

# 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”)

v2.2c [2020/10/04]






---

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	10
3	和文フォントの変更	39
4	フォントサイズ	40
5	レイアウト	46
5.1	ページレイアウト . . . . .	47
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	61
7	ページスタイル	62
8	文書のマークアップ	65
8.1	表題 . . . . .	65
8.2	章・節 . . . . .	71
8.3	リスト環境 . . . . .	83
8.4	パラメータの設定 . . . . .	90
8.5	フロート . . . . .	92
8.6	キャプション . . . . .	93
9	フォントコマンド	94

10	相互参照	97
10.1	目次の類 . . . . .	97
10.2	参考文献 . . . . .	102
10.3	索引 . . . . .	104
10.4	脚注 . . . . .	105
11	段落の頭へのグルー挿入禁止	108
12	いろいろなロゴ	112
13	amsmath との衝突の回避	112
14	初期設定	113
付録 A	和文ドライバの仕様 	117
付録 B	和文ドライバ : minimal 	118
B.1	補助マクロ . . . . .	118
B.2	(u)pTeX 用の設定 . . . . .	120
B.3	pdfTeX 用の処理 . . . . .	125
B.4	X <sub>Y</sub> TeX 用の処理 . . . . .	126
B.5	後処理 (エンジン共通) . . . . .	126
付録 C	和文ドライバ : standard 	129
C.1	準備 . . . . .	130
C.2	和文ドライバパラメタ . . . . .	130
C.3	共通処理 (1) . . . . .	131
C.4	pTeX 用設定 . . . . .	138
C.5	pdfTeX 用設定 : CJK + bxcjkatype . . . . .	142
C.6	X <sub>Y</sub> TeX 用設定 : xeCJK + zxjatype . . . . .	144
C.7	LuaTeX 用設定 : LuaTeX-ja . . . . .	146
C.8	共通処理 (2) . . . . .	150
付録 D	和文ドライバ : modern 	151
D.1	フォント設定 . . . . .	151
D.2	fixltx2e 読込 . . . . .	151
D.3	和文カテゴリコード . . . . .	152
D.4	完了 . . . . .	152
付録 E	和文ドライバ : pandoc 	152
E.1	準備 . . . . .	152
E.2	和文ドライバパラメタ . . . . .	152
E.3	dupload システム . . . . .	153

E.4	lang 変数 . . . . .	154
E.5	geometry 変数 . . . . .	157
E.6	CJKmainfont 変数 . . . . .	157
E.7	Option clash 対策 . . . . .	157
E.8	paragraph のマーク . . . . .	157
E.9	全角空白文字 . . . . .	158
E.10	hyperref 対策 . . . . .	158
E.11	Pandoc 要素に対する和文用の補正 . . . . .	159
E.12	完了 . . . . .	160
付録 F	補助パッケージ一覧 罨	160
付録 G	補助パッケージ : bxjscompat 罨	160
G.1	準備 . . . . .	160
G.2	X <sub>Y</sub> TeX 部分 . . . . .	161
G.3	LuaTeX 部分 . . . . .	161
G.4	完了 . . . . .	163
付録 H	補助パッケージ : bxjscjkat 罨	163
H.1	準備 . . . . .	163
H.2	和文カテゴリコードの設定 . . . . .	164
H.3	ギリシャ・キリル文字の扱い . . . . .	165
H.4	初期設定 . . . . .	172
H.5	完了 . . . . .	172
付録 I	補助パッケージ : bxjspandoc 罨	172
I.1	準備 . . . . .	172
I.2	パッケージオプション . . . . .	173
I.3	パッケージ読込の阻止 . . . . .	173
I.4	fixltx2e パッケージ . . . . .	173
I.5	cmap パッケージ . . . . .	174
I.6	microtype パッケージ . . . . .	174
I.7	Unicode 文字変換対策 . . . . .	174
I.8	PandoLa モジュール . . . . .	175
I.9	完了 . . . . .	175

## 1 はじめに

---

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

<code>&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。

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

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

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

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

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

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

## ■ BXJS クラス特有の設定

---

長さ値の指定で式を利用可能にするため `calc` を読み込む。

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

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

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

クラスの本体ではこの他に `geometry` パッケージが読み込まれる。

**TODO:** 依存パッケージの情報。

互換性のための補助パッケージを読み込む。

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

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

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

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

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

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

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

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

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

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

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

非サポートのエンジンの場合は強制終了させる。

※ NTT `jTeX` と `Omega` 系。

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

`LuaTeX` の場合、本クラス用の `Lua` モジュールを用意する。

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

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

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

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

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

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

※ `LuaTeX` 0.8x 版でのプリミティブ名変更に対応。

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

```

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

```

`\ifbxjs@TUenc` [スイッチ] L<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 X<sub>Ǝ</sub>L<sup>A</sup>T<sub>E</sub>X または LuaT<sub>E</sub>X で動かしている場合は、既定のエンコーディングが TU になる。それ以外の場合は、既定のエンコーディングは OT1 である。

```

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

```

`\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。

```

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

```

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

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

```

79 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
80   #1\expandafter\@firstoftwo
81   \else\expandafter\@secondoftwo
82   \fi}

```

`\bxjs@cslet` `\bxjs@cslet{〈名前 1〉}\制御綴 :`

```

83 \def\bxjs@cslet#1{%
84   \expandafter\let\csname#1\endcsname}

```

`\bxjs@csletcs` `\bxjs@csletcs{〈名前 1〉}{〈名前 2〉} :`

```

85 \def\bxjs@csletcs#1#2{%
86   \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}

```

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

```

87 \def\bxjs@catopt#1#2{%
88   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

```

`\bxjs@ifplus` `\@ifstar` の + 版。

```

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

```

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

```

90 \let\c@bxjs@tempcnta\@tempcnta
91 \def\bxjs@gset@tempcnta{\setcounter{bxjs@tempcnta}}

\jsSetQHLengh \jsSetQHLengh\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）。

92 \def\jsSetQHLengh#1#2{%
93   \begingroup
94     \bxjs@parse@qh{#2}%
95     \ifx\bxjs@tmpb\relax
96       \setlength\@tempdima{#2}%
97       \xdef\bxjs@g@tmpa{\the\@tempdima}%
98     \else \global\let\bxjs@g@tmpa\bxjs@tmpb
99     \fi
100   \endgroup
101   #1=\bxjs@g@tmpa\relax}

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

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

```



今の段階では Q/H だけが使用可能。

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

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

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

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

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

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

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

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

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

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

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

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

```
135 \def\jsAtEndOfClass{%
```

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

一時的な手続き用の制御綴。

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

```
138 \@onlypreamble\bxjs@tmpdo@a
```

```
139 \@onlypreamble\bxjs@tmpdo@b
```

```
140 \@onlypreamble\bxjs@tmpdo@c
```

```
141 \@onlypreamble\bxjs@tmpdo@d
```

Lua<sub>T</sub><sub>E</sub>X の場合、原版のコード中のコントロールワード中に現れる日本語文字のカテゴリコードを一時的に 11 に変更する。クラス読み込み終了時点で元に戻される。

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

```
142 \if 1\jsEngine
```

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

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

```
145 \bxjs@pre@jadriver@hook
```

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

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

```
148 \@tfor\bxjs@tmpa:=和西暦\do
```

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

```
150 \fi
```

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

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

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

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

```

153 \if@compatibility
154   \ClassError\bxjs@clsname
155   {Something went chaotic!\MessageBreak
156     (How come '\string\documentstyle' is there?)\MessageBreak
157     I cannot go a single step further...}
158   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
159     then there'll still be hope....}
160   \expandafter\@firstofone
161 \else \expandafter\@gobble
162 \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。

```
163 \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。

```
164 \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。

```
165 %<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。

```
166 %<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。

---

```
167 %<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。

---

```
168 \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載しては mm 単位に切り捨てたものが A1, A2, …です。

B 判は JIS と ISO で定義が異なります。JIS では B0 判の面積が  $1.5\text{ m}^2$  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， $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$  の `b5paper` は  $250\text{ mm} \times 176\text{ mm}$  ですが， $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$  の `b5paper` は  $257\text{ mm} \times 182\text{ mm}$  になっています。ここでは  $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$  にならって JIS に従いました。

デフォルトは `a4paper` です。

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

---

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

```
169 \onlypreamble\bxjs@setpaper
170 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
171 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
172 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
173 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
174 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
175 \DeclareOption{b4paper}{\bxjs@setpaper{b4j}}
176 \DeclareOption{b5paper}{\bxjs@setpaper{b5j}}
177 \DeclareOption{b6paper}{\bxjs@setpaper{b6j}}
178 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
179 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
180 \DeclareOption{b4j}{\bxjs@setpaper{b4j}}
181 \DeclareOption{b5j}{\bxjs@setpaper{b5j}}
182 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
183 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
184 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
185 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
186 \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。

```
187 \def\bxjs@tmpb#1#2{\DeclareOption{#1}{\bxjs@setpaper{#2}}}
188 \@for\bxjs@tmpa:={%
189   a0,a1,a2,c0,c1,c2,c3,c4,c5,c6,ansia,ansib,ansic,ansid,ansie%
190 }\do{\edef\bxjs@next{%
191   \noexpand\bxjs@tmpb{\bxjs@tmpa paper}{\bxjs@tmpa paper}%
192 }\bxjs@next}
193 \bxjs@tmpb{screen}{screen}
```

ただし `b?paper` は JIS の B 列に従う。

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

B?paper で ISO の B 列を指定できるようにする。

```
197 \@for\bxjs@tmpa:={0,1,2,3,4,5,6}\do{\edef\bxjs@next{%
198   \noexpand\bxjs@tmpb{A\bxjs@tmpa paper}{a\bxjs@tmpa paper}%
199   \noexpand\bxjs@tmpb{B\bxjs@tmpa paper}{b\bxjs@tmpa paper}%
200 }\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。

```
201 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210trueemm}{283trueemm}}}
202 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182trueemm}{230trueemm}}}
203 \DeclareOption{screenpaper}{\bxjs@setpaper{screen}}
204 \DeclareOption{custompaper}{}
```

---

■横置き 用紙の縦と横の長さを入れ換えます。

```
205 \newif\if@landscape
206 \@landscapefalse
207 \DeclareOption{landscape}{\@landscapetrue}
```

■slide オプション slide を新設しました。

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

```
208 \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。

```
209 %<!slide>\@slidefalse
210 %<slide>\@slidetrue
```

---

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

---

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

```
211 \newcommand{\@ptsize}{0}
212 \def\bxjs@param@basefontsize{10pt}
213 \def\jsUnusualPtSize{-20}
```

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

```
214 \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。

```
215 \jsSetQHLength\@tempdima{#1}%
216 \edef\bxjs@param@basefontsize{\the\@tempdima}%
217 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
218 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
219 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
220 \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。

```
221 \def\bxjs@setjbasefontsize#1{%
222 \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。

```
223 \newif\ifjsc@mag
224 \newif\ifjsc@mag@xreal
225 %\let\jsc@magscale\undefined

226 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
227 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
228 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
229 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
230 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
231 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
232 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
233 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
234 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
235 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
236 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
237 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
238 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
239 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
240 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
```

```

241 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
242 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
243 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
244 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

245 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
246 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
247 \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>La</sub>T<sub>E</sub>X 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>La</sub>T<sub>E</sub>X 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。

```

248 \if j\jsEngine
249 \hour\time \divide\hour by 60\relax
250 \@tempcnta\hour \multiply\@tempcnta 60\relax
251 \minute\time \advance\minute-\@tempcnta
252 \DeclareOption{tombow}{%
253   \tombowtrue \tombowdatetrue
254   \setlength{\@tombowwidth}{.1\p}%
255   \@bannertoken{%
256     \jobname\space(\number\year-\two@digits\month-\two@digits\day
257     \space\two@digits\hour:\two@digits\minute)}}%
258   \maketombowbox}
259 \DeclareOption{tombo}{%
260   \tombowtrue \tombowdatefalse
261   \setlength{\@tombowwidth}{.1\p}%
262   \maketombowbox}
263 \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。

```

264 \if j\jsEngine
265 \DeclareOption{mentuke}{%
266   \tombowtrue \tombowdatefalse
267   \setlength{\@tombowwidth}{\z@}%
268   \maketombowbox}
269 \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。

```
270 \DeclareOption{oneside}{\@twosidefalse \mparswitchfalse}
271 \DeclareOption{twoside}{\@twosidetrue \mparswitchtrue}
272 \DeclareOption{vartwoside}{\@twosidetrue \mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
273 \DeclareOption{onecolumn}{\@twocolumnfalse}
274 \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。

```
275 \DeclareOption{titlepage}{\@titlepagetrue}
276 \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。

```
277 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
278 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
279 %<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`  $\text{\LaTeX}$  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。

```
280 \def\eqnarray{%
281   \stepcounter{equation}%
282   \def\@currentlabel{\p@equation\theequation}%
283   \global\@eqnswtrue
284   \m@th
285   \global\@eqcnt\z@
286   \tabskip\@centering
287   \let\\\@eqnocr
288   $$\everycr{}\halign to\displaywidth\bgroup
289     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
290     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
291     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
292     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
293     \tabskip\z@skip
294   \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。

```
295 \DeclareOption{leqno}{\input{leqno.clo}}
296 \DeclareOption{fleqn}{\input{fleqn.clo}}
297 % fleqn 用の eqnarray 環境の再定義
298 \def\eqnarray{%
299   \stepcounter{equation}%
300   \def\@currentlabel{\p@equation\theequation}%
301   \global\@eqnswtrue\m@th
302   \global\@eqcnt\z@
303   \tabskip\mathindent
304   \let\@=\@eqnocr
305   \setlength\abovedisplayskip{\topsep}%
306   \ifvmode
307     \addtolength\abovedisplayskip{\partopsep}%
308   \fi
309   \addtolength\abovedisplayskip{\parskip}%
310   \setlength\belowdisplayskip{\abovedisplayskip}%
311   \setlength\belowdisplayshortskip{\abovedisplayskip}%
312   \setlength\abovedisplayshortskip{\abovedisplayskip}%
313   $$\everycr{}\halign to\linewidth% $$
314   \bgroup
315     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
316     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
317     &\global\@eqcnt\tw@
318     $\displaystyle{##}$\hfil \tabskip\@centering
319     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
320   \tabskip\z@skip\cr
321 }
```

■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。

```
322 % \DeclareOption{openbib}{%
323 %   \AtEndOfPackage{%
324 %     \renewcommand\@openbib@code{%
325 %       \advance\leftmargin\bibindent
326 %       \itemindent -\bibindent
327 %       \listparindent \itemindent
328 %       \parsep \z@}%
329 %     \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：

```
330 %\let\bxjs@enablejfam\undefined
```

`enablejfam` オプションの処理。

```
331 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
332 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
333 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
334 \define@key{bxjs}{enablejfam}[true]{%
335   \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。

```
336 \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。

```
337 \newif\ifjsDraft
338 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
339 \DeclareOption{final}{\jsDraftfalse \overfullrule=0pt }
```

---

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

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

---

BXJS クラスではここは和文ドライバの管轄。

---

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

---

BXJS クラスでは geometry パッケージがこの処理を行う。

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

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

```
340 \newif\ifbxjs@papersize
341 \bxjs@papersizetrue
342 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
343 \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。

---

```
344 \newif\if@english
345 \@englishfalse
346 \DeclareOption{english}{\@englishttrue}
```

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

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

---

BXJS では当初から bxjsreport クラスが用意されている。

---


■jslogo パッケージの読み込み L<sup>A</sup>T<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。

---

```
347 \newif\if@jslogo \@jslogofalse
348 \DeclareOption{jslogo}{\@jslogotrue}
349 \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。

```
350 \mathchardef\bxjs@isc@ll=128
351 \mathchardef\bxjs@isc@sl=259
352 \def\bxjs@isc@sl@h{65539 }
353 \def\bxjs@invscale#1#2{%
354   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
355   \ifdim\@tempdima<\bxjs@isc@ll\p@
356     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
357     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
358   \else
359     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
360     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
361   \fi
362   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
363   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
364   \@tempdimb\@tempcnta\@ne
365   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
366   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
367   \@whiledim\@tempdimb<\@tempdimc\do{%
368     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
369     \advance\@tempcntb\@ne \divide\@tempcntb\@tw@
370     \ifdim #2\@tempcntb>\@tempdima
371       \advance\@tempcntb\@ne \@tempdimc=\@tempcntb\@ne
372     \else \@tempdimb=\@tempcntb\@ne \fi}%
373   \xdef\bxjs@gtmpa{\the\@tempdimb}%
374   \endgroup #1=\bxjs@gtmpa\relax}
```

---

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

`pandoc` オプションは、Pandoc で  $\text{\LaTeX}$  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。

```
375 \DeclareOption{pandoc}{%
```

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

※実際の和文ドライバ・エンジン設定より優先される。

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

ドライバオプションを `dvi=dvipdfmx` 相当に変更する。

※これは実際のドライバ設定で上書きできる (オプション宣言順に注意)。

**TODO:** できない気がする…。

```
380 \def\bxjs@driver@opt{dvipdfmx}%  
381 \bxjs@dvi@opttrue}
```

---

## ■エンジン・ドライバオプション

---

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

```
382 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
383 %\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。

```
384 \DeclareOption{autodetect-engine}{%  
385   \let\bxjs@engine@given=*}  
386 \DeclareOption{latex}{%  
387   \def\bxjs@engine@opt{latex}%  
388   \let\bxjs@engine@given=n}  
389 \DeclareOption{platex}{%  
390   \def\bxjs@engine@opt{platex}%  
391   \let\bxjs@engine@given=j}  
392 \DeclareOption{uplatex}{%  
393   \def\bxjs@engine@opt{uplatex}%  
394   \let\bxjs@engine@given=u}  
395 \DeclareOption{xelatex}{%  
396   \def\bxjs@engine@opt{xelatex}%  
397   \let\bxjs@engine@given=x}  
398 \DeclareOption{pdflatex}{%  
399   \def\bxjs@engine@opt{pdflatex}%  
400   \let\bxjs@engine@given=p}  
401 \DeclareOption{lualatex}{%  
402   \def\bxjs@engine@opt{lualatex}%  
403   \let\bxjs@engine@given=l}  
404 \DeclareOption{platex-ng}{%  
405   \def\bxjs@engine@opt{platex-ng}%  
406   \let\bxjs@engine@given=g}  
407 \DeclareOption{platex-ng*}{%  
408   \def\bxjs@engine@opt{platex-ng*}%  
409   \let\bxjs@platexng@nodrv=t%  
410   \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。

```
411 %\let\bxjs@driver@given\@undefined  
412 \let\bxjs@driver@@dvimode=0
```

```

413 \let\bxjs@driver@@dvipdfmx=1
414 \let\bxjs@driver@@pdfmode=2
415 \let\bxjs@driver@@xetex=3
416 \let\bxjs@driver@@dvips=4
417 \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。

```

418 %\let\bxjs@driver@opt\@undefined

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

```

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

※ 2.0 版より dvipdfmx-if-dvi は非推奨となった。

```

443 \DeclareOption{dvipdfmx-if-dvi}{%
444   \ClassWarningNoLine\bxjs@clsname
445     {The old option 'dvipdfmx-if-dvi' is DEPRECATED\MessageBreak
446       and may be abolished in future!\MessageBreak
447       You should write 'dvi=dvipdfmx' instead}%
448   \setkeys{bxjs}{dvi=dvipdfmx}}
449

```

---

## ■その他の BXJS 独自オプション

---

**TODO:** 互換用オプションを分離する (2.0 版で?)。

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

※ 2.0 版より、既定値を常に真とする。

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

`nobigcode` / `bigcode` オプションの定義。

```
451 \DeclareOption{nobigcode}{%
```

```
452   \bxjs@bigcodefalse}
```

```
453 \DeclareOption{bigcode}{%
```

```
454   \bxjs@bigcodetrue}
```

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

```
455 \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。

```
456 \DeclareOption{nooldfontcommands}{%
```

```
457   \bxjs@oldfontcommandsfalse}
```

```
458 \DeclareOption{oldfontcommands}{%
```

```
459   \bxjs@oldfontcommandstrue}
```

---

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

```
460 \DeclareOption{winjis}{%
```

```
461   \ClassWarningNoLine\bxjs@clsname
```

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

```
463 }
```

```
464 \DeclareOption{mingoth}{%
```

```
465   \ClassWarningNoLine\bxjs@clsname
```

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


```
467 }
```

```
468 \DeclareOption{jis}{%
```

```
469   \ClassWarningNoLine\bxjs@clsname
```

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

```
471 }
```

■ `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。

```
472 \DeclareOption*{%
```

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

```
474   \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。

```

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

```

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

```

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

```

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

```

486 \@onlypreamble\bxjs@declare@bool@option
487 \def\bxjs@declare@bool@option#1#2{%
488   \define@key{bxjs}{#1}[true]{%
489     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
490     \bxjs@error@keyval{#1}{##1}%
491     \else \@nameuse{bxjs@#2##1}%
492     \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。

```

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

```

`\jsScale` 〔実数値マクロ〕 和文スケール値。

```

504 \def\jsScale{0.924715}

```

`\bxjs@base@opt` 明示された base オプションの値。

```

505 %\let\bxjs@base@opt\@undefined

```

base オプションの処理。

```

506 \define@key{bxjs}{base}{%
507   \edef\bxjs@base@opt{#1}%
508   \bxjs@setbasefontsize{#1}}
509 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

```

\bxjs@jbase@opt 明示された jbase オプションの値。

```

510 %\let\bxjs@jbase@opt\@undefined

```

jbase オプションの処理。

```

511 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
512 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}

```

\bxjs@scale@opt 明示された scale オプションの値。

```

513 %\let\bxjs@scale@opt\@undefined

```

scale オプションの処理。

```

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

```

noscale オプションの処理。

```

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

```

\bxjs@param@mag mag オプションの値。

```

519 \let\bxjs@param@mag\relax

```

mag オプションの処理。

```

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

```

paper オプションの処理。

```

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

```

\bxjs@jadriver 和文ドライバの名前。

```

522 \let\bxjs@jadriver\relax
523 %\let\bxjs@jadriver@opt\@undefined

```

ja オプションの処理。

※jadriver は 0.9 版で用いられた旧称。

※単なる ja という指定は無視される (Pandoc 対策)。

```

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

```

\jsJaFont 和文フォント設定の名前。

```

527 \let\jsJaFont\@empty

```

jafont オプションの処理。

```

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

```



`\jsJaParam` 和文ドライバパラメタの文字列。

```
529 \let\jsJaParam\@empty
```

`japaram` オプションの処理。

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

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

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

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

```
533 \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。

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

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

```
536 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

`\bxjs@magstyle@default` は既定の値を表す。

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

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

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

```
540 \fi\fi
```

```
541 \ifjsWithpTeXng
```

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

```
543 \fi
```

```
544 \let\bxjs@magstyle\bxjs@magstyle@default
```

`magstyle` オプションの処理。

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

```
546 \expandafter\let\expandafter\bxjs@magstyle\csname
```

```
547 bxjs@magstyle@#1\endcsname
```

```
548 \ifx\bxjs@magstyle\relax
```

```
549 \ClassError\bxjs@clsname
```

```
550 {Invalid value '#1' for option magstyle}\@ehc
```

```
551 \let\bxjs@magstyle\bxjs@magstyle@default
```

```
552 \fi}
```

`\bxjs@geometry` `geometry` オプションの値。

```
553 \let\bxjs@geometry@class=c
```

```
554 \let\bxjs@geometry@user=u
```

```
555 \let\bxjs@geometry\bxjs@geometry@class
```

`geometry` オプションの処理。

```
556 \define@key{bxjs}{geometry}{%
```

```
557 \expandafter\let\expandafter\bxjs@geometry\csname
```

```
558 bxjs@geometry@#1\endcsname
```

```
559 \ifx\bxjs@geometry\relax
```

```
560 \ClassError\bxjs@clsname
```

```

561     {Invalid value '#1' for option geometry}\@ehc
562     \let\bxjs@geometry\bxjs@geometry@class
563     \fi}

\ifbxjs@fancyhdr 〔スイッチ〕 fancyhdr パッケージに対する調整を行うか。
564 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

    fancyhdr オプションの処理。
565 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
566 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
567 \define@key{bxjs}{fancyhdr}[true]{%
568     \bxjs@set@keyval{fancyhdr}{#1}{}}

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

    DVI モードのドライバとドライバ種別との対応。
570 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
571 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
572 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
573 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
574 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

    dvi オプションの処理。
575 \define@key{bxjs}{dvi}{%
576     \expandafter\let\expandafter\bxjs@tmpa\csname
577         bxjs@dvidriver@@#1\endcsname
578     \ifx\bxjs@tmpa\relax
579         \ClassError\bxjs@clsname
580         {Invalid value '#1' for option dvi}\@ehc
581     \else

        \bxjs@driver@given を未定義にしていることに注意。
582         \def\bxjs@driver@opt{#1}%
583         \let\bxjs@driver@given\undefined
584         \bxjs@dvi@opttrue
585     \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。
586 \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。
587 \newif\ifbxjs@force@chapterabstract
588 %<book>\bxjs@force@chapterabstracttrue

    layout オプションの処理。
589 \@namedef{bxjs@kv@layout@v1}{%

```

```

590 %<book>\bxjs@layout@buggyhmargintrue
591 %<report>\bxjs@force@chapterabstracttrue
592 }
593 \@namedef{bxjs@kv@layout@v2}{%
594 %<book>\bxjs@layout@buggyhmarginfalse
595 %<report>\bxjs@force@chapterabstractfalse
596 }
597 \define@key{bxjs}{layout}{%
598   \bxjs@set@keyval{layout}{#1}{}}

```

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

```

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

```

`\bxjs@textwidth@opt` textwidth の指定値。

```

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

```

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

```

605 %\let\bxjs@number@of@lines@opt\@undefined
606 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
607 \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。

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

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

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

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

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

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

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

```
648 \newif\ifbxjs@usezw \bxjs@usezwtrue
649 \bxjs@declare@bool@option{use-zw}{usezw}
650 \DeclareOption{nozw}{\setkeys{bxjs}{use-zw=false}}
651 \DeclareOption{zw}{\setkeys{bxjs}{use-zw=true}}
```

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

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

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

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

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

```

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

```

---

## ■ 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。

```

664 \let\bxjs@org@removeelement\@removeelement
665 \def\@removeelement#1#2#3{%
666   \def\reserved@a{#2}%
667   \ifx\reserved@a\@empty \let#3\@empty
668   \else \bxjs@org@removeelement{#1}{#2}{#3}%
669   \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。

```

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

```

後処理

```

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

```

## ■ グローバルオプションの整理

---

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

```

684 \def\bxjs@tmpdo{%

```

```

685 \def\bxjs@tmpa{\@gobble}%
686 \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
687 \let\@classoptionslist\bxjs@tmpa}
688 \def\bxjs@tmpdo@a#1,{%
689 \ifx\@nil#1\relax\else
690 \bxjs@tmpdo@b#1{}\@nil
691 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
692 \expandafter\bxjs@tmpdo@a
693 \fi}
694 \def\bxjs@tmpdo@b#1#\bxjs@tmpdo@c}
695 \def\bxjs@tmpdo@c#1\@nil{%
696 \ifx\@nil#1\@nil \@tempwattrue \else \@tempswafalse \fi}
697 \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。

```

698 \@expandtwoargs\@removeelement
699 {papersize}\@classoptionslist\@classoptionslist
700 \@expandtwoargs\@removeelement
701 {10pt}\@classoptionslist\@classoptionslist
702 \@expandtwoargs\@removeelement
703 {noscale}\@classoptionslist\@classoptionslist

```

---

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

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

---

ここのコードを削除。

---

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

---

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

```

704 \ifx\bxjs@jadriver@opt\@undefined\else
705 \let\bxjs@jadriver\bxjs@jadriver@opt
706 \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。

```

707 \let\bxjs@tmpb\jsEngine
708 \ifx j\bxjs@tmpb\ifjsWithpTeXng

```

```

709 \let\bxjs@tmpb=g
710 \fi\fi
711 \ifx j\bxjs@tmpb\ifjsWithupTeX
712 \let\bxjs@tmpb=u
713 \fi\fi
714 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
715 \let\bxjs@tmpb=n
716 \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。)

```

717 \ifx *\bxjs@engine@given
718 \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。

```

719 \ifx j\bxjs@engine@given
720 \g@addto@macro\@classoptionslist{,latex}
721 \else\ifx u\bxjs@engine@given
722 \g@addto@macro\@classoptionslist{,uplatex}
723 \fi\fi
724 \fi
725 \ifx\bxjs@engine@given\@undefined\else
726 \ifx\bxjs@engine@given\bxjs@tmpb\else
727 \ClassError\bxjs@clsname
728 {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
729 \fi
730 \fi


```

エンジンが pT<sub>E</sub>X-ng の場合、グローバルオプションに uplatex を追加する。

```

731 \ifjsWithpTeXng
732 \g@addto@macro\@classoptionslist{,uplatex}
733 \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。

```

734 \@tempswattrue
735 \ifx \bxjs@driver@given\@undefined\else
736 \ifjsInPdfMode
737 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
738 \@tempswafalse
739 \fi
740 \else\ifx x\jsEngine
741 \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
742 \@tempswafalse
743 \fi
744 \else
745 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
746 \@tempswafalse

```

```

747 \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
748 \@tempswafalse
749 \fi\fi
750 \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
751 \@tempswafalse
752 \fi\fi
753 \fi\fi
754 \fi
755 \if@tempswa\else
756 \ClassError\bxjs@clsname
757 {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
758 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

759 \ifjsInPdfMode \@tempswafalse
760 \else\ifx x\jsEngine \@tempswafalse
761 \else\ifjsWithpTeXng \@tempswafalse
762 \else \@tempswatrue
763 \fi\fi\fi
764 \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)。

```

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

```

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

```

769 \ifbxjs@dvi@opt
770 \edef\bxjs@next{%
771 \let\noexpand\bxjs@driver@given
772 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
773 \noexpand\g@addto@macro\noexpand\@classoptionslist
774 {,\bxjs@driver@opt}%
775 }\bxjs@next
776 \fi
777 \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。

```

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

```




784 \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。

785 \ifx\bxjs@driver@given\bxjs@driver@none

786 \bxjs@papersizefalse

787 \fi

■その他の BXJS 特有の後処理  `\bxjs@jadriver` の正規化。値が未指定の場合は `minimal` に変える。ただしエンジンが (u)pTeX である場合は `standard` に変える。

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

788 \def\bxjs@@minimal{minimal}

789 \ifx\bxjs@jadriver\relax

790 \ifx j\jsEngine

791 \def\bxjs@jadriver{standard}

792 \else

793 \ClassWarningNoLine\bxjs@clsname

794 {The option 'ja' is MISSING!!\MessageBreak

795 So 'ja=minimal' is assumed as fallback, but\MessageBreak

796 such implicit setting is now DEPRECATED!\MessageBreak

797 You should write 'ja=minimal' explicitly,\MessageBreak

798 if it is intended}

799 \let\bxjs@jadriver\bxjs@@minimal

800 \fi

801 \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。

802 \ifx\bxjs@jadriver@opt\@undefined\else

803 \ifx\bxjs@engine@given\@undefined

804 \ClassError\bxjs@clsname

805 {An engine option must be explicitly given}%

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

807 engine option.\MessageBreak\@ehc}

808 \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。)

809 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else

810 \ifx\bxjs@magstyle\bxjs@magstyle@mag

811 \let\bxjs@magstyle\bxjs@magstyle@default

812 \ClassError\bxjs@clsname

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

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

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

816 \fi

817 \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。

```
818 \ifx\bxjs@jbase@opt\@undefined\else
819   \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。

```
820   \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
821   \bxjs@invscale\@tempdima\jsScale
822   \bxjs@setbasefontsize{\@tempdima}%
823 \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。

```
824   \ifx\bxjs@scale@opt\@undefined\else
825     \ClassWarningNoLine\bxjs@clsname
826     {Redundant 'scale' option is ignored}%
827   \fi
828   \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
829   \@tempdimb=\bxjs@param@basefontsize\relax
830   \edef\jsScale{\strip@pt\@tempdimb}%
831   \bxjs@invscale\@tempdima\jsScale
832   \edef\jsScale{\strip@pt\@tempdima}%
833 \fi
834 \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。

```
835 \let\Cjascale\jsScale
```

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

```
836 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
837   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
838     \catcode\@tempcnta\active
839     \advance\@tempcnta\@ne
840   \repeat
841 \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。

```
842 \ifbxjs@disguise@js
843 %<book|report>\def\bxjs@js@clsname{jsbook}
844 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
845   \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
846 \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。

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

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

```
848 \ifbxjs@oldfontcommands
```

```
849 \AtEndOfClass{\allowoldfontcommands}
```

```
850 \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

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

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

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

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

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

---

BXJS では出力用紙サイズ記録は `geometry` パッケージが行う。

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

---

```
851 \newdimen\stockwidth \newdimen\stockheight
```

```
852 \begingroup\expandafter\expandafter\expandafter\endgroup
```

```
853 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
```

```
854 % \newdimen\stockwidth \newdimen\stockheight
```

```
855 \setlength{\stockwidth}{\paperwidth}
```

```
856 \setlength{\stockheight}{\paperheight}
```

```
857 \advance \stockwidth 2in
```

```
858 \advance \stockheight 2in
```

```
859 \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。

```
860 %<slide>\def\n@baseline{13}%
```

```

861 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
862 %<!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。

```

863 \ifx\bxjs@magstyle\bxjs@magstyle@mag
864   \jsc@magtrue
865 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
866   \jsc@mag@xrealtrue
867 \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。

```

868 \ifx\bxjs@param@mag\relax
869   \@tempdima=\bxjs@param@basefontsize
870   \advance\@tempdima.001pt \multiply\@tempdima25
871   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
872   \edef\bxjs@param@mag{\the\@tempcnta}
873 \else
874 % mag 値が直接指定された場合
875   \bxjs@gset@tempcnta{\bxjs@param@mag}
876   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
877 % 有効な mag 値の範囲は 1--32768
878   \edef\bxjs@param@mag{\the\@tempcnta}
879   \advance\@tempcnta100000
880   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
881   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
882   \edef\bxjs@param@basefontsize{\the\@tempdima}
883 \fi
884 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
885 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
886 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
887 \edef\jsc@magscale{\strip@pt\@tempdima}
888 \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。

```
889 \newdimen\jsc@mpt
890 \newdimen\jsc@mmm
891 \ifjsc@mag
892   \jsc@mpt=1\p@
893   \jsc@mmm=1mm
894   \catcode`\?=9 % \p@? read as \p@
895 \else
896   \jsc@mpt=\jsc@magscale\p@
897   \jsc@mmm=\jsc@magscale mm
898   \catcode`\?=11 \let\p@?\jsc@mpt
899 \fi
900 \chardef\bxjs@qmc=\catcode`\?\relax
901 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode`\?=12\relax}
```

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

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

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

`\zwspace` 全角幅の水平空き。

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

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

```
908 \ifjsc@mag@xreal
909   \RequirePackage{type1cm}
910   \let\jsc@invscale\bxjs@invscale
```

ムニャムニャムニャ……。

---

```
911 \ifbxjs@TUenc
912   \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
913 \else
914   \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
915 \fi
916 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
```

```

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

```

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

```

\jsc@smallskip
\jsc@medskip 955 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 956 %\def\jsc@medskip{\vspace\jsc@medskipamount}
957 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount
\jsc@bigskipamount

```

```

958 \newskip\jsc@smallskipamount
959 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
960 %\newskip\jsc@medskipamount
961 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
962 %\newskip\jsc@bigskipamount
963 %\jsc@bigskipamoun =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

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

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

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

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

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

```

964 % \ifpapersize
965 %   \setlength{\@tempdima}{\paperwidth}
966 %   \setlength{\@tempdimb}{\paperheight}
967 %   \iftombow
968 %     \advance \@tempdima 2truein
969 %     \advance \@tempdimb 2truein
970 %   \fi
971 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
972 % \fi

```

### 3 和文フォントの変更

---

和文フォントの設定は和文ドライバの管轄。

---

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

[2016-07-14] 2015-01-01 の L<sup>A</sup>T<sub>E</sub>X 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。
-

```

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

```

## 4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

---

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



```

981 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
982 \expandafter\bxjs@tmpa\expandafter{%
983   \set@fontsize{#1}{#2}{#3}%
984 % 末尾にコードを追加
985   \expandafter\def\expandafter\size@update\expandafter{%
986     \size@update
987     \jsFontSizeChanged}%
988 }

```

`\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。

```

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

```

`\jsResetDimen` ユーザ定義用のフック。

```

998 \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。

```

999 \ifjsc@mag
1000   \let\jsc@setfontsize\@setfontsize
1001 \else
1002   \def\jsc@setfontsize#1#2#3{%
1003     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
1004 % microtype 対策
1005   \ifjsWithTeX\if j\jsEngine\else
1006     \def\jsc@setfontsize#1#2#3{%
1007       \edef\bxjs@sfs@next{%
1008         \unexpanded{\@setfontsize#1}%
1009         {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
1010       }\bxjs@sfs@next}
1011   \fi\fi
1012 \fi

```

これらのグルーをもっ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。

---

1013 `\emergencystretch 3\jsZw`

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

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

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

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

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

---

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

---

```
1014 \newif\ifnarrowbaselines
1015 \if@english
1016   \narrowbaselinestrue
1017 \fi
1018 \def\narrowbaselines{%
1019   \narrowbaselinestrue
1020   \skip0=\abovedisplayskip
1021   \skip2=\abovedisplayshortskip
1022   \skip4=\belowdisplayskip
1023   \skip6=\belowdisplayshortskip
1024 % 一時的に警告を無効化する
1025   \let\bxjs@save@nomath\@nomath
1026   \let\@nomath\@gobble
1027   \@currsize\selectfont
1028   \let\@nomath\bxjs@save@nomath
1029   \abovedisplayskip=\skip0
1030   \abovedisplayshortskip=\skip2
1031   \belowdisplayskip=\skip4
1032   \belowdisplayshortskip=\skip6\relax}
1033 \def\widebaselines{\narrowbaselinesfalse\@currsize\selectfont}
```

---

`microtype` パッケージを読み込んだ場合、`\normalsize` 等のフォントサイズ変更命令の定義の中に `if` 文が使われていると、不可解なエラーが発生する。これは `microtype` が邪悪なトリックを使用しているせいなのだが、一応こちら側で対策をとることにする。

```
1034 \def\bxjs@ifnarrowbaselines{%
1035   \ifnarrowbaselines\expandafter\@firstoftwo
1036   \else \expandafter\@secondoftwo
1037   \fi
```

---

`\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。

---

```
1039 \renewcommand{\normalsize}{%
1040   \bxjs@if@narrowbaselines{%
1041     \jsc@setfontsize\normalsize\@xpt\@xipt
1042   }{%else
1043     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1044   }%
```

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

[2003-02-16] ちょっと変えました。

[2009-08-26] T<sub>E</sub>X Q & A 52569 から始まる議論について逡巡していましたが、結局、微調節してみることになりました。

```
1045   \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1046   \abovedisplayshortskip \z@ \@plus3\p@?
1047   \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1048   \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。

```
1049   \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。

```
1050 \normalsize
```

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

---

`\Chs`

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

```
1051 \ifx\Cht\undefined \newdimen\Cht \fi
1052 \ifx\Cdp\undefined \newdimen\Cdp \fi
```

```

1053 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1054 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1055 \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。

---

```

1056 \setlength\Cht{0.88\jsZw}
1057 \setlength\Cdp{0.12\jsZw}
1058 \setlength\Cwd{1\jsZw}
1059 \setlength\Cvs{\baselineskip}
1060 \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。

---

```

1061 \newcommand{\small}{%
1062   \bxjs@if@narrowbaselines{%
1063     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1064     %<kiyou>   \jsc@setfontsize\small{8.8888}{11}%
1065   }{%else
1066     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1067     %<kiyou>   \jsc@setfontsize\small{8.8888}{13.2418}%
1068   }%
1069   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1070   \abovedisplayshortskip \z@ \@plus3\p@?
1071   \belowdisplayskip \abovedisplayskip
1072   \belowdisplayshortskip \belowdisplayskip
1073   \def\@listi{\leftmargin\leftmargin
1074             \topsep \z@
1075             \parsep \z@
1076             \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。

```

1077 \newcommand{\footnotesize}{%
1078   \bxjs@if@narrowbaselines{%
1079     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1080     %<kiyou>   \jsc@setfontsize\footnotesize{8.8888}{11}%
1081   }{%else
1082     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1083     %<kiyou>   \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1084   }%

```

```

1085 \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1086 \abovedisplayshortskip \z@ \@plus2\p@?
1087 \belowdisplayskip \abovedisplayskip
1088 \belowdisplayshortskip \belowdisplayskip
1089 \def\@listi{\leftmargin\leftmargin
1090         \topsep \z@
1091         \parsep \z@
1092         \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行が揃うようにします。

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1093 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1094 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 1095 \if@twocolumn
\HUGE 1096 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
1097 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1098 \else
1099 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
1100 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1101 \fi
1102 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1103 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1104 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1105 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1106 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1107 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

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

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

```

1108 \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。『 $\text{\LaTeX}$  2<sub>ε</sub> 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。

```

1109 % \newcommand{\headfont}{\bfseries}

```

```

1110 \newcommand{\headfont}{\sffamily}
1111 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

## 5 レイアウト

### ■二段組

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

```

1112 %<kiyou>\setlength\columnsep{2\Cwd}
1113 %<kiyou>\setlength\columnsep{28truebp}
1114 \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  
`\normallineskip` 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 1115 \setlength\lineskip{1\jsc@empt}
1116 \setlength\normallineskip{1\jsc@empt}
\normallineskiplimit 1117 \setlength\lineskiplimit{1\jsc@empt}
1118 \setlength\normallineskiplimit{1\jsc@empt}

```

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

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

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

```
1119 \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。

```

1120 \setlength\parskip{\z@}
1121 \if@slide
1122 \setlength\parindent{0\p@}
1123 \else
1124 \setlength\parindent{1\Cwd}
1125 \fi

```

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

```

\@highpenalty 1126 \@lowpenalty 51
1127 \@medpenalty 151
1128 \@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。

```
1129 % \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。

```
1130 % \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。

```
1131 \@onlypreamble\bxjs@bd@pre@geometry@hook
1132 \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。

```
1133 \ifjsc@mag
1134 \mag=\bxjs@param@mag
1135 \fi
1136 \setlength{\topskip}{10\p@?}
```

`\jsSetQHLength` のための和文単位の定義。

```
1137 \def\bxjs@unit@trueQ{0.25truemm}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1138 \def\bxjs@unit@zw{\jsZw}\let\bxjs@unit@zh\bxjs@unit@zw
```

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

```
1139 \def\bxjs@tmpdo{\futurelet\bxjs@tmpa\bxjs@tmpdo@a}
1140 \def\bxjs@tmpdo@a{%
1141   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@tmpdo@b
1142   \else \expandafter\bxjs@tmpdo@c \fi}
1143 \def\bxjs@tmpdo@b#1#2#3\@nil{\edef\bxjs@param@paper{papersize={#1,#2}}}
1144 \def\bxjs@tmpdo@c#1\@nil{\bxjs@tmpdo@d#1,,\@nil}
1145 \def\bxjs@tmpdo@d#1,#2,#3\@nil{%
1146   \ifx\@nil#3\@nil\else \edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1147 \expandafter\bxjs@tmpdo\bxjs@param@paper\@empty\@empty\@nil
```

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

```
1148 \edef\bxjs@layout@paper{%
1149   \ifjsc@mag truedimen,\fi
1150   \if@landscape landscape,\fi
1151   \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。

```

1152 %<*article|report>
1153 \def\bxjs@layout@base{%
1154   headheight=\topskip,footskip=0.03367\paperheight,%
1155   headsep=\footskip-\topskip,includeheadfoot,%
1156 }
1157 \edef\bxjs@layout{\bxjs@layout@base
1158   hscale=0.76,hmarginratio=1:1,%
1159   vscale=0.83,vmarginratio=1:1,%
1160 }
1161 %</article|report>
1162 %<*book>
1163 \def\bxjs@layout@base{%
1164   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1165 }
1166 \ifbxjs@layout@buggyhmargin      %---
1167 % アレ
1168 \edef\bxjs@layout{\bxjs@layout@base
1169   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1170   vscale=0.83,vmarginratio=1:1,%
1171 }
1172 \else                            %---
1173 % 非アレ
1174 \edef\bxjs@layout{\bxjs@layout@base
1175   hmargin=18\jsc@mmm,%
1176   vscale=0.83,vmarginratio=1:1,%
1177 }
1178 \fi                              %---
1179 %</book>
1180 %<*slide>
1181 \def\bxjs@layout@base{%
1182   noheadfoot,%
1183 }
1184 \edef\bxjs@layout{\bxjs@layout@base
1185   hscale=0.9,hmarginratio=1:1,%
1186   vscale=0.944,vmarginratio=1:1,%
1187 }
1188 %</slide>

```

textwidth オプションの設定を反映する。

```

1189 %<!*book>
1190 \ifx\bxjs@textwidth@opt\@undefined\else
1191   \jsSetQHLengh\@tempdima{\bxjs@textwidth@opt}
1192   \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1193 \fi
1194 %</!*book>
1195 \ifx\bxjs@number@of@lines@opt\@undefined\else
1196   \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1197   \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1198 \fi

```



`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1199 \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。

```
1200 %<*book>
1201 \newcommand\jsTextWidthLimit{40}
1202 \@tempdima=\jsTextWidthLimit\Cwd
1203 \ifx\bxjs@textwidth@limit@opt\@undefined\else
1204   \bxjs@set@tempcnta{\bxjs@textwidth@limit@opt}
1205   \@tempdima=\@tempcnta\Cwd
1206 \fi
1207 \ifx\bxjs@textwidth@opt\@undefined\else
1208   \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1209 \fi
1210 \edef\bxjs@textwidth@limit{\the\@tempdima}
1211 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1212   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1213   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1214 \fi
1215 %</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。

```
1216 \def\bxjs@preproc@layout{%
1217   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mppt}
```

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

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

```

1230 \advance\evensidemargin 0.5\@tempdimb
1231 \fi
1232 \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。

```

1233 %<*book>
1234 \@tempdima=\bxjs@textwidth@limit\relax
1235 \ifbxjs@whole@zw@lines
1236 \advance\@tempdima.005pt\relax
1237 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1238 \fi
1239 \ifdim\textwidth>\@tempdima
1240 \textwidth=\@tempdima
1241 \addtolength\evensidemargin{\fullwidth-\textwidth}
1242 \fi
1243 %</book>

```

\textheight 関連の調整。

```

1244 \@tempdimb=\textheight
1245 \advance\textheight-\topskip
1246 \advance\textheight.005pt\relax
1247 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1248 \advance\textheight\topskip
1249 \advance\@tempdimb-\textheight
1250 \advance\topmargin0.5\@tempdimb

```

\headheight 関連の調整。

```

1251 \@tempdima=\topskip
1252 \advance\headheight\@tempdima
1253 \advance\topmargin-\@tempdima

```

marginpar 関連の調整。

```

1254 \setlength\marginparsep{\columnsep}
1255 \setlength\marginparpush{\baselineskip}
1256 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1257 -\textwidth-10\jsc@mmm-\marginparsep}
1258 \ifbxjs@whole@zw@lines
1259 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1260 \fi

```

連動する変数。

```

1261 \maxdepth=.5\topskip
1262 \stockwidth=\paperwidth
1263 \stockheight=\paperheight
1264 }

```

\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。

```

1265 \edef\jsGeometryOptions{%

```

```
1266 \bxjs@layout@paper,\bxjs@layout}
```

---

## ■geometry パッケージを読み込む

---

ムニャムニャ。

```
1267 \def\bxjs@geometry@name{geometry}
1268 \ifbxjs@old@hook@system
1269 \let\bxjs@apply@bd@pre@geometry@hook\AtBeginDocument
1270 \else
1271 \def\bxjs@apply@bd@pre@geometry@hook{%
1272 \AddToHook{begindocument}[\bxjs@geometry@name]}
1273 \fi
```

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

```
1274 \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。

```
1275 \ifbxjs@papersize
1276 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1277 \PassOptionsToPackage{dvipdfm}{geometry}
1278 \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1279 \PassOptionsToPackage{dvipdfm}{geometry}
1280 \fi\fi
1281 \let\bxPapersizeSpecialDone=t
1282 \else
1283 \PassOptionsToPackage{driver=none}{geometry}
1284 \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。

```
1285 \bxjs@apply@bd@pre@geometry@hook
1286 \bxjs@apply@bd@pre@geometry@hook{\bxjs@bd@pre@geometry@hook}
1287 \bxjs@apply@bd@pre@geometry@hook{\ImposeOldLuaTeXBehavior}
1288 \bxjs@preproc@layout
1289 \edef\bxjs@next{%
1290 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1291 }\bxjs@next
1292 \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。

```
1293 \let\bxjs@geometry@driver\Gm@driver
```

1294 \bxjs@postproc@layout

geometry のドライバ自動判別に対する前処理。

1295 \g@addto@macro\bxjs@bd@pre@geometry@hook{%

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

1296 \ifpackagelater{geometry}{2010/02/12}{\}{\else

1297 \PackageError\bxjs@clsname

1298 {Your 'geometry' package is too old (< v5.0)}%

1299 {\@ehc}%

1300 \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。

1301 \ifjsWithpTeXng

1302 \ifx\Gm@driver\@empty

1303 \def\Gm@driver{pdftex}%

1304 \fi

1305 \fi}

\setpagelayout ページレイアウト設定のためのユーザ命令。

1306 \def\setpagelayout{%

1307 \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{\else

1308 \@ifstar{\bxjs@setpagelayout@a\@ne}{\bxjs@setpagelayout@a\@z}}}

1309 \def\bxjs@setpagelayout@a#1#2{%

1310 \ifcase#1% modify

1311 \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%

1312 \or% reset(\*)

1313 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%

1314 \or% semireset(+)

1315 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%

1316 \fi

1317 \bxjs@preproc@layout


1318 \edef\bxjs@next{%

1319 \noexpand\geometry{\bxjs@next}%

1320 }\bxjs@next

1321 \bxjs@postproc@layout}

---

■ geometry パッケージを読み込まない 

---

geometry=user の場合の処理。

1322 \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。

1323 \g@addto@macro\bxjs@begin@document@hook{%

```

1324 \ifdim\textwidth=.5\maxdimen
1325 \ClassError\bxjs@clsname
1326 {Page layout is not properly set}%
1327 {\@ehd}%
1328 \fi}
1329 \def\jsUseMinimalPageLayout{%
1330 \setlength{\textwidth}{6.5in}%
1331 \setlength{\textheight}{8in}}

\setpagelayout はとりあえず無効にしておく。

1332 \let\bxjs@geometry@driver\relax
1333 \def\setpagelayout{%
1334 \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1335 \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1336 \def\bxjs@pagelayout@a#1{%
1337 \ClassError\bxjs@clsname
1338 {Command '\string\setpagelayout' is not supported,\MessageBreak
1339 because 'geometry' value is not 'class'}\@eha}
1340 %
1341 \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。

---

```
1342 %<*jsclasses>
```

## ■ 縦方向のスペース

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

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

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

```

1343 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1344 \if@slide

```

```

1345 \setlength\headheight{0\jsc@empt}
1346 \else
1347 \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
    06-26)
1348 \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) となっていたましたが、ここでは A4 判のときちょうど 1cm となるように、`\paperheight` の 0.03367 倍 (最小 `\baselineskip`) としました。書籍については、フッタは使わないことにして、ゼロにしました。

```

1349 %<*article|kiyou>
1350 \if@slide
1351 \setlength\footskip{0pt}
1352 \else
1353 \setlength\footskip{0.03367\paperheight}
1354 \ifdim\footskip<\baselineskip
1355 \setlength\footskip{\baselineskip}
1356 \fi
1357 \fi
1358 %</article|kiyou>
1359 %<jspf>\setlength\footskip{9\jsc@mmm}
1360 %<*book>
1361 \if@report
1362 \setlength\footskip{0.03367\paperheight}
1363 \ifdim\footskip<\baselineskip
1364 \setlength\footskip{\baselineskip}
1365 \fi
1366 \else
1367 \setlength\footskip{0pt}
1368 \fi
1369 %</book>
1370 %<*report>
1371 \setlength\footskip{0.03367\paperheight}
1372 \ifdim\footskip<\baselineskip
1373 \setlength\footskip{\baselineskip}
1374 \fi
1375 %</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)。

```

1376 %<*article>
1377 \if@slide
1378 \setlength\headsep{0\jsc@empt}
1379 \addtolength\headsep{-\topskip}%% added (2016-10-08)

```

```

1380 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1381 \else
1382 \setlength\headsep{\footskip}
1383 \addtolength\headsep{-\topskip}
1384 \fi
1385 %</article>
1386 %<*book>
1387 \if@report
1388 \setlength\headsep{\footskip}
1389 \addtolength\headsep{-\topskip}
1390 \else
1391 \setlength\headsep{6\jsc@mmm}
1392 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1393 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1394 \fi
1395 %</book>
1396 %<*report>
1397 \setlength\headsep{\footskip}
1398 \addtolength\headsep{-\topskip}
1399 %</report>
1400 %<*jspf>
1401 \setlength\headsep{9\jsc@mmm}
1402 \addtolength\headsep{-\topskip}
1403 %</jspf>
1404 %<*kiyou>
1405 \setlength\headheight{0\jsc@mpt}
1406 \setlength\headsep{0\jsc@mpt}
1407 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1408 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1409 %</kiyou>

```

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

```

1410 \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。

```

1411 \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。

```
1412 %<*article>
1413 \if@slide
1414   \setlength\fullwidth{0.9\paperwidth}
1415 \else
1416   \setlength\fullwidth{0.76\paperwidth}
1417 \fi
1418 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1419 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1420 \setlength\textwidth{\fullwidth}
1421 %</article>
1422 %<*book>
1423 \if@report
1424   \setlength\fullwidth{0.76\paperwidth}
1425 \else
1426   \setlength\fullwidth{\paperwidth}
1427   \addtolength\fullwidth{-36\jsc@mmm}
1428 \fi
1429 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1430 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1431 \setlength\textwidth{\fullwidth}
1432 \if@report \else
1433   \if@twocolumn \else
1434     \ifdim \fullwidth>40zw
1435       \setlength\textwidth{40zw}
1436     \fi
1437   \fi
1438 \fi
1439 %</book>
1440 %<*report>
1441 \setlength\fullwidth{0.76\paperwidth}
1442 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1443 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1444 \setlength\textwidth{\fullwidth}
1445 %</report>
1446 %<*jspf>
1447 \setlength\fullwidth{50zw}
1448 \addtolength\fullwidth{8\jsc@mmm}
1449 \setlength\textwidth{\fullwidth}
1450 %</jspf>
1451 %<*kiyou>
1452 \setlength\fullwidth{48zw}
1453 \addtolength\fullwidth{\columnsep}
```



```

1454 \setlength\textwidth{\fullwidth}
1455 %</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)。

```

1456 %<*article|book|report>
1457 \if@slide
1458   \setlength{\textheight}{0.95\paperheight}
1459 \else
1460   \setlength{\textheight}{0.83\paperheight}
1461 \fi
1462 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1463 \addtolength{\textheight}{-\headsep}
1464 \addtolength{\textheight}{-\footskip}
1465 \addtolength{\textheight}{-\topskip}
1466 \divide\textheight\baselineskip
1467 \multiply\textheight\baselineskip
1468 %</article|book|report>
1469 %<jspf>\setlength{\textheight}{51\baselineskip}
1470 %<kiyou>\setlength{\textheight}{47\baselineskip}
1471 \addtolength{\textheight}{\topskip}
1472 \addtolength{\textheight}{0.1\jsc@mpt}
1473 %<jspf>\setlength{\mathindent}{10\jsc@mmm}

```

`\flushbottom` [2016-07-18] `\textheight` に念のため 0.1 ポイント余裕を持たせているのと同様に、`\flushbottom` にも余裕を持たせます。元の L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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。

```

1474 \def\flushbottom{%

```

```

1475 \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1476 \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。

```

1477 \setlength\marginparsep{\columnsep}
1478 \setlength\marginparpush{\baselineskip}

```

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

```

1479 \setlength{\oddsidemargin}{\paperwidth}
1480 \addtolength{\oddsidemargin}{-\fullwidth}
1481 \setlength{\oddsidemargin}{.5\oddsidemargin}
1482 \iftombow
1483 \addtolength{\oddsidemargin}{-1in}
1484 \else
1485 \addtolength{\oddsidemargin}{-\inv@mag in}
1486 \fi
1487 \setlength{\evensidemargin}{\oddsidemargin}
1488 \if@mparswitch
1489 \addtolength{\evensidemargin}{\fullwidth}
1490 \addtolength{\evensidemargin}{-\textwidth}
1491 \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。

```

1492 \setlength\marginparwidth{\paperwidth}
1493 \addtolength\marginparwidth{-\oddsidemargin}
1494 \addtolength\marginparwidth{-\inv@mag in}
1495 \addtolength\marginparwidth{-\textwidth}
1496 \addtolength\marginparwidth{-10\jsc@mmm}
1497 \addtolength\marginparwidth{-\marginparsep}
1498 \@tempdima=1zw
1499 \divide\marginparwidth\@tempdima
1500 \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)。

```

1501 \setlength\topmargin{\paperheight}

```

```

1502 \addtolength\topmargin{-\textheight}
1503 \if@slide
1504   \addtolength\topmargin{-\headheight}
1505 \else
1506   \addtolength\topmargin{-10\jsc@empt}}%% from -\topskip (2016-10-08); from -
      \headheight (2003-06-26)
1507 \fi
1508 \addtolength\topmargin{-\headsep}
1509 \addtolength\topmargin{-\footskip}
1510 \setlength\topmargin{0.5\topmargin}
1511 %<kiyou>\setlength\topmargin{81truebp}
1512 \iftombow
1513   \addtolength\topmargin{-1in}
1514 \else
1515   \addtolength\topmargin{-\inv@mag in}
1516 \fi
1517 %</jsclasses>

```

#### ■脚注

`\footnotesep` 各脚注の頭に入る支柱 (strut) の高さです。脚注間に余分のアキが入らないように、`\footnotesize` の支柱の高さ (行送りの 0.7 倍) に等しくします。

---

ここは元々は

```
{\footnotesize\global\setlength\footnotesep{\baselineskip}}
```

としていたが、そもそも `\global\setlength~` は calc 使用時には有意義な動作をしない。`\global\footnotesep` だと所望の値が得られるが、同時に `\footnotesize` のフォントを固定させてしまうという副作用をもつ。なので、実際の設定値を直接使うことにする。

---

```
1518 \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。

```
1519 \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  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}_{2_{\epsilon}}$  本体で定義されていますが、ここで設定変更します。本文ページ (本文とフロートが共存するページ) とフロートだけのページで設定が異なります。ちなみに、カウンタは内部では `\c@` を名前に冠したマクロになっています。

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

[2003-08-23] ちょっと増やしました。

```
1520 \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。

```
1521 \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。

```
1522 \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。

```
1523 \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。

```
1524 \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。

```
1525 \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。

```
1526 \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。

```
1527 \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。

```
1528 \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。

```
1529 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1530 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1531 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1532 \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 1533 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1534 \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`

```

1535 \setlength\@fptop{0\p@? \@plus 1fil}
1536 \setlength\@fpsep{8\p@? \@plus 2fil}
1537 \setlength\@fpbot{0\p@? \@plus 1fil}

```

\@dblftop 段抜きフロートについての値です。

```

\@dblfpsep 1538 \setlength\@dblftop{0\p@? \@plus 1fil}
\@dblfpbot 1539 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1540 \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 pT<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。

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

となっています。

```

1541 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1542 % \ifodd\c@page
1543 % \iftdir
1544 % \hbox{}\thispagestyle{empty}\newpage
1545 % \if@twocolumn\hbox{}\newpage\fi
1546 % \fi
1547 % \else
1548 % \ifydir
1549 % \hbox{}\thispagestyle{empty}\newpage
1550 % \if@twocolumn\hbox{}\newpage\fi
1551 % \fi
1552 % \fi\fi}
1553 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1554 % \ifodd\c@page
1555 % \ifydir
1556 % \hbox{}\thispagestyle{empty}\newpage
1557 % \if@twocolumn\hbox{}\newpage\fi
1558 % \fi
1559 % \else
1560 % \iftdir
1561 % \hbox{}\thispagestyle{empty}\newpage
1562 % \if@twocolumn\hbox{}\newpage\fi
1563 % \fi
1564 % \fi\fi}
1565 \def\pltx@cleartooddpage{\clearpage\if@twoside
1566 \ifodd\c@page\else
1567 \hbox{}\thispagestyle{empty}\newpage

```

```

1568 \if@twocolumn\hbox{}\newpage\fi
1569 \fi\fi}
1570 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1571 \ifodd\c@page
1572 \hbox{}\thispagestyle{empty}\newpage
1573 \if@twocolumn\hbox{}\newpage\fi
1574 \fi\fi}

```

---

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

```

1575 \let\pltx@cleartorightpage\pltx@cleartooddpage
1576 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

---

`\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。

```

1577 %<*book|report>
1578 \if@openleft
1579 \let\cleardoublepage\pltx@cleartoleftpage
1580 \else\if@openright
1581 \let\cleardoublepage\pltx@cleartorightpage
1582 \fi\fi
1583 %</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。

```

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

```

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

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

`\ps@empty` `empty` ページスタイルの定義です。L<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。

```
1584 % \def\ps@empty{%
1585 %   \let\mkboth\gobbletwo
1586 %   \let\@oddhead\empty
1587 %   \let\@oddfoot\empty
1588 %   \let\@evenhead\empty
1589 %   \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。

```
1590 \def\ps@plainfoot{%
1591   \let\mkboth\gobbletwo
1592   \let\@oddhead\empty
1593   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1594   \let\@evenhead\empty
1595   \let\@evenfoot\@oddfoot}
1596 \def\ps@plainhead{%
1597   \let\mkboth\gobbletwo
1598   \let\@oddfoot\empty
1599   \let\@evenfoot\empty
1600   \def\@evenhead{%
1601     \ifmparswitch \hss \fi
1602     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1603     \ifmparswitch\else \hss \fi}%
1604   \def\@oddhead{%
1605     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1606 %<book>\let\ps@plain\ps@plainhead
1607 %<!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。

```
1608 %<*article|slide>
1609 \if@twoside
1610   \def\ps@headings{%
1611     \let\@oddfoot\empty
1612     \let\@evenfoot\empty
1613     \def\@evenhead{\ifmparswitch \hss \fi
1614       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1615     }
```

```

1615     \if@mparswitch\else \hss \fi}%
1616 \def\@oddhead{%
1617     \underline{%
1618         \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1619 \let\@mkboth\markboth
1620 \def\sectionmark##1{\markboth{%
1621     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1622     ##1}\}}%
1623 \def\subsectionmark##1{\markright{%
1624     \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1625     ##1}}%
1626 }
1627 \else % if not twoside
1628 \def\ps@headings{%
1629     \let\@oddfoot\@empty
1630 \def\@oddhead{%
1631     \underline{%
1632         \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1633 \let\@mkboth\markboth
1634 \def\sectionmark##1{\markright{%
1635     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1636     ##1}}%
1637 \fi
1638 %</article|slide>

```

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

---

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

---

```

1639 %<*book|report>
1640 \def\bxjs@maybe@autoxspacing{%
1641     \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1642 \newif\if@omit@number
1643 \def\ps@headings{%
1644     \let\@oddfoot\@empty
1645     \let\@evenfoot\@empty
1646 \def\@evenhead{%
1647     \if@mparswitch \hss \fi
1648     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1649         \textbf{\thepage}\hfil\leftmark}}}%
1650     \if@mparswitch\else \hss \fi}%
1651 \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1652     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1653 \let\@mkboth\markboth
1654 \def\chaptermark##1{\markboth{%
1655     \ifnum \c@secnumdepth >\@m@ne

```



```

1656     \if@mainmatter
1657     \if@omit@number\else
1658     \@chapapp\thechapter\@chappos\hskip1\jsZw
1659     \fi
1660     \fi
1661     \fi
1662     ##1}{}}%
1663 \def\sectionmark##1{\markright{%
1664     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1665     ##1}}}%
1666 %</book|report>

```

最後は学会誌の場合です。

```

1667 %<*jspf>
1668 \def\ps@headings{%
1669     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1670     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1671     \def\@oddhead{\normalfont\hfil \@title \hfil}
1672     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1673 %</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。

```

1674 \def\ps@myheadings{%
1675     \let\@oddfoot\@empty\let\@evenfoot\@empty
1676     \def\@evenhead{%
1677         \if@mparswitch \hss \fi%
1678         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1679         \if@mparswitch\else \hss \fi}%
1680     \def\@oddhead{%
1681         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1682     \let\@mkboth\@gobbletwo
1683 %<book|report> \let\chaptermark\@gobble
1684 \let\sectionmark\@gobble
1685 %<!book&!report> \let\subsectionmark\@gobble
1686 }

```

## 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 1687 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1688 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1689 % \newcommand*{\date}[1]{\gdef\@date{#1}}

```

1690 % \date{\today}

---

\subtitle 副題を設定する。

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

本体を \jsSubtitle として定義する。

1691 \newcommand\*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}

1692 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1693 \renewcommand\*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}

1694 \g@addto@macro\bxjs@begin@document@hook{\bxjs@decl@subtitle}

1695 \def\bxjs@decl@subtitle{%

1696 \global\let\bxjs@decl@subtitle\relax

1697 \ifx\subtitle\@undefined

1698 \global\let\subtitle\jsSubtitle

1699 \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。

1700 \def\bxjs@annihilate@subtitle{%

1701 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi

1702 \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 1703 %<\*\jspf>

\keywords 1704 \newcommand\*{\etitle}[1]{\gdef\@etitle{#1}}

1705 \newcommand\*{\eauthor}[1]{\gdef\@eauthor{#1}}

1706 \newcommand\*{\keywords}[1]{\gdef\@keywords{#1}}

1707 \newcommand\*{\email}[1]{\gdef\authors@mail{#1}}

1708 \newcommand\*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}

1709 %</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。

1710 \def\plainifnotempty{%

1711 \ifx \@oddhead \@empty

1712 \ifx \@oddfoot \@empty

1713 \else

1714 \thispagestyle{plainfoot}%

```

1715     \fi
1716   \else
1717     \thispagestyle{plainhead}%
1718   \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。

```

1719 %<*article|book|report|slide>
1720 \if@titlepage
1721   \newcommand{\maketitle}{%
1722     \begin{titlepage}%
1723       \let\footnotesize\small
1724       \let\footnoterule\relax
1725       \let\footnote\thanks
1726       \null\vfil
1727       \if@slide
1728         {\footnotesize \@date}%
1729         \begin{center}
1730           \mbox{} \[\[1\jsZw]
1731           \large
1732           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1733           \jsc@smallskip
1734           \@title
1735           \ifx\bxjs@subtitle\@undefined\else
1736             \par\vskip\z@
1737             {\small \bxjs@subtitle\par}
1738           \fi
1739           \jsc@smallskip
1740           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1741           \vfill
1742           {\small \@author}%
1743         \end{center}
1744       \else
1745         \vskip 60\p@?
1746         \begin{center}%
1747           {\LARGE \@title \par}%
1748           \ifx\bxjs@subtitle\@undefined\else
1749             \vskip5\p@?
1750             {\normalsize \bxjs@subtitle\par}
1751           \fi
1752           \vskip 3em%
1753           {\large
1754             \lineskip .75em
1755             \begin{tabular}[t]{c}%
1756               \@author

```

```

1757         \end{tabular}\par}%
1758         \vskip 1.5em
1759         {\large \@date \par}%
1760     \end{center}%
1761     \fi
1762     \par
1763     \@thanks\vfil\null
1764 \end{titlepage}%
1765 \setcounter{footnote}{0}%
1766 \global\let\thanks\relax
1767 \global\let\maketitle\relax
1768 \global\let\@thanks\@empty
1769 \global\let\@author\@empty
1770 \global\let\@date\@empty
1771 \global\let\@title\@empty
1772 \global\let\title\relax
1773 \global\let\author\relax
1774 \global\let\date\relax
1775 \global\let\and\relax
1776 \bxjs@annihilate@subtitle
1777 }%
1778 \else
1779 \newcommand{\maketitle}{\par
1780     \begingroup
1781         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1782         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1783         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1784             \parindent 1\jsZw\noindent
1785             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1786         \if@twocolumn
1787             \ifnum \col@number=\@ne
1788                 \@maketitle
1789             \else
1790                 \twocolumn[\@maketitle]%
1791             \fi
1792         \else
1793             \newpage
1794             \global\@topnum\z@ % Prevents figures from going at top of page.
1795             \@maketitle
1796         \fi
1797         \plainifnotempty
1798         \@thanks
1799     \endgroup
1800     \setcounter{footnote}{0}%
1801     \global\let\thanks\relax
1802     \global\let\maketitle\relax
1803     \global\let\@thanks\@empty
1804     \global\let\@author\@empty
1805     \global\let\@date\@empty

```

```

1806 \global\let\@title\@empty
1807 \global\let\title\relax
1808 \global\let\author\relax
1809 \global\let\date\relax
1810 \global\let\and\relax
1811 \bxjs@annihilate@subtitle
1812 }

```

\@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。

```

1813 \def\@maketitle{%
1814 \newpage\null
1815 \vskip 2em
1816 \begin{center}%
1817 \let\footnote\thanks
1818 {\LARGE \@title \par}%
1819 \ifx\bxjs@subtitle\@undefined\else
1820 \vskip3\p@?
1821 {\normalsize \bxjs@subtitle\par}
1822 \fi
1823 \vskip 1.5em
1824 {\large
1825 \lineskip .5em
1826 \begin{tabular}[t]{c}%
1827 \@author
1828 \end{tabular}\par}%
1829 \vskip 1em
1830 {\large \@date}%
1831 \end{center}%
1832 \par\vskip 1.5em
1833 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1834 }
1835 \fi
1836 %</article|book|report|slide>
1837 %<*jspf>
1838 \newcommand{\maketitle}{\par
1839 \begingroup
1840 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1841 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1842 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1843 \parindent 1\jsZw\noindent
1844 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1845 \twocolumn[\@maketitle]%
1846 \plainifnotempty
1847 \@thanks
1848 \endgroup
1849 \setcounter{footnote}{0}%
1850 \global\let\thanks\relax
1851 \global\let\maketitle\relax
1852 \global\let\@thanks\@empty

```

```

1853 \global\let\@author\@empty
1854 \global\let\@date\@empty
1855 % \global\let\@title\@empty % \@title は柱に使う
1856 \global\let\title\relax
1857 \global\let\author\relax
1858 \global\let\date\relax
1859 \global\let\and\relax
1860 \ifx\authors@mail\@undefined\else{%
1861   \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1862   \footnotetext[0]{\itshape\authors@mail}%
1863 } \fi
1864 \global\let\authors@mail\@undefined}
1865 \def\@maketitle{%
1866   \newpage\null
1867   \vskip 6em % used to be 2em
1868   \begin{center}
1869     \let\footnote\thanks
1870     \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1871     \lineskip .5em
1872     \ifx\@author\@undefined\else
1873       \vskip 1em
1874       \begin{tabular}[t]{c}%
1875         \@author
1876       \end{tabular}\par
1877     \fi
1878     \ifx\@etitle\@undefined\else
1879       \vskip 1em
1880       {\large \@etitle \par}%
1881     \fi
1882     \ifx\@eauthor\@undefined\else
1883       \vskip 1em
1884       \begin{tabular}[t]{c}%
1885         \@eauthor
1886       \end{tabular}\par
1887     \fi
1888     \vskip 1em
1889     \@date
1890   \end{center}
1891   \vskip 1.5em
1892   \centerline{\box\@abstractbox}
1893   \ifx\@keywords\@undefined\else
1894     \vskip 1.5em
1895     \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1896   \fi
1897   \vskip 1.5em}
1898 %</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。

```
1899 \def\bxjs@label@sect#1{%
1900   \expandafter\ifx\csname label#1\endcsname\relax
1901     \csname the#1\endcsname
1902   \else \csname label#1\endcsname
1903   \fi}
1904 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}
```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```
1905 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1906 \def\@secapp{\presectionname}
1907 \def\@secpos{\postsectionname}
1908 \fi
```

`\labelsection` 節番号の出力書式。

```
1909 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1910 \def\labelsection{\@secapp\thesection\@secpos}
1911 \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。

```

1912 \def\@startsection#1#2#3#4#5#6{%
1913   \if@noskipsec \leavevmode \fi
1914   \par
1915 % 見出し上の空きを \@tempskipa にセットする
1916   \@tempskipa #4\relax
1917 % \@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
1918   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1919 % 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
1920   \ifdim \@tempskipa <\z@
1921     \@tempskipa -\@tempskipa \@afterindentfalse
1922   \fi
1923   \if@nobreak
1924 %   \everypar{\everyparhook}% これは間違い
1925     \everypar{}%
1926   \else
1927     \addpenalty\@secpenalty
1928 % 次の行は削除
1929 %   \addvspace\@tempskipa
1930 % 次の \noindent まで追加
1931     \ifdim \@tempskipa >\z@
1932       \if@slide\else
1933         \null
1934         \vspace*{-\baselineskip}%
1935       \fi
1936       \vskip\@tempskipa
1937     \fi
1938   \fi
1939   \noindent
1940 % 追加終わり
1941   \@ifstar
1942     {\@ssect{#3}{#4}{#5}{#6}}%
1943     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}
```

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

---

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

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



```

1944 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1945   \let\bxjs@if@ceph\@firstofone
1946 \else \let\bxjs@if@ceph\@gobble
1947 \fi

1948 \def\@sect#1#2#3#4#5#6[#7]#8{%
1949   \ifnum #2>\c@secnumdepth
1950     \let\@svsec\@empty
1951   \else
1952     \refstepcounter{#1}%
1953     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1954   \fi
1955 % 見出し後の空きを \@tempskipa にセット
1956   \@tempskipa #5\relax
1957 % 条件判断の順序を入れ替えました
1958   \ifdim \@tempskipa<\z@
1959     \def\@svsechd{%
1960       #6{\hskip #3\relax
1961         \@svsec #8}%
1962       \csname #1mark\endcsname{#7}%
1963       \addcontentsline{toc}{#1}{%
1964         \ifnum #2>\c@secnumdepth \else
1965           \protect\numberline{\bxjs@label@sect{#1}}%
1966         \fi
1967       #7}}% 目次にフルネームを載せるなら #8
1968   \else
1969     \begingroup
1970     \interlinepenalty \@M % 下から移動
1971     #6{%
1972       \@hangfrom{\hskip #3\relax\@svsec}%
1973 %     \interlinepenalty \@M % 上に移動
1974     #8\@@par}%
1975   \endgroup
1976   \csname #1mark\endcsname{#7}%
1977   \addcontentsline{toc}{#1}{%
1978     \ifnum #2>\c@secnumdepth \else
1979       \protect\numberline{\bxjs@label@sect{#1}}%
1980     \fi
1981     #7}% 目次にフルネームを載せるならここは #8
1982   \fi
1983   \@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。

```

1984 \def\@xsect#1{%
1985 % 見出しの後ろの空きを \@tempskipa にセット

```

```

1986 \@tempskipa #1\relax
1987 % 条件判断の順序を変えました
1988 \ifdim \@tempskipa<\z@
1989 \@nbreakfalse
1990 \global\@noskipsectrue
1991 \everypar{%
1992 \if@noskipsec
1993 \global\@noskipsecfalse
1994 {\setbox\z@\lastbox}%
1995 \clubpenalty\@M
1996 \begingroup \svsechd \endgroup
1997 \unskip
1998 \@tempskipa #1\relax
1999 \hskip -\@tempskipa
2000 \else
2001 \clubpenalty \@clubpenalty
2002 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
2003 \fi\bxjs@if@ceph\everyparhook}%
2004 \else
2005 \par \nbreak
2006 \vskip \@tempskipa
2007 \@afterheading
2008 \fi
2009 \if@slide
2010 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
2011 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
2012 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
2013 \fi
2014 \par % 2000-12-18
2015 \ignorespaces}
2016 \def\@ssect#1#2#3#4#5{%
2017 \@tempskipa #3\relax
2018 \ifdim \@tempskipa<\z@
2019 \def\@svsechd{#4{\hskip #1\relax #5}}%
2020 \else
2021 \begingroup
2022 #4{%
2023 \@hangfrom{\hskip #1}%
2024 \interlinepenalty \@M #5\@@par}%
2025 \endgroup
2026 \fi
2027 \@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 2028 \newcommand*\chaptermark[1]{}
\subsubsectionmark 2029 % \newcommand*\sectionmark[1]{}
\paragraphmark
\subparagraphmark

```

```

2030 % \newcommand*{\subsectionmark}[1]{}
2031 % \newcommand*{\subsubsectionmark}[1]{}
2032 % \newcommand*{\paragraphmark}[1]{}
2033 % \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。

```

2034 %<!book&!report>\setcounter{secnumdepth}{3}
2035 %<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 2036 \newcounter{part}
2037 %<book|report>\newcounter{chapter}
\c@subsubsection 2038 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2039 %<!book&!report>\newcounter{section}
\c@subparagraph 2040 \newcounter{subsection}[section]
2041 \newcounter{subsubsection}[subsection]
2042 \newcounter{paragraph}[subsubsection]
2043 \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。

```

2044 \renewcommand{\thepart}{\@Roman\c@part}
2045 %<*&!book&!report>
2046 \ifnum\bxjs@label@section=\bxjs@label@section@compat
2047 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2048 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2049 \else
2050 \renewcommand{\thesection}{\@arabic\c@section}
2051 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2052 \fi
2053 %</&!book&!report>
2054 %<*&book|report>
2055 \renewcommand{\thechapter}{\@arabic\c@chapter}
2056 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2057 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}

```

```

2058 %</book|report>
2059 \renewcommand{\thesubsubsection}{%
2060   \thesubsection.\@arabic\c@subsubsection}
2061 \renewcommand{\theparagraph}{%
2062   \thesubsubsection.\@arabic\c@paragraph}
2063 \renewcommand{\thesubparagraph}{%
2064   \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。

```

2065 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2066 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

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

```

2067 %<*book|report>
2068 \newcommand\frontmatter{%
2069   \pltx@cleartooddpage
2070   \@mainmatterfalse
2071   \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。

```

2072 \newcommand\mainmatter{%
2073   \pltx@cleartooddpage
2074   \@mainmattertrue
2075   \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。

```

2076 \newcommand\backmatter{%
2077   \if@openleft
2078     \cleardoublepage
2079   \else\if@openright
2080     \cleardoublepage
2081   \else
2082     \clearpage
2083   \fi\fi
2084   \@mainmatterfalse}
2085 %</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。

```
2086 %<!*book&!report>
2087 \newcommand\part{%
2088   \if@noskipsec \leavevmode \fi
2089   \par
2090   \addvspace{4ex}%
2091   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2092   \secdef\@part\@spart}
2093 %</!*book&!report>
```

`book` および `report` クラスの場合は、少し複雑です。

```
2094 %<*book|report>
2095 \newcommand\part{%
2096   \if@openleft
2097     \cleardoublepage
2098   \else\if@openright
2099     \cleardoublepage
2100   \else
2101     \clearpage
2102   \fi\fi
2103   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2104   \if@twocolumn
2105     \onecolumn
2106     \@restonecoltrue
2107   \else
2108     \@restonecolfalse
2109   \fi
2110   \null\vfil
2111   \secdef\@part\@spart}
2112 %</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。

```
2113 %<*!book&!report>
2114 \def\@part[#1]#2{%
2115   \ifnum \c@secnumdepth >\m@ne
2116     \refstepcounter{part}%
2117     \addcontentsline{toc}{part}{%
2118       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2119   \else
2120     \addcontentsline{toc}{part}{#1}%
2121   \fi
2122   \markboth{}{}%
2123   {\parindent\z@
2124     \raggedright
2125     \interlinepenalty \@M
2126     \normalfont
2127     \ifnum \c@secnumdepth >\m@ne
2128       \Large\headfont\prepartname\thepart\postpartname
2129       \par\nobreak
2130     \fi
2131     \huge \headfont #2%
2132     \markboth{}{}\par}%
2133   \nobreak
2134   \vskip 3ex
2135   \@afterheading}
2136 %</!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。

```
2137 %<*book|report>
2138 \def\@part[#1]#2{%
2139   \ifnum \c@secnumdepth >-2\relax
2140     \refstepcounter{part}%
2141     \addcontentsline{toc}{part}{%
2142       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2143   \else
2144     \addcontentsline{toc}{part}{#1}%
2145   \fi
2146   \markboth{}{}%
2147   {\centering
2148     \interlinepenalty \@M
2149     \normalfont
2150     \ifnum \c@secnumdepth >-2\relax
2151       \huge\headfont \prepartname\thepart\postpartname
2152       \par\vskip20\p@?
2153     \fi
2154     \Huge \headfont #2\par}%
2155   \@endpart}
2156 %</book|report>
```

\@spart 番号を付けない部です。

```

2157 %<*!book&!report>
2158 \def\@spart#1{%
2159     \parindent \z@ \raggedright
2160     \interlinepenalty \@M
2161     \normalfont
2162     \huge \headfont #1\par}%
2163 \nobreak
2164 \vskip 3ex
2165 \@afterheading}
2166 %</!book&!report>
2167 %<*book|report>
2168 \def\@spart#1{%
2169     \centering
2170     \interlinepenalty \@M
2171     \normalfont
2172     \Huge \headfont #1\par}%
2173 \@endpart}
2174 %</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。

```

2175 %<*book|report>
2176 \def\@endpart{\vfil\newpage
2177     \if@twoside
2178     \if@openleft %% added (2017/02/24)
2179     \null\thispagestyle{empty}\newpage
2180     \else\if@openright %% added (2016/12/13)
2181     \null\thispagestyle{empty}\newpage
2182     \fi\fi %% added (2016/12/13, 2017/02/24)
2183 \fi
2184 \if@restonecol
2185     \twocolumn
2186 \fi}
2187 %</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。

```

2188 %<*book|report>
2189 \newcommand{\chapter}{%
2190     \if@openleft\cleardoublepage\else
2191     \if@openright\cleardoublepage\else\clearpage\fi\fi
2192     \plainifnotempty % 元: \thispagestyle{plain}

```

```

2193 \global\@topnum\z@
2194 \if@english \afterindentfalse \else \afterindenttrue \fi
2195 \secdef
2196     {\@omit@numberfalse\@chapter}%
2197     {\@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。

```

2198 \def\@chapter[#1]#2{%
2199     \ifnum \c@secnumdepth >\m@ne
2200         \if@mainmatter
2201             \refstepcounter{chapter}%
2202             \typeout{\@chapapp\thechapter\@chappos}%
2203             \addcontentsline{toc}{chapter}%
2204                 {\protect\numberline
2205 %             %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2206                 {\@chapapp\thechapter\@chappos}%
2207                 #1}%
2208         \else\addcontentsline{toc}{chapter}{#1}\fi
2209     \else
2210         \addcontentsline{toc}{chapter}{#1}%
2211     \fi
2212     \chaptermark{#1}%
2213     \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
2214     \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
2215     \if@twocolumn
2216         \@topnewpage[\@makechapterhead{#2}]%
2217     \else
2218         \@makechapterhead{#2}%
2219         \@afterheading
2220     \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。

```

2221 \def\@makechapterhead#1{%
2222     \vspace*{2\Cvs}% 欧文は 50pt
2223     {\parindent \z@ \raggedright \normalfont
2224         \ifnum \c@secnumdepth >\m@ne
2225             \if@mainmatter
2226                 \huge\headfont \@chapapp\thechapter\@chappos
2227                 \par\nobreak
2228                 \vskip \Cvs % 欧文は 20pt
2229             \fi
2230         \fi
2231         \interlinepenalty\@M
2232         \Huge \headfont #1\par\nobreak
2233         \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。

```

2234 \def\@schapter#1{%

```



```

2235 \chaptermark{#1}%
2236 \if@twocolumn
2237 \topnewpage[\@makeschapterhead{#1}]%
2238 \else
2239 \@makeschapterhead{#1}\@afterheading
2240 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2241 \def\@makeschapterhead#1{%
2242 \vspace*{2\Cvs}% 欧文は 50pt
2243 {\parindent \z@ \raggedright
2244 \normalfont
2245 \interlinepenalty\@M
2246 \Huge \headfont #1\par\nobreak
2247 \vskip 3\Cvs}} % 欧文は 40pt
2248 %</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。

```

2249 \if@twocolumn
2250 \newcommand{\section}{%
2251 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2252 \@startsection{section}{1}{\z@}%
2253 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2254 %<kiyou> {\Cvs}{0.5\Cvs}%
2255 % {\normalfont\large\headfont\@secapp}}
2256 {\normalfont\large\headfont\raggedright}}
2257 \else
2258 \newcommand{\section}{%
2259 \if@slide\clearpage\fi
2260 \@startsection{section}{1}{\z@}%
2261 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2262 {.5\Cvs \@plus.3\Cdp}% 後アキ
2263 % {\normalfont\Large\headfont\@secapp}}
2264 {\normalfont\Large\headfont\raggedright}}
2265 \fi

```

`\subsection` 同上です。

```

2266 \if@twocolumn
2267 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2268 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2269 {\normalfont\normalsize\headfont}}
2270 \else
2271 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2272 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ

```

```

2273     {.5\Cvs \@plus.3\Cdp}% 後アキ
2274     {\normalfont\large\headfont}}
2275 \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)。

```

2276 \if@twocolumn
2277   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2278     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2279     {\normalfont\normalsize\headfont}}
2280 \else
2281   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2282     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2283     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2284     {\normalfont\normalsize\headfont}}
2285 \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。

---

```

2286 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2287 \let\bxjs@org@paragraph@mark\jsParagraphMark
2288 \ifx\bxjs@paragraph@mark\@empty
2289   \let\jsParagraphMark\@empty
2290 \else\ifx\bxjs@paragraph@mark\@undefined\else
2291   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2292 \fi\fi
2293 \let\jsJaChar\@empty
2294 \if@twocolumn
2295   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2296     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2297     %<jspf> {\normalfont\normalsize\headfont}}
2298     %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2299 \else
2300   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%

```

```

2301      {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2302      {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2303 %<jspf>      {\normalfont\normalsize\headfont}}
2304 %<!jspf>      {\normalfont\normalsize\headfont\jsParagraphMark}}
2305 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2306 \if@twocolumn
2307   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2308     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2309     {\normalfont\normalsize\headfont}}
2310 \else
2311   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2312     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2313     {\normalfont\normalsize\headfont}}
2314 \fi

```

### 8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2315 \if@slide
2316   \setlength\leftmargini{1\jsZw}
2317 \else
2318   \if@twocolumn
2319     \setlength\leftmargini{2\jsZw}
2320   \else
2321     \setlength\leftmargini{3\jsZw}
2322   \fi
2323 \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 2324 \if@slide
\leftmarginv 2325   \setlength\leftmarginii {1\jsZw}
2326   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2327   \setlength\leftmarginiv {1\jsZw}
2328   \setlength\leftmarginv  {1\jsZw}
2329   \setlength\leftmarginvi {1\jsZw}
2330 \else
2331   \setlength\leftmarginii {2\jsZw}
2332   \setlength\leftmarginiii{2\jsZw}

```

```

2333 \setlength\leftmarginiv {2\jsZw}
2334 \setlength\leftmarginv {1\jsZw}
2335 \setlength\leftmarginvi {1\jsZw}
2336 \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。

```

2337 \setlength \labelsep {0.5\jsZw} % .5em
2338 \setlength \labelwidth{\leftmargini}
2339 \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。

```

2340 \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 2341 \@beginparpenalty -\@lowpenalty
\@itempenalty 2342 \@endparpenalty -\@lowpenalty
2343 \@itempenalty -\@lowpenalty

```

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

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

```

2344 \def\@listi{\leftmargin\leftmargini
2345 \parsep \z@
2346 \topsep 0.5\baselineskip
2347 \itemsep \z@ \relax}
2348 \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）。

```

2349 \@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 2350 \def\@listii{\leftmargin\leftmarginii
\@listiv 2351 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2352 \topsep \z@
\@listv 2353 \parsep \z@
\@listvi 2354 \itemsep\parsep}
2355 \def\@listiii{\leftmargin\leftmarginiii
2356 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2357 \topsep \z@
2358 \parsep \z@

```

```

2359 \itemsep\parsep}
2360 \def\@listiv {\leftmargin\leftmarginiv
2361             \labelwidth\leftmarginiv
2362             \advance\labelwidth-\labelsep}
2363 \def\@listv {\leftmargin\leftmarginv
2364             \labelwidth\leftmarginv
2365             \advance\labelwidth-\labelsep}
2366 \def\@listvi {\leftmargin\leftmarginvi
2367             \labelwidth\leftmarginvi
2368             \advance\labelwidth-\labelsep}

```

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

`\theenumi` 出力する番号の書式を設定します。これらは L<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。

```

2369 \renewcommand{\theenumi}{\@arabic\c@enumi}
2370 \renewcommand{\theenumii}{\@alph\c@enumii}
2371 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2372 \renewcommand{\theenumiv}{\@Alph\c@enumiv}

```

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

---

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

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

---

```

2373 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2374 \newcommand*\jsInJaParen[1]{%
2375   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2376 \newcommand{\labelenumi}{\theenumi.}
2377 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2378 \newcommand{\labelenumiii}{\theenumiii.}
2379 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 2380 \renewcommand{\p@enumii}{\theenumi}
2381 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2382 \renewcommand{\p@enumiv}{\p@enumiii\theenumiii}

```

## ■itemize 環境

`\labelitemi` itemize 環境の第  $n$  レベルのラベルを作るコマンドです。

```
\labelitemii 2383 \newcommand\labelitemi{\textbullet}
\labelitemiii 2384 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiv 2385 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemv 2386 \newcommand\labelitemiv{\textperiodcentered}
```

## ■description 環境

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

```
2387 \newenvironment{description}{%
2388   \list{}{%
2389     \labelwidth=\leftmargin
2390     \labelsep=1\jsZw
2391     \advance \labelwidth by -\labelsep
2392     \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。

```
2393 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}
```

## ■概要

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

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

---

`bxjsreport` クラスの `abstract` 環境は：

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

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

```
2394 %<*book|report>
2395 \newenvironment{chapterabstract}{%
2396   \begin{list}{}{%
2397     \listparindent=1\jsZw
2398     \itemindent=\listparindent
```

```

2399 \rightmargin=0pt
2400 \leftmargin=5\jsZw\item[]\end{list}\vspace{\baselineskip}}
2401 %</book|report>

```

“普通の” abstract 環境の定義。

```

2402 %<*article|report|slide>
2403 \newbox\@abstractbox
2404 \if@titlepage
2405 \newenvironment{abstract}{%
2406 \titlepage
2407 \null\vfil
2408 \@beginparpenalty\@lowpenalty
2409 \begin{center}%
2410 \headfont \abstractname
2411 \@endparpenalty\@M
2412 \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。

```

2413 \par}%
2414 {\par\vfil\null\endtitlepage}
2415 \else
2416 \newenvironment{abstract}{%
2417 \if@twocolumn
2418 \ifx\maketitle\relax
2419 \section*\abstractname}%
2420 \else
2421 \global\setbox\@abstractbox\hbox\bgroup
2422 \begin{minipage}[b]{\textwidth}
2423 \small\parindent1\jsZw
2424 \begin{center}%
2425 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2426 \end{center}%
2427 \list{}{%
2428 \listparindent\parindent
2429 \itemindent \listparindent
2430 \rightmargin \leftmargin}%
2431 \item\relax
2432 \fi
2433 \else
2434 \small
2435 \begin{center}%
2436 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2437 \end{center}%
2438 \list{}{%
2439 \listparindent\parindent
2440 \itemindent \listparindent
2441 \rightmargin \leftmargin}%
2442 \item\relax
2443 \fi}{\if@twocolumn

```

```

2444 \ifx\maketitle\relax
2445 \else
2446 \endlist\end{minipage}\egroup
2447 \fi
2448 \else
2449 \endlist
2450 \fi}
2451 \fi
2452 %</article|report|slide>
2453 %<*jspf>
2454 \newbox\@abstractbox
2455 \newenvironment{abstract}{%
2456 \global\setbox\@abstractbox\hbox\bgroup
2457 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2458 \small
2459 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2460 {\end{minipage}\egroup}
2461 %</jspf>

```

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

```

2462 %<*book|report>
2463 \ifbxjs@force@chapterabstract
2464 \let\abstract\chapterabstract
2465 \let\endabstract\endchapterabstract
2466 \fi
2467 %</book|report>

```

## ■キーワード

**keywords** キーワードを準備する環境です。実際の出力は `\maketitle` で行われます。

```

2468 %<*jspf>
2469 %\newbox\@keywordsbox
2470 %\newenvironment{keywords}{%
2471 % \global\setbox\@keywordsbox\hbox\bgroup
2472 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2473 % \small\parindent0\jsZw}%
2474 % {\end{minipage}\egroup}
2475 %</jspf>

```

## ■verse 環境

**verse** 詩のための `verse` 環境です。

```

2476 \newenvironment{verse}{%
2477 \let \=\@centercr
2478 \list{}{%
2479 \itemsep \z@

```



```

2480 \itemindent -2\jsZw % 元: -1.5em
2481 \listparindent\itemindent
2482 \rightmargin \z@
2483 \advance\leftmargin 2\jsZw}% 元: 1.5em
2484 \item\relax}{\endlist}

```

#### ■quotation 環境

**quotation** 段落の頭の字下げ量を 1.5em から \parindent に変えました。また、右マージンを 0 にしました。

```

2485 \newenvironment{quotation}{%
2486 \list{}{%
2487 \listparindent\parindent
2488 \itemindent\listparindent
2489 \rightmargin \z@}%
2490 \item\relax}{\endlist}

```

#### ■quote 環境

**quote** quote 環境は、段落がインデントされないことを除き、quotation 環境と同じです。

```

2491 \newenvironment{quote}%
2492 {\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。

```

2493 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2494 \item[\hskip \labelsep{\headfont #1\ #2}]}
2495 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2496 \item[\hskip \labelsep{\headfont #1\ #2 (#3)}]}

```

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

[2017-02-24] コミュニティ版 pL<sup>A</sup>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 pL<sup>A</sup>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。

```

2497 \newenvironment{titlepage}{%

```

```

2498 %<book>      \pltx@cleartooddpage %% 2017-02-24
2499      \if@twocolumn
2500          \@restonecoltrue\onecolumn
2501      \else
2502          \@restonecolfalse\newpage
2503      \fi
2504      \thispagestyle{empty}%
2505      \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
24
2506  }%
2507  {\if@restonecol\twocolumn \else \newpage \fi
2508      \if@twoside\else
2509          \setcounter{page}\@ne
2510      \fi}

```

## ■付録

`\appendix` 本文と付録を分離するコマンドです。

```

2511 %<!*book&!report>
2512 \newcommand{\appendix}{\par
2513     \setcounter{section}{0}%
2514     \setcounter{subsection}{0}%
2515     \ifnum\bxjs@label@section=\bxjs@label@section@compat
2516     \gdef\presectionname{\appendixname}%
2517     \gdef\postsectionname{}}%
2518 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2519 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2520 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2521 \else
2522 \gdef\@secapp{\appendixname}%
2523 \gdef\@secpos{}}%
2524 \gdef\thesection{\@Alph\c@section}%
2525 \fi}
2526 %</!*book&!report>
2527 %<*book|report>
2528 \newcommand{\appendix}{\par
2529     \setcounter{chapter}{0}%
2530     \setcounter{section}{0}%
2531     \gdef\@chapapp{\appendixname}%
2532     \gdef\@chappos{}}%
2533 \gdef\thechapter{\@Alph\c@chapter}}
2534 %</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。

2535 `\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。

2536 `\setlength\tabcolsep{6\p@?}`

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

2537 `\setlength\arrayrulewidth{.4\p@}`

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

2538 `\setlength\doublerulesep{2\p@}`

### ■tabbing 環境

`\tabbingsep` \ ' コマンドで入るアキです。

2539 `\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。

2540 `\skip\@mpfootins = \skip\footins`

### ■framebox 環境

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

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

2541 `\setlength\fboxsep{3\p@?}`

2542 `\setlength\fboxrule{.4\p@}`

### ■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

2543 `%<!book&!report>\renewcommand \theequation {\@arabic\c@equation}`

2544 `%<*book|report>`

2545 `\@addtoreset{equation}{chapter}`

2546 `\renewcommand\theequation`

2547 `{\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}`

2548 `%</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。

2549 `% \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。

2550 `% \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。

```
2551 % \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。

```
2552 %<*!book&!report>
2553 \newcounter{figure}
2554 \renewcommand \thefigure {\@arabic\c@figure}
2555 %</!book&!report>
2556 %<*book|report>
2557 \newcounter{figure}[chapter]
2558 \renewcommand \thefigure
2559     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2560 %</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 2561 \def\fps@figure{tbp}
2562 \def\ftype@figure{1}
\fnum@figure 2563 \def\ext@figure{lof}
2564 \def\fnum@figure{\figurename\nobreak\thefigure}
```

**figure** \* 形式は段抜きのフロートです。

```
figure* 2565 \newenvironment{figure}%
2566     {\@float{figure}}%
2567     {\endfloat}
2568 \newenvironment{figure*}%
2569     {\@dblfloat{figure}}%
2570     {\enddblfloat}
```

### ■table 環境

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

```
2571 %<!*book&!report>
2572 \newcounter{table}
2573 \renewcommand\thetable{\@arabic\c@table}
2574 %</!book&!report>
2575 %<*book|report>
2576 \newcounter{table}[chapter]
2577 \renewcommand \thetable
2578     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2579 %</book|report>
```

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

```
\ext@table 2580 \def\fps@table{tbp}
2581 \def\ftype@table{2}
\fnun@table 2582 \def\ext@table{lot}
2583 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` \* は段抜きのフロートです。

```
table* 2584 \newenvironment{table}%
2585     {\@float{table}}%
2586     {\end@float}
2587 \newenvironment{table*}%
2588     {\dblfloat{table}}%
2589     {\enddblfloat}
```

## 8.6 キャプション

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

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

```
2590 \newlength\abovecaptionskip
2591 \newlength\belowcaptionskip
2592 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2593 \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。

```
2594 %<!*jspf>
2595 % \long\def\@makecaption#1#2{{\small
2596 %     \advance\leftskip10\jsc@mmm
```

```

2597 % \advance\rightskip10\jsc@mmm
2598 % \vskip\abovcaptionskip
2599 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2600 % \ifdim \wd\@tempboxa >\hsize
2601 % #1\hskip1\jsZw\relax #2\par
2602 % \else
2603 % \global \@minipagefalse
2604 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2605 % \fi
2606 % \vskip\belowcaptionskip}}
2607 \long\def\@makecaption#1#2{\small
2608 \advance\leftskip .0628\linewidth
2609 \advance\rightskip .0628\linewidth
2610 \vskip\abovcaptionskip
2611 \sbox\@tempboxa{#1\zwspace#2}%
2612 \ifdim \wd\@tempboxa <\hsize \centering \fi
2613 #1\zwspace#2\par
2614 \vskip\belowcaptionskip}}
2615 %</!jspf>
2616 %<*jspf>
2617 \long\def\@makecaption#1#2{%
2618 \vskip\abovcaptionskip
2619 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2620 \ifdim \wd\@tempboxa >\hsize
2621 {\small\sffamily
2622 \list{#1}{%
2623 \renewcommand{\makelabel}[1]{##1\hfil}
2624 \itemsep \z@
2625 \itemindent \z@
2626 \labelsep \z@
2627 \labelwidth 11\jsc@mmm
2628 \listparindent\z@
2629 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2630 \else
2631 \global \@minipagefalse
2632 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2633 \fi
2634 \vskip\belowcaptionskip}
2635 %</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。
```

---

```
2636 \newif\if@jsc@warnoldfontcmd
2637 \@jsc@warnoldfontcmdtrue
2638 \newif\if@jsc@warnoldfontcmdexception
2639 \@jsc@warnoldfontcmdexceptionfalse
```

`\jsc@DeclareOldFontCommand`

```
2640 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2641   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2642   \DeclareOldFontCommand{#1}{%
2643     \bxjs@oldfontcmd{#1}#2%
2644   }{%
2645     \bxjs@oldfontcmd{#1}#3%
2646   }%
2647 }
2648 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2649   \ClassInfo\bxjs@clsname
2650   {Old font command '\string#1' is used!!\MessageBreak
2651     The first occurrence is}%
2652 }
```

---

`\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。

```
2653 \newcommand*{\allowoldfontcommands}{%
2654   \@jsc@warnoldfontcmdexceptiontrue}
2655 \newcommand*{\disallowoldfontcommands}{%
2656   \@jsc@warnoldfontcmdexceptionfalse}

2657 \let\bxjs@oldfontcmd@list\@empty
2658 \def\bxjs@oldfontcmd#1{%
2659   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2660 \def\bxjs@oldfontcmd@a#1#2{%
2661   \if@jsc@warnoldfontcmdexception\else
2662     \global\@jsc@warnoldfontcmdfalse
2663     \ifx#1\relax
2664       \global\let#1=t%
2665       \jsc@warnoldfontcmd{#2}%
2666     \fi
2667   \fi}
```

```

2668 \def\bxjs@warnoldfontcmd@final{%
2669 % \par
2670 \global\let\bxjs@warnoldfontcmd@final\@empty
2671 \let\@tempa\@empty
2672 \def\do##1{%
2673   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2674     \edef\@tempa{\@tempa \space\string##1}\fi}
2675 \bxjs@oldfontcmd@list
2676 \ifx\@tempa\@empty\else
2677   \ClassWarningNoLine\bxjs@clsname
2678     {Some old font commands were used in text:\MessageBreak
2679     \space\@tempa\MessageBreak
2680     You should note, that since 1994 LaTeX2e provides a\MessageBreak
2681     new font selection scheme called NFSS2 with several\MessageBreak
2682     new, combinable font commands. The
2683     class provides\MessageBreak
2684     the old font commands only for compatibility}
2685 \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。

```

2686 \def\bxjs@warnoldfontcmd@kick@final{%
2687   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2688 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2689 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2690 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 2691 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\tt 2692 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2693 \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。

```

2694 \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。

```

2695 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2696 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2697 \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`



```
2698 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2699 \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。

```
2700 \newcommand\@pnumwidth{1.55em}
2701 \newcommand\@tocrmarg{2.55em}
2702 \newcommand\@dotsep{4.5}
2703 %<!book&!report>\setcounter{tocdepth}{2}
2704 %<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)

```

2705 \newdimen\jsc@tocl@width
2706 \newcommand{\tableofcontents}{%
2707 %<*book|report>
2708   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2709   \settowidth\@tempdima{\headfont\appendixname}%
2710   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2711   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2712   \if@twocolumn
2713     \@restonecoltrue\onecolumn
2714   \else
2715     \@restonecolfalse
2716   \fi
2717   \chapter*{\contentsname}%
2718   \@mkboth{\contentsname}{}%
2719 %</book|report>
2720 %<*&book&!report>
2721   \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2722   \settowidth\@tempdima{\headfont\appendixname}%
2723   \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2724   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2725   \section*{\contentsname}%
2726   \@mkboth{\contentsname}{\contentsname}%
2727 %</&book&!report>
2728   \@starttoc{toc}%
2729 %<book|report> \if@restonecol\twocolumn\fi
2730 }
```

`\l@part` 部の目次です。

```

2731 \newcommand*{\l@part}[2]{%
2732   \ifnum \c@tocdepth >-2\relax
2733 %<!book&!report> \addpenalty\@secpenalty
2734 %<book|report> \addpenalty{-\@highpenalty}%
2735   \addvspace{2.25em \@plus\p@}%
2736   \begingroup
2737     \parindent \z@
2738 % \@pnumwidth should be \@tocrmarg
2739 % \rightskip \@pnumwidth
2740 \rightskip \@tocrmarg
2741 \parfillskip -\rightskip
2742 {\leavevmode
2743   \large \headfont
2744   \setlength\@lnumwidth{4\jsZw}%
2745   #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2746   \nobreak
```

```

2747 %<book|report>      \global\@nbreaktrue
2748 %<book|report>      \everypar{\global\@nbreakfalse\everypar{}}%
2749      \endgroup
2750      \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)

```

2751 %<*book|report>
2752 \newcommand*{\l@chapter}[2]{%
2753   \ifnum \c@tocdepth >\m@ne
2754     \addpenalty{-\@highpenalty}%
2755     \addvspace{1.0em \@plus\p@?}
2756 %   \vskip 1.0em \@plus\p@    % book.cls では↑がこうなっている
2757   \begingroup
2758     \parindent\z@
2759 %   \rightskip\@pnumwidth
2760   \rightskip\@tocrmarg
2761   \parfillskip-\rightskip
2762   \leavevmode\headfont
2763 %   % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2764   \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2765   \advance\leftskip\@lnumwidth \hskip-\leftskip
2766   #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2767   \penalty\@highpenalty
2768   \endgroup
2769   \fi}
2770 %</book|report>

```

\l@section 節の目次です。

```

2771 %<*!book&!report>
2772 \newcommand*{\l@section}[2]{%
2773   \ifnum \c@tocdepth >\z@
2774     \addpenalty{\@secpenalty}%
2775     \addvspace{1.0em \@plus\p@?}%
2776   \begingroup
2777     \parindent\z@
2778 %   \rightskip\@pnumwidth
2779   \rightskip\@tocrmarg
2780   \parfillskip-\rightskip
2781   \leavevmode\headfont
2782 %   % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2783   \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2784   \advance\leftskip\@lnumwidth \hskip-\leftskip
2785   #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2786   \endgroup
2787   \fi}
2788 %</!book&!report>

```

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

```
2789 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

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

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

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

```
\l@subparagraph 2790 %<!*book&!report>
2791 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2792 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2793 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2794 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2795 %
2796 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2797 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2798 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2799 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2800 %
2801 \newcommand*{\l@subsection}{%
2802     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2803     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2804 \newcommand*{\l@subsubsection}{%
2805     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2806     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2807 \newcommand*{\l@paragraph}{%
2808     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2809     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2810 \newcommand*{\l@subparagraph}{%
2811     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2812     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2813 %</!*book&!report>
2814 %<*book|report>
2815 % \newcommand*{\l@subsection} {\@dottedtocline{2}{3.8em}{3.2em}}
2816 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2817 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2818 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2819 \newcommand*{\l@section}{%
2820     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2821     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2822 \newcommand*{\l@subsection}{%
2823     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2824     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2825 \newcommand*{\l@subsubsection}{%
2826     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2827     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2828 \newcommand*{\l@paragraph}{%
2829     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2830     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
```

```

2831 \newcommand*{\l@subparagraph}{%
2832     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2833     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2834 %</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。

```

2835 \newdimen\@lnumwidth
2836 \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` に変えています。

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

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

とします。

```

2837 \def\jsTocLine{\leaders\hbox{%
2838     $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2839 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2840     \vskip \z@ \@plus.2\p@?
2841     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2842       \parindent #2\relax\@afterindenttrue
2843       \interlinepenalty\@M
2844       \leavevmode
2845       \@lnumwidth #3\relax
2846       \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2847       {#4}\nobreak
2848       \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2849         \hfil\normalfont \normalcolor #5}\par}\fi}

```

## ■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2850 \newcommand{\listoffigures}{%
2851 %<*book|report>
2852     \if@twocolumn\@restonecoltrue\onecolumn
2853     \else\@restonecolfalse\fi
2854     \chapter*{\listfigurename}%
2855     \@mkboth{\listfigurename}{}%
2856 %</book|report>
2857 %<!*book&!report>
2858     \section*{\listfigurename}%
2859     \@mkboth{\listfigurename}{\listfigurename}%

```

```

2860 %</!book&!report>
2861 \starttoc{lof}%
2862 %<book|report> \if@restonecol\twocolumn\fi
2863 }

```

`\l@figure` 図目次の項目を出力します。

```

2864 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2865 \newcommand{\listoftables}{%
2866 %<*book|report>
2867 \if@twocolumn\@restonecoltrue\onecolumn
2868 \else\@restonecolfalse\fi
2869 \chapter*{\listtablename}%
2870 \@mkboth{\listtablename}{}%
2871 %</book|report>
2872 %<*!book&!report>
2873 \section*{\listtablename}%
2874 \@mkboth{\listtablename}{\listtablename}%
2875 %</!book&!report>
2876 \starttoc{lot}%
2877 %<book|report> \if@restonecol\twocolumn\fi
2878 }

```

`\l@table` 表目次は図目次と同じです。

```

2879 \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。

```

2880 \newdimen\bibindent
2881 \setlength\bibindent{2\jsZw}

```

`thebibliography` 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。

```

2882 \newenvironment{thebibliography}[1]{%
2883 \@jsc@warnoldfontcmdexceptiontrue
2884 \global\let\presectionname\relax
2885 \global\let\postsectionname\relax
2886 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2887 %<*kiyou>
2888 \vspace{1.5\baselineskip}
2889 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2890 \vspace{0.5\baselineskip}
2891 %</kiyou>

```

```

2892 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2893 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2894 \list{\@biblabel{\@arabic{c@enumiv}}}%
2895 {\settowidth\labelwidth{\@biblabel{#1}}%
2896 \leftmargin\labelwidth
2897 \advance\leftmargin\labelsep
2898 \@openbib@code
2899 \usecounter{enumiv}%
2900 \let\p@enumiv\@empty
2901 \renewcommand\theenumiv{\@arabic{c@enumiv}}%
2902 %<kiyou> \small
2903 \sloppy
2904 \clubpenalty4000
2905 \@clubpenalty\clubpenalty
2906 \widowpenalty4000%
2907 \sfcode`.\@m}
2908 {\def\@noitemerr
2909 {\@latex@warning{Empty 'thebibliography' environment}}}%
2910 \endlist}

```

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

```

2911 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

```

`\@openbib@code` `\@openbib@code` はデフォルトでは何もありません。この定義は `openbib` オプションによって変更されます。

```

2912 \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。

```

2913 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

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

```

2914 % \def\@citex[#1]#2{\leavevmode
2915 % \let\@citea\@empty
2916 % \@cite{\for\@citeb:=#2\do
2917 % {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2918 % \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2919 % \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2920 % \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2921 % \G@refundefinedtrue
2922 % \@latex@warning
2923 % {Citation ``\@citeb' on page \thepage \space undefined}}%
2924 % {\@citeofmt{\csname b@\@citeb\endcsname}}}{#1}}
2925 % \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。

```
2926 % \DeclareRobustCommand\cite{\unskip
2927 %   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]
2928 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2929 %   , \jsInhibitGlue\ #2\fi}) }}$}
```

### 10.3 索引

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

```
2930 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2931   \if@twocolumn
2932     \onecolumn\@restonecolfalse
2933   \else
2934     \clearpage\@restonecoltrue
2935   \fi
2936   \columnseprule.4pt \columnsep 2\jsZw
2937   \ifx\multicols\@undefined
2938 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}]%
2939 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2940 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2941 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2942   \else
2943     \ifdim\textwidth<\fullwidth
2944       \setlength{\evensidemargin}{\oddsidemargin}
2945       \setlength{\textwidth}{\fullwidth}
2946       \setlength{\linewidth}{\fullwidth}
2947 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}]%
2948 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2949 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2950 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2951   \else
2952 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}]%
2953 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2954 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2955 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2956   \fi
2957   \fi
2958 %<book|report>      \@mkboth{\indexname}{}%
2959 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2960   \plainifnotempty % \thispagestyle{plain}
2961   \parindent\z@
2962   \parskip\z@ \@plus .3\p@?\relax
2963   \let\item\@idxitem
2964   \raggedright
2965   \footnotesize\narrowbaselines
```



```

2966 }{
2967   \ifx\multicols\@undefined
2968     \if@restonecol\onecolumn\fi
2969   \else
2970     \end{multicols}
2971   \fi
2972   \clearpage
2973 }

```

`\@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 2974 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2975 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2976 \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。

```

2977 \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。

```

2978 \newcommand\seename{\if@english see\else \rightarrow\fi}
2979 \newcommand\alsoname{\if@english see also\else \rightarrow\fi}

```

## 10.4 脚注

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

---

```

2980 \ifx\pltx@foot@penalty\@undefined
2981   \let\footnotes@ve=\footnote
2982   \def\footnote{\jsInhibitGlue\footnotes@ve}
2983   \let\footnotemarks@ve=\footnotemark
2984   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2985 \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 pL<sup>A</sup>T<sub>E</sub>X の変更に追随しました (Thanks: 角藤さん)。pL<sup>A</sup>T<sub>E</sub>X の日付が 2016/04/17 より新しい場合は、このパッチが不要なのであてません。

---

pT<sub>E</sub>X 依存のコードなので、minimal 和文ドライバ実装に移動。

---

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

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

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

```
2986 \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。

```
2987 % \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。

```
2988 \renewcommand{\footnoterule}{%
2989   \kern-2.6\p@? \kern-.4\p@
2990   \hrule width .4\columnwidth
2991   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2992 %<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<sub>E</sub>X and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9)

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

[2016-09-08] コミュニティ版 pL<sup>A</sup>T<sub>E</sub>X のバグ修正に追随しました。

[2016-11-29] 古い pL<sup>A</sup>T<sub>E</sub>X で使用された場合を考慮してコードを改良。

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

```
2993 \long\def\@footnotetext{%
2994   \insert\footins\bgroup
2995     \normalfont\footnotesize
2996     \interlinepenalty\interfootnotelinepenalty
2997     \splittopskip\footnotesep
2998     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2999     \hsize\columnwidth \@parboxrestore
3000     \protected@edef\@currentlabel{%
```

```

3001      \csname p@footnote\endcsname\@thefnmark
3002    }%
3003    \color@begingroup
3004      \@makefnmark{%
3005        \rule{z@\footnotesep\ignorespaces}%
3006        \futurelet\jsc@next\jsc@fo@t}
3007 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@f@t
3008               \else \let\jsc@next\jsc@f@t\fi \jsc@next}
3009 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3010 \def\jsc@f@t#1{#1\jsc@@foot}
3011 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
3012   \ifx\pltx@foot@penalty\@undefined\else
3013     \ifhmode\null\fi
3014     \ifnum\pltx@foot@penalty=z@\else
3015       \penalty\pltx@foot@penalty
3016       \pltx@foot@penaltyz@
3017     \fi
3018   \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。

```

3019 \newcommand\@makefnmark[1]{%
3020   \advance\leftskip 3\jsZw
3021   \parindent 1\jsZw
3022   \noindent
3023   \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。

```

3024 % \def\@xfootnotenext[#1]{%
3025 %   \begingroup
3026 %     \ifnum#1>z@
3027 %       \csname c@\@mpfn\endcsname #1\relax
3028 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
3029 %     \else
3030 %       \unrestored@protected@xdef\@thefnmark{}%
3031 %     \fi
3032 %   \endgroup
3033 %   \@footnotetext}

```

---

ここまでのコードは JS クラスを踏襲する。

---

## 11 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

---

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

```
3034 \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。

```
3035 \def\everyparhook{\jsInhibitGlueAtParTop}
3036 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
3037 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3038 \fi
```

---

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

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

[2017-02-13] `\jsc@tempa` は実はテンポラリではなく「この処理専用のユニーク制御綴」である必要があります。間違っ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。

---

```
3039 \ifx j\jsEngine
3040 \def\@inhibitglue{%
3041   \futurelet\@let@token\@@inhibitglue}
3042 \begingroup
3043 \let\GDEF=\gdef
3044 \let\CATCODE=\catcode
3045 \let\ENDGROUP=\endgroup
3046 \CATCODE`k=12
3047 \CATCODE`a=12
3048 \CATCODE`n=12
3049 \CATCODE`j=12
3050 \CATCODE`i=12
3051 \CATCODE`c=12
3052 \CATCODE`h=12
3053 \CATCODE`r=12
3054 \CATCODE`t=12
```

```

3055 \CATCODE`e=12
3056 \GDEF\KANJI@CHARACTER{kanji character }
3057 \ENDGROUP
3058 \def\@@inhibitglue{%
3059   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
3060 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3061   \def\jsc@ig@temp{#1}%
3062   \ifx\jsc@ig@temp\@empty
3063     \ifnum\the\inhibitxspcode`#2=2\relax
3064       \inhibitglue
3065     \fi
3066   \fi}
3067 \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。

---

```

3068 \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。

```

3069 \def\@doendpe{%
3070   \@endpetrue
3071   \def\par{%
3072     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3073   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3074 \def\@setminipage{%
3075   \@minipagetrue
3076   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3077 }

```

`\item` 命令の直後です。

```

3078 \def\@item[#1]{%
3079   \if@noparitem
3080     \@donoparitem
3081   \else
3082     \if@inlabel
3083       \indent \par
3084     \fi
3085     \ifhmode
3086       \unskip\unskip \par
3087     \fi
3088     \if@newlist

```

```

3089     \if@nobreak
3090         \@nbitem
3091     \else
3092         \addpenalty\@beginparpenalty
3093         \addvspace\@topsep
3094         \addvspace{-\parskip}%
3095     \fi
3096 \else
3097     \addpenalty\@itempenalty
3098     \addvspace\itemsep
3099 \fi
3100 \global\@inlabeltrue
3101 \fi
3102 \everypar{%
3103     \@minipagefalse
3104     \global\@newlistfalse
3105     \if@inlabel
3106         \global\@inlabelfalse
3107         {\setbox\z@\lastbox
3108         \ifvoid\z@
3109             \kern-\itemindent
3110         \fi}%
3111     \box\@labels
3112     \penalty\z@
3113 \fi
3114 \if@nobreak
3115     \@nobreakfalse
3116     \clubpenalty \@M
3117 \else
3118     \clubpenalty \@clubpenalty
3119     \everypar{\everyparhook}%
3120 \fi
3121 \everyparhook}%
3122 \if@noitemarg
3123     \@noitemargfalse
3124     \if@nmbrlist
3125         \refstepcounter\@listctr
3126     \fi
3127 \fi
3128 \sbox\@tempboxa{\makelabel{#1}}%
3129 \global\setbox\@labels\hbox{%
3130     \unhbox\@labels
3131     \hskip \itemindent
3132     \hskip -\labelwidth
3133     \hskip -\labelsep
3134     \ifdim \wd\@tempboxa >\labelwidth
3135         \box\@tempboxa
3136     \else
3137         \hbox to\labelwidth {\unhbox\@tempboxa}%

```

```

3138 \fi
3139 \hskip \labelsep}%
3140 \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。

```

3141 \def\@afterheading{%
3142 \@nbreaktrue
3143 \everypar{%
3144 \if@nbreak
3145 \@nbreakfalse
3146 \clubpenalty \@M
3147 \if@afterindent \else
3148 {\setbox\z@\lastbox}%
3149 \fi
3150 \else
3151 \clubpenalty \@clubpenalty
3152 \everypar{\everyparhook}%
3153 \fi\everyparhook}}

```

---

「`\everyparhook` 用のパッチ処理」はここまで。

---

```

3154 \fi

```

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

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

---

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

---

```

3155 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3156 \AtEndOfPackage{%
3157 \def\@gnewline #1{%
3158 \ifvmode
3159 \@nolnerr
3160 \else
3161 \unskip \reserved@e {\reserved@f#1}\nbreak \hfil \break \null
3162 \jsInhibitGlue \ignorespaces
3163 \fi}
3164 }
3165 \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。

---

```
3166 \if@jslogo
3167   \IfFileExists{jslogo.sty}{%
3168     \RequirePackage{jslogo}%
3169   }{%
3170     \ClassWarningNoLine{bxjs@clsname}
3171       {The package 'jslogo' is not installed.\MessageBreak
3172        It is included in the recent release of\MessageBreak
3173        the 'jsclasses' bundle}
3174   }
3175 \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` 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: しっぱ愛好家さん)。

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

```
3176 %\let\ltx@ifnextchar\@ifnextchar
3177 %\def\ProvidesFile#1{%
3178 %  \begingroup
3179 %    \catcode`\ 10 %
3180 %    \ifnum \endlinechar<256 %
3181 %      \ifnum \endlinechar>\m@ne
3182 %        \catcode\endlinechar 10 %
3183 %      \fi
3184 %    \fi
3185 %    \@makeother\/%
```



```

3186 % \makeother\&%
3187 % \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

## 14 初期設定

### ■いろいろな語

```

\prepartname
\postpartname 3188 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3189 \newcommand{\postpartname}{\if@english\else 部\fi}
3190 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3191 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3192 \newcommand{\presectionname}{}% 第
\postsectionname 3193 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3194 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3195 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3196 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3197 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3198 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3199 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3200 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3201 %<jspf>\newcommand{\figurename}{Fig.~}
3202 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3203 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3204 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3205 \newcommand{\appendixname}{\if@english \else 付録\fi}
3206 %<book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

---

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

※ 2.0 版より、完全に `bxwareki` に任せる。

---

```

3207 \@onlypreamble\bxjs@decl@Seireki@cmds

```

```

3208 \@tempswafalse
3209 \if p\jsEngine \@tempwattrue \fi
3210 \if n\jsEngine \@tempwattrue \fi
3211 \bxjs@cond\if@tempswa\fi{%
3212 % 欧文 8bitTeX の場合
3213 \newif\ifjsSeireki \jsSeirekitrue
3214 \def\bxjs@decl@Seireki@cmds{%
3215   \def\西暦{\jsSeirekitrue}%
3216   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3217 \def\Seireki{\jsSeirekitrue}
3218 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3219 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3220 \def\bxjs@iai{\noexpand~}
3221 }{%
3222 \newif\if 西暦 \西暦 true
3223 \def\bxjs@decl@Seireki@cmds{%
3224   \def\西暦{\西暦 true}%
3225   \def\和暦{\西暦 false\bxjs@wareki@used}}
3226 \def\Seireki{\西暦 true}
3227 \def\Wareki{\西暦 false\bxjs@wareki@used}
3228 \def\bxjs@if@use@seireki{\bxjs@cond\if 西暦\fi}
3229 \let\bxjs@iai@empty
3230 }
3231 \bxjs@decl@Seireki@cmds
3232 \let\bxjs@unxp@firstofone \let\bxjs@onxp@firstofone
3233 \bxjs@test@engine\unexpanded{%
3234   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}

```

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

```

3235 \newif\ifbxjs@bxwareki@avail
3236 \IfFileExists{bxwareki.sty}{%
3237   \RequirePackage{bxwareki}[]%
3238   \bxjs@bxwareki@availtrue}{%

```

`\bxjs@wareki@used` 和暦が非対応の場合に警告を出す。

```

3239 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\empty
3240 \else
3241   \bxjs@robust@def\bxjs@wareki@used{%
3242     \global\let\bxjs@wareki@used\empty
3243     \ClassWarning\bxjs@clsname
3244       {Wareki mode is not supported, since\MessageBreak
3245         'bxwareki' is unavailable, found}}
3246   \g@addto@macro\bxjs@begin@document@hook{%
3247     \let\bxjs@wareki@used\empty}
3248 \fi

```

`\jayear` 和暦における年の表記（元号 + 年数）。

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

```

3249 \ifbxjs@bxwareki@avail
3250   \let\jyear\warekiyear
3251   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3252     \newcount\heisei \heisei=\value{warekiyear}
3253   \fi

    bxwareki が使えない場合は西暦表示にする。

3254 \else
3255   \edef\jyear{\the\year \bxjs@iai}
3256 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3257 \let\bxjs@next\relax
3258 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3259   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3260 \fi\fi
3261 \edef\bxjs@today{%
3262   \if@english
3263     \ifcase\month\or
3264       January\or February\or March\or April\or May\or June\or
3265       July\or August\or September\or October\or November\or December\fi
3266     \space\number\day, \number\year
3267   \else
3268     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3269     \else \noexpand\bxjs@if@use@seireki
3270   \fi {%
3271     \number\year\bxjs@iai\bxjs@unxp{年}%
3272     \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3273     \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3274   }\bxjs@next}%
3275 \fi}
3276 \let\today\bxjs@today

```

---

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

```

3277 \g@addto@macro\bxjs@begin@document@hook{%
3278   \ifx\bb1@jpn@maybekansuji\@undefined\else
3279     \bxjs@decl@Seireki@cmds
3280     \g@addto@macro\datejapanese{%
3281       \let\today\bxjs@today}%
3282   \fi}

```

---

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


```

3283 \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。

```
3284 %<slide>\pagestyle{empty}%
3285 %<article|report>\pagestyle{plain}%
3286 %<book>\pagestyle{headings}%
3287 \pagenumbering{arabic}
3288 \if@twocolumn
3289   \twocolumn
3290   \sloppy
3291   \flushbottom
3292 \else
3293   \onecolumn
3294   \raggedbottom
3295 \fi
3296 %<*slide>
3297 \renewcommand\familydefault{\sfdefault}
3298 \raggedright
3299 %</slide>
```

■ BXJS 独自の追加処理 

---

フックを実行する。

```
3300 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3301 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

```
3302 %</cls>
```

---

以上です。

## 付録 A 和文ドライバの仕様

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

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

```
3303 %<*drv>
```

## 付録 B 和文ドライバ : minimal

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

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

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

### B.1 補助マクロ

```
3304 %<*minimal>
```

```
3305 %% 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.
```

```
3306 \def\DeclareJaTextFontCommand#1#2{%
```

```
3307   \DeclareRobustCommand#1[1]{%
```

```
3308     \relax
```

```
3309     \ifmmode \expandafter\nfss@text \fi
```

```
3310     {#2##1}}%
```

```
3311 }
```

```
\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。
```

```
3312 \def\DeclareJaMathFontCommand#1#2{%
```

```
3313   \DeclareRobustCommand#1[1]{%
```

```
3314     \relax
```

```
3315     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3316     \nfss@text{\fontfamily\familydefault
```

```
3317         \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3318         #2##1}}%
```

```

3319 }%
3320 }

```

`\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。

```

3321 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3322 \@onlypreamble\bxjs@if@sf@default
3323 \def\bxjs@if@sf@default#1{%
3324   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3325   \g@addto@macro\bxjs@begin@document@hook{%
3326     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3327 }

```

`\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。

```

3328 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3329 \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。

```

3330 \def\jsLetHeadChar#1#2{%
3331   \begingroup
3332     \escapechar=`\ %
3333     \let\bxjs@tmpa={% brace-match-hack
3334     \bxjs@let@hchar@exp#2}%
3335   \endgroup
3336   \let#1\bxjs@g@tmpa}
3337 \def\bxjs@let@hchar@exp{%
3338   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3339 \def\bxjs@let@hchar@exp@a{%
3340   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3341   \bxjs@let@hchar@out\let\relax
3342   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3343   \bxjs@let@hchar@out\let\space%
3344   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3345   \bxjs@let@hchar@out\let\@backslashchar
3346   }{\bxjs@let@hchar@exp@b}}}%
3347 \def\bxjs@let@hchar@exp@b#1{%
3348   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3349 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3350 %\message{<#1#2>}%
3351   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3352 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3353 \bxjs@let@hchar@out\let\relax
3354 }{%else
3355 \expandafter\bxjs@let@hchar@exp
3356 }%
3357 }{%else
3358 \bxjs@let@hchar@chr#1%
3359 }}
3360 \def\bxjs@let@hchar@chr#1{%
3361 \bxjs@let@hchar@out\def{{#1}}}%
3362 \def\bxjs@let@hchar@out#1#2{%
3363 \global#1\bxjs@g@tmpa#2\relax
3364 \toks@{bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3365 \chardef\bxjs@let@hchar@csta=128
3366 \chardef\bxjs@let@hchar@cstb=192
3367 \chardef\bxjs@let@hchar@cstc=224
3368 \chardef\bxjs@let@hchar@cstd=240
3369 \chardef\bxjs@let@hchar@cste=248
3370 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3371 \def\bxjs@let@hchar@chr@ue#1{%
3372 \@tempcnta=`#1\relax
3373 %\message{\the\@tempcnta}%
3374 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3375 \bxjs@let@hchar@chr@ue@a#1%
3376 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3377 \bxjs@let@hchar@out\let\relax
3378 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3379 \bxjs@let@hchar@chr@ue@b
3380 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3381 \bxjs@let@hchar@chr@ue@c
3382 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3383 \bxjs@let@hchar@chr@ue@d
3384 }{%else
3385 \bxjs@let@hchar@out\let\relax
3386 }}}}}%
3387 \def\bxjs@let@hchar@chr@ue@a#1{%
3388 \bxjs@let@hchar@out\def{{#1}}}%
3389 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3390 \bxjs@let@hchar@out\def{{#1#2}}}%
3391 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3392 \bxjs@let@hchar@out\def{{#1#2#3}}}%
3393 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3394 \bxjs@let@hchar@out\def{{#1#2#3#4}}}%

```

## B.2 (u)pTeX 用の設定



```
3395 \ifx j\jsEngine
```

基本的に、JS クラスのコードの中で、「和文コンポーネントの管轄」として BXJS クラスで除外されている部分に相当するが、若干の変更が加えられている。

■補助マクロ `\jsLetHeadChar` を和文文字トークンに対応させる。

```
3396 \def\bxjs@let@hchar@chr@pp#1#2{%
3397   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3398 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3399   %\message{(\meaning#3:\meaning#4)}%
3400   \bxjs@cond\if#1k\fi{%
3401     \bxjs@let@hchar@out\def{#{#4}}%
3402   }{%else
3403     \bxjs@let@hchar@chr@ue#3#4%
3404   }}
3405 \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。

```
3406 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3407 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3408 \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。

```
3409 \@onlypreamble\bxjs@declarefontshape
3410 \ifjsWithupTeX
3411 \def\bxjs@declarefontshape{%
3412   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3413   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3414   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3415   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3416 }
3417 \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。

```
3418 \else
3419 \def\bxjs@declarefontshape{%
3420   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3421   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3422   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3423 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3424 }
3425 \def\bxjs@sizereference{jis}
3426 \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。

```

3427 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3428   \def\bxjs@tmpb{#5}}
3429 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3430 \else
3431 \expandafter\expandafter\expandafter\bxjs@next
3432 \expandafter\string\the\jfont\relax
3433 \fi
3434 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3435                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3436   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3437    \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。

```

3438 \begingroup
3439 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3440 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3441 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3442 % 幅が丁度 10pt なら補正は不要
3443 \ifdim\wd\z@=10pt
3444   \global\let\bxjs@scale\jsScale
3445 \else
3446 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3447   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3448   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3449   \bxjs@invscale\@tempdima\bxjs@tmpa
3450   \xdef\bxjs@scale{\strip@pt\@tempdima}
3451 \fi
3452 \endgroup
3453 %\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。

```

3454 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3455 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3456 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3457 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3458 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3459 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3460 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3461 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3462 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3463 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3464 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3465 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3466 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3467 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3468 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3469 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{-}
3470 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3471 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
3472 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
3473 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3474 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3475 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3476 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3477 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3478 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3479 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3480 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3481 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3482 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3483 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3484 \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。

```

3485 \ifx\@rmfamilyhook\@undefined % old
3486 \DeclareRobustCommand\rmfamily
3487   {\not@math@alphabet\rmfamily\mathrm
3488     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3489 \DeclareRobustCommand\sffamily
3490   {\not@math@alphabet\sffamily\mathsf
3491     \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
3492 \DeclareRobustCommand\ttfamily
3493   {\not@math@alphabet\ttfamily\mathtt
3494     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3495 \g@addto@macro\bxjs@begin@document@hook{%
3496   \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3497     % my definitions above should have been overwritten, recover it!
3498     % \selectfont is executed twice but I don't care about speed...
3499     \expandafter\g@addto@macro\csname rmfamily \endcsname
3500       {\kanjifamily\mcdefault\selectfont}%
3501     \expandafter\g@addto@macro\csname sffamily \endcsname
3502       {\kanjifamily\gtdefault\selectfont}%

```

```

3503 \expandafter\g@addto@macro\csname ttfamily \endcsname
3504 {\kanjifamily\gtdefault\selectfont}%
3505 \fi}
3506 \else % 2020-02-02
3507 \g@addto@macro\@rmfamilyhook
3508 {\prepare@family@series@update@kanji{mc}\mcdefault}
3509 \g@addto@macro\@sffamilyhook
3510 {\prepare@family@series@update@kanji{gt}\gtdefault}
3511 \g@addto@macro\@ttfamilyhook
3512 {\prepare@family@series@update@kanji{gt}\gtdefault}
3513 \fi
3514 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3515 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3516 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3517 \fi
3518 \bxjs@if@sf@default{%
3519 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3520 \selectfont

```

`\bxjs@parse@qh` の処理は pTeX 系では不要になるので無効化する（つまり `\jsSetQHLlength` は `\setlength` と等価になる）。

```

3521 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3522 \let\bxjs@parse@qh@a\@undefined
3523 \let\bxjs@parse@qh@b\@undefined

```

#### ■パラメタの設定

```

3524 \prebreakpenalty\jis"2147=10000
3525 \postbreakpenalty\jis"2148=10000
3526 \prebreakpenalty\jis"2149=10000
3527 \inhibitxspcode`!=1
3528 \inhibitxspcode`〒=2
3529 \xspcode`+=3
3530 \xspcode`%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3531 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3532 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3533 \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。

```

3534 \begingroup
3535 \catcode`\!=0

```

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

```
3536 \gdef\bxjs@ptex@dir{%
3537   !iftdir t%
3538   !else!ifydir y%
3539   !else ?%
3540   !fi!fi}
```

新版の p<sub>T</sub>E<sub>X</sub> で脚注番号の周囲の空きが過大になる現象への対処。

※現在の p<sub>T</sub>E<sub>X</sub> 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。

```
3541 % 古い \@makefnmark の定義
3542 \long\def\bxjs@tmpa{\hbox{%
3543   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3544   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3545 \ifx\@makefnmark\bxjs@tmpa
3546 \long\gdef\@makefnmark{%
3547   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3548   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3549 \fi
3550 \endgroup
```

### B.3 pdf<sub>T</sub>E<sub>X</sub> 用の処理

```
3551 \else\ifx p\jsEngine
3552 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3553 \@onlypreamble\bxjs@cjk@loaded
3554 \def\bxjs@cjk@loaded{%
3555   \def\@footnotemark{%
3556     \leavevmode
3557     \ifhmode
3558       \edef\@x@sf{\the\spacefactor}%
3559       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3560         \unkern\unkern
3561         \ifdim\lastskip>\z@ \unskip \fi
3562       \fi\fi
3563       \nobreak
3564     \fi
3565     \@makefnmark
3566     \ifhmode \spacefactor\@x@sf \fi
3567     \relax}%
3568   \let\bxjs@cjk@loaded\relax
3569 }
3570 \g@addto@macro\bxjs@begin@document@hook{%
3571   \@ifpackageloaded{CJK}{%
3572     \bxjs@cjk@loaded
3573   }{}%
3574 }
```

## B.4 XeTeX 用の処理

3575 \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。

```
3576 \def\bxjs@let@hchar@chr#1{%
3577   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3578   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3579     \bxjs@let@hchar@chr@xe
3580   }\bxjs@let@hchar@out\def{{#1}}}}
3581 \def\bxjs@let@hchar@chr@xe#1{%
3582   \lccode`0=`#1\relax
3583   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3584 \@onlypreamble\bxjs@do@precisetext
3585 \ifx\XeTeXgenerateactualtext\@undefined\else
3586   \def\bxjs@do@precisetext{%
3587     \XeTeXgenerateactualtext=\@ne}
3588 \fi
```

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

```
3589 \@onlypreamble\bxjs@do@simplejasetup
3590 \def\bxjs@do@simplejasetup{%
3591   \ifnum\XeTeXinterchartokenstate>\z@
3592   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3593     \jsSimpleJaSetup
3594     \ClassInfo\bxjs@clsname
3595     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3596   \fi\fi}
```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```
3597 \newcommand*{\jsSimpleJaSetup}{%
3598   \XeTeXlinebreaklocale "ja"\relax
3599   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3600   \XeTeXlinebreakpenalty=0\relax}
```

## B.5 後処理（エンジン共通）

3601 \fi\fi\fi

simplejasetup オプションの処理。

```
3602 \ifx\bxjs@do@simplejasetup\@undefined\else
3603   \g@addto@macro\bxjs@begin@document@hook{%
3604     \ifbxjs@simplejasetup
3605       \bxjs@do@simplejasetup
3606     \fi}
3607 \fi
```

precisetext オプションの処理。

```

3608 \ifbxjs@precisetext
3609   \ifx\bxjs@do@precisetext\@undefined
3610     \ClassWarning\bxjs@clsname
3611       {The current engine does not support the\MessageBreak
3612         'precise-text' option\@gobble}
3613   \else
3614     \bxjs@do@precisetext
3615   \fi
3616 \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。

```

3617 \g@addto@macro\bxjs@begin@document@hook{%
3618   \ifx\jsInhibitGlueAtParTop\@empty
3619     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3620     \ifx\everyparhook\bxjs@tmpa
3621       \let\everyparhook\@empty
3622     \fi
3623 \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。

```

3624 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3625 \let\bxjs@everypar\everypar
3626 \newtoks\everypar
3627 \everypar\bxjs@everypar

```

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

```

3628 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3629 \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。

```

3630 \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。

```

3631 \@onlypreamble\bxjs@adjust@fancyhdr
3632 \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。

```
3633 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3634 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3635 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3636 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3637 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3638 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3639 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3640 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3641 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3642 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3643 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3644 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3645 \def\bxjs@tmpa{\rm\thepage\strut}%
3646 \def\bxjs@tmpb{\thepage\strut}%
3647 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3648 \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。

```
3649 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3650 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3651 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3652 }\bxjs@tmpa
3653 \fi\fi
3654 \PackageInfo\bxjs@clsname
3655 {Patch to fancyhdr is applied\@gobble}}
```

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

```
3656 \def\bxjs@pagestyle@hook{%
3657 \ifpackageloaded{fancyhdr}{%
3658 \bxjs@adjust@fancyhdr
3659 \global\let\bxjs@adjust@fancyhdr\relax
3660 }{}}
```

\pagestyle にフックを入れ込む。

```
3661 \let\bxjs@org@pagestyle\pagestyle
3662 \def\pagestyle{%
3663 \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。

```
3664 \g@addto@macro\bxjs@begin@document@hook{%
3665 \bxjs@pagestyle@hook
3666 \global\let\bxjs@pagestyle@hook\relax}
```

## ■和文空白命令

```
3667 \ifbxjs@jaspace@cmd
```



`\jaenspace` 半角幅の水平空き。

```
3668 \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。

```
3669 \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。

```
3670 \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3671 \DeclareRobustCommand*{\jaspace}[1]{%
3672   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3673   \ClassError\bxjs@clsname
3674   {Unknown jaspacespace: #1}{\@eha}%
3675   \else
3676   \csname bxjs@jaspace@@#1\endcsname
3677   \fi}
3678 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3679 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3680 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3681 \fi
```

終わり。

```
3682 \fi
```

以上で終わり。

```
3683 %</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。

```
3684 %<*standard>
3685 %% このファイルは日本語文字を含みます
3686 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3687 \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。

```
3688 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3689 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3690 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3691 \define@key{bxjsStd}{jis2004}[true]{%
3692   \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。

```
3693 \newif\ifbxjs@jp@units

units オプションの処理。
3694 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3695 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3696 \define@key{bxjsStd}{units}[true]{%
3697   \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。

```
3698 \let\bxjs@jp@font\@empty

font オプションの処理。
3699 \define@key{bxjsStd}{font}{%
3700   \edef\bxjs@jp@font{#1}}
```

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

```
3701 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue

strong-cmd オプションの処理。
3702 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
3703 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
3704 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}
```

実際の japaram の値を適用する。

```
3705 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3706 \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。

```
3707 \ifbxjs@jp@jismmiv
3708 \g@addto@macro\@classoptionslist{,jis2004}
3709 % \@ifpackagewith 判定への対策
3710 \PassOptionsToPackage{jis2004}{otf}
3711 \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。

```
3712 \if j\jsEngine
3713 \bxjs@jp@unitsfalse
3714 \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。

```
3715 \ifbxjs@jp@units
3716 \IfFileExists{bxcalc.sty}{%
3717 \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3718 \ifx\usepTeXunits\@undefined
3719 \PackageWarningNoLine\bxjs@clsname
3720 {Cannot support pTeX units (zw etc.), since\MessageBreak
3721 the package 'bxcalc' is too old}%
3722 \bxjs@jp@unitsfalse
3723 \else \usepTeXunits
3724 \fi
3725 }{%else
3726 \PackageWarningNoLine\bxjs@clsname
3727 {Cannot support pTeX units (zw etc.), since\MessageBreak
3728 the package 'bxcalc' is unavailable}%
3729 \bxjs@jp@unitsfalse
3730 }
3731 \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。

```
3732 \ifbxjs@jp@units
3733 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3734 \let\bxjs@parse@qh@a\@undefined
3735 \let\bxjs@parse@qh@b\@undefined
3736 \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。

```
3737 \ifbxjs@jp@units
3738   \def\bxjs@let@lenexpr#1#2{%
3739     \edef#1{#2}%
3740     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3741 \else
3742   \def\bxjs@let@lenexpr{\edef}
3743 \fi
```

### ■ `\strong` 命令の補填

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

※

```
3744 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3745   \ifx\strong\undefined\ifx\strongenv\undefined
3746     \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3747     \DeclareTextFontCommand{\strong}{\strongenv}%
```

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

※`\strongfontdeclare` は試験的機能とする。

```
3748   \@ifdefinable{\strongfontdeclare}{\bxjs@protected\def\strongfontdeclare#1{%
3749     \bxjs@strong@font@declare@a#1,\@nil}}%
3750   \def\bxjs@strong@font@declare@a#1,#2\@nil{\def\bxjs@strong@font{#1}}%
3751   \def\bxjs@strong@font{\bfseries}%
3752   \fi\fi
3753 }\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。

```
3754 \ifjsWitheTeX
```

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

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

```
\jH3755   \@tempdima=0.25mm
3756   \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3757   \let\jH\jQ
```

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

```
\trueH3758   \ifjsc@mag
3759     \@tempdimb=\jsBaseFontSize\relax
3760     \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
```

```

3761 \tempdima=2.5mm
3762 \bxjs@invscale\tempdima\bxjs@tmpa
3763 \protected\edef\trueQ{\dimexpr\the\tempdima\relax}
3764 \tempdima=10pt
3765 \bxjs@invscale\tempdima\bxjs@tmpa
3766 \protected\edef\bxjs@truept{\dimexpr\the\tempdima\relax}
3767 \else \let\trueQ\jQ \let\bxjs@truept\p@
3768 \fi
3769 \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。

```

3770 \tempdima\trueQ \bxjs@invscale\tempdima\jsScale
3771 \protected\edef\ascQ{\dimexpr\the\tempdima\relax}
3772 \tempdima\bxjs@truept \bxjs@invscale\tempdima\jsScale
3773 \protected\edef\ascpt{\dimexpr\the\tempdima\relax}
3774 \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。

```

3775 \def\jafontsize#1#2{%
3776 \begingroup
3777 \bxjs@jafontsize@a{#1}%
3778 \tempdimb\jsInverseScale\tempdima
3779 \bxjs@jafontsize@a{#2}%
3780 \xdef\bxjs@g@tmpa{%
3781 \noexpand\fontsize{\the\tempdimb}{\the\tempdima}}%
3782 \endgroup\bxjs@g@tmpa}
3783 \def\bxjs@jafontsize@a#1{%
3784 \bxjs@parse@qh{#1}%
3785 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3786 \@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。

```

3787 \def\bxjs@kanjiskip{0pt}

```

\setkanjiskip 和文間空白の量を設定する。

```

3788 \newcommand*\setkanjiskip[1]{%
3789 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3790 \bxjs@reset@kanjiskip}

```

\getkanjiskip 和文間空白の量を表すテキストに展開する。

```

3791 \newcommand*\getkanjiskip{%
3792 \bxjs@kanjiskip}

```

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

```
3793 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

`\bxjs@enable@kanjiskip` 和文間空白の挿入を有効／無効にする。(p<sub>T</sub>E<sub>X</sub> 以外)

```
\bxjs@disable@kanjiskip 3794 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3795   \bxjs@kanjiskip@enabledtrue
3796   \bxjs@reset@kanjiskip}
3797 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3798   \bxjs@kanjiskip@enabledfalse
3799   \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。

```
3800 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3801   \ifbxjs@kanjiskip@enabled
3802     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3803   \else \@tempskipa\z@
3804   \fi
3805   \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 3806 \def\bxjs@xkanjiskip{0pt}
\getxkanjiskip 3807 \newcommand*\setxkanjiskip[1]{%
3808   \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3809   \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3810 \newcommand*\getxkanjiskip{%
3811   \bxjs@xkanjiskip}
\bxjs@disable@xkanjiskip 3812 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3813 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3814   \bxjs@xkanjiskip@enabledtrue
3815   \bxjs@reset@xkanjiskip}
3816 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3817   \bxjs@xkanjiskip@enabledfalse
3818   \bxjs@reset@xkanjiskip}
3819 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3820   \ifbxjs@xkanjiskip@enabled
3821     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3822   \else \@tempskipa\z@
3823   \fi
3824   \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。

```
3825 \g@addto@macro\jsResetDimen{%
3826   \bxjs@reset@kanjiskip
3827   \bxjs@reset@xkanjiskip}
3828 \let\bxjs@apply@kanjiskip\relax
3829 \let\bxjs@apply@xkanjiskip\relax
```

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

\bxjs@adjust@jafont \jsJaFont に入っている和文フォント設定の値を“調整”して、その結果を \bxjs@tmpa に返す。#1 が f の場合は“非埋込 (noEmbed)”の設定が禁止される。この禁止の場合も含め、何か異常がある場合は \bxjs@tmpa は空になる。

```

3830 \@onlypreamble\bxjs@adjust@jafont
3831 \def\bxjs@adjust@jafont#1{%
3832   \ifx\jsJaFont\bxjs@@auto
3833     \bxjs@get@kanjiEmbed
3834     \ifx\bxjs@kanjiEmbed\relax
3835       \let\bxjs@tmpa\@empty
3836     \else
3837       \let\bxjs@tmpa\bxjs@kanjiEmbed
3838     \fi
3839   \else
3840     \let\bxjs@tmpa\jsJaFont
3841   \fi
3842   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3843     \ClassWarningNoLine\bxjs@clsname
3844     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3845      not available on the current situation}%
3846     \let\bxjs@tmpa\@empty
3847   \fi\fi
3848 }
3849 \def\bxjs@@auto{auto}
3850 \def\bxjs@@noEmbed{noEmbed}

```

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

```

3851 \let\bxjs@kanjiEmbed\relax

```

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

```

3852 \@onlypreamble\bxjs@get@kanjiEmbed
3853 \def\bxjs@get@kanjiEmbed{%
3854   \begingroup\setbox\z@=\hbox{%
3855     \global\let\bxjs@g@tmpa\relax
3856     \endlinechar\m@ne
3857     \let\do\@makeother\dospecials
3858     \catcode32=10 \catcode12=10 %form-feed
3859     \let\bxjs@tmpa\@empty
3860     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3861     \ifeof\@inputcheck\else
3862       \read\@inputcheck to\bxjs@tmpa
3863     \closein\@inputcheck

```

```

3864 \fi
3865 \ifx\bxjs@tmpa\@empty\else
3866 \openin\@inputcheck="\bxjs@tmpa"\relax
3867 \@tempswatrue
3868 \loop\if@tempswa
3869 \read\@inputcheck to\bxjs@tmpa
3870 \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3871 \ifx\bxjs@tmpb\relax\else
3872 \global\let\bxjs@g@tmpa\bxjs@tmpb
3873 \@tempswafalse
3874 \fi
3875 \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3876 \ifx\bxjs@tmpb\relax\else
3877 \global\let\bxjs@g@tmpa\bxjs@tmpb
3878 \@tempswafalse
3879 \fi
3880 \ifeof\@inputcheck \@tempswafalse \fi
3881 \repeat
3882 \fi
3883 }\endgroup
3884 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3885 }
3886 \@onlypreamble\bxjs@get@ke@a
3887 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3888 \ifx$#1$\def\bxjs@tmpb{#2}%
3889 \else \let\bxjs@tmpb\relax
3890 \fi}
3891 \@onlypreamble\bxjs@get@ke@b
3892 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3893 \ifx$#1$\def\bxjs@tmpb{#2}%
3894 \else \let\bxjs@tmpb\relax
3895 \fi}

```

\bxjs@resolve@jafont@paren jafont パラメタ値内の()を解決する。 \bxjs@resolve@jafont@paren\CS で、 \CS の内容中の (...) を \bxjs@jafont@paren{...} に置き換える。

```

3896 \@onlypreamble\bxjs@resolve@jafont@paren
3897 \def\bxjs@resolve@jafont@paren#1{%
3898 \def\bxjs@tmpb{\let#1}%
3899 \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3900 \@onlypreamble\bxjs@resolve@jafont@paren@a
3901 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3902 \ifx\relax#4\relax \bxjs@tmpb\jJaFont
3903 \else
3904 \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3905 \bxjs@tmpb\bxjs@tmpa
3906 \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。

```
3907 \newcommand*\jachar[1]{%
3908   \begingroup
      \jsLetHeadChar で先頭の“文字”を拾ってそれを \bxjs@jachar に渡す。
3909   \jsLetHeadChar\bxjs@tmpa{#1}%
3910   \ifx\bxjs@tmpa\relax
3911     \ClassWarningNoLine\bxjs@clsname
3912       {Illegal argument given to \string\jachar}%
3913   \else
3914     \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3915   \fi
3916 \endgroup}

\jsJaChar を \jachar と等価にする。
3917 \let\jsJaChar\jachar

下請けの \bxjs@jachar の実装はエンジンにより異なる。
3918 \let\bxjs@jachar\@firstofone
```

■hyperref 対策 出力ページサイズに館する処理は geometry パッケージが行うので、hyperref 側の処理は無効にしておく。

```
3919 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3920 \@onlypreamble\bxjs@fix@hyperref@unicode
3921 \def\bxjs@fix@hyperref@unicode#1{%
3922   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3923   \@namedef{KV@Hyp@bxjs/hook}##1{%
3924     \KV@Hyp@unicode{##1}%
3925     \def\KV@Hyp@unicode####1{%
3926       \expandafter\ifx\csname if##1\expandafter\endcsname
3927         \csname if####1\endcsname\else
3928         \ClassWarningNoLine\bxjs@clsname
3929           {Blcoked hyperref option 'unicode=####1'}%
3930       \fi
3931     }%
3932   }%
3933 }
```

`\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。

```
3934 \@onlypreamble\jsCheckHyperrefUnicode
3935 \let\jsCheckHyperrefUnicode\@empty
3936 \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。

```
3937 \@onlypreamble\bxjs@check@hyperref@unicode
3938 \def\bxjs@check@hyperref@unicode#1{%
3939   \g@addto@macro\jsCheckHyperrefUnicode{%
```

```

3940 \@tempwafalse
3941 \begingroup
3942 \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3943 \aftergroup\@tempwattrue \fi
3944 \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3945 \csname if#1\endcsname
3946 \aftergroup\@tempwattrue \fi
3947 \endgroup
3948 \if@tempwa\else
3949 \ClassError\bxjs@clsname
3950 {The value of hyperref 'unicode' key is not suitable\MessageBreak
3951 for the present engine (must be #1)}%
3952 {\@ehc}%
3953 \fi}}

```

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

```

3954 \@onlypreamble\bxjs@urgent@special
3955 \@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。

```

3956 \ifbxjs@old@hook@system
3957 \def\bxjs@urgent@special#1{%
3958 \AtBeginDvi{\special{#1}}%
3959 \g@addto@macro\bxjs@begin@document@hook{%
3960 \@ifpackageloaded{atbegshi}{%
3961 \begingroup
3962 \toks\z{\special{#1}}%
3963 \toks\tw{\expandafter{\AtBegShi@HookFirst}}%
3964 \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw}%
3965 \endgroup
3966 }{}%
3967 }%
3968 }

```

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。

```

3969 \else
3970 \def\bxjs@urgent@special#1{%
3971 \bxjs@urgent@special@a
3972 \AddToHook{shipout/firstpage}[pxjahyper/enc]{\special{#1}}
3973 \def\bxjs@urgent@special@a{%
3974 \DeclareHookRule{shipout/firstpage}[pxjahyper/enc]{<}{hyperref}%
3975 \global\let\bxjs@urgent@special@a\relax}
3976 \fi

```

## C.4 pT<sub>E</sub>X 用設定

```
3977 \if j\jsEngine
```

#### ■ 共通命令の実装

```
3978 \def\bxjs@apply@kanjiskip{%
3979   \kanjiskip\@tempskipa}
3980 \def\bxjs@apply@xkanjiskip{%
3981   \xkanjiskip\@tempskipa}
```

\jaJaChar のサブマクロ。

```
3982 \def\bxjs@jachar#1{%
3983   \bxjs@jachar@a#1...\@nil}
3984 \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。

```
3985   \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。

```
3986   \else\ifx.#3%
3987     \@tempcnta`#1 \multiply\@tempcnta64
3988     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3989     \bxjs@jachar@b
3990   \else\ifx.#4%
3991     \@tempcnta`#1 \multiply\@tempcnta64
3992     \advance\@tempcnta`#2 \multiply\@tempcnta64
3993     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3994     \bxjs@jachar@b
3995   \else
3996     \@tempcnta`#1 \multiply\@tempcnta64
3997     \advance\@tempcnta`#2 \multiply\@tempcnta64
3998     \advance\@tempcnta`#3 \multiply\@tempcnta64
3999     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
4000     \bxjs@jachar@b
4001   \fi\fi\fi}
```

符号値が \@tempcnta の和文文字を出力する処理。

```
4002 \ifjsWithupTeX
4003   \def\bxjs@jachar@b{\kchar\@tempcnta}
4004 \else
4005   \def\bxjs@jachar@b{%
4006     \ifx\bxUInt\@undefined\else
4007       \bxUInt{\@tempcnta}%
4008     \fi}
4009 \fi
```

和欧文間空白の命令 \jathinspace の実装。

```
4010 \ifbxjs@jaspace@cmd
4011   \def\jathinspace{\hskip\xkanjiskip}
4012 \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。

```
4013 \ifbxjs@jp@jismmiv
4014   \PassOptionsToPackage{prefer2004jis}{pxchfon}
4015   \PassOptionsToPackage{prefer2004jis}{pxbabel}
4016 \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。

```
4017 \let\bxjs@jafont@paren\@firstofone
4018 \let\bxjs@tmpa\jsJaFont
4019 \ifx\bxjs@tmpa\bxjs@@auto
4020   \let\bxjs@tmpa\@empty
4021 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
4022   \def\bxjs@tmpa{noembed}
4023 \fi\fi
4024 \bxjs@resolve@jafont@paren\bxjs@tmpa
4025 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4026 \ifx\bxjs@tmpa\@empty\else
4027   \edef\bxjs@next{%
4028     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4029   }\bxjs@next
4030 \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……。)

```
4031 \begingroup
4032   \global\let\bxjs@g@tmpa\relax
4033   \catcode`\|=0 \catcode`\|=12
4034   |def|bxjs@tmpdo#1|@nil{%
4035     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4036   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4037     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4038   |catcode`\|=0 \catcode`\|=12
4039   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4040     \ifx$#2$\else
4041       \xdef\bxjs@g@tmpa{%
4042         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4043     \fi}
4044 \@firstofone{%
4045   \catcode10=12 \endlinechar\m@ne
```

```

4046 \let\do\@makeother \dospecials \catcode32=10
4047 \openin\@inputcheck=otf.sty\relax
4048 \@tempswatrue
4049 \loop\if@tempswa
4050   \ifeof\@inputcheck \@tempswafalse \fi
4051   \if@tempswa
4052     \read\@inputcheck to\bxjs@next
4053     \expandafter\bxjs@tmpdo\bxjs@next\@nil
4054   \fi
4055 \repeat
4056 \closein\@inputcheck
4057 \endgroup}
4058 \bxjs@g@tmpa

```

■hyperref 対策 unicode にはいけない。

```

4059 \ifbxjs@hyperref@enc
4060   \bxjs@check@hyperref@unicode{false}
4061 \fi

```

tounicode special 命令を出力する。

```

4062 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4063   \else\ifjsWithpTeXng T\else F\fi\fi T%
4064   \ifnum\jis"2121="A1A1 %euc
4065     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4066   \else\ifnum\jis"2121="8140 %sjis
4067     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4068   \else\ifnum\jis"2121="3000 %uptex
4069     \ifbxjs@bigcode
4070       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4071       \PassOptionsToPackage{bigcode}{pxjahyper}
4072     \else
4073       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4074     \fi
4075   \fi\fi\fi
4076 \let\bxToUnicodeSpecialDone=t
4077 \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。

```

4078 \ifx f\bxjs@enablejfam\else
4079   \@enablejfamtrue
4080 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

4081 \if@enablejfam
4082   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4083   \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4084   \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4085   \jfam\symmincho

```

```

4086 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4087 \g@addto@macro\bxjs@begin@document@hook{%
4088   \ifx\reDeclareMathAlphabet\@undefined\else
4089     \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4090     \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4091     \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4092   \fi}
4093 \fi

```

## C.5 pdfTeX 用設定 : CJK + bxcjkatype

```
4094 \else\if p\jsEngine
```

■bxcjkatype パッケージの読み込 \jsJaFont が指定されている場合は、その値を bxcjkatype のオプション（プリセット指定）に渡す。（auto ならば \bxjs@get@kanjiEmbed を実行する。）スケール値（\jsScale）の反映は bxcjkatype の側で行われる。

※ Pandoc モードでは autotilde を指定しない。

```

4095 \bxjs@adjust@jafont{f}
4096 \let\bxjs@jafont@paren\@firstofone
4097 \bxjs@resolve@jafont@paren\bxjs@tmpa
4098 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4099 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4100 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4101   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4102 \fi
4103 \edef\bxjs@next{%
4104   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4105 }\bxjs@next
4106 \bxjs@cjk@loaded

```

■hyperref 対策 bxcjkatype 使用時は unicode にするべき。

※取りあえず固定はしない。

```

4107 \ifbxjs@hyperref@enc
4108   \PassOptionsToPackage{unicode}{hyperref}
4109 \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。

```

4110 \ifx\bxjkatypeHyperrefPatchDone\@undefined
4111 \begingroup
4112   \CJK@input{UTF8.bdg}
4113 \endgroup
4114 \g@addto@macro\pdfstringdefPreHook{%

```

```

4115 \@nameuse{CJK@UTF8Binding}%
4116 }
4117 \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。

```

4118 \ifx\bxCJKtypeHyperrefPatchDone\@undefined
4119 \g@addto@macro\pdfstringdefPreHook{%
4120   \ifx~\bxjs@@CJKtilde
4121     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4122     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4123     \let~\@empty
4124   \fi
4125 }
4126 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4127 \def\bxjs@tildecmd{~}
4128 \def\bxjs@LetUnexpandableSpace#1{%
4129   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4130     \bxjs@org@LetUnexpandableSpace#1%
4131   \fi}
4132 \fi

```

#### ■ 共通命令の実装

```

4133 \newskip\jsKanjiSkip
4134 \newskip\jsXKanjiSkip
4135 \ifx\CJKecglue\@undefined
4136   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4137 \fi
4138 \let\autospacing\bxjs@enable@kanjiskip
4139 \let\noautospacing\bxjs@disable@kanjiskip
4140 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4141 \def\bxjs@apply@kanjiskip{%
4142   \jsKanjiSkip\@tempskipa
4143   \let\CJKglue\bxjs@CJKglue}
4144 \let\autoxspacing\bxjs@enable@xkanjiskip
4145 \let\noautoxspacing\bxjs@disable@xkanjiskip
4146 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4147 \def\bxjs@apply@xkanjiskip{%
4148   \jsXKanjiSkip\@tempskipa
4149   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4150 \def\bxjs@jachar#1{%
4151   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4152 \ifbxjs@jaspace@cmd
4153   \protected\def\jathinspace{\CJKecglue}
4154 \fi

```

■和文数式ファミリー CJK パッケージは（恐らく）数式文字として CJK 文字をサポートしていない。従って @enablejfam は常に偽になる。

```
4155 \ifx t\bxjs@enablejfam
4156   \ClassWarningNoLine\bxjs@clsname
4157   {You cannot use 'enablejfam=true', since the\MessageBreak
4158     CJK package does not support Japanese math}
4159 \fi
```

## C.6 Xe<sub>La</sub>TeX 用設定：xeCJK + zxjatype

```
4160 \else\if x\jsEngine
```

■zxjatype パッケージの読み込 スケール値 (\jsScale) の反映は zxjatype の側で行われる。

```
4161 \RequirePackage{zxjatype}
4162 \PassOptionsToPackage{no-math}{fontspec}%!
4163 \PassOptionsToPackage{xetex}{graphicx}%!
4164 \PassOptionsToPackage{xetex}{graphics}%!
4165 \ifx\zxJaFamilyName\@undefined
4166   \ClassError\bxjs@clsname
4167   {xeCJK or zxjatype is too old}\@ehc
4168 \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。

```
4169 \bxjs@adjust@jafont{f}
4170 \let\bxjs@jafont@paren\@gobble
4171 \bxjs@resolve@jafont@paren\bxjs@tmpa
4172 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4173 \ifx\bxjs@tmpa\@empty
4174   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4175   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4176 \else
4177   \edef\bxjs@next{%
4178     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4179   }\bxjs@next
4180 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X<sub>e</sub>L<sub>A</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 の方針としては、X<sub>e</sub>L<sub>A</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



と見なしていて警告が発生する。

これを踏まえて、ここでは、「Xe<sub>La</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。

※取りあえず固定はしない。

```
4181 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4182   \ifbxjs@hyperref@enc
4183     \PassOptionsToPackage{unicode}{hyperref}
4184   \fi
4185 \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。

```
4186 \AtEndOfPackage{%
4187 \def\@gnewline #1{%
4188   \ifvmode \@nolnerr
4189   \else
4190     \unskip \reserved@a {\reserved@a#1}\nobreak \hfil \break \null
4191     \nobreak \hskip-1sp\hskip1sp\relax
4192     \ignorespaces
4193   \fi}
4194 }
```

#### ■共通命令の実装

```
4195 \newskip\jsKanjiSkip
4196 \newskip\jsXKanjiSkip
4197 \ifx\CJKecglue\@undefined
4198   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4199 \fi
4200 \let\autospacing\bxjs@enable@kanjiskip
4201 \let\noautospacing\bxjs@disable@kanjiskip
4202 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4203 \def\bxjs@apply@kanjiskip{%
4204   \jsKanjiSkip\@tempskipa
4205   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4206 \let\autoxspacing\bxjs@enable@xkanjiskip
4207 \let\noautoxspacing\bxjs@disable@xkanjiskip
4208 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4209 \def\bxjs@apply@xkanjiskip{%
4210   \jsXKanjiSkip\@tempskipa
4211   \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。

```
4212 \ifx\mcfamily\@undefined
4213   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4214   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4215 \fi
```

`\jachar` のサブマクロの実装。

```
4216 \def\bxjs@jachar#1{%
4217   \xeCJKDeclareCharClass{CJK}{\#1}\relax
4218   #1}
```

`\jathinspace` の実装。

```
4219 \ifbxjs@jaspace@cmd
4220   \protected\def\jathinspace{\CJKecglue}
4221 \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。

```
4222 \ifx t\bxjs@enablejfam
4223   \@enablejfamtrue
4224 \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。

```
4225 \if@enablejfam
4226   \xeCJKsetup{CJKmath=true}
4227 \fi
```

## C.7 Lua<sub>T</sub>E<sub>X</sub> 用設定 : Lua<sub>T</sub>E<sub>X</sub>-ja

```
4228 \else\if 1\jsEngine
```

■Lua<sub>T</sub>E<sub>X</sub>-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。

```
4229 \let\zw\@undefined
4230 \RequirePackage{luatexja}
4231 \edef\bxjs@next{%
4232   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4233 }\bxjs@next
```

フォント代替の明示的定義。

```
4234 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{-}
4235 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
```

```

4236 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4237 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4238 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4239 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4240 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4241 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4242 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4243 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4244 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4245 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4246 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4247 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4248 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4249 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4250 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4251 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4252 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4253 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4254 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4255 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4256 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4257 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4258 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4259 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4260 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4261 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4262 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4263 \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。

```

4264 \bxjs@adjust@jafont{t}
4265 \ifx\bxjs@tmpa\bxjs@noEmbed
4266   \def\bxjs@tmpa{noembed}
4267 \fi
4268 \let\bxjs@jafont@paren\@gobble
4269 \bxjs@resolve@jafont@paren\bxjs@tmpa
4270 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4271 \ifx\bxjs@tmpa\@empty
4272   \defaultjfontfeatures{ Kerning=Off }
4273   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4274   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4275 \else
4276   \edef\bxjs@next{%

```

```

4277 \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4278 }\bxjs@next
4279 \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。

```

4280 \ifpackagelater{luatexja}{2016/03/31}{\else
4281 \DeclareRobustCommand\rmfamily
4282 {\not@math@alphabet\rmfamily\mathrm
4283 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4284 \DeclareRobustCommand\sffamily
4285 {\not@math@alphabet\sffamily\mathsf
4286 \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
4287 \DeclareRobustCommand\ttfamily
4288 {\not@math@alphabet\ttfamily\mathtt
4289 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4290 }
4291 \long\def\jttdefault{\gtdefault}
4292 \unless\ifx\@ltj@match@familytrue\@undefined
4293 \@ltj@match@familytrue
4294 \fi
4295 \g@addto@macro\bxjs@begin@document@hook{%
4296 \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4297 \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4298 \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4299 \bxjs@if@sf@default{%
4300 \renewcommand\kanjifamilydefault{\gtdefault}}

```

#### ■和文パラメタの設定

```

4301 % 次の3つは既定値の通り
4302 %\ltjsetparameter{prebreakpenalty={`',10000}}
4303 %\ltjsetparameter{postbreakpenalty={`",10000}}
4304 %\ltjsetparameter{prebreakpenalty={`",10000}}
4305 \ltjsetparameter{jaxspmode={`!',1}}
4306 \ltjsetparameter{jaxspmode={`〒,2}}
4307 \ltjsetparameter{alxspmode={`+,3}}
4308 \ltjsetparameter{alxspmode={`%,3}}

```

■段落頭でのグルー挿入禁止 基本的に現状の `ltjs*` クラスの処理に合わせる。

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の LuaTeX-jā で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合に備えて同等のものを用意する。

```

4309 \ifx\ltjfakeparbegin\@undefined
4310 \protected\def\ltjfakeparbegin{%
4311 \ifhmode
4312 \relax\directlua{%
4313 luatexja.jfmglue.create_beginpar_node()}}

```

```

4314 \fi}
4315 \fi

```

ltjs\* クラスの定義と同等になるようにパッチを当てる。

```

4316 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4317 \begingroup
4318 \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4319 \@gobble\if\def\bxjs@tmpa{\@{\everypar}\fi}
4320 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4321 \@gobble\if\def\bxjs@tmpa{\@{\everypar\everyparhook}\fi}\fi
4322 \directlua{
4323   local function patchcmd(cs, code, from, to)
4324     tex.sprint(code:gsub(from:gsub("\%W", "\%\%\%0"), "\%0"..to)
4325       :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4326   end
4327   patchcmd(\@xsect, [[\meaning\xsect]],
4328     \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4329   patchcmd(\@item, [[\meaning\@item]],
4330     \bxjs@tmpa, \@ltjfakeparbegin)}
4331 \endgroup
4332 \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。

```

4333 \ifbxjs@hyperref@enc
4334 \PassOptionsToPackage{unicode}{hyperref}
4335 \bxjs@check@hyperref@unicode{true}
4336 \fi

```

■共通命令の実装

```

4337 \protected\def\autospacing{%
4338 \ltjsetparameter{autospacing=true}}
4339 \protected\def\noautospacing{%
4340 \ltjsetparameter{autospacing=false}}
4341 \protected\def\autoxspacing{%
4342 \ltjsetparameter{autoxspacing=true}}
4343 \protected\def\noautoxspacing{%
4344 \ltjsetparameter{autoxspacing=false}}
4345 \def\bxjs@apply@kanjiskip{%
4346 \ltjsetparameter{kanjiskip={\@tempskipa}}}
4347 \def\bxjs@apply@xkanjiskip{%
4348 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4349 \def\bxjs@jachar#1{%
4350 \ltjjachar`#1\relax}

```

\jathinspace の実装。

```

4351 \ifbxjs@jaspace@cmd

```

```

4352 \protected\def\jathinspace{%
4353   \hskip\ltjgetparameter{xkanjiskip}\relax}
4354 \fi

```

■和文数式ファミリー LuaTeX-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。

```

4355 \ifx f\bxjs@enablejfam
4356   \ClassWarningNoLine\bxjs@clsname
4357   {You cannot use 'enablejfam=false', since the\MessageBreak
4358     LuaTeX-ja always provides Japanese math families}
4359 \fi

```

## C.8 共通処理 (2)

```

4360 \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 4361 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4362   \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4363   \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4364 \fi

```

\mathmc この時点で未定義である場合に限り、\DeclareJaMathFontCommand を利用したフォール

\mathgt バックの定義を行う。

```

4365 \ifx\mathmc\@undefined
4366   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4367   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4368 \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)。

```

4369 \ifbxjs@jaspace@cmd
4370   \bxjs@protected\def\bxjs@choice@jathinspace{%
4371     \relax\ifmmode \mskip\medmuskip
4372     \else \jathinspace\ignorespaces
4373     \fi}
4374   \jsAtEndOfClass{%
4375     \ifjswitheTeX \let\>\bxjs@choice@jathinspace
4376     \else \def\>\protect\bxjs@choice@jathinspace}%
4377   \fi}
4378 \fi

```

### ■和文・和欧文間空白の初期値

```

4379 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4380 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4381 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4382 \fi

```

以上で終わり。

```

4383 %</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。

```

4384 %<*modern>
4385 \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。

```

4386 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4387 \def\encodingdefault{T1}%
4388 \input{t1enc.def}%
4389 \fontencoding\encodingdefault\selectfont
4390 \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。

```

4391 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4392 \renewcommand{\rmdefault}{lmr}
4393 \renewcommand{\sfdefault}{lms}
4394 \renewcommand{\ttdefault}{lmtt}
4395 \fi

```

大型演算子用の数式フォントの設定。

※`amsmaths` パッケージと同等にする。

```

4396 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4397   <-7.5>cmex7<7.5-8.5>cmex8%
4398   <8.5-9.5>cmex9<9.5->cmex10}{}%
4399 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

```

`amsmaths` 読込時に上書きされるのを防ぐ。

```

4400 \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。

```

4401 \ifx\@IncludeInRelease\@undefined
4402 \RequirePackage{fixltx2e}
4403 \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。

```

4404 \RequirePackage{bxjscjkat}

```

### D.4 完了

おしまい。

```

4405 %</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。

```

4406 %<*pandoc>
4407 \input{bxjsja-standard.def}

```

bxjspandoc パッケージを読み込む。

```

4408 \RequirePackage{bxjspandoc}

```

`\bxjs@endpreamble@hook` etoolbox の `\AtEndPreamble` で実行される BXJS クラス用のフック。

※  $\varepsilon$ -TeX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```

4409 \@onlypreamble\bxjs@endpreamble@hook
4410 \let\bxjs@endpreamble@hook\@empty
4411 \ifjsWithTeX
4412   \RequirePackage{etoolbox}[2010/08/21]% v2.0
4413   \AtEndPreamble{\bxjs@endpreamble@hook}
4414 \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。

```

4415 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue

```



fix-strong オプションの処理。

```
4416 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
4417 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
4418 \define@key{bxjsPan}{fix-strong}[true]{%
4419   \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。

```
4420 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue
```

fix-code オプションの処理。

```
4421 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
4422 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
4423 \define@key{bxjsPan}{fix-code}[true]{%
4424   \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。

```
4425 \chardef\bxjs@jp@strong=0
```

strong オプションの処理。

```
4426 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
4427 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
4428 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }
4429 \define@key{bxjsPan}{strong}{%
4430   \bxjs@set@keyval{strong}{#1}{}}
```

実際の japaram の値を適用する。

```
4431 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}
4432 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

### E.3 dupload システム

パッケージが重複して読み込まれたときに“option clash”の検査をスキップする。この時に何らかのコードを実行させることができる。

\bxjs@set@dupload@proc \bxjs@set@dupload@proc{〈ファイル名〉}{〈定義本体〉} 特定のファイルの読込が \@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。

```
4433 \@onlypreamble\bxjs@set@dupload@proc
4434 \def\bxjs@set@dupload@proc#1{%
4435   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
4436 \@onlypreamble\bxjs@set@dupload@proc@a
4437 \def\bxjs@set@dupload@proc@a#1{%
4438   \@onlypreamble#1\def#1##1}
4439 \def\bxjs@unset@dupload@proc#1{%
4440   \expandafter\let\csname bxjs@dlp/#1\endcsname\@undefined}
```

\@if@options \@if@options の再定義。

```

4441 \@onlypreamble\bxjs@org@if@options
4442 \let\bxjs@org@if@options\@if@options
4443 \@onlypreamble\bxjs@org@reset@options
4444 \let\bxjs@org@reset@options\relax
4445 \def\@if@options#1#2#3{%
4446   \let\bxjs@next\@secondoftwo
4447   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@current}%
4448   \ifx\bxjs@tmpa\bxjs@tmpb
4449     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4450       \let\bxjs@next\@firstoftwo \fi
4451   \fi
4452   \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4453 \g@addto@macro\bxjs@begin@document@hook{%
4454   \let\@if@options\bxjs@org@if@options}
4455 \@onlypreamble\bxjs@do@dupload@proc
4456 \def\bxjs@do@dupload@proc#1#2#3{%
4457   \ifx\bxjs@org@reset@options\relax
4458     \let\bxjs@org@reset@options\@reset@options
4459   \fi
4460   \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4461   \def\@reset@options{%
4462     \let\@reset@options\bxjs@org@reset@options
4463     \@reset@options
4464     \bxjs@next{#3}}%
4465   \@firstoftwo}

```

## E.4 lang 変数

`lang=ja` という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐための対策を行う。

`\bxjs@polyglossia@options` Polyglossia のオプション列のテキスト。“実際には読み込まれていない” 場合は `\relax` になる。

```
4466 \let\bxjs@polyglossia@options\relax
```

`\bxjs@babel@options` Babel のオプション列のテキスト。“実際には読み込まれていない” 場合は `\relax` になる。

```
4467 \let\bxjs@babel@options\relax
```

■Polyglossia について つまり Xe<sub>La</sub>TeX・Lua<sub>La</sub>TeX の場合。

※この場合 `etoolbox` が使用可能になっている。

```
4468 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0
```

パッケージの読込を検知するため読込済のマークを付けて `dupload` の処理を仕込む。

```

4469 \pandocSkipLoadPackage{polyglossia}
4470 \bxjs@set@dupload@proc{polyglossia.sty}{%
4471   \bxjs@unset@dupload@proc{polyglossia.sty}%
4472   \ClassWarning\bxjs@clsname

```

```

4473   {Package polyglossia is required}%
4474   \def\bxjs@polyglossia@options{#1}%

```

polyglossia の読込が指示された場合、直後に \setmainlanguage が実行されることを想定して、フック用の \setmainlanguage を定義する。

※先に \setmainlanguage 以外が実行された場合はエラーになる。

```

4475   \newcommand*\setmainlanguage[2] []{%

```

もし、言語名が空の \setmainlanguage{} が実行された場合は、lang=ja が指定されたと見なす。

```

4476   \ifblank{##2}{%
4477     \ClassWarning\bxjs@clsname
4478     {Main language is invalid, thus the fallback\MessageBreak
4479     definitions will be employed}%
4480     \bxjs@pandoc@polyglossia@ja

```

言語名が非空ならば、本来の polyglossia の処理を実行する。

```

4481   }{%else
4482     \ClassWarning\bxjs@clsname
4483     {Main language is valid (##2),\MessageBreak
4484     thus polyglossia will be loaded}%
4485     \csundef{ver@polyglossia.sty}%
4486     \edef\bxjs@next{%
4487       \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia} []%
4488     }\bxjs@next
4489     \setmainlanguage[##1]{##2}%
4490   }}}

```

プレアンブルで polyglossia の読込が指示されなかった場合、Polyglossia と連携するパッケージの誤動作を防ぐため、読込済マークを外す。

```

4491 \g@addto@macro\bxjs@endpreamble@hook{%
4492   \ifx\bxjs@polyglossia@options\relax
4493     \csundef{ver@polyglossia.sty}%
4494   \fi}

```

\bxjs@pandoc@polyglossia@ja Pandoc 側で lang=ja が指定されていた場合の処理。この場合は Polyglossia の処理を無効化するためにダミーの定義を行う。その時点でダミーの \setotherlanguage(s) を定義する。

```

4495 \onlypreamble\bxjs@pandoc@polyglossia@ja
4496 \def\bxjs@pandoc@polyglossia@ja{%
4497   \renewcommand*\setmainlanguage[2] []{%
4498     \newcommand*\setotherlanguage[2] []{%
4499       \ifblank{##2}{}%else
4500       \cslet{##2}\@empty \cslet{end##2}\@empty
4501       \cslet{text##2}\@firstofone}}%
4502   \newcommand*\setotherlanguages[2] []{%
4503     \@for\bxjs@tmpa:={##2}\do{%
4504       \setotherlanguage{\bxjs@tmpa}}}%

```

Polyglossia の読込済マークは外れるようにしておく。

```
4505 \let\bxjs@polyglossia@options\relax}%  
4506 \fi
```

■Babel について Xe<sub>La</sub>TeX 以外の場合。

※ Pandoc 2.6 版において、Lua<sub>TeX</sub> で用いられる多言語パッケージが Polyglossia から Babel に変更された。

```
4507 \if x\jsEngine\else
```

パッケージの読込を検知するため読込済のマークを付けて dupload の処理を仕込む。

```
4508 \pandocSkipLoadPackage{babel}  
4509 \bxjs@set@dupload@proc{babel.sty}{%  
4510 \bxjs@unset@dupload@proc{babel.sty}%  
4511 \ClassWarning\bxjs@clsname  
4512 {Package babel is required}}%
```

パッケージオプションに言語名が空の main= があるかを調べる。ある場合は lang=ja 対策を実行する。

※\bxjs@babel@options には main= を除いたオプション列を格納する。

```
4513 \@tempswafalse \let\bxjs@babel@options\@empty  
4514 \def\bxjs@tmpb{main=}%  
4515 \@for\bxjs@tmpa:=#1\do{%  
4516 \ifx\bxjs@tmpa\bxjs@tmpb \@tempswattrue  
4517 \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%  
4518 \fi}%  
4519 \if@tempswa  
4520 \ClassWarning\bxjs@clsname  
4521 {Main language is invalid, so fallback\MessageBreak  
4522 definitions will be employed}%  
4523 \bxjs@pandoc@polyglossia@ja
```

ない場合は、本来の babel の処理を実行する。

```
4524 \else  
4525 \ClassWarning\bxjs@clsname  
4526 {Main language is valid,\MessageBreak  
4527 thus babel will be loaded}%  
4528 \expandafter\let\csname ver@babel.sty\endcsname\@undefined  
4529 \RequirePackage[#1]{babel}[]%  
4530 \fi}
```

プレアンブルで babel の読込が指示されなかった場合、読込済マークを外す。

```
4531 \g@addto@macro\bxjs@endpreamble@hook{%  
4532 \ifx\bxjs@babel@options\relax  
4533 \expandafter\let\csname ver@babel.sty\endcsname\@undefined  
4534 \fi}
```

\bxjs@pandoc@babel@ja Pandoc 側で lang=ja が指定されていた場合の処理。

```
4535 \@onlypreamble\bxjs@pandoc@babel@ja
```

```

4536 \def\bxjs@pandoc@babel@ja{%
4537   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4538   \edef\bxjs@next{%
4539     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4540   }\bxjs@next
4541   \if j\jsEngine
4542     \RequirePackage[main=japanese]{pxbabel}[]%
4543   \else
4544     \RequirePackage{bxorigcapt}[]%
4545   \fi}

```

lang 対策はこれで終わり。

```
4546 \fi
```

## E.5 geometry 変数

geometry を “再度読み込んだ” 場合に、そのパラメタで \setpagelayout\* が呼ばれるようにする。

```

4547 \bxjs@set@dupload@proc{geometry.sty}{%
4548   \setpagelayout*{#1}}

```

## E.6 CJKmainfont 変数

LuaTeX (+ LuaTeX-ja) の場合に CJKmainfont 変数が指定された場合は \setmainfont の指定にまわす。

```

4549 \if l\jsEngine
4550   \pandocSkipLoadPackage{xeCJK}
4551   \providecommand*\setCJKmainfont{\setmainfont}
4552 \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。

```

4553 \if x\jsEngine
4554   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4555     ,space}
4556 \fi

```

## E.8 paragraph のマーク

BXJS クラスでは \paragraph の見出しの前に \jsParagraphMark で指定したマークが付加され、既定ではこれは “■” である。しかし、この規定は \paragraph が本来のレイア

ウトを保っている、すなわち「行内見出しである」「節番号が付かない」ことが前提になっていると考えられる。Pandocはこの規定を変更することがある（特に既定で `\paragraph` を別行見出しに再定義する）ため、変更された場合は `\jsParagraphMark` の既定値を空にする。

Pandoc がプレアンブルで行う再定義の結果を調べるため、`begin-document` フックを利用する。

```
4557 \g@addto@macro\bxjs@begin@document@hook{%
4558   \@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。

```
4559   \ifx\oldparagraph\@undefined\else
4560     \@tempswatruetrue
4561   \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4562   \ifnum\c@secnumdepth>3
4563     \@tempswatruetrue
4564   \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。

```
4565   \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4566     \let\jsParagraphMark\@empty
4567   \fi\fi}
```

## E.9 全角空白文字

```
4568 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4569   \catcode"3000=\active
4570   \begingroup \catcode`\!=7
4571   \protected\gdef!!!3000{\zwspace}
4572   \endgroup
4573 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4574   \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4575   \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4576 \fi\fi
```

## E.10 hyperref 対策

`hyperref` の `unicode` オプションの固定を行う。

```
4577 \if j\jsEngine
4578   \bxjs@fix@hyperref@unicode{false}
4579 \else
4580   \bxjs@fix@hyperref@unicode{true}
4581 \fi
```

## E.11 Pandoc 要素に対する和文用の補正

■重要要素 重要 (Strong) 要素に対する 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)。

```
4582 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
4583   \let\orgtextbf\textbf
4584   \DeclareRobustCommand\bxjstextbf[1]{%
4585     \begingroup
4586       \let\textbf\orgtextbf
4587       \strong{#1}%
4588     \endgroup}%
4589   \g@addto@macro\bxjs@begin@document@hook{%
4590     \let\textbf\bxjstextbf}
4591 \fi\fi
```

`\strong` の書体を設定する。

```
4592 \jsAtEndOfClass{%
4593   \ifx\strongfontdeclare\@undefined\else
4594     \ifcase\bxjs@jp@strong
4595       \or \strongfontdeclare{\sffamily}%
4596       \or \strongfontdeclare{\sffamily\bfseries}%
4597     \fi
4598   \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。

```
4599 \ifbxjs@jp@fix@code
4600   \chardef\bxjs@eghost@c=23
4601   \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4602   \else\ifx l\jsEngine \ltjsetparameter{alxspmode={\bxjs@eghost@c,3}}
4603   \else\ifx x\jsEngine %no-op
4604   \else \let\bxjs@eghost@c\@undefined
4605   \fi\fi\fi
4606   \ifx\bxjs@eghost@c\@undefined\else
4607     \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4608     \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4609     \let\orgtexttt\texttt
4610     \DeclareRobustCommand\bxjstexttt[1]{%
4611       \ifmmode \nfss@text{\ttfamily #1}%
4612       \else
4613         \ifvmode \leavevmode \fi
4614         \bxjs@pan@eghost\null
4615         \begingroup \ttfamily #1\endgroup
4616         \null\bxjs@pan@eghost
4617       \fi}
4618     \g@addto@macro\bxjs@begin@document@hook{%
4619       \let\texttt\bxjstexttt}
```

さらに `\verb` の外側にも欧文ゴーストが入るようにする。

※`bxjaghost` の実装を参考にした。

```
4620 \expandafter\def\expandafter\verb\expandafter{%
4621 \expandafter\bxjs@pan@eghost\verb}
4622 \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4623 \fi
4624 \fi
```

## E.12 完了

おしまい。

```
4625 %</pandoc>
```

和文ドライバ実装はここまで。

```
4626 %</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。

```
4627 %<*anc>
```

## 付録 G 補助パッケージ : `bxjscompat`

ムニャムニャムニャ……。

### G.1 準備

```
4628 %<*compat>
4629 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。
4630 \let\bxac@engine=n
4631 \def\bxac@do#1#2{%
4632 \edef\bxac@tmpa{\string#1}%
4633 \edef\bxac@tmpb{\meaning#1}%
4634 \ifx\bxac@tmpa\bxac@tmpb #2\fi}
4635 \bxac@do\XeTeXversion{\let\bxac@engine=x}
4636 \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。

```
4637 \ifx\jsAtEndOfClass\@undefined
4638   \let\bxac@delayed@if@bxjs\@firstofone
4639 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
4640 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 4641 \newif\ifbxac@in@old@behavior
4642 \let\ImposeOldLuaTeXBehavior\relax
4643 \let\RevokeOldLuaTeXBehavior\relax
```

## G.2 XeTeX 部分

```
4644 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。
4645 \@onlypreamble\bxac@adjust@charclass
4646 \bxac@delayed@if@bxjs{%
4647   \@ifpackageloaded{xeCJK}{-}{%else
4648     \ifx\Xe@alloc@intercharclass\@undefined\else
4649       \ifnum\Xe@alloc@intercharclass=\z@
4650         \PackageInfo\bxac@pkgname
4651           {Setting up interchar class for CJK...\@gobble}%
4652         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4653           \Xe@alloc@intercharclass=3
4654         }{%else
4655           \PackageWarning\bxac@pkgname
4656             {Cannot find file 'load-unicode-xetex-classes.tex'%
4657             \@gobble}%
4658         }%
4659   \fi\fi
4660   \ifnum\XeTeXcharclass"3041=\z@
4661     \PackageInfo\bxac@pkgname
4662       {Adjusting interchar class for CJK...\@gobble}%
4663     \@for\bxac@x:={%
4664       3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4665       3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4666       30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4667       31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4668       31FF%
4669     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4670   \fi
4671 }%
4672 }
```

以上。

```
4673 \fi
```

## G.3 LuaTeX 部分

```

4674 \ifx l\bxac@engine
      ムニヤムニヤ。
4675 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4676   \chardef\pdftexversion=200
4677   \def\pdftexrevision{0}
4678   \let\pdftexbanner\luatexbanner
4679 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior 4680 \begingroup\expandafter\expandafter\expandafter\endgroup
4681 \expandafter\ifx\csname outputmode\endcsname\relax\else
4682 \def\bxac@ob@list{%
4683   \do{\let}\pdfoutput{\outputmode}%
4684   \do{\let}\pdfpagewidth{\pagewidth}%
4685   \do{\let}\pdfpageheight{\pageheight}%
4686   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
4687   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
4688 \def\bxac@ob@do#1#2{\begingroup
4689   \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
4690 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4691   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4692   \else \let#3#1\relax \let#1\@undefined
4693   \fi}
4694 \protected\def\ImposeOldLuaTeXBehavior{%
4695   \unless\ifbxac@in@old@behavior
4696     \bxac@in@old@behaviortrue
4697     \let\do\bxac@ob@do \bxac@ob@list
4698   \fi}
4699 \protected\def\RevokeOldLuaTeXBehavior{%
4700   \ifbxac@in@old@behavior
4701     \bxac@in@old@behaviorfalse
4702     \let\do\bxac@ob@do \bxac@ob@list
4703   \fi}
4704 \fi

```

漢字および完成形ハングルのカテゴリコードのムニヤムニヤ。

```

4705 \ifnum\luatexversion>64 \directlua{
4706   local function range(cs, ce, cc, ff)
4707     if ff or not tex.getcatcode(cs) == cc then
4708       local setcc = tex.setcatcode
4709       for c = cs, ce do setcc(c, cc) end
4710     end
4711   end
4712   range(0x3400, 0x4DB5, 11, false)
4713   range(0x4DB5, 0x4DBF, 11, true)
4714   range(0x4E00, 0x9FCC, 11, false)
4715   range(0x9FCD, 0x9FFF, 11, true)
4716   range(0xAC00, 0xD7A3, 11, false)
4717   range(0x20000, 0x2A6D6, 11, false)

```

```

4718     range(0x2A6D7, 0x2A6FF, 11, true)
4719     range(0x2A700, 0x2B734, 11, false)
4720     range(0x2B735, 0x2B73F, 11, true)
4721     range(0x2B740, 0x2B81D, 11, false)
4722     range(0x2B81E, 0x2B81F, 11, true)
4723     range(0x2B820, 0x2CEA1, 11, false)
4724     range(0x2CEA2, 0x2FFFD, 11, true)
4725 } \fi

```

以上。

```
4726 \fi
```

## G.4 完了

おしまい。

```
4727 %</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

```

4728 %<*cjkcat>
4729 \def\bxjx@pkgname{bxjscjkat}
4730 \newcount\bxjx@canta
4731 \@onlypreamble\bxjx@tmpdo
4732 \@onlypreamble\bxjx@tmpdo@a
4733 \@onlypreamble\bxjx@tmpdo@b

```

\bxjx@engine エンジンの種別。

```

4734 \let\bxjx@engine=n
4735 \def\bxjx@tmpdo#1#2{%
4736   \edef\bxjx@tmpa{\string#1}%
4737   \edef\bxjx@tmpb{\meaning#1}%
4738   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4739 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4740 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4741 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4742 \bxjx@tmpdo\pdfTeXversion{\let\bxjx@engine=p}
4743 \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。

```

4744 \def\bxjx@tmpdo#1#2{%
4745   \if#1\bxjx@engine
4746     \@ifpackageloaded{#2}{-}{%else

```

```

4747 \PackageError\bxjx@pkgname
4748 {Package '#2' must be loaded}%
4749 {Package loading is aborted.\MessageBreak\@ehc}%
4750 \endinput}
4751 \fi}
4752 \bxjx@tmpdo{p}{bxCJKatype}
4753 \bxjx@tmpdo{x}{xeCJK}
4754 \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。

```

4755 \ifx\TextOrMath\undefined
4756 \RequirePackage{fixltx2e}
4757 \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。

```

4758 \if u\bxjx@engine
4759 \@for\bxjx@tmpa:={%
4760 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4761 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4762 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4763 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4764 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4765 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4766 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4767 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4768 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4769 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4770 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4771 FE70,FFF0,%
4772 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4773 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4774 10480,104B0,10500,10530,10600,10800,10840,10860,%
4775 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4776 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4777 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4778 11200,11280,112B0,11300,11400,11480,11580,11600,%
4779 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4780 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4781 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4782 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4783 1EE00,1F000,1F030,1F0A0,1F300,1F600,1F650,1F680,%
4784 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4785 00C0%
4786 }\do{%

```

```

4787 \@tempcnta="\bxjx@tmpa\relax
4788 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4789 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4790 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4791 \fi

```

### H.3 ギリシャ・キリル文字の扱い

「特定 CJK 曖昧文字」について、和文・欧文扱いを制御できるようにする。ここで「特定 CJK 曖昧文字」とは以下に該当する文字の集合を指す：

- Unicode と JIS X 0213 に共通して含まれるギリシャ文字・キリル文字。
- Latin-1 の上位部分と JIS X 0208 に共通して含まれる文字（LuaT<sub>E</sub>X-j<sub>a</sub> 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。

```

4792 \@onlypreamble\bxjx@grkcyr@list
4793 \def\bxjx@grkcyr@list{%
4794 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4795 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4796 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4797 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4798 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4799 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4800 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
4801 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4802 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4803 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4804 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4805 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4806 \do{039D}{LGR}{\textNu}{N}%             % GR. C. L. NU
4807 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
4808 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
4809 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
4810 \do{03A1}{LGR}{\textRho}{P}%            % GR. C. L. RHO
4811 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4812 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
4813 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4814 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
4815 \do{03A7}{LGR}{\textChi}{X}%            % GR. C. L. CHI
4816 \do{03A8}{LGR}{\textPsi}{\Psi}%         % GR. C. L. PSI
4817 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA
4818 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
4819 \do{03B2}{LGR}{\textbeta}{\beta}%       % GR. S. L. BETA
4820 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA

```

4821	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4822	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4823	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4824	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4825	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4826	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4827	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4828	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4829	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4830	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4831	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4832	\do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
4833	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4834	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4835	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4836	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4837	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4838	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4839	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4840	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4841	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4842	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4843	\do{0401}{T2A}{\CYRYO}{}	% CY. C. L. IO
4844	\do{0410}{T2A}{\CYRA}{}	% CY. C. L. A
4845	\do{0411}{T2A}{\CYRB}{}	% CY. C. L. BE
4846	\do{0412}{T2A}{\CYRV}{}	% CY. C. L. VE
4847	\do{0413}{T2A}{\CYRG}{}	% CY. C. L. GHE
4848	\do{0414}{T2A}{\CYRD}{}	% CY. C. L. DE
4849	\do{0415}{T2A}{\CYRE}{}	% CY. C. L. IE
4850	\do{0416}{T2A}{\CYRZH}{}	% CY. C. L. ZHE
4851	\do{0417}{T2A}{\CYRZ}{}	% CY. C. L. ZE
4852	\do{0418}{T2A}{\CYRI}{}	% CY. C. L. I
4853	\do{0419}{T2A}{\CYRISHRT}{}	% CY. C. L. SHORT I
4854	\do{041A}{T2A}{\CYRK}{}	% CY. C. L. KA
4855	\do{041B}{T2A}{\CYRL}{}	% CY. C. L. EL
4856	\do{041C}{T2A}{\CYRM}{}	% CY. C. L. EM
4857	\do{041D}{T2A}{\CYRN}{}	% CY. C. L. EN
4858	\do{041E}{T2A}{\CYRO}{}	% CY. C. L. O
4859	\do{041F}{T2A}{\CYRP}{}	% CY. C. L. PE
4860	\do{0420}{T2A}{\CYRR}{}	% CY. C. L. ER
4861	\do{0421}{T2A}{\CYRS}{}	% CY. C. L. ES
4862	\do{0422}{T2A}{\CYRT}{}	% CY. C. L. TE
4863	\do{0423}{T2A}{\CYRU}{}	% CY. C. L. U
4864	\do{0424}{T2A}{\CYRF}{}	% CY. C. L. EF
4865	\do{0425}{T2A}{\CYRH}{}	% CY. C. L. HA
4866	\do{0426}{T2A}{\CYRC}{}	% CY. C. L. TSE
4867	\do{0427}{T2A}{\CYRCH}{}	% CY. C. L. CHE
4868	\do{0428}{T2A}{\CYRSH}{}	% CY. C. L. SHA
4869	\do{0429}{T2A}{\CYRSHCH}{}	% CY. C. L. SHCHA

4870 \do{042A}{T2A}{\CYRHRDSN}{}	% CY. C. L. HARD SIGN
4871 \do{042B}{T2A}{\CYRERY}{}	% CY. C. L. YERU
4872 \do{042C}{T2A}{\CYRSFTSN}{}	% CY. C. L. SOFT SIGN
4873 \do{042D}{T2A}{\CYREREV}{}	% CY. C. L. E
4874 \do{042E}{T2A}{\CYRYU}{}	% CY. C. L. YU
4875 \do{042F}{T2A}{\CYRYA}{}	% CY. C. L. YA
4876 \do{0430}{T2A}{\cyra}{}	% CY. S. L. A
4877 \do{0431}{T2A}{\cyrb}{}	% CY. S. L. BE
4878 \do{0432}{T2A}{\cyrv}{}	% CY. S. L. VE
4879 \do{0433}{T2A}{\cyrg}{}	% CY. S. L. GHE
4880 \do{0434}{T2A}{\cyrd}{}	% CY. S. L. DE
4881 \do{0435}{T2A}{\cyre}{}	% CY. S. L. IE
4882 \do{0436}{T2A}{\cyrrzh}{}	% CY. S. L. ZHE
4883 \do{0437}{T2A}{\cyrrz}{}	% CY. S. L. ZE
4884 \do{0438}{T2A}{\cyri}{}	% CY. S. L. I
4885 \do{0439}{T2A}{\cyrishrt}{}	% CY. S. L. SHORT I
4886 \do{043A}{T2A}{\cyrk}{}	% CY. S. L. KA
4887 \do{043B}{T2A}{\cyrl}{}	% CY. S. L. EL
4888 \do{043C}{T2A}{\cyrm}{}	% CY. S. L. EM
4889 \do{043D}{T2A}{\cyrn}{}	% CY. S. L. EN
4890 \do{043E}{T2A}{\cyro}{}	% CY. S. L. O
4891 \do{043F}{T2A}{\cyrrp}{}	% CY. S. L. PE
4892 \do{0440}{T2A}{\cyrr}{}	% CY. S. L. ER
4893 \do{0441}{T2A}{\cyrs}{}	% CY. S. L. ES
4894 \do{0442}{T2A}{\cyrt}{}	% CY. S. L. TE
4895 \do{0443}{T2A}{\cyru}{}	% CY. S. L. U
4896 \do{0444}{T2A}{\cyrf}{}	% CY. S. L. EF
4897 \do{0445}{T2A}{\cyrrh}{}	% CY. S. L. HA
4898 \do{0446}{T2A}{\cyrc}{}	% CY. S. L. TSE
4899 \do{0447}{T2A}{\cyrrch}{}	% CY. S. L. CHE
4900 \do{0448}{T2A}{\cyrrsh}{}	% CY. S. L. SHA
4901 \do{0449}{T2A}{\cyrrshch}{}	% CY. S. L. SHCHA
4902 \do{044A}{T2A}{\cyrrhdsn}{}	% CY. S. L. HARD SIGN
4903 \do{044B}{T2A}{\cyrrery}{}	% CY. S. L. YERU
4904 \do{044C}{T2A}{\cyrrsftsn}{}	% CY. S. L. SOFT SIGN
4905 \do{044D}{T2A}{\cyrrerev}{}	% CY. S. L. E
4906 \do{044E}{T2A}{\cyrryu}{}	% CY. S. L. YU
4907 \do{044F}{T2A}{\cyrrya}{}	% CY. S. L. YA
4908 \do{0451}{T2A}{\cyryo}{}	% CY. S. L. IO
4909 \do{00A7}{TS1}{\textsection}{\mathsection}%	SECTION SYMBOL
4910 \do{00A8}{TS1}{\textasciidieresis}{}	% DIAERESIS
4911 \do{00B0}{TS1}{\textdegree}{\mathdegree}%	% DEGREE SIGN
4912 \do{00B1}{TS1}{\textpm}{\pm}%	% PLUS-MINUS SIGN
4913 \do{00B4}{TS1}{\textasciicircum}{}	% ACUTE ACCENT
4914 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}%	PILCROW SIGN
4915 \do{00D7}{TS1}{\texttimes}{\times}%	% MULTIPLICATION SIGN
4916 \do{00F7}{TS1}{\textdiv}{\div}%	% DIVISION SIGN
4917 }	

`\mathdegree` 面倒なので補っておく。

```
4918 \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。

```
4919 \newif\ifbxjx@gcc@CJK
```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```
4920 \newcommand*\greekasCJK{%
```

```
4921   \bxjx@gcc@CJKtrue}
```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```
4922 \newcommand*\nogreekasCJK{%
```

```
4923   \bxjx@gcc@CJKfalse}
```

`\bxjx@fake@grk` `\bxjx@fake@grk{〈出力文字〉}{〈基準文字〉}` : ラテン文字で代用される数式ギリシャ文字の出力を行う。〈基準文字〉(`\mathchardef` の制御綴) の数式クラスと数式ファミリを引き継いで、〈出力文字〉(ASCII 文字トークン) の文字コードの数式文字を出力する。例えば、`\Pi` の意味が `\mathchar"7005` である場合、`\bxjx@fake@grk{B}{\Pi}` は `\mathchar"7042` を実行する。

※フォントパッケージ使用時の再定義を考慮して、〈基準文字〉が `\mathchardef` であるかを検査し、そうでない場合はフォールバックとして単に 〈出力文字〉 を実行する。

```
4924 \def\bxjx@tmpdo#1\relax{%
```

```
4925   \def\bxjx@fake@grk##1##2{%
```

```
4926     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
```

```
4927   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
```

```
4928     \ifx\##1\%
```

```
4929       \bxjx@canta##4\divide\bxjx@canta\@cclvi
```

```
4930       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
```

```
4931       \mathchar\bxjx@canta
```

```
4932     \else ##3\fi}
```

```
4933 }\expandafter\bxjx@tmpdo\string\mathchar\relax
```

## ■pdfTeX・upTeX の場合

```
4934 \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。

```
4935 \@ifpackageloaded{inputenc}{}{%else
```

```
4936   \RequirePackage[utf8]{inputenc}}
```

```
4937 \def\bxjx@tmpa{utf8}
```

```
4938 \ifx\bxjx@tmpa\inputencdoingname
```

```
4939   \PackageWarningNoLine\bxjx@pkgnam
```



```

4940   {Input encoding changed to utf8}%
4941   \inputencoding{utf8}%
4942 \fi

```

upTeX の場合に、「特定曖昧 CJK 文字」を含むブロックの和文カテゴリコードを変更する。

```

4943 \if u\bxjx@engine
4944 \kcatcode"0370=15
4945 \kcatcode"0400=15
4946 \kcatcode"0500=15
4947 \fi

```

各文字について \DeclareUnicodeCharacter を実行する。

```

4948 \def\bxjx@tmpdo#1{%
4949   \@tempcnta="#1\relax
4950   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4951 \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。

```

4952   \ifx\#5\%
4953     \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。

```

4954   \else\ifcat A\noexpand#5%
4955     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4956       {\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。

```

4957   \else \def\bxjx@tmpa{#5}%
4958   \fi\fi
4959   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
4960   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

4961 \if u\bxjx@engine
4962 \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。

```

4963   \kchardef#1=\@tempcnta
4964   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4965   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdf<sub>T</sub><sub>E</sub>X の場合も処理はほとんど同じ。ただし、和文用の定義として `\UTF{〈符号値〉}` を使う (`\UTF` は `bxCJKatype` の命令)。`\[bxjx@KC/...]` は使わないが定義済にする必要がある。

```
4966 \else\if p\bxjx@engine
4967 \def\bxjx@tmpdo@b#1#2#3#4#5{%
4968   \mathchardef#1=\@tempcnta
4969   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4970   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4971 \fi\fi
```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```
4972 \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。

```
4973 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4974 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4975 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4976 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4977   \count@=#1\relax
4978   \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4979     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4980   \else
4981     \wlog{ \space\space skipped defining Unicode char U+#1}%
4982   \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。

```
4983 \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。

```
4984   \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。

```
4985   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4986   \else \UseTextSymbol{#2}{#3}%
4987   \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。

```
4988 \begingroup
4989 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4990 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4991   \noexpand\bxjx@swap@DUC@cmd
4992   \the\toks@
4993   \noexpand\bxjx@swap@DUC@cmd}}
4994 \endgroup\next
4995 \def\bxjx@swap@DUC@cmd{%
4996   \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```

4997 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4998 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
4999 \let\bxjx@tmpa\relax}

```

以上。

### ■Xe<sub>La</sub>TeX・LuaTeX の場合

```

5000 \else\ifnum0\if x\bxjx@engine1\fi\if 1\bxjx@engine1\fi>0

```

各文字について、数式中の動作を定義する。

```

5001 \def\bxjx@tmpdo#1{%
5002   \bxjx@cmta="#1\relax
5003   \begingroup
5004     \lccode`~=\bxjx@cmta
5005   \lowercase{\endgroup
5006     \bxjx@tmpdo@a{~}}{#1}}
5007 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

〈数式 LICR〉が空なら何もしない。空でない場合、up<sub>La</sub>TeX の場合と同じ方法で“数式中の動作”を決定し、当該の文字を math active にしてその動作を設定する。

```

5008   \ifx\\#5\\\let\bxjx@tmpa\relax
5009   \else\ifcat A\noexpand#5%
5010     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5011       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
5012   \else \def\bxjx@tmpa{#5}%
5013   \fi\fi
5014   \ifx\bxjx@tmpa\relax\else
5015     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
5016   \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。

```

5017 \mathchardef\bxjx@tmpa="119
5018 \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。

```

5019 \if 1\bxjx@engine
5020   \protected\def\greekasCJK{%
5021     \bxjx@gcc@cjktrue
5022     \ltjsetparameter{jacharrange={+2, +8}}
5023   \protected\def\nogreekasCJK{%
5024     \bxjx@gcc@cjkfalse
5025     \ltjsetparameter{jacharrange={-2, -8}}
5026 \fi

```

Xe<sub>La</sub>TeX の場合、xeCJK は Xe<sub>La</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。

```

5027 \if x\bxjx@engine
5028   \let\bxjx@gcc@cjkc@list\@empty
5029   \def\do#1#2#3#4{%
5030     \edef\bxjx@gcc@cjkc@list{\bxjx@gcc@cjkc@list
5031       \noexpand\XeTeXcharclass"#1\bxjx@cncnta}}
5032   \bxjx@grkcyr@list
5033   \protected\def\greekasCJK{%
5034     \bxjx@gcc@cjkctrue
5035     \bxjx@cncnta=\@ne \bxjx@gcc@cjkc@list}
5036   \protected\def\nogreekasCJK{%
5037     \bxjx@gcc@cjkcfalse
5038     \bxjx@cncnta=\z@ \bxjx@gcc@cjkc@list}
5039 \fi

    以上。
5040 \fi\fi

```

## H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5041 \nogreekasCJK
```

## H.5 完了

おしまい。

```
5042 %</cjkc@>
```

## 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

```

5043 %<*ancpandoc>
5044 %% このファイルは日本語文字を含みます。
5045 \def\bxjsp@pkgname{bxjscjkc@}

\bxjsp@engine エンジンの種別。

5046 \let\bxjsp@engine=n
5047 \@onlypreamble\bxjsp@do
5048 \def\bxjsp@do#1#2{%
5049   \edef\bxjsp@tmpa{\string#1}%

```

```

5050 \edef\bxjsp@tmpb{\meaning#1}%
5051 \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
5052 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
5053 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
5054 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
5055 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}

```

## 1.2 パッケージオプション

`english` オプションが指定されている場合、`\ldots` の調整を抑止する。

※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  $\text{\LaTeX}$  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。

```

5056 \newif\ifbxjsp@english
5057 \DeclareOption{english}{\bxjsp@englishttrue}

```

オプション定義はおしまい。

```

5058 \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。

```

5059 \@onlypreamble\pandocSkipLoadFile
5060 \newcommand*\pandocSkipLoadFile[1]{%
5061   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5062 \def\bxjsp@skip@load@file@a#1#2{%
5063   \ifx#1\relax
5064     \def#1{2001/01/01}%
5065     \PackageInfo\bxjsp@pkgname
5066       {File '#2' marked as loaded\@gobble}%
5067   \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。

```

5068 \@onlypreamble\pandocSkipLoadPackage
5069 \newcommand*\pandocSkipLoadPackage[1]{%
5070   \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。

```

5071 \ifx\@IncludeInRelease\@undefined\else
5072   \pandocSkipLoadPackage{fixltx2e}
5073 \fi

```

## 1.5 cmap パッケージ

エンジンが (u)pL<sup>A</sup>T<sub>E</sub>X 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。)

```

5074 \if j\bxjps@engine
5075   \pandocSkipLoadPackage{cmap}
5076 \fi

```

## 1.6 microtype パッケージ

警告が多すぎなので消す。

```

5077 \if j\bxjps@engine \else
5078   \PassOptionsToPackage{verbose=silent}{microtype}
5079 \fi

```

エンジンが (u)pL<sup>A</sup>T<sub>E</sub>X 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。

```

5080 \if j\bxjps@engine
5081   \pandocSkipLoadPackage{microtype}
5082   \newcommand*\UseMicrotypeSet[2][]{ }
5083 \fi

```

## 1.7 Unicode 文字変換対策

Pandoc で L<sup>A</sup>T<sub>E</sub>X 形式に書き出す場合は、元データ中の一部の Unicode 文字を「L<sup>A</sup>T<sub>E</sub>X の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

```

…→\ldots{ }   ‘→`   ’→'   “→` `   ”→' '

```

日本語 L<sup>A</sup>T<sub>E</sub>X では「L<sup>A</sup>T<sub>E</sub>X 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。

```

5084 \DeclareRobustCommand{\pandocLdots}{%
5085   \relax\ifmmode \expandafter\bxjsp@org@ldots
5086   \else \expandafter\bxjsp@ja@ellipsis
5087   \fi}
5088 \def\bxjsp@ja@ellipsis{…}
5089 \let\bxjsp@org@ldots\ldots

```

`\ldots` の実装を置き換える。

※ Benglish オプション指定時は置き換えない。

```

5090 \AtBeginDocument{%
5091   \let\bxjsp@org@ldots\ldots
5092   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}

```

`\ldots` の直後の文字が非英字の場合、Pandoc は「`\ldots。`」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが  $\text{Xe}\text{L}\text{A}\text{TeX}$ ・ $\text{Lua}\text{TeX}$  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。

```

5093 \chardef\bxjsp@cc@other=12
5094 \@onlypreamble\bxjsp@makeother@range
5095 \def\bxjsp@makeother@range#1#2{%
5096   \@tempcnta"#1\relax \@tempcntb"#2\relax
5097   \loop\ifnum\@tempcnta<\@tempcntb
5098     \catcode\@tempcnta\bxjsp@cc@other
5099     \advance\@tempcnta\@ne
5100   \repeat}
5101 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
5102   \catcode"1F23B=\bxjsp@cc@other
5103   \bxjsp@makeother@range{9FCD}{A000}
5104   \bxjsp@makeother@range{1B002}{1B170}
5105   \bxjsp@makeother@range{2B820}{2EBF0}
5106 \fi

```

## I.8 Pandola モジュール

インストール済であれば読み込む。

```

5107 \IfFileExists{bxpandola.sty}{%
5108   \RequirePackage{bxpandola}\relax
5109   \PackageInfo\bxjsp@pkgname
5110   {PandoLa module is loaded\@gobble}
5111 }{}

```

## I.9 完了

おしまい。

```

5112 %</ancpandoc>

```

補助パッケージ実装はここまで。

5113 %</anc>