehook のフックで「パッケージ読込中は偽装を解除する」ことを実現している。

```
4860 \jsAtEndOfClass{\bxjs@forge@ifPDFTeX}
4861 \ifjsWitheTeX
4862   \AtBeginOfFile{\bxjs@check@frontier\bxjs@unforge@ifPDFTeX}%
4863   \AtEndOfFile{\bxjs@check@frontier\bxjs@forge@ifPDFTeX}%
4864   \g@addto@macro\bxjs@endpreamble@hook{\bxjs@unload@forge@ifPDFTeX}
4865 \else
4866   \g@addto@macro\bxjs@begin@document@hook{\bxjs@unload@forge@ifPDFTeX}
4867 \fi
4868 \fi
```

## E.14 完了

おしまい。

```
4869 %</pandoc>
```

和文ドライバ実装はここまで。

```
4870 %</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。

4871 %<\*anc>

## 付録 G 補助パッケージ：bxjscompat 🤖

古いやつをどうにかするためのムニャムニャ。

### G.1 準備

4872 %<\*compat>

4873 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4874 \let\bxac@engine=n

4875 \def\bxac@do#1#2{%

4876 \edef\bxac@tmpa{\string#1}%

4877 \edef\bxac@tmpb{\meaning#1}%

4878 \ifx\bxac@tmpa\bxac@tmpb #2\fi}

4879 \bxac@do\XeTeXversion{\let\bxac@engine=x}

4880 \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。

4881 \ifx\jsAtEndOfClass\undefined

4882 \let\bxac@delayed@if@bxjs\@firstofone

4883 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass

4884 \fi

\ImposeOldLuaTeXBehavior \ImposeOldLuaTeXBehavior は 0.85 版以降の LuaTeX を一時的に pdfTeX と互換である

\RevokeOldLuaTeXBehavior ように見せかける。 \RevokeOldLuaTeXBehavior で元に戻ることができる。

※エンジンが LuaTeX 以外の場合は何もしない。

4885 \newif\ifbxac@in@old@behavior

4886 \let\ImposeOldLuaTeXBehavior\relax

4887 \let\RevokeOldLuaTeXBehavior\relax

### G.2 XeTeX 部分

4888 \ifx x\bxac@engine

■文字クラスの設定 XeTeX の文字クラス (\XeTeXcharclass) の Unicode 規定に基づく設定は、初期の版ではフォーマットに組み込まれていたが、2016/02/01 以降の L<sup>A</sup>T<sub>E</sub>X カーネルでは「必要に応じて後から設定用のファイルを読み込む」方式に変更された。ここでは「設定されている状態」を担保する。

※ちなみに、Xe<sub>g</sub>TeX に「文字間トークン挿入」の機能が導入されたのは 0.997 版（2007 年頃）からのようだ。

ただし xeCJK が読込済ならば（そちらが適切に設定しているはずなので）何もしない。

```
4889 \ifx\XeTeXcharclass\@undefined\else
4890 \bxac@delayed@if@bxjs{%
4891   \@ifpackageloaded{xeCJK}{-}{-}%else
```

設定が未実行の状態ならば、設定用のファイルを読む。

```
4892   \ifx\XeTeXalloc@intercharclass\@undefined\else
4893     \ifnum\XeTeXalloc@intercharclass=\z@
4894       \PackageInfo\bxac@pkgname
4895         {Setting up interchar class for CJK...\@gobble}%
4896       \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4897         \XeTeXalloc@intercharclass=3
4898       }{%else
4899       \PackageWarning\bxac@pkgname
4900         {Cannot find file 'load-unicode-xetex-classes.tex'%
4901         \@gobble}%
4902     }%
4903   \fi\fi
```

フォーマット組込んだ時代の設定は不完全なところがあるので補正する。

```
4904   \ifnum\XeTeXcharclass"3041=\z@
4905     \PackageInfo\bxac@pkgname
4906       {Adjusting interchar class for CJK...\@gobble}%
4907     \@for\bxac@tmpb:={%
4908       3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4909       3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4910       30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4911       31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4912       31FF%
4913     }\do{\XeTeXcharclass"\bxac@tmpb=\@ne}%
4914   \fi
4915 }%
4916 }
4917 \fi
```

漢字および完成形ハングルのカテゴリコードが確実に 11 になっているようにする。

```
4918 \chardef\bxac@tmpb=11
4919 \def\bxac@do#1#2{%
4920   \@tempcnta=#1\relax
4921   \unless\ifnum\catcode\@tempcnta=\bxac@tmpb
4922     \chardef\bxac@tmpa=#2\relax
4923     \@whilenum{\@tempcnta<\bxac@tmpa}\do{%
4924       \catcode\@tempcnta\bxac@tmpb \advance\@tempcnta\@ne}%
4925     \fi}
4926 \bxac@do{"4E00}{-9FCD}
```

以上。

```
4927 \fi
```

### G.3 LuaTeX 部分

```
4928 \ifx l\bxac@engine
```

0.82~0.84 版の LuaTeX を (0.81 版以前と同様に) 「pdfTeX の拡張である」 ように見せかける処理。

※恐らく必要な場面はなかったと思われるので、外しておく。

```
4929 %\unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4930 % \chardef\pdftexversion=200
4931 % \def\pdftexrevision{0}
4932 % \let\pdftexbanner\luatexbanner
4933 %\fi\fi
```

`\ImposeOldLuaTeXBehavior` 0.85 版以降であるかを検査する。

```
\RevokeOldLuaTeXBehavior 4934 \begingroup\expandafter\expandafter\expandafter\endgroup
4935 \expandafter\ifx\csname outputmode\endcsname\relax\else
```

該当する場合、以下の 5 つの pdfTeX 拡張プリミティブを復帰させることになる。

```
4936 \def\bxac@ob@list{%
4937   \do{\let}\pdfoutput{\outputmode}%
4938   \do{\let}\pdfpagewidth{\pagewidth}%
4939   \do{\let}\pdfpageheight{\pageheight}%
4940   \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4941   \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}%
4942 \def\bxac@ob@do#1#2{\begingroup
4943   \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4944 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4945   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4946   \else \let#3#1\relax \let#1\@undefined
4947   \fi}
4948 \protected\def\ImposeOldLuaTeXBehavior{%
4949   \unless\ifbxac@in@old@behavior
4950     \bxac@in@old@behaviortrue
4951     \let\do\bxac@ob@do \bxac@ob@list
4952   \fi}
4953 \protected\def\RevokeOldLuaTeXBehavior{%
4954   \ifbxac@in@old@behavior
4955     \bxac@in@old@behaviorfalse
4956     \let\do\bxac@ob@do \bxac@ob@list
4957   \fi}
4958 \fi
```

漢字および完成形ハングルのカテゴリコードが確実に 11 になっているようにする。

```
4959 \directlua{
4960   local function range(cs, ce, cc, ff)
4961     if ff or not tex.getcatcode(cs) == cc then
4962       local setcc = tex.setcatcode
4963       for c = cs, ce do setcc(c, cc) end
4964     end
```

```

4965     end
4966     range(0x3400, 0x4DB5, 11, false)
4967     \ifnum\luatexversion>64
4968     range(0x4DB5, 0x4DBF, 11, true)
4969     range(0x4E00, 0x9FCC, 11, false)
4970     range(0x9FCD, 0x9FFF, 11, true)
4971     range(0xAC00, 0xD7A3, 11, false)
4972     range(0x20000, 0x2A6D6, 11, false)
4973     range(0x2A6D7, 0x2A6FF, 11, true)
4974     range(0x2A700, 0x2B734, 11, false)
4975     range(0x2B735, 0x2B73F, 11, true)
4976     range(0x2B740, 0x2B81D, 11, false)
4977     range(0x2B81E, 0x2B81F, 11, true)
4978     range(0x2B820, 0x2CEA1, 11, false)
4979     range(0x2CEA2, 0x2FFFD, 11, true)
4980     \fi
4981 }

```

以上。

```
4982 \fi
```

## G.4 完了

おしまい。

```
4983 %</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

```

4984 %<*cjkcat>
4985 \def\bxjx@pkgname{bxjscjkat}
4986 \newcount\bxjx@canta
4987 \@onlypreamble\bxjx@tmpdo
4988 \@onlypreamble\bxjx@tmpdo@a
4989 \@onlypreamble\bxjx@tmpdo@b

```

\bxjx@engine エンジンの種別。

```

4990 \let\bxjx@engine=n
4991 \def\bxjx@tmpdo#1#2{%
4992   \edef\bxjx@tmpa{\string#1}%
4993   \edef\bxjx@tmpb{\meaning#1}%
4994   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4995 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4996 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4997 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}

```

```

4998 \bxjx@tmpdo\pdftexversion{\let\bxjx@engine=p}
4999 \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。

```

5000 \def\bxjx@tmpdo#1#2{%
5001   \if#1\bxjx@engine
5002     \@ifpackageloaded{#2}{-}{%else
5003       \PackageError\bxjx@pkgname
5004         {Package '#2' must be loaded}%
5005         {Package loading is aborted.\MessageBreak\@ehc}%
5006     \endinput}
5007   \fi}
5008 \bxjx@tmpdo{p}{bxcjkatype}
5009 \bxjx@tmpdo{x}{xeCJK}
5010 \bxjx@tmpdo{l}{luatexja}

```

古い L<sup>A</sup>T<sub>E</sub>X の場合、`\TextOrMath` は `fixltx2e` パッケージで提供される。

```

5011 \ifx\TextOrMath\@undefined
5012   \RequirePackage{fixltx2e}
5013 \fi

```

## H.2 和文カテゴリコードの設定

upL<sup>A</sup>T<sub>E</sub>X の場合、和文カテゴリコードの設定を LuaT<sub>E</sub>X-ja と（ほぼ）等価なものに変更  
 する。

※ LuaT<sub>E</sub>X-ja との相違点：A830、A960、1B000。

```

5014 \if u\bxjx@engine
5015 \@for\bxjx@tmpa:={%
5016 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
5017 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
5018 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
5019 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
5020 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
5021 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
5022 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
5023 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
5024 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
5025 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
5026 AB30,AB70,ABCO,D800,DB80,DC00,E000,FB00,FB50,FE00,%
5027 FE70,FFF0,%
5028 10000,10080,10100,10140,10190,101D0,10280,102A0,%
5029 102E0,10300,10330,10350,10380,103A0,10400,10450,%
5030 10480,104B0,10500,10530,10600,10800,10840,10860,%
5031 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
5032 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
5033 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
5034 11200,11280,112B0,11300,11400,11480,11580,11600,%

```

```

5035 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
5036 11C70,11D00,12000,12400,12480,13000,14400,16800,%
5037 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
5038 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
5039 1EE00,1F000,1F030,1F0A0,1F300,1F600,1F650,1F680,%
5040 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
5041 00C0%
5042 }\do{%
5043 \@tempcnta="\bxjx@tmpa\relax
5044 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
5045 \chardef\bxjx@tmpb\kcatcode\@tempcntb
5046 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
5047 \fi

```

### H.3 ギリシャ・キリル文字の扱い

「特定 CJK 曖昧文字」について、和文・欧文扱いを制御できるようにする。ここで「特定 CJK 曖昧文字」とは以下に該当する文字の集合を指す：

- Unicode と JIS X 0213 に共通して含まれるギリシャ文字・キリル文字。
- Latin-1 の上位部分と JIS X 0208 に共通して含まれる文字（LuaTeX-jd 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。

```

5048 \@onlypreamble\bxjx@grkcyr@list
5049 \def\bxjx@grkcyr@list{%
5050 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
5051 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
5052 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
5053 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
5054 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
5055 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
5056 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
5057 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
5058 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
5059 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
5060 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
5061 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
5062 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
5063 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
5064 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
5065 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
5066 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
5067 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
5068 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU

```

|      |  |                         |
|------|--|-------------------------|
| 5069 | $\backslash\mathrm{do}\{03\mathrm{A}5\}\{\mathrm{LGR}\}\{\backslash\mathrm{textUpsilon}\}\{\backslash\mathrm{Upsilon}\}\%$         | % GR. C. L. UPSILON     |
| 5070 | $\backslash\mathrm{do}\{03\mathrm{A}6\}\{\mathrm{LGR}\}\{\backslash\mathrm{textPhi}\}\{\backslash\mathrm{Phi}\}\%$                 | % GR. C. L. PHI         |
| 5071 | $\backslash\mathrm{do}\{03\mathrm{A}7\}\{\mathrm{LGR}\}\{\backslash\mathrm{textChi}\}\{\backslash\mathrm{X}\}\%$                   | % GR. C. L. CHI         |
| 5072 | $\backslash\mathrm{do}\{03\mathrm{A}8\}\{\mathrm{LGR}\}\{\backslash\mathrm{textPsi}\}\{\backslash\mathrm{Psi}\}\%$                 | % GR. C. L. PSI         |
| 5073 | $\backslash\mathrm{do}\{03\mathrm{A}9\}\{\mathrm{LGR}\}\{\backslash\mathrm{textOmega}\}\{\backslash\mathrm{Omega}\}\%$             | % GR. C. L. OMEGA       |
| 5074 | $\backslash\mathrm{do}\{03\mathrm{B}1\}\{\mathrm{LGR}\}\{\backslash\mathrm{textalpha}\}\{\backslash\mathrm{alpha}\}\%$             | % GR. S. L. ALPHA       |
| 5075 | $\backslash\mathrm{do}\{03\mathrm{B}2\}\{\mathrm{LGR}\}\{\backslash\mathrm{textbeta}\}\{\backslash\mathrm{beta}\}\%$               | % GR. S. L. BETA        |
| 5076 | $\backslash\mathrm{do}\{03\mathrm{B}3\}\{\mathrm{LGR}\}\{\backslash\mathrm{textgamma}\}\{\backslash\mathrm{gamma}\}\%$             | % GR. S. L. GAMMA       |
| 5077 | $\backslash\mathrm{do}\{03\mathrm{B}4\}\{\mathrm{LGR}\}\{\backslash\mathrm{textdelta}\}\{\backslash\mathrm{delta}\}\%$             | % GR. S. L. DELTA       |
| 5078 | $\backslash\mathrm{do}\{03\mathrm{B}5\}\{\mathrm{LGR}\}\{\backslash\mathrm{textepsilon}\}\{\backslash\mathrm{epsilon}\}\%$         | % GR. S. L. EPSILON     |
| 5079 | $\backslash\mathrm{do}\{03\mathrm{B}6\}\{\mathrm{LGR}\}\{\backslash\mathrm{textzeta}\}\{\backslash\mathrm{zeta}\}\%$               | % GR. S. L. ZETA        |
| 5080 | $\backslash\mathrm{do}\{03\mathrm{B}7\}\{\mathrm{LGR}\}\{\backslash\mathrm{texteta}\}\{\backslash\mathrm{eta}\}\%$                 | % GR. S. L. ETA         |
| 5081 | $\backslash\mathrm{do}\{03\mathrm{B}8\}\{\mathrm{LGR}\}\{\backslash\mathrm{texttheta}\}\{\backslash\mathrm{theta}\}\%$             | % GR. S. L. THETA       |
| 5082 | $\backslash\mathrm{do}\{03\mathrm{B}9\}\{\mathrm{LGR}\}\{\backslash\mathrm{textiota}\}\{\backslash\mathrm{iota}\}\%$               | % GR. S. L. IOTA        |
| 5083 | $\backslash\mathrm{do}\{03\mathrm{B}\mathrm{A}\}\{\mathrm{LGR}\}\{\backslash\mathrm{textkappa}\}\{\backslash\mathrm{kappa}\}\%$    | % GR. S. L. KAPPA       |
| 5084 | $\backslash\mathrm{do}\{03\mathrm{B}\mathrm{B}\}\{\mathrm{LGR}\}\{\backslash\mathrm{textlambda}\}\{\backslash\mathrm{lambda}\}\%$  | % GR. S. L. LAMDA       |
| 5085 | $\backslash\mathrm{do}\{03\mathrm{B}\mathrm{C}\}\{\mathrm{LGR}\}\{\backslash\mathrm{textmu}\}\{\backslash\mathrm{mu}\}\%$          | % GR. S. L. MU          |
| 5086 | $\backslash\mathrm{do}\{03\mathrm{B}\mathrm{D}\}\{\mathrm{LGR}\}\{\backslash\mathrm{textnu}\}\{\backslash\mathrm{nu}\}\%$          | % GR. S. L. NU          |
| 5087 | $\backslash\mathrm{do}\{03\mathrm{B}\mathrm{E}\}\{\mathrm{LGR}\}\{\backslash\mathrm{textxi}\}\{\backslash\mathrm{xi}\}\%$          | % GR. S. L. XI          |
| 5088 | $\backslash\mathrm{do}\{03\mathrm{B}\mathrm{F}\}\{\mathrm{LGR}\}\{\backslash\mathrm{textomicron}\}\{\backslash\mathrm{o}\}\%$      | % GR. S. L. OMICRON     |
| 5089 | $\backslash\mathrm{do}\{03\mathrm{C}0\}\{\mathrm{LGR}\}\{\backslash\mathrm{textpi}\}\{\backslash\mathrm{pi}\}\%$                   | % GR. S. L. PI          |
| 5090 | $\backslash\mathrm{do}\{03\mathrm{C}1\}\{\mathrm{LGR}\}\{\backslash\mathrm{textrho}\}\{\backslash\mathrm{rho}\}\%$                 | % GR. S. L. RHO         |
| 5091 | $\backslash\mathrm{do}\{03\mathrm{C}2\}\{\mathrm{LGR}\}\{\backslash\mathrm{textvarsigma}\}\{\backslash\mathrm{varsigma}\}\%$       | % GR. S. L. FINAL SIGMA |
| 5092 | $\backslash\mathrm{do}\{03\mathrm{C}3\}\{\mathrm{LGR}\}\{\backslash\mathrm{textsigma}\}\{\backslash\mathrm{sigma}\}\%$             | % GR. S. L. SIGMA       |
| 5093 | $\backslash\mathrm{do}\{03\mathrm{C}4\}\{\mathrm{LGR}\}\{\backslash\mathrm{texttau}\}\{\backslash\mathrm{tau}\}\%$                 | % GR. S. L. TAU         |
| 5094 | $\backslash\mathrm{do}\{03\mathrm{C}5\}\{\mathrm{LGR}\}\{\backslash\mathrm{textupsilon}\}\{\backslash\mathrm{upsilon}\}\%$         | % GR. S. L. UPSILON     |
| 5095 | $\backslash\mathrm{do}\{03\mathrm{C}6\}\{\mathrm{LGR}\}\{\backslash\mathrm{textphi}\}\{\backslash\mathrm{phi}\}\%$                 | % GR. S. L. PHI         |
| 5096 | $\backslash\mathrm{do}\{03\mathrm{C}7\}\{\mathrm{LGR}\}\{\backslash\mathrm{textchi}\}\{\backslash\mathrm{chi}\}\%$                 | % GR. S. L. CHI         |
| 5097 | $\backslash\mathrm{do}\{03\mathrm{C}8\}\{\mathrm{LGR}\}\{\backslash\mathrm{textpsi}\}\{\backslash\mathrm{psi}\}\%$                 | % GR. S. L. PSI         |
| 5098 | $\backslash\mathrm{do}\{03\mathrm{C}9\}\{\mathrm{LGR}\}\{\backslash\mathrm{textomega}\}\{\backslash\mathrm{omega}\}\%$             | % GR. S. L. OMEGA       |
| 5099 | $\backslash\mathrm{do}\{0401\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{Y}\mathrm{O}\}\{\backslash\}\%$             | % CY. C. L. IO          |
| 5100 | $\backslash\mathrm{do}\{0410\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{A}\}\{\backslash\}\%$                       | % CY. C. L. A           |
| 5101 | $\backslash\mathrm{do}\{0411\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{B}\}\{\backslash\}\%$                       | % CY. C. L. BE          |
| 5102 | $\backslash\mathrm{do}\{0412\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{V}\}\{\backslash\}\%$                       | % CY. C. L. VE          |
| 5103 | $\backslash\mathrm{do}\{0413\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{G}\}\{\backslash\}\%$                       | % CY. C. L. GHE         |
| 5104 | $\backslash\mathrm{do}\{0414\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{D}\}\{\backslash\}\%$                       | % CY. C. L. DE          |
| 5105 | $\backslash\mathrm{do}\{0415\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{E}\}\{\backslash\}\%$                       | % CY. C. L. IE          |
| 5106 | $\backslash\mathrm{do}\{0416\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{Z}\mathrm{H}\}\{\backslash\}\%$             | % CY. C. L. ZHE         |
| 5107 | $\backslash\mathrm{do}\{0417\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{Z}\}\{\backslash\}\%$                       | % CY. C. L. ZE          |
| 5108 | $\backslash\mathrm{do}\{0418\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{I}\}\{\backslash\}\%$                       | % CY. C. L. I           |
| 5109 | $\backslash\mathrm{do}\{0419\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{I}\mathrm{S}\mathrm{HRT}\}\{\backslash\}\%$ | % CY. C. L. SHORT I     |
| 5110 | $\backslash\mathrm{do}\{041\mathrm{A}\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{K}\}\{\backslash\}\%$              | % CY. C. L. KA          |
| 5111 | $\backslash\mathrm{do}\{041\mathrm{B}\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{L}\}\{\backslash\}\%$              | % CY. C. L. EL          |
| 5112 | $\backslash\mathrm{do}\{041\mathrm{C}\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{M}\}\{\backslash\}\%$              | % CY. C. L. EM          |
| 5113 | $\backslash\mathrm{do}\{041\mathrm{D}\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{N}\}\{\backslash\}\%$              | % CY. C. L. EN          |
| 5114 | $\backslash\mathrm{do}\{041\mathrm{E}\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{O}\}\{\backslash\}\%$              | % CY. C. L. O           |
| 5115 | $\backslash\mathrm{do}\{041\mathrm{F}\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{P}\}\{\backslash\}\%$              | % CY. C. L. PE          |
| 5116 | $\backslash\mathrm{do}\{0420\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{R}\}\{\backslash\}\%$                       | % CY. C. L. ER          |
| 5117 | $\backslash\mathrm{do}\{0421\}\{\mathrm{T}2\mathrm{A}\}\{\backslash\mathrm{CYR}\mathrm{S}\}\{\backslash\}\%$                       | % CY. C. L. ES          |



|  |                       |
|--|-----------------------|
| 5118 \do{0422}{T2A}{\CYRT}{}%                    | % CY. C. L. TE        |
| 5119 \do{0423}{T2A}{\CYRU}{}%                    | % CY. C. L. U         |
| 5120 \do{0424}{T2A}{\CYRF}{}%                    | % CY. C. L. EF        |
| 5121 \do{0425}{T2A}{\CYRH}{}%                    | % CY. C. L. HA        |
| 5122 \do{0426}{T2A}{\CYRC}{}%                    | % CY. C. L. TSE       |
| 5123 \do{0427}{T2A}{\CYRCH}{}%                   | % CY. C. L. CHE       |
| 5124 \do{0428}{T2A}{\CYRSH}{}%                   | % CY. C. L. SHA       |
| 5125 \do{0429}{T2A}{\CYRSHCH}{}%                 | % CY. C. L. SHCHA     |
| 5126 \do{042A}{T2A}{\CYRHRSN}{}%                 | % CY. C. L. HARD SIGN |
| 5127 \do{042B}{T2A}{\CYRERY}{}%                  | % CY. C. L. YERU      |
| 5128 \do{042C}{T2A}{\CYRSFTSN}{}%                | % CY. C. L. SOFT SIGN |
| 5129 \do{042D}{T2A}{\CYREREV}{}%                 | % CY. C. L. E         |
| 5130 \do{042E}{T2A}{\CYRYU}{}%                   | % CY. C. L. YU        |
| 5131 \do{042F}{T2A}{\CYRYA}{}%                   | % CY. C. L. YA        |
| 5132 \do{0430}{T2A}{\cyra}{}%                    | % CY. S. L. A         |
| 5133 \do{0431}{T2A}{\cyrb}{}%                    | % CY. S. L. BE        |
| 5134 \do{0432}{T2A}{\cyrv}{}%                    | % CY. S. L. VE        |
| 5135 \do{0433}{T2A}{\cyrg}{}%                    | % CY. S. L. GHE       |
| 5136 \do{0434}{T2A}{\cyrd}{}%                    | % CY. S. L. DE        |
| 5137 \do{0435}{T2A}{\cyre}{}%                    | % CY. S. L. IE        |
| 5138 \do{0436}{T2A}{\cyrz}{}%                    | % CY. S. L. ZHE       |
| 5139 \do{0437}{T2A}{\cyrz}{}%                    | % CY. S. L. ZE        |
| 5140 \do{0438}{T2A}{\cyri}{}%                    | % CY. S. L. I         |
| 5141 \do{0439}{T2A}{\cyrishrt}{}%                | % CY. S. L. SHORT I   |
| 5142 \do{043A}{T2A}{\cyrk}{}%                    | % CY. S. L. KA        |
| 5143 \do{043B}{T2A}{\cyr1}{}%                    | % CY. S. L. EL        |
| 5144 \do{043C}{T2A}{\cyr}{}%                     | % CY. S. L. EM        |
| 5145 \do{043D}{T2A}{\cyrn}{}%                    | % CY. S. L. EN        |
| 5146 \do{043E}{T2A}{\cyro}{}%                    | % CY. S. L. O         |
| 5147 \do{043F}{T2A}{\cyrp}{}%                    | % CY. S. L. PE        |
| 5148 \do{0440}{T2A}{\cyrr}{}%                    | % CY. S. L. ER        |
| 5149 \do{0441}{T2A}{\cyr}{}%                     | % CY. S. L. ES        |
| 5150 \do{0442}{T2A}{\cyrt}{}%                    | % CY. S. L. TE        |
| 5151 \do{0443}{T2A}{\cyru}{}%                    | % CY. S. L. U         |
| 5152 \do{0444}{T2A}{\cyrf}{}%                    | % CY. S. L. EF        |
| 5153 \do{0445}{T2A}{\cyrh}{}%                    | % CY. S. L. HA        |
| 5154 \do{0446}{T2A}{\cyrc}{}%                    | % CY. S. L. TSE       |
| 5155 \do{0447}{T2A}{\cyrch}{}%                   | % CY. S. L. CHE       |
| 5156 \do{0448}{T2A}{\cyrsh}{}%                   | % CY. S. L. SHA       |
| 5157 \do{0449}{T2A}{\cyrshch}{}%                 | % CY. S. L. SHCHA     |
| 5158 \do{044A}{T2A}{\cyrhrdsn}{}%                | % CY. S. L. HARD SIGN |
| 5159 \do{044B}{T2A}{\cyrery}{}%                  | % CY. S. L. YERU      |
| 5160 \do{044C}{T2A}{\cyrfts}{}%                  | % CY. S. L. SOFT SIGN |
| 5161 \do{044D}{T2A}{\cyrrerev}{}%                | % CY. S. L. E         |
| 5162 \do{044E}{T2A}{\cyryu}{}%                   | % CY. S. L. YU        |
| 5163 \do{044F}{T2A}{\cyrya}{}%                   | % CY. S. L. YA        |
| 5164 \do{0451}{T2A}{\cyryo}{}%                   | % CY. S. L. IO        |
| 5165 \do{00A7}{TS1}{\textsection}{\mathsection}% | SECTION SYMBOL        |
| 5166 \do{00A8}{TS1}{\textasciidieresis}{}%       | % DIAERESIS           |

```

5167 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
5168 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN
5169 \do{00B4}{TS1}{\textasciicute}{'}% % ACUTE ACCENT
5170 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
5171 \do{00D7}{TS1}{\textttimes}{\times}% % MULTIPLICATION SIGN
5172 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN
5173 }

```

`\mathdegree` 面倒なので補っておく。

```

5174 \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。

```

5175 \newif\ifbxjx@gcc@CJK

```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```

5176 \newcommand*\greekasCJK{%
5177   \bxjx@gcc@CJKtrue}

```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```

5178 \newcommand*\nogreekasCJK{%
5179   \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ーを引き継いで、〈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。

```

5180 \def\bxjx@tmpdo#1\relax{%
5181   \def\bxjx@fake@grk##1##2{%
5182     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
5183   \def\bxjx@fake@grk@a##1##2\@nil##3##4{%
5184     \ifx\##1\%
5185       \bxjx@canta##4\divide\bxjx@canta\@cclvi
5186       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
5187       \mathchar\bxjx@canta
5188     \else ##3\fi}
5189 }\expandafter\bxjx@tmpdo\string\mathchar\relax

```

## ■pdfTeX・upTeX の場合

```

5190 \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<sup>A</sup>T<sub>E</sub>X においても、必ず「inputenc が明示的に読み込まれた」状態になる。

```
5191 \ifpackageloaded{inputenc}{}{%else
5192   \RequirePackage[utf8]{inputenc}}
5193 \def\bxjx@tmpa{utf8}
5194 \ifx\bxjx@tmpa\inputencdoingname
5195   \PackageWarningNoLine\bxjx@pkgname
5196     {Input encoding changed to utf8}%
5197   \inputencoding{utf8}%
5198 \fi
```

upT<sub>E</sub>X 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。

```
5199 \if u\bxjx@engine
5200 \kcatcode"0370=15
5201 \kcatcode"0400=15
5202 \kcatcode"0500=15
5203 \fi
```

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

```
5204 \def\bxjx@tmpdo#1{%
5205   \@tempcnta="#1\relax
5206   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
5207 \def\bxjx@tmpdo@a#1#2#3#4#5{%
```

引数 = \[bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式 LICR>}

“数式中の動作”を決定する。<数式 LICR> が空（数式非対応）なら警告を出す。

```
5208   \ifx\\#5\\%
5209     \def\bxjx@tmpa{\@inmathwarn#4}%
```

<数式 LICR> が英字である場合は \bxjx@fake@grk で出力する。大文字なら \Pi、小文字なら \pi を基準文字にする。

```
5210   \else\ifcat A\noexpand#5%
5211     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5212       {\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。

```
5213   \else \def\bxjx@tmpa{#5}%
5214   \fi\fi
5215   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
5216   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
```

以降はエンジン種別で分岐する。upT<sub>E</sub>X の場合。

```
5217 \if u\bxjx@engine
5218 \def\bxjx@tmpdo@b#1#2#3#4#5{%
```

引数 = \[bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式中の動作>}

当該の Unicode 文字の動作は「テキストでは <LICR>、数式では <数式中の動作>」となる。LICR は現在エンコーディングで有効な定義がある場合はそれが実行されるはずである。（つまり、現在が LGR である場合はギリシャ文字は常に欧文扱いになる。）それ以外の場合は

LICR を `\bxjx@ja@or@not` に帰着させる。この際に、和文用の定義として当該の `kchardef` を使用し、その制御綴として `\[bxjx@KC/...]` を流用している。

```
5219 \kchardef#1=\@tempcnta
5220 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
5221 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}}
```

pdfTeX の場合も処理はほとんど同じ。ただし、和文用の定義として `\UTF{⟨符号値⟩}` を使う (`\UTF` は `bxckjatype` の命令)。`\[bxjx@KC/...]` は使わないが定義済にする必要がある。

```
5222 \else\if p\bxjx@engine
5223 \def\bxjx@tmpdo@b#1#2#3#4#5{%
5224 \mathchardef#1=\@tempcnta
5225 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
5226 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}}%
5227 \fi\fi
```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```
5228 \let\do\bxjx@tmpdo \bxjx@grkcyr@list
```

`\bxjx@DeclareUnicodeCharacter` `\bxjx@DeclareUnicodeCharacter` を改変して、「特定 CJK 曖昧文字」の場合に再定義を抑制したもの。

```
5229 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
5230 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
5231 \@onlypreamble\bxjx@DeclareUnicodeCharacter
5232 \def\bxjx@DeclareUnicodeCharacter#1#2{%
5233 \count@=#1\relax
5234 \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
5235 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
5236 \else
5237 \wlog{ \space\space skipped defining Unicode char U+#1}%
5238 \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。

```
5239 \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。

```
5240 \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。

```
5241 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
5242 \else \UseTextSymbol{#2}{#3}%
5243 \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。

```
5244 \begingroup
5245 \toks@{\expandafter\DeclareFontEncoding@{#1}{#2}{#3}}
5246 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
```

```

5247 \noexpand\bxjx@swap@DUC@cmd
5248 \the\toks@
5249 \noexpand\bxjx@swap@DUC@cmd}}
5250 \endgroup\next
5251 \def\bxjx@swap@DUC@cmd{%
5252 \let\bxjx@tmpa\DeclareUnicodeCharacter
5253 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
5254 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
5255 \let\bxjx@tmpa\relax}

```

以上。

### ■X<sub>3</sub>TeX・LuaTeX の場合

```
5256 \else\ifnum0\if x\bxjx@engine1\fi\if l\bxjx@engine1\fi>0
```

各文字について、数式中の動作を定義する。

```

5257 \def\bxjx@tmpdo#1{%
5258 \bxjx@cmta="#1\relax
5259 \begingroup
5260 \lccode`~=\bxjx@cmta
5261 \lowercase{\endgroup
5262 \bxjx@tmpdo@a{~}}{#1}}
5263 \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、upL<sup>A</sup>T<sub>E</sub>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。

```

5264 \ifx\\#5\\\let\bxjx@tmpa\relax
5265 \else\ifcat A\noexpand#5%
5266 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5267 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
5268 \else \def\bxjx@tmpa{#5}%
5269 \fi\fi
5270 \ifx\bxjx@tmpa\relax\else
5271 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
5272 \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。

```

5273 \mathchardef\bxjx@tmpa="119
5274 \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。

LuaTeX の場合は、LuaTeX-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。

```

5275 \if l\bxjx@engine
5276 \protected\def\greekasCJK{%
5277 \bxjx@gcc@cjktrue
5278 \ltjsetparameter{jacharrange={+2, +8}}

```

```

5279 \protected\def\nogreekasCJK{%
5280 \bxjx@gcc@cjkgfalse
5281 \ltjsetparameter{jacharrange={-2, -8}}
5282 \fi

```

X<sub>q</sub>TeX の場合、xeCJK は X<sub>q</sub>TeX の文字クラス定義を参照しているので、対象文字の文字クラスを変更する。

```

5283 \if x\bxjx@engine
5284 \let\bxjx@gcc@cjkglist\@empty
5285 \def\do#1#2#3#4{%
5286 \edef\bxjx@gcc@cjkglist{\bxjx@gcc@cjkglist
5287 \noexpand\XeTeXcharclass"#1\bxjx@cmta}}
5288 \bxjx@grkcyrglist
5289 \protected\def\greekasCJK{%
5290 \bxjx@gcc@cjkgtrue
5291 \bxjx@cmta=\@ne \bxjx@gcc@cjkglist}
5292 \protected\def\nogreekasCJK{%
5293 \bxjx@gcc@cjkgfalse
5294 \bxjx@cmta=\z@ \bxjx@gcc@cjkglist}
5295 \fi

```

以上。

```
5296 \fi\fi
```

## H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5297 \nogreekasCJK
```

## H.5 完了

おしまい。

```
5298 %</cjkgcat>
```

## 付録 I 補助パッケージ：bxjspandoc 🐼

Pandoc の L<sup>A</sup>T<sub>E</sub>X 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<sub>E</sub>X 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

```

5299 %<*ancpandoc>
5300 %% このファイルは日本語文字を含みます。

```

```

5301 \def\bxjsp@pkgname{bxjscjkat}

\bxjsp@engine エンジンの種別。

5302 \let\bxjsp@engine=n
5303 \@onlypreamble\bxjsp@do
5304 \def\bxjsp@do#1#2{%
5305   \edef\bxjsp@tmpa{\string#1}%
5306   \edef\bxjsp@tmpb{\meaning#1}%
5307   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
5308 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
5309 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
5310 \bxjsp@do\pdfTeXversion{\let\bxjsp@engine=p}
5311 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}

```

\bxjsp@begin@document@hook 文書本体開始時フック。

```

5312 \@onlypreamble\bxjsp@begin@document@hook
5313 \let\bxjsp@begin@document@hook\empty
5314 \AtBeginDocument{\bxjsp@begin@document@hook}

```

\ifbxjsp@babel@used [スイッチ] Babel が読み込まれたか。

```

5315 \newif\ifbxjsp@babel@used
5316 \g@addto@macro\bxjsp@begin@document@hook{%
5317   \@ifpackageloaded{babel}{\bxjsp@babel@usedtrue}{}}

```

## 1.2 パッケージオプション

english オプションが指定されている場合、\ldots の調整を抑止する。

※つまり、「グローバルの english オプション」が指定されている場合も抑止の対象になる。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<sup>A</sup>T<sub>E</sub>X 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。

```

5318 \newif\ifbxjsp@english
5319 \DeclareOption{english}{\bxjsp@englishttrue}

```

オプション定義はおしまい。

```

5320 \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。

```

5321 \@onlypreamble\pandocSkipLoadFile
5322 \newcommand*\pandocSkipLoadFile[1]{%
5323   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5324 \def\bxjsp@skip@load@file@a#1#2{%
5325   \ifx#1\relax
5326     \def#1{2001/01/01}%
5327     \PackageInfo\bxjsp@pkgname

```

```

5328      {File '#2' marked as loaded\@gobble}%
5329    \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。

```

5330 \@onlypreamble\pandocSkipLoadPackage
5331 \newcommand*\pandocSkipLoadPackage[1]{%
5332   \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  $\text{\LaTeX}$  ではこれで警告が出る。これを抑止する。

$\text{\LaTeX}$  カーネルが新しい場合は `fixltx2e` を読込済にする。

```

5333 \ifx\@IncludeInRelease\undefined\else
5334   \pandocSkipLoadPackage{fixltx2e}
5335 \fi

```

## 1.5 cmap パッケージ

エンジンが  $(\text{u})\text{\pLaTeX}$  のときに `cmap` パッケージが読み込まれるのを阻止する。（実際は警告が出るだけで無害であるが。）

```

5336 \if j\bxjisp@engine
5337   \pandocSkipLoadPackage{cmap}
5338 \fi

```

## 1.6 microtype パッケージ

警告が多すぎなので消す。

```

5339 \if j\bxjisp@engine \else
5340   \PassOptionsToPackage{verbose=silent}{microtype}
5341 \fi

```

エンジンが  $(\text{u})\text{\pLaTeX}$  のときに `microtype` パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は `standard` ドライバでこの処理を行っていたが、元来は `Pandoc` 用の処理なので、1.5 版で `pandoc` に移動。

```

5342 \if j\bxjisp@engine
5343   \pandocSkipLoadPackage{microtype}
5344   \newcommand*\UseMicrotypeSet[2][{}]{%
5345     \fi

```



## I.7 Unicode 文字変換対策

Pandoc で  $\text{\LaTeX}$  形式に書き出す場合は、元データ中の一部の Unicode 文字を「 $\text{\LaTeX}$  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。

```
…→\ldots{} ‘→` ’→' “→`` ”→''
```

日本語  $\text{\LaTeX}$  では「 $\text{\LaTeX}$  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、「`\ldots` を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の `\ldots` の実装。非数式である場合は代わりに … を実行する。

※以前は「Pandoc が必ず `\ldots{}` の形で書き出す」ことを利用して後続に `{}` があるかで「元が … であるか」を判断していた。ところが、Pandoc 2.7 版で `{}` を必ずしも付けなくなったため、1.9f 版で非数式の `\ldots` を全て … に戻す動作に変更した。

```
5346 \DeclareRobustCommand{\pandocLdots}{%
5347   \let\bxjsp@do\bxjsp@ja@ellipsis
5348   \ifmmode \let\bxjsp@do\bxjsp@org@ldots
5349   \else\ifbxjsp@babel@used
5350     \expandafter\ifx\csname bxjsp@ld/\language\endcsname\relax
5351     \let\bxjsp@do\bxjsp@org@ldots \fi
5352   \fi\fi \bxjsp@do}
5353 \@namedef{bxjsp@ld/japanese}{1}
5354 \def\bxjsp@ja@ellipsis{…}
5355 \let\bxjsp@org@ldots\ldots
```

`\ldots` の実装を `\pandocLdots` に置き換える。

```
5356 \g@addto@macro\bxjsp@begin@document@hook{%
5357   \let\bxjsp@org@ldots\ldots
```

もしここで `\newcommand\pandocLdots{\ldots}` という定義である場合は置き換えない。

```
5358 \long\def\bxjsp@tmpa{\ldots}%
5359 \ifx\pandocLdots\bxjsp@tmpa\else
```

`english` オプションが指定されていてかつ Babel が読み込まれていない場合も置き換えない。

```
5360   \ifnum0\ifbxjsp@english\ifbxjsp@babel@used\else1\fi\fi=0
5361   \let\ldots\pandocLdots
5362   \fi
5363 \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  $\text{Xe}\text{\LaTeX}$ ・ $\text{Lua}\text{\LaTeX}$  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。

```
5364 \chardef\bxjsp@cc@other=12
5365 \@onlypreamble\bxjsp@makeother@range
5366 \def\bxjsp@makeother@range#1#2{%
5367   \@tempcnta"#1\relax \@tempcntb"#2\relax
5368   \loop\ifnum\@tempcnta<\@tempcntb
5369     \catcode\@tempcnta\bxjsp@cc@other
5370     \advance\@tempcnta\@ne
5371   \repeat}
5372 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
5373   \catcode"1F23B=\bxjsp@cc@other
5374   \bxjsp@makeother@range{9FCD}{A000}
5375   \bxjsp@makeother@range{1B002}{1B170}
5376   \bxjsp@makeother@range{2B820}{2EBF0}
5377 \fi
```

## 1.8 PandoLa モジュール

インストール済であれば読み込む。

```
5378 \IfFileExists{bxpandola.sty}{%
5379   \RequirePackage{bxpandola}\relax
5380   \PackageInfo{bxjsp@pkgname
5381     {PandoLa module is loaded\@gobble}
5382 }{}
```

## 1.9 完了

おしまい。

```
5383 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
5384 %</anc>
```