

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”)

v2.4 [2020/10/16]

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

目次

1	はじめに	3
2	オプション	10
3	和文フォントの変更	39
4	フォントサイズ	40
5	レイアウト	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	63
8	文書のマークアップ	66
8.1	表題	66
8.2	章・節	71
8.3	リスト環境	83
8.4	パラメータの設定	91
8.5	フロート	92
8.6	キャプション	94
9	フォントコマンド	95

10	相互参照	97
10.1	目次の類	97
10.2	参考文献	103
10.3	索引	104
10.4	脚注	106
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	113
14	初期設定	113
付録 A	和文ドライバの仕様 	118
付録 B	和文ドライバ : minimal 	119
B.1	補助マクロ	119
B.2	(u)pTeX 用の設定	121
B.3	pdfTeX 用の処理	126
B.4	X _Y TeX 用の処理	127
B.5	後処理 (エンジン共通)	128
付録 C	和文ドライバ : standard 	131
C.1	準備	131
C.2	和文ドライバパラメタ	131
C.3	共通処理 (1)	132
C.4	pTeX 用設定	140
C.5	pdfTeX 用設定 : CJK + bxcjkatype	143
C.6	X _Y TeX 用設定 : xeCJK + zxjatype	145
C.7	LuaTeX 用設定 : LuaTeX-ja	148
C.8	共通処理 (2)	151
付録 D	和文ドライバ : modern 	152
D.1	フォント設定	152
D.2	fixltx2e 読込	153
D.3	和文カテゴリコード	153
D.4	完了	153
付録 E	和文ドライバ : pandoc 	153
E.1	準備	154
E.2	和文ドライバパラメタ	154
E.3	dupload システム	155

E.4	lang 変数	156
E.5	geometry 変数	159
E.6	CJKmainfont 変数	159
E.7	Option clash 対策	159
E.8	レイアウト上書き禁止	160
E.9	paragraph のマーク	161
E.10	全角空白文字	161
E.11	hyperref 対策	161
E.12	Pandoc 要素に対する和文用の補正	162
E.13	完了	163
付録 F	補助パッケージ一覧 罫	163
付録 G	補助パッケージ : bxjscompat 罫	163
G.1	準備	163
G.2	X _Y TeX 部分	164
G.3	LuaTeX 部分	165
G.4	完了	166
付録 H	補助パッケージ : bxjscjkat 罫	166
H.1	準備	166
H.2	和文カテゴリコードの設定	167
H.3	ギリシャ・キリル文字の扱い	168
H.4	初期設定	175
H.5	完了	175
付録 I	補助パッケージ : bxjspandoc 罫	175
I.1	準備	175
I.2	パッケージオプション	176
I.3	パッケージ読込の阻止	176
I.4	fixltx2e パッケージ	177
I.5	cmap パッケージ	177
I.6	microtype パッケージ	177
I.7	Unicode 文字変換対策	177
I.8	PandoLa モジュール	178
I.9	完了	179

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なものである。

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

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

これは L^AT_EX 3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいてもともと奥村晴彦により作成されたものです。現在は日本語 T_EX 開発コミュニティにより GitHub で管理されています。

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

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

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

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

ここでは次のドキュメントクラス（スタイルファイル）を作ります。

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

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

以下では実際のコードに即して説明します。

minijs は、jsclasses に似た設定を行うパッケージです。

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

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

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

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

```
16 \newif\ifjsc@needsp@tch
17 \jsc@needsp@tchfalse
```

■BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため `calc` を読み込む。

```
18 \RequirePackage{calc}
```

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

```
19 \RequirePackage{keyval}
```

クラスの本体ではこの他に `geometry` パッケージが読み込まれる。

TODO: 依存パッケージの情報。

互換性のための補助パッケージを読み込む。

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

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

```
24 \let\jsArticle=a
25 \let\jsBook=b
26 \let\jsReport=r
27 \let\jsSlide=s
28 %<article>\let\jsDocClass\jsArticle
29 %<book>\let\jsDocClass\jsBook
30 %<report>\let\jsDocClass\jsReport
31 %<slide>\let\jsDocClass\jsSlide
```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

非サポートのエンジンの場合は強制終了させる。

※ NTT `jTeX` と `Omega` 系。

```

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

```

`LuaTeX` の場合、本クラス用の `Lua` モジュールを用意する。

```

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

```

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

```

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

```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

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

`\ifjsInPdfMode` [スイッチ] pdf_{TEX} / Lua_{TEX} が PDF モードで動作しているか。

※ Lua_{TEX} 0.8x 版でのプリミティブ名変更に対応。

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

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

※ 2017 年 1 月以降の L_AT_EX カーネルにおいて「Unicode を表す L_AT_EX 公式のフォントエンコーディング」である“TU”が導入され、これ以降の L_AT_EX を X_YL_AT_EX または Lua_{TEX} 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。

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

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

※カーネルの 2020/10/01 版で導入された。

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

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

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

```
88 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
89   #1\expandafter\@firstoftwo
90   \else\expandafter\@secondoftwo
91   \fi}
```

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

```
92 \def\bxjs@cslet#1{%
93   \expandafter\let\csname#1\endcsname}
```

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

```
94 \def\bxjs@csletcs#1#2{%
95   \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。

```
96 \def\bxjs@catopt#1#2{%
97   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}
```

`\bxjs@ifplus` `\@ifstar` の + 版。

```
98 \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。

```
99 \let\c@bxjs@tempcnta\@tempcnta
100 \def\bxjs@gset@tempcnta{\setcounter{bxjs@tempcnta}}
```

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

```
101 \def\jsSetQHLlength#1#2{%
102   \begingroup
103     \bxjs@parse@qh{#2}%
104     \ifx\bxjs@tmpb\relax
105       \setlength\@tempdima{#2}%
106       \xdef\bxjs@g@tmpa{\the\@tempdima}%
107     \else \global\let\bxjs@g@tmpa\bxjs@tmpb
108     \fi
109   \endgroup
110   #1=\bxjs@g@tmpa\relax}
```

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

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

```
111 \if j\jsEngine \def\bxjs@parse@qh@units{zw,zh}
112 \else \def\bxjs@parse@qh@units{trueQ,trueH,Q,H,zw,zh}
113 \fi
114 \def\bxjs@parse@qh#1{%
115   \let\bxjs@tmpb\relax
116   \@for\bxjs@tmpa:=\bxjs@parse@qh@units\do{%
117     \ifx\bxjs@tmpb\relax
118       \edef\bxjs@next{{\bxjs@tmpa}{#1}}%
119       \expandafter\bxjs@parse@qh@a\cname bxjs@unit@\bxjs@tmpa\expandafter
120       \endcsname\bxjs@next
121     \fi}}
122 \def\bxjs@parse@qh@a#1#2#3{%
123   \def\bxjs@next##1#2\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}#1}%
124   \bxjs@next#3\@nil#2\@nil\@nnil}
125 \def\bxjs@parse@qh@b#1#2#3{%
126   \ifx\@nnil#2\@nnil\else
```



```

127 \ifx#3\relax
128 \ClassError\bxjs@clsname
129 {You cannot use '\bxjs@tmpa' here}{\@ehc}%
130 \def\bxjs@tmpb{Opt}%
131 \else
132 \@tempdimb#3\relax \@tempdimb#1\@tempdimb
133 \edef\bxjs@tmpb{\the\@tempdimb}%
134 \fi
135 \fi}

```

今の段階では Q/H だけが使用可能。

```

136 \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。

```

137 \@onlypreamble\bxjs@begin@document@hook
138 \let\bxjs@begin@document@hook\@empty
139 \AtBeginDocument{\bxjs@begin@document@hook}

```

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

```

140 \@onlypreamble\bxjs@post@option@hook
141 \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。

```

142 \@onlypreamble\bxjs@pre@jadriver@hook
143 \let\bxjs@pre@jadriver@hook\@empty

```

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

```

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

```

一時的な手続き用の制御綴。

```

146 \@onlypreamble\bxjs@tmpdo
147 \@onlypreamble\bxjs@tmpdo@a
148 \@onlypreamble\bxjs@tmpdo@b
149 \@onlypreamble\bxjs@tmpdo@c
150 \@onlypreamble\bxjs@tmpdo@d

```

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

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

```

151 \if 1\jsEngine
152 \def\bxjs@tmpdo#1{%
153 \xdef\bxjs@pre@jadriver@hook{%
154 \bxjs@pre@jadriver@hook
155 \catcode`#1=\the\catcode`#1\relax}%
156 \catcode`#1=11\relax}
157 \@tfor\bxjs@tmpa:=和西曆\do

```

```

158 {\expandafter\bxjs@tmpdo\bxjs@tmpa}
159 \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。

```

160 \bxjs@robust@def\jsInhibitGlue{%
161   \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。

```

162 \if@compatibility
163   \ClassError\bxjs@clsname
164   {Something went chaotic!\MessageBreak
165     (How come '\string\documentstyle' is there?)\MessageBreak
166     I cannot go a single step further...}
167   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
168     then there'll still be hope....}
169   \expandafter\@firstofone
170 \else \expandafter\@gobble
171 \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。

```
172 \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。

```
173 \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。

```
174 %<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。

```
175 %<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。

```
176 %<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。

```
177 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

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

```
178 \@onlypreamble\bxjs@setpaper
179 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
180 \newif\ifbxjs@iso@bsize
181 \DeclareOption{iso-bsize}{\bxjs@iso@bsizetrue}
182 \@onlypreamble\bxjs@setpaper@bsize
183 \def\bxjs@setpaper@bsize#1{\edef\bxjs@param@paper{%
184   b#1\ifbxjs@iso@bsize paper\else jfi}}
185 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
186 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
187 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
188 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
189 \DeclareOption{b4paper}{\bxjs@setpaper@bsize{4}}
190 \DeclareOption{b5paper}{\bxjs@setpaper@bsize{5}}
191 \DeclareOption{b6paper}{\bxjs@setpaper@bsize{6}}
192 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
193 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
194 \DeclareOption{b4j}{\bxjs@setpaper{b4j}}
195 \DeclareOption{b5j}{\bxjs@setpaper{b5j}}
196 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
197 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
198 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
199 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
200 \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。

```
201 \@for\bxjs@tmpa:={%
202   a0,a1,a2,c0,c1,c2,c3,c4,c5,c6,ansia,ansib,ansic,ansid,ansie%
203 }\do{\edef\bxjs@next{%
204   \noexpand\DeclareOption{\bxjs@tmpa paper}%
205   {\noexpand\bxjs@setpaper{\bxjs@tmpa paper}}%
206 }\bxjs@next}
207 \DeclareOption{screen}{\bxjs@setpaper{screen}}
```

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

```
208 \@for\bxjs@tmpa:={0,1,2,3}\do{\edef\bxjs@next{%
209   \noexpand\DeclareOption{b\bxjs@tmpa paper}%
210   {\noexpand\bxjs@setpaper@bsize{\bxjs@tmpa}}%
211 }\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。

```
212 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
213 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
214 \DeclareOption{screenpaper}{\bxjs@setpaper{screen}}
215 \DeclareOption{custompaper}{}
```

■横置き 用紙の縦と横の長さを入れ換えます。

```
216 \newif\if@landscape
217 \@landscapefalse
218 \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。

```
219 \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。

```
220 %<!slide>\@slidefalse
221 %<slide>\@slidetrue
```

■サイズオプション 10pt, 11pt, 12pt のほかに、8pt, 9pt, 14pt, 17pt, 21pt, 25pt, 30pt, 36pt, 43pt を追加しました。これは等比数列になるように選んだものです（従来の

20pt も残しました)。`\@ptsize` の定義が変だったのでご迷惑をおかけしましたが、標準的なドキュメントクラスと同様にポイント数から 10 を引いたものに直しました。

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

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

```
222 \newcommand{\@ptsize}{0}
223 \def\bxjs@param@basefontsize{10pt}
224 \def\jsUnusualPtSize{-20}
```

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

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

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

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

```
226 \jsSetQHLength\@tempdima{#1}%
227 \edef\bxjs@param@basefontsize{\the\@tempdima}%
228 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
229 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
230 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
231 \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。

```
232 \def\bxjs@setjbasefontsize#1{%
233 \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。

```
234 \newif\ifjsc@mag
235 \newif\ifjsc@mag@xreal
236 %\let\jsc@magscale\@undefined

237 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
238 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
239 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
240 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
241 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
242 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
243 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
```

```

244 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
245 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
246 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
247 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
248 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
249 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
250 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
251 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
252 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
253 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
254 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
255 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

256 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@@usemag}
257 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@@nomag}
258 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@@xreal}

```

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

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

```

259 \if j\jsEngine
260 \hour\time \divide\hour by 60\relax
261 \@tempcnta\hour \multiply\@tempcnta 60\relax
262 \minute\time \advance\minute-\@tempcnta
263 \DeclareOption{tombow}{%
264   \tombowtrue \tombowdatetrue
265   \setlength{\@tombowwidth}{.1\p@}%
266   \@bannertoken{%
267     \jobname\space(\number\year-\two@digits\month-\two@digits\day
268     \space\two@digits\hour:\two@digits\minute)}}%
269   \maketombowbox}
270 \DeclareOption{tombo}{%
271   \tombowtrue \tombowdatefalse
272   \setlength{\@tombowwidth}{.1\p@}%
273   \maketombowbox}
274 \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。

```

275 \if j\jsEngine

```

```

276 \DeclareOption{mentuke}{%
277   \tombowtrue \tombowdatefalse
278   \setlength{\@tombowwidth}{\z@}%
279   \maketombowbox}
280 \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。

```

281 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
282 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
283 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 `twocolumn` で二段組になります。

```

284 \DeclareOption{onecolumn}{\@twocolumnfalse}
285 \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。

```

286 \DeclareOption{titlepage}{\@titlepagetrue}
287 \DeclareOption{notitlepage}{\@titlepagefalse}

```

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

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

```

288 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
289 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
290 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}

```

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

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

```

291 \def\eqnarray{%
292   \stepcounter{equation}%
293   \def\@currentlabel{\p@equation\theequation}%
294   \global\@eqnswtrue
295   \m@th
296   \global\@eqcnt\z@
297   \tabskip\@centering
298   \let\\\@eqnocr
299   $$\everycr{}\halign to\displaywidth\bgroup
300     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1

```

```

301      &\global\@eqcnt\@ne \hfil$\displaystyle{##{}}$\hfil
302      &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
303      &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
304      \tabskip\z@skip
305      \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。

```

306 \DeclareOption{leqno}{\input{leqno.clo}}
307 \DeclareOption{fleqn}{\input{fleqn.clo}}
308 % fleqn 用の eqnarray 環境の再定義
309 \def\eqnarray{%
310   \stepcounter{equation}%
311   \def\@currentlabel{\p@equation\theequation}%
312   \global\@eqnswtrue\m@th
313   \global\@eqcnt\z@
314   \tabskip\mathindent
315   \let\@=\@eqncr
316   \setlength\abovedisplayskip{\topsep}%
317   \ifvmode
318     \addtolength\abovedisplayskip{\partopsep}%
319   \fi
320   \addtolength\abovedisplayskip{\parskip}%
321   \setlength\belowdisplayskip{\abovedisplayskip}%
322   \setlength\belowdisplayshortskip{\abovedisplayskip}%
323   \setlength\abovedisplayshortskip{\abovedisplayskip}%
324   $$\everycr{}\halign to\linewidth% $$
325   \bgroup
326   \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnsele
327   &\global\@eqcnt\@ne \hfil$\displaystyle{##{}}$\hfil
328   &\global\@eqcnt\tw@
329   $\displaystyle{##}$\hfil \tabskip\@centering
330   &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
331   \tabskip\z@skip\cr
332   }}

```

■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。

```

333 % \DeclareOption{openbib}{%
334 %   \AtEndOfPackage{%
335 %     \renewcommand\@openbib@code{%
336 %       \advance\leftmargin\bibindent
337 %       \itemindent -\bibindent
338 %       \listparindent \itemindent
339 %       \parsep \z@}%
340 %     \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。AMSTeX 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：

```
341 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。

342 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
343 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
344 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
345 \define@key{bxjs}{enablejfam}[true]{%
346   \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。

```
347 \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。

```
348 \newif\ifjsDraft
349 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
350 \DeclareOption{final}{\jsDraftfalse \overfullrule=0pt }
```

■和文フォントメトリックの選択 このクラスファイルでは、和文 TFM として東京書籍印刷の小林肇さんの作られた JIS フォントメトリック (jis, jisg) を標準で使うことにしますが、従来の min10, goth10 などを使いたいときは mingoth というオプションを指定します。また、winjis オプションで winjis メトリック (OTF パッケージと同じ psitau さん

作；ソースに書かれた Windows の機種依存文字が dvips, dvipdfmx などでも出力出来るようになる) が使えます。

[2018-02-04] winjis オプションはコ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。

```
351 \newif\ifbxjs@papersize
352 \bxjs@papersizetrue
353 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
354 \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。

```
355 \newif\if@english
356 \@englishfalse
357 \DeclareOption{english}{\@englishtrue}
```

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

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

BXJS では当初から bxjsreport クラスが用意されている。

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

BXJS クラスでは、nojslogo を既定とする。

```

358 \newif\if@jslogo \@jslogofalse
359 \DeclareOption{jslogo}{\@jslogotrue}
360 \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 128pt 以上の場合でも動作するように修正した。

```

361 \mathchardef\bxjs@isc@ll=128
362 \mathchardef\bxjs@isc@sl=259
363 \def\bxjs@isc@sl@h{65539 }
364 \def\bxjs@invscale#1#2{%
365   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
366   \ifdim\@tempdima<\bxjs@isc@ll\p@
367     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
368     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
369   \else
370     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
371     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
372   \fi
373   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
374   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
375   \@tempdimb\@tempcnta\@ne
376   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
377   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
378   \@whiledim\@tempdimb<\@tempdimc\do{%
379     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
380     \advance\@tempcntb\@ne \divide\@tempcntb\@tw@
381     \ifdim #2\@tempcntb>\@tempdima
382       \advance\@tempcntb\@ne \@tempdimc=\@tempcntb\@ne
383     \else \@tempdimb=\@tempcntb\@ne \fi}%
384   \xdef\bxjs@gtmpa{\the\@tempdimb}%
385   \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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 用の既定テンプレートをを用いて他形式から $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ (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。

```

386 \DeclareOption{pandoc}{%
387   \bxjs@apply@pandoc@opt}
388 \@onlypreamble\bxjs@apply@pandoc@opt

```

```
389 \def\bxjs@apply@pandoc@opt{%
```

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

※実際の和文ドライバ・エンジン設定より優先される。

```
390 \g@addto@macro\bxjs@post@option@hook{%
391   \bxjs@oldfontcommandstrue
392   \setkeys{bxjs}{ja=pandoc}%
393   \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）。

```
394 \ifx\bxjs@driver@opt\@undefined
395   \def\bxjs@driver@opt{dvipdfmx}%
396   \bxjs@dvi@opttrue
397 \fi
398 \global\let\bxjs@apply@pandoc@opt\relax}
```

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

```
399 \DeclareOption{pandoc+}{%
400   \g@addto@macro\bxjs@post@option@hook{%
401     \edef\jsJaParam{\bxjs@catopt{_plus}\jsJaParam}}%
402   \ExecuteOptions{pandoc}}
```

■エンジン・ドライバオプション

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

```
403 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
404 %\let\bxjs@engine@opt\@undefined
```

エンジン明示指定のオプションの処理。

※ 0.9pre 版の暫定仕様と異なり、エンジン名は `...latex` に限定する。`xetex` や `pdftex` は一般的な L^AT_EX の慣習に従って「ドライバの指定」とみなすべきだから。

```
405 \DeclareOption{autodetect-engine}{%
406   \let\bxjs@engine@given=}%
407 \DeclareOption{latex}{%
408   \def\bxjs@engine@opt{latex}%
409   \let\bxjs@engine@given=n}%
410 \DeclareOption{platex}{%
411   \def\bxjs@engine@opt{platex}%
412   \let\bxjs@engine@given=j}%
413 \DeclareOption{uplatex}{%
414   \def\bxjs@engine@opt{uplatex}%
415   \let\bxjs@engine@given=u}
```

```

416 \DeclareOption{xelatex}{%
417   \def\bxjs@engine@opt{xelatex}%
418   \let\bxjs@engine@given=x}
419 \DeclareOption{pdflatex}{%
420   \def\bxjs@engine@opt{pdflatex}%
421   \let\bxjs@engine@given=p}
422 \DeclareOption{lualatex}{%
423   \def\bxjs@engine@opt{lualatex}%
424   \let\bxjs@engine@given=l}
425 \DeclareOption{platex-ng}{%
426   \def\bxjs@engine@opt{platex-ng}%
427   \let\bxjs@engine@given=g}
428 \DeclareOption{platex-ng*}{%
429   \def\bxjs@engine@opt{platex-ng*}%
430   \let\bxjs@platexng@nodrv=t%
431   \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。

```

432 %\let\bxjs@driver@given\@undefined
433 \let\bxjs@driver@@dvimode=0
434 \let\bxjs@driver@@dvipdfmx=1
435 \let\bxjs@driver@@pdfmode=2
436 \let\bxjs@driver@@xetex=3
437 \let\bxjs@driver@@dvips=4
438 \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。

```

439 %\let\bxjs@driver@opt\@undefined

440 \DeclareOption{dvips}{%
441   \def\bxjs@driver@opt{dvips}%
442   \let\bxjs@driver@given\bxjs@driver@@dvips}
443 \DeclareOption{dviout}{%
444   \def\bxjs@driver@opt{dviout}%
445   \let\bxjs@driver@given\bxjs@driver@@dvimode}
446 \DeclareOption{xdvi}{%
447   \def\bxjs@driver@opt{xdvi}%
448   \let\bxjs@driver@given\bxjs@driver@@dvimode}
449 \DeclareOption{dvipdfmx}{%
450   \def\bxjs@driver@opt{dvipdfmx}%
451   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
452 \DeclareOption{nodvidriver}{%
453   \def\bxjs@driver@opt{nodvidriver}%
454   \let\bxjs@driver@given\bxjs@driver@@none}
455 \DeclareOption{pdftex}{%
456   \def\bxjs@driver@opt{pdftex}%
457   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
458 \DeclareOption{luatex}{%
459   \def\bxjs@driver@opt{luatex}%

```

```

460 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
461 \DeclareOption{xetex}{%
462 \def\bxjs@driver@opt{xetex}%
463 \let\bxjs@driver@given\bxjs@driver@@xetex}

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

```

■その他の BXJS 独自オプション

TODO: 互換用オプションを分離する (2.0 版で?)。

`\bxjs@depre@opt` 非推奨のオプションについて警告を出す。

```

\bxjs@depre@opt@do 465 \@onlypreamble\bxjs@depre@opt
466 \def\bxjs@depre@opt#1#2{%
467 \ClassWarningNoLine\bxjs@clsname
468 {The old option '#1' is DEPRECATED\MessageBreak
469 and may be abolished in future!\MessageBreak
470 You should instead write:\MessageBreak
471 \space\space #2}}
472 \@onlypreamble\bxjs@depre@opt@do
473 \def\bxjs@depre@opt@do#1#2{%
474 \bxjs@depre@opt{#1}{#2}%
475 \setkeys{bxjs}{#2}}

```

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

※ 2.0 版より、既定値を常に真とする。

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

`nobigcode` / `bigcode` オプションの定義。

```

477 \DeclareOption{nobigcode}{%
478 \bxjs@bigcodefalse}
479 \DeclareOption{bigcode}{%
480 \bxjs@bigcodetrue}

```

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

```
481 \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。

```

482 \DeclareOption{nooldfontcommands}{%
483 \bxjs@oldfontcommandsfalse}

```

```

484 \DeclareOption{oldfontcommands}{%
485   \bxjs@oldfontcommandstrue}


```

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

```

486 \DeclareOption{winjis}{%
487   \ClassWarningNoLine\bxjs@clsname
488   {This class does not support `winjis' option}%
489 }
490 \DeclareOption{mingoth}{%
491   \ClassWarningNoLine\bxjs@clsname
492   {This class does not support `mingoth' option}%
493 }
494 \DeclareOption{jis}{%
495   \ClassWarningNoLine\bxjs@clsname
496   {This class does not support `jis' option}%
497 }

```

■`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。

```

498 \DeclareOption*{%
499   \def\bxjs@next{\bxjs@safe@setkeys\bxjs}%
500   \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。

```

501 \def\bxjs@safe@setkeys#1#2{%
502   \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
503   \setkeys{#1}{#2}%
504   \let\KV@errx\bxjs@save@KV@errx}

```

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

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

```

505 \@onlypreamble\bxjs@declare@enum@option
506 \def\bxjs@declare@enum@option#1#2#3{%
507   \bxjs@csletcs{bxjs@#2}{bxjs@#2@@#3}%
508   \define@key{bxjs}{#1}{%
509     \expandafter\ifx\csname bxjs@#2@@##1\endcsname\relax
510     \bxjs@error@keyval{#1}{##1}%
511     \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@@##1}%
512     \fi}}

```

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

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

```

513 \@onlypreamble\bxjs@declare@bool@option
514 \def\bxjs@declare@bool@option#1#2#3{%
515   \csname newif\expandafter\endcsname\csname ifbxjs@#2\endcsname
516   \@nameuse{bxjs@#2#3}%
517   \define@key{bxjs}{#1}[true]{%
518     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
519       \bxjs@error@keyval{#1}{##1}%
520     \else \@nameuse{bxjs@#2##1}%
521     \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。
522 \def\bxjs@set@keyval#1#2#3{%
523   \bxjs@csletcs{bxjs@next}{bxjs@kv@#1@#2}%
524   \ifx\bxjs@next\relax
525     \bxjs@error@keyval{#1}{#2}%
526     #3%
527   \else \bxjs@next
528   \fi}
529 \@onlypreamble\bxjs@error@keyval
530 \def\bxjs@error@keyval#1#2{%
531   \ClassError\bxjs@clsname
532   {Invalid value '#2' for option #1}\@ehc}

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

\bxjs@base@opt 明示された base オプションの値。
534 %\let\bxjs@base@opt\@undefined

    base オプションの処理。
535 \define@key{bxjs}{base}{%
536   \edef\bxjs@base@opt{#1}%
537   \bxjs@setbasefontsize{#1}}
538 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

\bxjs@jbase@opt 明示された jbase オプションの値。
539 %\let\bxjs@jbase@opt\@undefined

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

\bxjs@scale@opt 明示された scale オプションの値。
542 %\let\bxjs@scale@opt\@undefined

    scale オプションの処理。
543 \define@key{bxjs}{scale}{%
544   \edef\bxjs@scale@opt{#1}%

```



```

545 \let\jsScale\bxjs@scale@opt}
546 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}

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

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

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

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

\bxjs@jadriver 和文ドライバの名前。
551 \let\bxjs@jadriver\relax
552 %\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)。
553 \define@key{bxjs}{jadriver}{%
554 \bxjs@depre@opt{jadriver}{ja=#1}\edef\bxjs@jadriver@opt{#1}}
555 \define@key{bxjs}{ja}{\relax}%
556 \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}

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

    jafont オプションの処理。
558 \define@key{bxjs}{jafont}{\edef\jsJaFont{#1}}

\jsJaParam 和文ドライバパラメタの文字列。
559 \let\jsJaParam\@empty

    japaram オプションの処理。
560 \define@key{bxjs}{japaram}{%
561 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}

    引数をもつ pandoc・pandoc+ オプションは、その引数を和文パラメタの指定と見なす。
562 \define@key{bxjs}{pandoc}[]{%
563 \ExecuteOptions{pandoc}%
564 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}
565 \define@key{bxjs}{pandoc+}[]{%
566 \ExecuteOptions{pandoc+}%
567 \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}}

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

```

```

568 \let\bxjs@magstyle@@mag=m
569 \let\bxjs@magstyle@@real=r
570 \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。

```

571 \let\bxjs@magstyle@@usemag\bxjs@magstyle@@mag
572 \let\bxjs@magstyle@@nomag\bxjs@magstyle@@real
573 \bxjs@cslet{bxjs@magstyle@@nomag*}\bxjs@magstyle@@xreal

```

\bxjs@magstyle@@default は既定の値を表す。

```

574 \let\bxjs@magstyle@@default\bxjs@magstyle@@usemag
575 \ifx l\jsEngine \ifnum\luatexversion>86
576   \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
577 \fi\fi
578 \ifjsWithpTeXng
579   \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
580 \fi
581 \let\bxjs@magstyle\bxjs@magstyle@@default

```

magstyle オプションの処理。

```

582 \define@key{bxjs}{magstyle}{%
583   \bxjs@csletcs{bxjs@magstyle}{bxjs@magstyle@@#1}%
584   \ifx\bxjs@magstyle\relax
585     \bxjs@error@keyval{magstyle}{#1}%
586   \let\bxjs@magstyle\bxjs@magstyle@@default
587 \fi}

```

\bxjs@geometry geometry オプションの指定値。

```

588 \let\bxjs@geometry@@class=c
589 \let\bxjs@geometry@@user=u
590 \bxjs@declare@enum@option{geometry}{geometry}{class}

```

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

```

591 \bxjs@declare@bool@option{fancyhdr}{fancyhdr}{true}

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

592 \newif\ifbxjs@dvi@opt

```

DVI モードのドライバとドライバ種別との対応。

```

593 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
594 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
595 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
596 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
597 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

```

dvi オプションの処理。

```

598 \define@key{bxjs}{dvi}{%

```

```

599 \bxjs@csletcs{bxjs@tmpa}{bxjs@dvidriver@@#1}%
600 \ifx\bxjs@tmpa\relax
601 \bxjs@error@keyval{dvi}{#1}%
602 \else

\bxjs@driver@given を未定義にしていることに注意。

603 \def\bxjs@driver@opt{#1}%
604 \let\bxjs@driver@given\undefined
605 \bxjs@dvi@opttrue
606 \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。
607 \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。
608 \newif\ifbxjs@force@chapterabstract
609 %<book>\bxjs@force@chapterabstracttrue

layout オプションの処理。
610 \@namedef{bxjs@kv@layout@v1}{%
611 %<book>\bxjs@layout@buggyhmargintrue
612 %<report>\bxjs@force@chapterabstracttrue
613 }
614 \@namedef{bxjs@kv@layout@v2}{%
615 %<book>\bxjs@layout@buggyhmarginfalse
616 %<report>\bxjs@force@chapterabstractfalse
617 }
618 \define@key{bxjs}{layout}{%
619 \bxjs@set@keyval{layout}{#1}{}}

\bxjs@textwidth@limit textwidth-limit の指定値。
620 %\let\bxjs@textwidth@limit@opt\undefined
621 \define@key{bxjs}{textwidth-limit}{%
622 \bxjs@depre@opt{textwidth-limit}{textwidth=#1zw}%
623 \edef\bxjs@textwidth@limit@opt{#1}}

\bxjs@textwidth@opt textwidth の指定値。
624 %\let\bxjs@textwidth@opt\undefined
625 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
626 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}

\bxjs@number@of@lines@opt number-of-lines の指定値。
627 %\let\bxjs@number@of@lines@opt\undefined
628 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
629 \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。

```

```

630 %\let\bxjs@paragraph@mark\@undefined
631 \define@key{bxjs}{paragraph-mark}{%
632   \edef\bxjs@paragraph@mark{#1}}

\ifbxjs@whole@zw@lines 〔スイッチ〕 whole-zw-lines の指定値。
633 \bxjs@declare@bool@option{whole-zw-lines}{whole@zw@lines}{true}

\ifbxjs@jaspace@cmd 〔スイッチ〕 jspace-cmd の指定値。
634 \bxjs@declare@bool@option{jaspace-cmd}{jaspace@cmd}{true}
635 \define@key{bxjs}{xkanjiskip-cmd}[true]{\setkeys{bxjs}{jaspace-cmd=#1}}

\ifbxjs@fix@at@cmd 〔スイッチ〕 fix-at-cmd の指定値。
636 \bxjs@declare@bool@option{fix-at-cmd}{fix@at@cmd}{true}

\ifbxjs@hyperref@enc 〔スイッチ〕 hyperref-enc の指定値。
637 \bxjs@declare@bool@option{hyperref-enc}{hyperref@enc}{true}

\bxjs@everyparhook  everyparhook の指定値。
638 \chardef\bxjs@everyparhook@@none=0
639 \chardef\bxjs@everyparhook@@compat=1
640 \chardef\bxjs@everyparhook@@modern=2
641 \bxjs@declare@enum@option{everyparhook}{everyparhook}{%
642   \if j\jsEngine compat\else modern\fi}

\bxjs@label@section  label-section の指定値。
643 \chardef\bxjs@label@section@@none=0
644 \chardef\bxjs@label@section@@compat=1
645 \chardef\bxjs@label@section@@modern=2
646 \bxjs@declare@enum@option{label-section}{label@section}{compat}

\ifbxjs@usezw 〔スイッチ〕 use-zw の指定値。
647 \bxjs@declare@bool@option{use-zw}{usezw}{true}
648 \DeclareOption{noz}{\bxjs@depre@opt@do{noz}{use-zw=false}}
649 \DeclareOption{zw}{\bxjs@depre@opt@do{zw}{use-zw=true}}

\ifbxjs@disguise@js 〔スイッチ〕 disguise-js の指定値。
650 \bxjs@declare@bool@option{disguise-js}{disguise@js}{true}
651 \DeclareOption{nojs}{\bxjs@depre@opt@do{nojs}{disguise-js=false}}
652 \DeclareOption{js}{\bxjs@depre@opt@do{js}{disguise-js=true}}

\ifbxjs@precisetext 〔スイッチ〕 precise-text の指定値。
653 \bxjs@declare@bool@option{precise-text}{precisetext}{false}
654 \DeclareOption{noprecisetext}{\bxjs@depre@opt@do{noprecisetext}{precise-
  text=false}}
655 \DeclareOption{precisetext}{\bxjs@depre@opt@do{precisetext}{precise-
  text=true}}

\ifbxjs@simplejasetup 〔スイッチ〕 simple-ja-setup の指定値。
656 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}{true}

```

```

657 \DeclareOption{nosimplejasetup}{\bxjs@depre@opt@do{nosimplejasetup}{simple-jasetup=false}}
658 \DeclareOption{simplejasetup}{\bxjs@depre@opt@do{simplejasetup}{simple-jasetup=true}}

```

■ オプションの実行

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

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

```

659 \let\bxjs@org@removeelement\@removeelement
660 \def\@removeelement#1#2#3{%
661   \def\reserved@a{#2}%
662   \ifx\reserved@a\@empty \let#3\@empty
663   \else \bxjs@org@removeelement{#1}{#2}{#3}%
664   \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。

```

665 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
666 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
667 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
668 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
669 \ProcessOptions\relax
670 \bxjs@post@option@hook

```

後処理

```

671 \if@slide
672   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
673 \fi
674 \if@landscape
675   \setlength\@tempdima {\paperheight}
676   \setlength\paperheight{\paperwidth}
677   \setlength\paperwidth {\@tempdima}
678 \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。

```

679 \def\bxjs@tmpdo{%

```

```

680 \def\bxjs@tmpaf{\@gobble}%
681 \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
682 \let\@classoptionslist\bxjs@tmpa}
683 \def\bxjs@tmpdo@a#1,{%
684 \ifx\@nil#1\relax\else
685 \bxjs@tmpdo@b#1{}\@nil
686 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
687 \expandafter\bxjs@tmpdo@a
688 \fi}
689 \def\bxjs@tmpdo@b#1#\bxjs@tmpdo@c}
690 \def\bxjs@tmpdo@c#1\@nil{%
691 \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
692 \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。

```

693 \@expandtwoargs\@removeelement
694 {papersize}\@classoptionslist\@classoptionslist
695 \@expandtwoargs\@removeelement
696 {10pt}\@classoptionslist\@classoptionslist
697 \@expandtwoargs\@removeelement
698 {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。

```

699 \ifx\bxjs@jadriver@opt\undefined\else
700 \let\bxjs@jadriver\bxjs@jadriver@opt
701 \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。

```

702 \let\bxjs@tmpb\jsEngine
703 \ifx j\bxjs@tmpb\ifjsWithpTeXng

```

```

704 \let\bxjs@tmpb=g
705 \fi\fi
706 \ifx j\bxjs@tmpb\ifjsWithupTeX
707 \let\bxjs@tmpb=u
708 \fi\fi
709 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
710 \let\bxjs@tmpb=n
711 \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。)

```

712 \ifx *\bxjs@engine@given
713 \let\bxjs@engine@given\bxjs@tmpb

```

エンジン指定が autodetect-engine であり、かつ実際のエンジンが (u)pL^AT_EX だった場
合は、本来のエンジンオプションをグローバルオプションに加える。

```

714 \ifx j\bxjs@engine@given
715 \g@addto@macro\@classoptionslist{,latex}
716 \else\ifx u\bxjs@engine@given
717 \g@addto@macro\@classoptionslist{,uplatex}
718 \fi\fi
719 \fi
720 \ifx\bxjs@engine@given\@undefined\else
721 \ifx\bxjs@engine@given\bxjs@tmpb\else
722 \ClassError\bxjs@clsname
723 {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
724 \fi
725 \fi


```

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

```

726 \ifjsWithpTeXng
727 \g@addto@macro\@classoptionslist{,uplatex}
728 \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。

```

729 \@tempswatrue
730 \ifx \bxjs@driver@given\@undefined\else
731 \ifjsInPdfMode
732 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
733 \@tempswafalse
734 \fi
735 \else\ifx x\jsEngine
736 \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
737 \@tempswafalse
738 \fi
739 \else
740 \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
741 \@tempswafalse

```

```

742 \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
743 \@tempswafalse
744 \fi\fi
745 \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
746 \@tempswafalse
747 \fi\fi
748 \fi\fi
749 \fi
750 \if@tempswa\else
751 \ClassError\bxjs@clsname
752 {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
753 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

754 \ifjsInPdfMode \@tempswafalse
755 \else\ifx x\jsEngine \@tempswafalse
756 \else\ifjsWithpTeXng \@tempswafalse
757 \else \@tempswatrue
758 \fi\fi\fi
759 \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)。

```

760 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@opt\@undefined\else
761 \ClassWarningNoLine\bxjs@clsname
762 {No driver option is given}
763 \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。)

```

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

```

773 \ifjsWithpTeXng
774 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
775 \let\bxjs@platexng@nodrv\@undefined
776 \else\ifx t\bxjs@platexng@nodrv\else
777 \g@addto@macro\@classoptionslist{,dvipdfmx}
778 \fi\fi

```



779 \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。

780 \ifx\bxjs@driver@given\bxjs@driver@@none

781 \bxjs@papersizefalse

782 \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。

783 \def\bxjs@@minimal{minimal}

784 \ifx\bxjs@jadriver\relax

785 \ifx j\jsEngine

786 \def\bxjs@jadriver{standard}

787 \else

788 \ClassWarningNoLine\bxjs@clsname

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

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

791 such implicit setting is now DEPRECATED!\MessageBreak

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

793 if it is intended}

794 \let\bxjs@jadriver\bxjs@@minimal

795 \fi

796 \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。

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

798 \ifx\bxjs@engine@given\@undefined

799 \ClassError\bxjs@clsname

800 {An engine option must be explicitly given}%

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

802 engine option.\MessageBreak\@ehc}

803 \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。)

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

805 \ifx\bxjs@magstyle\bxjs@magstyle@@mag

806 \let\bxjs@magstyle\bxjs@magstyle@@default

807 \ClassError\bxjs@clsname

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

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

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

811 \fi

812 \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。

```
813 \ifx\bxjs@jbase@opt\@undefined\else
814   \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。

```
815     \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
816     \bxjs@invscale\@tempdima\jsScale
817     \bxjs@setbasefontsize{\@tempdima}%
818   \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。

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

```
830 \let\Cjascale\jsScale
```

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

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

```
837 \ifbxjs@disguise@js
838 %<book|report>\def\bxjs@js@clsname{jsbook}
839 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
840 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
841 \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。

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

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

```
843 \ifbxjs@oldfontcommands
```

```
844 \AtEndOfClass{\allowoldfontcommands}
```

```
845 \fi
```

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

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

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

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

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

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

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

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

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

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

```
846 \newdimen\stockwidth \newdimen\stockheight
```

```
847 \begingroup\expandafter\expandafter\expandafter\endgroup
```

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

```
849 % \newdimen\stockwidth \newdimen\stockheight
```

```
850 \setlength{\stockwidth}{\paperwidth}
```

```
851 \setlength{\stockheight}{\paperheight}
```

```
852 \advance \stockwidth 2in
```

```
853 \advance \stockheight 2in
```

```
854 \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。

```
855 %<slide>\def\n@baseline{13}%
856 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
857 %<!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。

```
858 \ifx\bxjs@magstyle\bxjs@magstyle@@mag
859   \jsc@magtrue
860 \else\ifx\bxjs@magstyle\bxjs@magstyle@@xreal
861   \jsc@mag@xrealtrue
862 \fi\fi
```

サイズの変更は $\mathrm{T\!E\!X}$ 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と若干縮めました。サイズについては全面的に見直しました。

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

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

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

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

```

880 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
881 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
882 \edef\jsc@magscale{\strip@pt\@tempdima}
883 \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。

```

884 \newdimen\jsc@mpt
885 \newdimen\jsc@mmm
886 \ifjsc@mag
887   \jsc@mpt=1\p@
888   \jsc@mmm=1mm
889   \catcode`\?=9 % \p@? read as \p@
890 \else
891   \jsc@mpt=\jsc@magscale\p@
892   \jsc@mmm=\jsc@magscale mm
893   \catcode`\?=11 \let\p@?\jsc@mpt
894 \fi
895 \chardef\bxjs@qmcc=\catcode`\?\relax
896 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode`\?=12\relax}

```

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

noz 非指定時は \zw を \jsZw と同義にする。

```

897 \newdimen\jsZw
898 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
899 \ifbxjs@usezw
900   \providecommand*\zw{\jsZw}
901 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

903 \ifjsc@mag@xreal
904   \RequirePackage{type1cm}
905   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

906 \ifbxjs@TUenc
907   \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax

```

```

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

```

[2016-11-16] latex.ltx (ltspace.dtx) で定義されている `\smallskip` の、単位 `pt` を `\jsc@mp` 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 950 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip

```

```

951 %\def\jsc@medskip{\vspace\jsc@medskipamount}
952 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

\jsc@smallskipamount

```

\jsc@medskipamount 953 \newskip\jsc@smallskipamount
\jsc@bigskipamount 954 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
955 %\newskip\jsc@medskipamount
956 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
957 %\newskip\jsc@bigskipamount
958 %\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。

```

959 % \ifpapersize
960 %   \setlength{\@tempdima}{\paperwidth}
961 %   \setlength{\@tempdimb}{\paperheight}
962 %   \iftombow
963 %     \advance \@tempdima 2truein
964 %     \advance \@tempdimb 2truein
965 %   \fi
966 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
967 % \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, IAT_EX の \def\@{\spacefactor\@m} という定義 (\@m は 1000) では I watch TV\@. と書くと V とピリオドのペアカーニングが効かなくなります。そこで, 次のような定義に直し, I watch TV.\@ と書くことにします。

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

BXJS クラスでの変更点:

- fix-at-cmd オプションが偽の場合は再定義しない。

- 固定の 3000 でなく実際のピリオドの sfcode 値を使う。
- 「防御的な \@」での不具合を防ぐため、大文字直後の \@ は標準と同等の動作にする。

```

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

```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

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

```
976 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
977 \expandafter\bxjs@tmpa\expandafter{%
978   \set@fontsize{#1}{#2}{#3}%
979 % 末尾にコードを追加
980   \expandafter\def\expandafter\size@update\expandafter{%
981     \size@update
982     \jsFontSizeChanged}%
983 }
```

`\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。

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

`\jsResetDimen` ユーザ定義用のフック。

```
993 \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。

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

これらのグルーをもってしても行分割ができない場合は、`\emergencystretch` に訴えます。

これはフォントサイズ非依存なので `\Cwd` で書くのが適当だが、`\Cwd` はまだ定義されていない。

1008 `\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。

```
1009 \newif\ifnarrowbaselines
1010 \if@english
1011   \narrowbaselinestrue
1012 \fi
1013 \def\narrowbaselines{%
1014   \narrowbaselinestrue
1015   \skip0=\abovedisplayskip
1016   \skip2=\abovedisplayshortskip
1017   \skip4=\belowdisplayskip
1018   \skip6=\belowdisplayshortskip
1019 % 一時的に警告を無効化する
1020   \let\bxjs@save@nomath\@nomath
1021   \let\@nomath\@gobble
1022   \@currsize\selectfont
1023   \let\@nomath\bxjs@save@nomath
1024   \abovedisplayskip=\skip0
1025   \abovedisplayshortskip=\skip2
1026   \belowdisplayskip=\skip4
1027   \belowdisplayshortskip=\skip6\relax}
1028 \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。

```
1029 \def\bxjs@ifnarrowbaselines{%
```

```

1030 \ifnarrowbaselines\expandafter\@firstoftwo
1031 \else \expandafter\@secondoftwo
1032 \fi
1033 }

```

`\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。

```

1034 \renewcommand{\normalsize}{%
1035   \bxjs@ifnarrowbaselines{%
1036     \jsc@setfontsize\normalsize\@xpt\@xipt
1037   }{%else
1038     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1039   }%

```

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

[2003-02-16] ちょっと変えました。

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

```

1040 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1041 \abovedisplayshortskip \z@ \@plus3\p@?
1042 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1043 \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。

```

1044 \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。

```

1045 %</class>
1046 %<*class|minijs>
1047 %% initialize
1048 \normalsize
1049 %</class|minijs>
1050 %<*class>

```

`\Cht` 基準となる長さの設定をします。pL^AT_EX 2_ε カーネル (`plfonts.dtx`) で宣言されているパ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`

`\Cvs`

`\Chs`

[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。

`\Cwd` 等の変数は p_TE_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 ± 2 , 2 ± 1 ポイントでしたが、ここではゼロ (`\z@`) にしました。

microtype 対策のため if 文を避ける。後の `\footnotesize` も同様。

```
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\leftmargini
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 ± 1 , 2 ± 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\leftmargin1
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
`\tiny` ます。特に注意すべきは `\large` で、これは二段組のときに節見出しのフォントとして使い、
`\large` 行送りを `\normalsize` と同じにすることによって、節見出しが複数行にわたっても段間で
`\Large` 行が揃うようにします。

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1093 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1094 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 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。『*LaTeX 2_ε 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@mp}
1116 \setlength\normallineskip{1\jsc@mp}
\normallineskiplimit 1117 \setlength\lineskiplimit{1\jsc@mp}
1118 \setlength\normallineskiplimit{1\jsc@mp}
```

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

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

とすると、行送りが通常の 2 倍になります。ただし、これを設定すると、たとえ `\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` が長さ指定の場合、`geometry` の形式 (`papersize={W,H}`) に変換する。`{W}{H}` の形式について。

```
1139 \@tempwafalse
1140 \def\bxjs@tmpdo{\@ifnextchar\bgroup\bxjs@tmpdo@a\remove@to@nnil}
1141 \def\bxjs@tmpdo@a#1{\edef\bxjs@tmpa{#1}%
1142   \@ifnextchar\bgroup\bxjs@tmpdo@b\remove@to@nnil}
1143 \def\bxjs@tmpdo@b#1{\edef\bxjs@tmpa{\bxjs@tmpa,#1}%
1144   \@ifnextchar\@nnil\bxjs@tmpdo@c\remove@to@nnil}
1145 \def\bxjs@tmpdo@c\@nnil{\@tempwattrue
1146   \edef\bxjs@param@paper{papersize={\bxjs@tmpa}}}
1147 \expandafter\bxjs@tmpdo\bxjs@param@paper\@nnil
```

W,H の形式について。

```
1148 \if@tempswa\else
1149   \def\bxjs@tmpa{\@nil,\@nil}
1150   \def\bxjs@tmpdo#1,#2,#3\@nnil{%
1151     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1152       \@tempswatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1153   \expandafter\bxjs@tmpdo\bxjs@param@paper,\@nil,\@nil\@nnil
1154 \fi
```

W*H の形式について。

```
1155 \if@tempswa\else
1156   \def\bxjs@tmpa{\@nil*\@nil}
1157   \def\bxjs@tmpdo#1*#2*#3\@nnil{%
1158     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1159       \@tempswatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1160   \expandafter\bxjs@tmpdo\bxjs@param@paper*\@nil*\@nil\@nnil
1161 \fi
```

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

```
1162 \edef\bxjs@layout@paper{%
1163   \ifjsc@mag truedimen,\fi
1164   \if@landscape landscape,\fi
1165   \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。

```
1166 %<*article|report>
1167 \def\bxjs@layout@base{%
1168   headheight=\topskip,footskip=0.03367\paperheight,%
1169   headsep=\footskip-\topskip,includeheadfoot,%
1170 }
1171 \edef\bxjs@layout{\bxjs@layout@base
1172   hscale=0.76,hmarginratio=1:1,%
1173   vscale=0.83,vmarginratio=1:1,%
1174 }
1175 %</article|report>
1176 %<*book>
1177 \def\bxjs@layout@base{%
1178   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1179 }
1180 \ifbxjs@layout@buggyhmargin %---
1181 % アレ
1182 \edef\bxjs@layout{\bxjs@layout@base
1183   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1184   vscale=0.83,vmarginratio=1:1,%
1185 }
1186 \else %---
1187 % 非アレ
1188 \edef\bxjs@layout{\bxjs@layout@base
1189   hmargin=18\jsc@mmm,%
```



```

1190 vscale=0.83,vmarginratio=1:1,%
1191 }
1192 \fi %---
1193 %</book>
1194 %<*slide>
1195 \def\bxjs@layout@base{%
1196 noheadfoot,%
1197 }
1198 \edef\bxjs@layout{\bxjs@layout@base
1199 hscale=0.9,hmarginratio=1:1,%
1200 vscale=0.944,vmarginratio=1:1,%
1201 }
1202 %</slide>

```

textwidth オプションの設定を反映する。

```

1203 %<*!book>
1204 \ifx\bxjs@textwidth@opt\undefined\else
1205 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1206 \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1207 \fi
1208 %</!book>
1209 \ifx\bxjs@number@of@lines@opt\undefined\else
1210 \bxjs@gsset@tempcnta{\bxjs@number@of@lines@opt}
1211 \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1212 \fi

```

\fullwidth [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1213 \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。

```

1214 %<*book>
1215 \newcommand\jsTextWidthLimit{40}
1216 \@tempdima=\jsTextWidthLimit\Cwd
1217 \ifx\bxjs@textwidth@limit@opt\undefined\else
1218 \bxjs@gsset@tempcnta{\bxjs@textwidth@limit@opt}
1219 \@tempdima=\@tempcnta\Cwd
1220 \fi
1221 \ifx\bxjs@textwidth@opt\undefined\else
1222 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1223 \fi
1224 \edef\bxjs@textwidth@limit{\the\@tempdima}
1225 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1226 \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1227 \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1228 \fi
1229 %</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。

```
1230 \def\bxjs@preproc@layout{%
1231   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mppt}
```

\bxjs@postproc@layout geometry の後処理。

```
1232 \def\bxjs@postproc@layout{%
    geometry のドライバを再設定する。
1233   \ifx\bxjs@geometry@driver\relax\else
1234     \let\Gm@driver\bxjs@geometry@driver
1235   \fi
    \ht\strutbox の値を元に戻す。
1236   \ht\strutbox=\bxjs@save@ht@strutbox\relax
    \textwidth の値を補正する。
1237   \ifbxjs@whole@zw@lines
1238     \@tempdimb=\textwidth
1239     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1240     \advance\textwidth.005pt\relax
1241     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1242     \advance\@tempdimb-\textwidth
1243     \advance\oddsidemargin 0.5\@tempdimb
1244     \advance\evensidemargin 0.5\@tempdimb
1245   \fi
1246   \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。

```
1247 %<*book>
1248   \@tempdima=\bxjs@textwidth@limit\relax
1249   \ifbxjs@whole@zw@lines
1250     \advance\@tempdima.005pt\relax
1251     \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1252   \fi
1253   \ifdim\textwidth>\@tempdima
1254     \textwidth=\@tempdima
1255     \addtolength\evensidemargin{\fullwidth-\textwidth}
1256   \fi
1257 %</book>
```

\textheight 関連の調整。

```
1258   \@tempdimb=\textheight
1259   \advance\textheight-\topskip
1260   \advance\textheight.005pt\relax
1261   \divide\textheight\baselineskip \multiply\textheight\baselineskip
1262   \advance\textheight\topskip
1263   \advance\@tempdimb-\textheight
```

```

1264 \advance\topmargin0.5\@tempdimb
    \headheight 関連の調整。
1265 \@tempdima=\topskip
1266 \advance\headheight\@tempdima
1267 \advance\topmargin-\@tempdima
    marginpar 関連の調整。
1268 \setlength\marginparsep{\columnsep}
1269 \setlength\marginparpush{\baselineskip}
1270 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1271     -\textwidth-10\jsc@mmm-\marginparsep}
1272 \ifbxjs@whole@zw@lines
1273     \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1274 \fi
    連動する変数。
1275 \maxdepth=.5\topskip
1276 \stockwidth=\paperwidth
1277 \stockheight=\paperheight
1278 }

```

`\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。

```

1279 \edef\jsGeometryOptions{%
1280     \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージを読み込む

ムニャムニャ。

```

1281 \def\bxjs@geometry@name{geometry}
1282 \ifbxjs@old@hook@system
1283     \let\bxjs@apply@bd@pre@geometry@hook\AtBeginDocument
1284 \else
1285     \def\bxjs@apply@bd@pre@geometry@hook{%
1286         \AddToHook{begindocument}{\bxjs@geometry@name}}
1287 \fi

```

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

```

1288 \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。

```

1289 \ifbxjs@papersize
1290     \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx

```

```

1291 \PassOptionsToPackage{dvipdfm}{geometry}
1292 \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1293 \PassOptionsToPackage{dvipdfm}{geometry}
1294 \fi\fi
1295 \let\bxPapersizeSpecialDone=t
1296 \else
1297 \PassOptionsToPackage{driver=none}{geometry}
1298 \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。

```

1299 \bxjs@apply@bd@pre@geometry@hook
1300 \bxjs@apply@bd@pre@geometry@hook{\bxjs@bd@pre@geometry@hook}
1301 \bxjs@apply@bd@pre@geometry@hook{\ImposeOldLuaTeXBehavior}
1302 \bxjs@preproc@layout
1303 \edef\bxjs@next{%
1304 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1305 }\bxjs@next
1306 \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。

```

1307 \let\bxjs@geometry@driver\Gm@driver
1308 \bxjs@postproc@layout

```

geometry のドライバ自動判別に対する前処理。

```

1309 \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。

```

1310 \ifpackagelater{geometry}{2010/02/12}{\else
1311 \PackageError\bxjs@clsname
1312 {Your 'geometry' package is too old (< v5.0)}%
1313 {\@ehc}%
1314 \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。

```

1315 \ifjsWithpTeXng
1316 \ifx\Gm@driver\@empty
1317 \def\Gm@driver{pdftex}%
1318 \fi
1319 \fi}

```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```

1320 \def\setpagelayout{%
1321 \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{\else
1322 \@ifstar{\bxjs@setpagelayout@a@one}{\bxjs@setpagelayout@a\z@}}
1323 \def\bxjs@setpagelayout@a#1#2{%
1324 \ifcase#1% modify

```

```

1325 \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1326 \or% reset(*)
1327 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1328 \or% semireset(+)
1329 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1330 \fi
1331 \bxjs@preproc@layout
1332 \edef\bxjs@next{%
1333 \noexpand\geometry{\bxjs@next}%
1334 }\bxjs@next
1335 \bxjs@postproc@layout}

```

■ geometry パッケージを読み込まない

geometry=user の場合の処理。

```
1336 \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。

```

1337 \g@addto@macro\bxjs@begin@document@hook{%
1338 \ifdim\textwidth=.5\maxdimen
1339 \ClassError\bxjs@clsname
1340 {Page layout is not properly set}%
1341 {\@ehd}%
1342 \fi}
1343 \def\jsUseMinimalPageLayout{%
1344 \setlength{\textwidth}{6.5in}%
1345 \setlength{\textheight}{8in}}

```

\setpagelayout はとりあえず無効にしておく。

```

1346 \let\bxjs@geometry@driver\relax
1347 \def\setpagelayout{%
1348 \bxjs@ifplus{\bxjs@pagelayout@a}{\%else
1349 \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1350 \def\bxjs@pagelayout@a#1{%
1351 \ClassError\bxjs@clsname
1352 {Command '\string\setpagelayout' is not supported,\MessageBreak
1353 because 'geometry' value is not 'class'}\@eha}
1354 %
1355 \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。

```
1356 %<*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。

```
1357 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1358 \if@slide
1359   \setlength\headheight{0\jsc@empt}
1360 \else
1361   \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
    06-26)
1362 \fi
```

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

```
1363 %<*article|kiyou>
1364 \if@slide
1365   \setlength\footskip{0pt}
1366 \else
1367   \setlength\footskip{0.03367\paperheight}
1368   \ifdim\footskip<\baselineskip
1369     \setlength\footskip{\baselineskip}
1370   \fi
1371 \fi
1372 %</article|kiyou>
1373 %<jspf>\setlength\footskip{9\jsc@mmm}
1374 %<*book>
1375 \if@report
1376   \setlength\footskip{0.03367\paperheight}
```

```

1377 \ifdim\footskip<\baselineskip
1378 \setlength\footskip{\baselineskip}
1379 \fi
1380 \else
1381 \setlength\footskip{0pt}
1382 \fi
1383 %</book>
1384 %<*report>
1385 \setlength\footskip{0.03367\paperheight}
1386 \ifdim\footskip<\baselineskip
1387 \setlength\footskip{\baselineskip}
1388 \fi
1389 %</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)。

```

1390 %<*article>
1391 \if@slide
1392 \setlength\headsep{0\jsc@empt}
1393 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1394 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1395 \else
1396 \setlength\headsep{\footskip}
1397 \addtolength\headsep{-\topskip}
1398 \fi
1399 %</article>
1400 %<*book>
1401 \if@report
1402 \setlength\headsep{\footskip}
1403 \addtolength\headsep{-\topskip}
1404 \else
1405 \setlength\headsep{6\jsc@mmm}
1406 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1407 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1408 \fi
1409 %</book>
1410 %<*report>
1411 \setlength\headsep{\footskip}
1412 \addtolength\headsep{-\topskip}
1413 %</report>
1414 %<*jspf>
1415 \setlength\headsep{9\jsc@mmm}
1416 \addtolength\headsep{-\topskip}
1417 %</jspf>
1418 %<*kiyou>

```

```

1419 \setlength\headheight{0\jsc@empt}
1420 \setlength\headsep{0\jsc@empt}
1421 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1422 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1423 %</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。

```

1424 \setlength\maxdepth{.5\topskip}

```

■本文の幅と高さ

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

```

1425 \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。

```

1426 %<*article>
1427 \if@slide
1428   \setlength\fullwidth{0.9\paperwidth}
1429 \else
1430   \setlength\fullwidth{0.76\paperwidth}
1431 \fi
1432 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1433 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1434 \setlength\textwidth{\fullwidth}
1435 %</article>
1436 %<*book>
1437 \if@report
1438   \setlength\fullwidth{0.76\paperwidth}
1439 \else
1440   \setlength\fullwidth{\paperwidth}
1441   \addtolength\fullwidth{-36\jsc@mmm}
1442 \fi
1443 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1444 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima

```



```

1445 \setlength\textwidth{\fullwidth}
1446 \if@report \else
1447   \if@twocolumn \else
1448     \ifdim \fullwidth>40zw
1449       \setlength\textwidth{40zw}
1450     \fi
1451   \fi
1452 \fi
1453 %</book>
1454 %<*report>
1455 \setlength\fullwidth{0.76\paperwidth}
1456 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1457 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1458 \setlength\textwidth{\fullwidth}
1459 %</report>
1460 %<*jspf>
1461 \setlength\fullwidth{50zw}
1462 \addtolength\fullwidth{8\jsc@mmm}
1463 \setlength\textwidth{\fullwidth}
1464 %</jspf>
1465 %<*kiyou>
1466 \setlength\fullwidth{48zw}
1467 \addtolength\fullwidth{\columnsep}
1468 \setlength\textwidth{\fullwidth}
1469 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

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

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

```

1470 %<*article|book|report>
1471 \if@slide
1472   \setlength{\textheight}{0.95\paperheight}
1473 \else
1474   \setlength{\textheight}{0.83\paperheight}
1475 \fi

```

```

1476 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1477 \addtolength{\textheight}{-\headsep}
1478 \addtolength{\textheight}{-\footskip}
1479 \addtolength{\textheight}{-\topskip}
1480 \divide\textheight\baselineskip
1481 \multiply\textheight\baselineskip
1482 %</article|book|report>
1483 %<jspf>\setlength{\textheight}{51\baselineskip}
1484 %<kiyou>\setlength{\textheight}{47\baselineskip}
1485 \addtolength{\textheight}{\topskip}
1486 \addtolength{\textheight}{0.1\jsc@mpt}
1487 %<jspf>\setlength{\mathindent}{10\jsc@mmm}

```

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

```

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

```

ですが、次のようにします。

```

1488 \def\flushbottom{%
1489     \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1490     \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。

```

1491 \setlength\marginparsep{\columnsep}
1492 \setlength\marginparpush{\baselineskip}

```

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

```

1493 \setlength{\oddsidemargin}{\paperwidth}
1494 \addtolength{\oddsidemargin}{-\fullwidth}
1495 \setlength{\oddsidemargin}{.5\oddsidemargin}
1496 \iftombow
1497     \addtolength{\oddsidemargin}{-1in}
1498 \else
1499     \addtolength{\oddsidemargin}{-\inv@mag in}
1500 \fi
1501 \setlength{\evensidemargin}{\oddsidemargin}
1502 \if@mparswitch
1503     \addtolength{\evensidemargin}{\fullwidth}
1504     \addtolength{\evensidemargin}{-\textwidth}
1505 \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。

```

1506 \setlength\marginparwidth{\paperwidth}
1507 \addtolength\marginparwidth{-\oddsidemargin}
1508 \addtolength\marginparwidth{-\inv@mag in}
1509 \addtolength\marginparwidth{-\textwidth}
1510 \addtolength\marginparwidth{-10\jsc@mmm}
1511 \addtolength\marginparwidth{-\marginparsep}
1512 \@tempdima=1zw
1513 \divide\marginparwidth\@tempdima
1514 \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)。

```

1515 \setlength\topmargin{\paperheight}
1516 \addtolength\topmargin{-\textheight}
1517 \if@slide
1518   \addtolength\topmargin{-\headheight}
1519 \else
1520   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
     \headheight (2003-06-26)
1521 \fi
1522 \addtolength\topmargin{-\headsep}
1523 \addtolength\topmargin{-\footskip}
1524 \setlength\topmargin{0.5\topmargin}
1525 %<kiyou>\setlength\topmargin{81truebp}
1526 \iftombow
1527   \addtolength\topmargin{-1in}
1528 \else
1529   \addtolength\topmargin{-\inv@mag in}
1530 \fi
1531 %</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。

1532 `\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。

1533 `\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
 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。

1534 `\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。

1535 `\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。

1536 `\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。

1537 `\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。

1538 `\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。

1539 `\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。

1540 `\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。

1541 `\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。

```
1542 \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。

```
1543 \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。

```
1544 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1545 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1546 \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 1547 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1548 \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 1549 \setlength\@fptop{0\p@? \@plus 1fil}
```

```
1550 \setlength\@fpsep{8\p@? \@plus 2fil}
```

```
1551 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblftop` 段抜きフロートについての値です。

```
\@dblfpsep 1552 \setlength\@dblftop{0\p@? \@plus 1fil}
```

```
\@dblfpbot 1553 \setlength\@dblfpsep{8\p@? \@plus 2fil}
```

```
1554 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

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

`\pltx@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。

```
1555 %\def\pltx@cleartorightpage{\clearpage\if@twoside
```

```
1556 % \ifodd\c@page
```

```
1557 % \iftdir
```

```
1558 % \hbox{}\thispagestyle{empty}\newpage
```

```
1559 % \if@twocolumn\hbox{}\newpage\fi
```

```

1560 % \fi
1561 % \else
1562 % \ifdir
1563 % \hbox{}\thispagestyle{empty}\newpage
1564 % \if@twocolumn\hbox{}\newpage\fi
1565 % \fi
1566 % \fi\fi}
1567 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1568 % \ifodd\c@page
1569 % \ifdir
1570 % \hbox{}\thispagestyle{empty}\newpage
1571 % \if@twocolumn\hbox{}\newpage\fi
1572 % \fi
1573 % \else
1574 % \iftdir
1575 % \hbox{}\thispagestyle{empty}\newpage
1576 % \if@twocolumn\hbox{}\newpage\fi
1577 % \fi
1578 % \fi\fi}
1579 \def\pltx@cleartooddpage{\clearpage\if@twoside
1580 \ifodd\c@page\else
1581 \hbox{}\thispagestyle{empty}\newpage
1582 \if@twocolumn\hbox{}\newpage\fi
1583 \fi\fi}
1584 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1585 \ifodd\c@page
1586 \hbox{}\thispagestyle{empty}\newpage
1587 \if@twocolumn\hbox{}\newpage\fi
1588 \fi\fi}

```

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

```

1589 \let\pltx@cleartorightpage\pltx@cleartooddpage
1590 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

`\cleardoublepage` [2017-02-24] コミュニティ版 p^LA^TE_X 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。

```

1591 %<*book|report>
1592 \if@openleft
1593 \let\cleardoublepage\pltx@cleartoleftpage
1594 \else\if@openright
1595 \let\cleardoublepage\pltx@cleartorightpage
1596 \fi\fi
1597 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

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

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

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

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

```
1598 % \def\ps@empty{%
1599 %   \let\mkboth\gobbletwo
1600 %   \let\oddhead\empty
1601 %   \let\oddfoot\empty
1602 %   \let\evenhead\empty
1603 %   \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。

```
1604 \def\ps@plainfoot{%
1605   \let\mkboth\gobbletwo
1606   \let\oddhead\empty
1607   \def\oddfoot{\normalfont\hfil\thepage\hfil}%
1608   \let\evenhead\empty}
```

```

1609 \let\@evenfoot\@oddfoot}
1610 \def\ps@plainhead{%
1611 \let\@mkboth\@gobbletwo
1612 \let\@oddfoot\@empty
1613 \let\@evenfoot\@empty
1614 \def\@evenhead{%
1615 \ifmparswitch \hss \fi
1616 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1617 \ifmparswitch\else \hss \fi}%
1618 \def\@oddhead{%
1619 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1620 %<book>\let\ps@plain\ps@plainhead
1621 %<!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。

```

1622 %<*article|slide>
1623 \if@twoside
1624 \def\ps@headings{%
1625 \let\@oddfoot\@empty
1626 \let\@evenfoot\@empty
1627 \def\@evenhead{\ifmparswitch \hss \fi
1628 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1629 \ifmparswitch\else \hss \fi}%
1630 \def\@oddhead{%
1631 \underline{%
1632 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1633 \let\@mkboth\markboth
1634 \def\sectionmark##1{\markboth{%
1635 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1636 ##1}{}}%
1637 \def\subsectionmark##1{\markright{%
1638 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1639 ##1}}%
1640 }
1641 \else % if not twoside
1642 \def\ps@headings{%
1643 \let\@oddfoot\@empty
1644 \def\@oddhead{%
1645 \underline{%
1646 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1647 \let\@mkboth\markboth
1648 \def\sectionmark##1{\markright{%
1649 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1650 ##1}}%
1651 \fi
1652 %</article|slide>

```


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

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

```
1653 %<*book|report>
1654 \def\bxjs@maybe@autoxspacing{%
1655   \ifx\autoxspacing\undefined\else \autoxspacing \fi}
1656 \newif\if@omit@number
1657 \def\ps@headings{%
1658   \let\@oddfoot\@empty
1659   \let\@evenfoot\@empty
1660   \def\@evenhead{%
1661     \if@mparswitch \hss \fi
1662     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1663       \textbf{\thepage}\hfil\leftmark}}}%
1664     \if@mparswitch\else \hss \fi}%
1665   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1666     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1667   \let\@mkboth\markboth
1668   \def\chaptermark##1{\markboth{%
1669     \ifnum \c@secnumdepth >\m@ne
1670       \if@mainmatter
1671         \if@omit@number\else
1672           \@chapapp\thechapter\@chappos\hskip1\jsZw
1673         \fi
1674       \fi
1675     \fi
1676     ##1}{}}}%
1677   \def\sectionmark##1{\markright{%
1678     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1679     ##1}}}%
1680 %</book|report>
```

最後は学会誌の場合です。

```
1681 %<*jspf>
1682 \def\ps@headings{%
1683   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1684   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1685   \def\@oddhead{\normalfont\hfil \@title \hfil}
1686   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1687 %</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。

```

1688 \def\ps@myheadings{%
1689   \let\@oddfoot\@empty\let\@evenfoot\@empty
1690   \def\@evenhead{%
1691     \if@mparswitch \hss \fi%
1692     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1693     \if@mparswitch\else \hss \fi}%
1694   \def\@oddhead{%
1695     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1696   \let\@mkboth\@gobbletwo
1697 %<book|report> \let\chaptermark\@gobble
1698 \let\sectionmark\@gobble
1699 %<!book&!report> \let\subsectionmark\@gobble
1700 }

```

8 文書のマークアップ

8.1 表題

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

```

\date 1701 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1702 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1703 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1704 % \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。

```

1705 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1706 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1707 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1708 \g@addto@macro\bxjs@begin@document@hook{\bxjs@decl@subtitle}
1709 \def\bxjs@decl@subtitle{%
1710   \global\let\bxjs@decl@subtitle\relax
1711   \ifx\subtitle\@undefined
1712     \global\let\subtitle\jsSubtitle
1713   \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。

```

1714 \def\bxjs@annihilate@subtitle{%

```

```

1715 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1716 \global\let\jsSubtitle\relax}

```

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

```

\author 1717 %<*jspf>
1718 \newcommand*\etitle}[1]{\gdef\etitle{#1}}
\keywords 1719 \newcommand*\eauthor}[1]{\gdef\eauthor{#1}}
1720 \newcommand*\keywords}[1]{\gdef\keywords{#1}}
1721 \newcommand*\email}[1]{\gdef\authors@mail{#1}}
1722 \newcommand*\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1723 %</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。

```

1724 \def\plainifnotempty{%
1725   \ifx \@oddhead \@empty
1726     \ifx \@oddfoot \@empty
1727       \else
1728         \thispagestyle{plainfoot}%
1729       \fi
1730     \else
1731       \thispagestyle{plainhead}%
1732     \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。

```

1733 %<*article|book|report|slide>
1734 \if@titlepage
1735   \newcommand*\maketitle{%
1736     \begin{titlepage}%
1737       \let\footnotesize\small
1738       \let\footnoterule\relax
1739       \let\footnote\thanks
1740       \null\vfil
1741       \if@slide
1742         {\footnotesize \@date}%
1743       \begin{center}
1744         \mbox{} \ \ [1\jsZw]
1745         \large

```

```

1746      {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1747      \jsc@smallskip
1748      \@title
1749      \ifx\bxjs@subtitle\@undefined\else
1750        \par\vskip\z@
1751        {\small \bxjs@subtitle\par}
1752      \fi
1753      \jsc@smallskip
1754      {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1755      \vfill
1756      {\small \@author}%
1757      \end{center}
1758    \else
1759      \vskip 60\p@?
1760      \begin{center}%
1761        {\LARGE \@title \par}%
1762        \ifx\bxjs@subtitle\@undefined\else
1763          \vskip5\p@?
1764          {\normalsize \bxjs@subtitle\par}
1765        \fi
1766        \vskip 3em%
1767        {\large
1768          \lineskip .75em
1769          \begin{tabular}[t]{c}%
1770            \@author
1771          \end{tabular}\par}%
1772        \vskip 1.5em
1773        {\large \@date \par}%
1774      \end{center}%
1775      \fi
1776      \par
1777      \@thanks\vfil\null
1778    \end{titlepage}%
1779    \setcounter{footnote}{0}%
1780    \global\let\thanks\relax
1781    \global\let\maketitle\relax
1782    \global\let\@thanks\@empty
1783    \global\let\@author\@empty
1784    \global\let\@date\@empty
1785    \global\let\@title\@empty
1786    \global\let\title\relax
1787    \global\let\author\relax
1788    \global\let\date\relax
1789    \global\let\and\relax
1790    \bxjs@annihilate@subtitle
1791  }%
1792 \else
1793   \newcommand{\maketitle}{\par
1794     \begingroup

```

```

1795 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1796 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1797 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1798 \parindent 1\jsZw\noindent
1799 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1800 \if@twocolumn
1801 \ifnum \col@number=\@ne
1802 \maketitle
1803 \else
1804 \twocolumn[\maketitle]%
1805 \fi
1806 \else
1807 \newpage
1808 \global\@topnum\z@ % Prevents figures from going at top of page.
1809 \maketitle
1810 \fi
1811 \plainifnotempty
1812 \@thanks
1813 \endgroup
1814 \setcounter{footnote}{0}%
1815 \global\let\thanks\relax
1816 \global\let\maketitle\relax
1817 \global\let\@thanks\@empty
1818 \global\let\@author\@empty
1819 \global\let\@date\@empty
1820 \global\let\@title\@empty
1821 \global\let\title\relax
1822 \global\let\author\relax
1823 \global\let\date\relax
1824 \global\let\and\relax
1825 \bxjs@annihilate@subtitle
1826 }

```

\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。

```

1827 \def\maketitle{%
1828 \newpage\null
1829 \vskip 2em
1830 \begin{center}%
1831 \let\footnote\thanks
1832 {\LARGE \@title \par}%
1833 \ifx\bxjs@subtitle\@undefined\else
1834 \vskip3\p@?
1835 {\normalsize \bxjs@subtitle\par}
1836 \fi
1837 \vskip 1.5em
1838 {\large
1839 \lineskip .5em
1840 \begin{tabular}[t]{c}%
1841 \author

```

```

1842         \end{tabular}\par}%
1843     \vskip 1em
1844     {\large \@date}%
1845     \end{center}%
1846     \par\vskip 1.5em
1847 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1848 }
1849 \fi
1850 %</article|book|report|slide>
1851 %<*jspf>
1852 \newcommand{\maketitle}{\par
1853     \begin{group}
1854         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1855         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1856         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1857             \parindent 1\jsZw\noindent
1858             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1859         \twocolumn[\@maketitle]%
1860         \plainifnotempty
1861         \@thanks
1862     \end{group}
1863     \setcounter{footnote}{0}%
1864     \global\let\thanks\relax
1865     \global\let\maketitle\relax
1866     \global\let\@thanks\@empty
1867     \global\let\@author\@empty
1868     \global\let\@date\@empty
1869 % \global\let\@title\@empty % \@title は柱に使う
1870     \global\let\title\relax
1871     \global\let\author\relax
1872     \global\let\date\relax
1873     \global\let\and\relax
1874     \ifx\authors@mail\@undefined\else{%
1875         \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1876         \footnotetext[0]{\itshape\authors@mail}%
1877     }\fi
1878     \global\let\authors@mail\@undefined}
1879 \def\@maketitle{%
1880     \newpage\null
1881     \vskip 6em % used to be 2em
1882     \begin{center}
1883         \let\footnote\thanks
1884         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1885         \lineskip .5em
1886         \ifx\@author\@undefined\else
1887             \vskip 1em
1888             \begin{tabular}[t]{c}%
1889                 \@author
1890             \end{tabular}\par

```

```

1891 \fi
1892 \ifx\@etitle\@undefined\else
1893 \vskip 1em
1894 {\large \@etitle \par}%
1895 \fi
1896 \ifx\@eauthor\@undefined\else
1897 \vskip 1em
1898 \begin{tabular}[t]{c}%
1899 \@eauthor
1900 \end{tabular}\par
1901 \fi
1902 \vskip 1em
1903 \@date
1904 \end{center}
1905 \vskip 1.5em
1906 \centerline{\box\@abstractbox}
1907 \ifx\@keywords\@undefined\else
1908 \vskip 1.5em
1909 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1910 \fi
1911 \vskip 1.5em}
1912 %</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。

```

1913 \def\bxjs@label@sect#1{%
1914 \expandafter\ifx\csname label#1\endcsname\relax
1915 \csname the#1\endcsname
1916 \else \csname label#1\endcsname
1917 \fi}
1918 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1919 \ifnum\bxjs@label@section=\bxjs@label@section@@compat\else
1920 \def\@secapp{\presectionname}
1921 \def\@secpos{\postsectionname}
1922 \fi

```

`\labelsection` 節番号の出力書式。

```

1923 \ifnum\bxjs@label@section=\bxjs@label@section@@modern
1924 \def\labelsection{\@secapp\thesection\@secpos}

```

■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側の空きです (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、完全に行送りがない \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。

```

1926 \def\startsection#1#2#3#4#5#6{%
1927   \if@noskipsec \leavevmode \fi
1928   \par
1929 % 見出し上の空きを \@tempskipa にセットする
1930   \@tempskipa #4\relax
1931 % \@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
1932   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1933 % 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
1934   \ifdim \@tempskipa <\z@
1935     \@tempskipa -\@tempskipa \@afterindentfalse
1936   \fi
1937   \if@nobreak
1938 %   \everypar{\everyparhook}% これは間違い
1939     \everypar{}%
1940   \else
1941     \addpenalty\@secpenalty
1942 % 次の行は削除

```



```

1943 % \addvspace\@tempskipa
1944 % 次の \noindent まで追加
1945 \ifdim \@tempskipa >\z@
1946 \if@slide\else
1947 \null
1948 \vspace*{-\baselineskip}%
1949 \fi
1950 \vskip\@tempskipa
1951 \fi
1952 \fi
1953 \noindent
1954 % 追加終わり
1955 \@ifstar
1956 {\@ssect{#3}{#4}{#5}{#6}}%
1957 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

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

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

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

```

1958 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
1959 \let\bxjs@if@ceph\@firstofone
1960 \else \let\bxjs@if@ceph\@gobble
1961 \fi

1962 \def\@sect#1#2#3#4#5#6[#7]#8{%
1963 \ifnum #2>\c@secnumdepth
1964 \let\@svsec\@empty
1965 \else
1966 \refstepcounter{#1}%
1967 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1968 \fi
1969 % 見出し後の空きを \@tempskipa にセット
1970 \@tempskipa #5\relax
1971 % 条件判断の順序を入れ替えました
1972 \ifdim \@tempskipa<\z@
1973 \def\@svsechd{%
1974 #6{\hskip #3\relax
1975 \@svsec #8}%
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 #8
1982 \else

```

```

1983 \begingroup
1984 \interlinepenalty \@M % 下から移動
1985 #6{%
1986 \hangfrom{\hskip #3\relax\@svsec}%
1987 % \interlinepenalty \@M % 上に移動
1988 #8\@@par}%
1989 \endgroup
1990 \csname #1mark\endcsname{#7}%
1991 \addcontentsline{toc}{#1}{%
1992 \ifnum #2>\c@secnumdepth \else
1993 \protect\numberline{\bxjs@label@sect{#1}}%
1994 \fi
1995 #7}% 目次にフルネームを載せるならここは #8
1996 \fi
1997 \@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。

```

1998 \def\@xsect#1{%
1999 % 見出しの後ろの空きを \@tempskipa にセット
2000 \@tempskipa #1\relax
2001 % 条件判断の順序を変えました
2002 \ifdim \@tempskipa<\z@
2003 \@nobreakfalse
2004 \global\@noskipsecttrue
2005 \everypar{%
2006 \if@noskipsec
2007 \global\@noskipsecfalse
2008 {\setbox\z@\lastbox}%
2009 \clubpenalty\@M
2010 \begingroup \@svsechd \endgroup
2011 \unskip
2012 \@tempskipa #1\relax
2013 \hskip -\@tempskipa
2014 \else
2015 \clubpenalty \@clubpenalty
2016 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
2017 \fi\bxjs@if@ceph\everyparhook}%
2018 \else
2019 \par \nobreak
2020 \vskip \@tempskipa
2021 \@afterheading
2022 \fi
2023 \if@slide
2024 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
2025 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt

```

```

2026     \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
2027 \fi
2028 \par % 2000-12-18
2029 \ignorespaces}
2030 \def\@ssect#1#2#3#4#5{%
2031   \@tempskipa #3\relax
2032   \ifdim \@tempskipa<\z@
2033     \def\@svsechd{#4{\hskip #1\relax #5}}%
2034   \else
2035     \begingroup
2036       #4{%
2037         \@hangfrom{\hskip #1}%
2038         \interlinepenalty \@M #5\@@par}%
2039     \endgroup
2040 \fi
2041 \@xsect{#3}}

```

■柱関係の命令

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

```

\subsectionmark 2042 \newcommand*\chaptermark[1]{}
\subsubsectionmark 2043 % \newcommand*\sectionmark[1]{}
2044 % \newcommand*\subsectionmark[1]{}
\paragraphmark 2045 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 2046 % \newcommand*\paragraphmark[1]{}
2047 % \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。

```

2048 %<!book&!report>\setcounter{secnumdepth}{3}
2049 %<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 2050 \newcounter{part}
\c@subsubsection 2051 %<book|report>\newcounter{chapter}
2052 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2053 %<!book&!report>\newcounter{section}
\c@subparagraph 2054 \newcounter{subsection}[section]
2055 \newcounter{subsubsection}[subsection]
2056 \newcounter{paragraph}[subsubsection]
2057 \newcounter{subparagraph}[paragraph]

```

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

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

`\thesection`

`\thesubsection`

`\thesubsubsection`

`\theparagraph`

`\thesubparagraph`

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

2058 \renewcommand{\thepart}{\@Roman\c@part}
2059 %<!*book&!report>
2060 \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2061 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2062 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2063 \else
2064 \renewcommand{\thesection}{\@arabic\c@section}
2065 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2066 \fi
2067 %</!book&!report>
2068 %<*book|report>
2069 \renewcommand{\thechapter}{\@arabic\c@chapter}
2070 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2071 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2072 %</book|report>
2073 \renewcommand{\thesubsubsection}{\%
2074   \thesubsection.\@arabic\c@subsubsection}
2075 \renewcommand{\theparagraph}{\%
2076   \thesubsubsection.\@arabic\c@paragraph}
2077 \renewcommand{\thesubparagraph}{\%
2078   \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。

```

2079 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2080 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

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

ていた問題です（コミュニティ版 p_LA_TE_X の標準クラス 2017/03/05 も参照）。

```
2081 %<*book|report>
2082 \newcommand\frontmatter{%
2083   \pltx@cleartooddpage
2084   \@mainmatterfalse
2085   \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。

```
2086 \newcommand\mainmatter{%
2087   \pltx@cleartooddpage
2088   \@mainmattertrue
2089   \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。

```
2090 \newcommand\backmatter{%
2091   \if@openleft
2092     \cleardoublepage
2093   \else\if@openright
2094     \cleardoublepage
2095   \else
2096     \clearpage
2097   \fi\fi
2098   \@mainmatterfalse}
2099 %</book|report>
```

■部

`\part` 新しい部を始めます。

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

`\secdef{星なし}{星あり}`

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

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

まず `book` と `report` のクラス以外です。

```
2100 %<*&!book&!report>
2101 \newcommand\part{%
2102   \if@noskipsec \leavevmode \fi
2103   \par
2104   \addvspace{4ex}%
2105   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2106   \secdef\@part\@spart}
2107 %</&!book&!report>
```

book および report クラスの場合は、少し複雑です。

```
2108 %<*book|report>
2109 \newcommand\part{%
2110   \if@openleft
2111     \cleardoublepage
2112   \else\if@openright
2113     \cleardoublepage
2114   \else
2115     \clearpage
2116   \fi\fi
2117   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2118   \if@twocolumn
2119     \onecolumn
2120     \@restonecoltrue
2121   \else
2122     \@restonecolfalse
2123   \fi
2124   \null\vfil
2125   \secdef\@part\@spart}
2126 %</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。

```
2127 %<!*book&!report>
2128 \def\@part[#1]#2{%
2129   \ifnum \c@secnumdepth >\m@ne
2130     \refstepcounter{part}%
2131     \addcontentsline{toc}{part}{%
2132       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2133   \else
2134     \addcontentsline{toc}{part}{#1}%
2135   \fi
2136   \markboth{}{}%
2137   {\parindent\z@
2138     \raggedright
2139     \interlinepenalty \@M
2140     \normalfont
2141     \ifnum \c@secnumdepth >\m@ne
2142       \Large\headfont\prepartname\thepart\postpartname
2143       \par\nobreak
2144     \fi
2145     \huge \headfont #2%
2146     \markboth{}{}\par}%
2147   \nobreak
2148   \vskip 3ex
2149   \@afterheading}
2150 %</!*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。

```
2151 %<*book|report>
2152 \def\@part[#1]#2{%
2153   \ifnum \c@secnumdepth >-2\relax
2154     \refstepcounter{part}%
2155     \addcontentsline{toc}{part}{%
2156       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2157   \else
2158     \addcontentsline{toc}{part}{#1}%
2159   \fi
2160   \markboth{}{}%
2161   {\centering
2162     \interlinepenalty \@M
2163     \normalfont
2164     \ifnum \c@secnumdepth >-2\relax
2165       \huge\headfont \prepartname\thepart\postpartname
2166       \par\vskip20\p@?
2167     \fi
2168     \Huge \headfont #2\par}%
2169   \@endpart}
2170 %</book|report>
```

\@spart 番号を付けない部です。

```
2171 %<*!book&!report>
2172 \def\@spart#1{%
2173   \parindent \z@ \raggedright
2174   \interlinepenalty \@M
2175   \normalfont
2176   \huge \headfont #1\par}%
2177   \nobreak
2178   \vskip 3ex
2179   \@afterheading}
2180 %</!book&!report>
2181 %<*book|report>
2182 \def\@spart#1{%
2183   \centering
2184   \interlinepenalty \@M
2185   \normalfont
2186   \Huge \headfont #1\par}%
2187   \@endpart}
2188 %</book|report>
```

\@endpart \@part と \@spart の最後で実行されるマクロです。両面印刷のときは白ページを追加します。二段組のときには、二段組に戻します。

[2016-12-13] openany のときには白ページが追加されるのは変なので、その場合は追加しないようにしました。このバグは L^AT_EX では classes.dtx v1.4b (2000/05/19) で修正されています。

```
2189 %<*book|report>
```

```

2190 \def\@endpart{\vfil\newpage
2191   \if@twoside
2192     \if@openleft %% added (2017/02/24)
2193       \null\thispagestyle{empty}\newpage
2194     \else\if@openright %% added (2016/12/13)
2195       \null\thispagestyle{empty}\newpage
2196     \fi\fi %% added (2016/12/13, 2017/02/24)
2197   \fi
2198   \if@restonecol
2199     \twocolumn
2200   \fi}
2201 %</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。

```

2202 %<*book|report>
2203 \newcommand{\chapter}{%
2204   \if@openleft\cleardoublepage\else
2205   \if@openright\cleardoublepage\else\clearpage\fi\fi
2206   \plainifnotempty % 元: \thispagestyle{plain}
2207   \global\@topnum\z@
2208   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2209   \secdef
2210     {\@omit@numberfalse\@chapter}%
2211     {\@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。

```

2212 \def\@chapter[#1]#2{%
2213   \ifnum \c@secnumdepth >\m@ne
2214     \if@mainmatter
2215       \refstepcounter{chapter}%
2216       \typeout{\@chapapp\thechapter\@chappos}%
2217       \addcontentsline{toc}{chapter}%
2218         {\protect\numberline
2219 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2220         {\@chapapp\thechapter\@chappos}%
2221         #1}%
2222     \else\addcontentsline{toc}{chapter}{#1}\fi
2223   \else
2224     \addcontentsline{toc}{chapter}{#1}%
2225   \fi
2226   \chaptermark{#1}%
2227   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}%
2228   \addtocontents{lot}{\protect\addvspace{10\jsc@empt}}%
2229   \if@twocolumn

```



```

2230 \topnewpage[\@makechapterhead{#2}]\%
2231 \else
2232 \@makechapterhead{#2}\%
2233 \afterheading
2234 \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。

```

2235 \def\@makechapterhead#1{%
2236 \vspace*{2\Cvs}\% 欧文は 50pt
2237 {\parindent \z@ \raggedright \normalfont
2238 \ifnum \c@secnumdepth >\m@ne
2239 \if@mainmatter
2240 \huge\headfont \@chapapp\thechapter\@chappos
2241 \par\nobreak
2242 \vskip \Cvs \% 欧文は 20pt
2243 \fi
2244 \fi
2245 \interlinepenalty\@M
2246 \Huge \headfont #1\par\nobreak
2247 \vskip 3\Cvs}} \% 欧文は 40pt

```

`\@chapter` `\chapter*{...}` コマンドの本体です。`\chaptermark` を補いました。

```

2248 \def\@chapter#1{%
2249 \chaptermark{#1}\%
2250 \if@twocolumn
2251 \topnewpage[\@makeschapterhead{#1}]\%
2252 \else
2253 \@makeschapterhead{#1}\@afterheading
2254 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2255 \def\@makeschapterhead#1{%
2256 \vspace*{2\Cvs}\% 欧文は 50pt
2257 {\parindent \z@ \raggedright
2258 \normalfont
2259 \interlinepenalty\@M
2260 \Huge \headfont #1\par\nobreak
2261 \vskip 3\Cvs}} \% 欧文は 40pt
2262 %</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。

```

2263 \if@twocolumn
2264 \newcommand{\section}{%
2265 %<jspf>\ifx\maketitle\relax\else\maketitle\fi

```

```

2266 \startsection{section}{1}{\z@}%
2267 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2268 %<kiyou> {\Cvs}{0.5\Cvs}%
2269 % {\normalfont\large\headfont\@secapp}}
2270 {\normalfont\large\headfont\raggedright}}
2271 \else
2272 \newcommand{\section}{%
2273 \if@slide\clearpage\fi
2274 \startsection{section}{1}{\z@}%
2275 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2276 {.5\Cvs \@plus.3\Cdp}% 後アキ
2277 % {\normalfont\Large\headfont\@secapp}}
2278 {\normalfont\Large\headfont\raggedright}}
2279 \fi

```

\subsection 同上です。

```

2280 \if@twocolumn
2281 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2282 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2283 {\normalfont\normalsize\headfont}}
2284 \else
2285 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2286 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2287 {.5\Cvs \@plus.3\Cdp}% 後アキ
2288 {\normalfont\large\headfont}}
2289 \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)。

```

2290 \if@twocolumn
2291 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2292 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2293 {\normalfont\normalsize\headfont}}
2294 \else
2295 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2296 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2297 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2298 {\normalfont\normalsize\headfont}}
2299 \fi

```

\paragraph 見出しの後ろで改行されません。

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

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

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

※ BXJS クラスでは、1.1 版 [2016-02-14] から `\jsParagraphMark` をサポートしている。
 段落のマーク (■) が必ず和文フォントで出力されるようにする。
`\jsJaChar` は standard 和文ドライバが読み込まれた場合は `\jachar` と同義になるが、
 それ以外は何もしない。

```

2300 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2301 \let\bxjs@paragraph@mark\jsParagraphMark
2302 \ifx\bxjs@paragraph@mark\@empty
2303   \let\jsParagraphMark\@empty
2304 \else\ifx\bxjs@paragraph@mark\undefined\else
2305   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2306 \fi\fi
2307 \let\jsJaChar\@empty
2308 \if@twocolumn
2309   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2310     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2311     <jspf> {\normalfont\normalsize\headfont}}
2312     <!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2313 \else
2314   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2315     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2316     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2317     <jspf> {\normalfont\normalsize\headfont}}
2318     <!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2319 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2320 \if@twocolumn
2321   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2322     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2323     {\normalfont\normalsize\headfont}}
2324 \else
2325   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2326     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2327     {\normalfont\normalsize\headfont}}
2328 \fi

```

8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```
2329 \if@slide
2330   \setlength\leftmargini{1\jsZw}
2331 \else
2332   \if@twocolumn
2333     \setlength\leftmargini{2\jsZw}
2334   \else
2335     \setlength\leftmargini{3\jsZw}
2336   \fi
2337 \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 2338 \if@slide
\leftmarginv 2339   \setlength\leftmarginii {1\jsZw}
2340   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2341   \setlength\leftmarginiv {1\jsZw}
2342   \setlength\leftmarginv {1\jsZw}
2343   \setlength\leftmarginvi {1\jsZw}
2344 \else
2345   \setlength\leftmarginii {2\jsZw}
2346   \setlength\leftmarginiii{2\jsZw}
2347   \setlength\leftmarginiv {2\jsZw}
2348   \setlength\leftmarginv {1\jsZw}
2349   \setlength\leftmarginvi {1\jsZw}
2350 \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。

```
2351 \setlength \labelsep {0.5\jsZw} % .5em
2352 \setlength \labelwidth{\leftmargini}
2353 \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。

```
2354 \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 2355 \@beginparpenalty -\@lowpenalty
\@itempenalty 2356 \@endparpenalty -\@lowpenalty
2357 \@itempenalty -\@lowpenalty
```

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

す。アスキーの標準スタイルではトップレベルの `itemize`, `enumerate` 環境でだけ最初と最後に行送りの半分の空きが入るようになっていました。

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

```
2358 \def\@listi{\leftmargin\leftmargini
2359   \parsep \z@
2360   \topsep 0.5\baselineskip
2361   \itemsep \z@ \relax}
2362 \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）。

```
2363 \@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 2364 \def\@listii{\leftmargin\leftmarginii
\@listiv 2365   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2366   \topsep \z@
\@listv 2367   \parsep \z@
\@listvi 2368   \itemsep\parsep}
2369 \def\@listiii{\leftmargin\leftmarginiii
2370   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2371   \topsep \z@
2372   \parsep \z@
2373   \itemsep\parsep}
2374 \def\@listiv {\leftmargin\leftmarginiv
2375   \labelwidth\leftmarginiv
2376   \advance\labelwidth-\labelsep}
2377 \def\@listv {\leftmargin\leftmarginv
2378   \labelwidth\leftmarginv
2379   \advance\labelwidth-\labelsep}
2380 \def\@listvi {\leftmargin\leftmarginvi
2381   \labelwidth\leftmarginvi
2382   \advance\labelwidth-\labelsep}
```

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

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

```
2383 \renewcommand{\theenumi}{\@arabic\c@enumi}
2384 \renewcommand{\theenumii}{\@alph\c@enumii}
2385 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2386 \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。

```
2387 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2388 \newcommand*\jsInJaParen[1]{%
2389   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2390 \newcommand\labelenumi{\theenumi.}
2391 \newcommand\labelenumii{\jsInJaParen\theenumii}
2392 \newcommand\labelenumiii{\theenumiii.}
2393 \newcommand\labelenumiv{\theenumiv.}
```

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

```
\p@enumiv 2394 \renewcommand\p@enumii{\theenumi}
2395 \renewcommand\p@enumiii{\theenumi\jsInhibitGlue (\theenumii )}
2396 \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 2397 \newcommand\labelitemi{\textbullet}
2398 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 2399 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2400 \newcommand\labelitemiv{\textperiodcentered}
```

■description 環境

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

```
2401 \newenvironment{description}{%
2402   \list{}{%
2403     \labelwidth=\leftmargin
2404     \labelsep=1\jsZw
2405     \advance \labelwidth by -\labelsep
2406     \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。

```
2407 \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。

```
2408 %<*book|report>
2409 \newenvironment{chapterabstract}{%
2410   \begin{list}{}{%
2411     \listparindent=1\jsZw
2412     \itemindent=\listparindent
2413     \rightmargin=0pt
2414     \leftmargin=5\jsZw}\item[]{}{\end{list}\vspace{\baselineskip}}
2415 %>/book|report>
```

“普通の” abstract 環境の定義。

```
2416 %<*article|report|slide>
2417 \newbox\@abstractbox
2418 \if@titlepage
2419   \newenvironment{abstract}{%
2420     \titlepage
2421     \null\vfil
2422     \@beginparpenalty\@lowpenalty
2423     \begin{center}%
2424       \headfont \abstractname
2425       \@endparpenalty\@M
2426     \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。

```
2427   \par}%
2428   {\par\vfil\@null\endtitlepage}
2429 \else
2430   \newenvironment{abstract}{%
2431     \if@twocolumn
2432       \ifx\maketitle\relax
2433         \section*{\abstractname}%
2434       \else
2435         \global\setbox\@abstractbox\hbox\bgroup

```

```

2436 \begin{minipage}[b]{\textwidth}
2437 \small\parindent1\jsZw
2438 \begin{center}%
2439 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2440 \end{center}%
2441 \list{}{%
2442 \listparindent\parindent
2443 \itemindent \listparindent
2444 \rightmargin \leftmargin}%
2445 \item\relax
2446 \fi
2447 \else
2448 \small
2449 \begin{center}%
2450 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2451 \end{center}%
2452 \list{}{%
2453 \listparindent\parindent
2454 \itemindent \listparindent
2455 \rightmargin \leftmargin}%
2456 \item\relax
2457 \fi}{\if@twocolumn
2458 \ifx\maketitle\relax
2459 \else
2460 \endlist\end{minipage}\egroup
2461 \fi
2462 \else
2463 \endlist
2464 \fi}
2465 \fi
2466 %</article|report|slide>
2467 %<*jspf>
2468 \newbox\@abstractbox
2469 \newenvironment{abstract}{%
2470 \global\setbox\@abstractbox\hbox\bgroup
2471 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2472 \small
2473 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2474 {\end{minipage}\egroup}
2475 %</jspf>

```

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

```

2476 %<*book|report>
2477 \ifbxjs@force@chapterabstract
2478 \let\abstract\chapterabstract
2479 \let\endabstract\endchapterabstract
2480 \fi
2481 %</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。

```
2482 %<*jspf>
2483 %\newbox\@keywordsbox
2484 %\newenvironment{keywords}{%
2485 % \global\setbox\@keywordsbox\hbox\bgroup
2486 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2487 % \small\parindent0\jsZw}%
2488 % {\end{minipage}\egroup}
2489 %</jspf>
```

■verse 環境

verse 詩のための `verse` 環境です。

```
2490 \newenvironment{verse}{%
2491 \let \=\@centercr
2492 \list{}{%
2493 \itemsep \z@
2494 \itemindent -2\jsZw % 元: -1.5em
2495 \listparindent\itemindent
2496 \rightmargin \z@
2497 \advance\leftmargin 2\jsZw}% 元: 1.5em
2498 \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。

```
2499 \newenvironment{quotation}{%
2500 \list{}{%
2501 \listparindent\parindent
2502 \itemindent\listparindent
2503 \rightmargin \z@}%
2504 \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。

```
2505 \newenvironment{quote}%
2506 {\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。

```
2507 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2508   \item[\hskip \labelsep{\headfont #1\ #2}]}
2509 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2510   \item[\hskip \labelsep{\headfont #1\ #2 (#3)}]}
```

`titlepage` タイトルを独立のページに出力するのに使われます。

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

```
2511 \newenvironment{titlepage}{%
2512 %<book> \pltx@cleartooddpage %% 2017-02-24
2513   \if@twocolumn
2514     \@restonecoltrue\onecolumn
2515   \else
2516     \@restonecolfalse\newpage
2517   \fi
2518   \thispagestyle{empty}%
2519   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2520   }%
2521   {\if@restonecol\twocolumn \else \newpage \fi
2522   \if@twoside\else
2523     \setcounter{page}\@ne
2524   \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2525 %<*&book&!report>
2526 \newcommand{\appendix}{\par
2527   \setcounter{section}{0}%
2528   \setcounter{subsection}{0}%
2529   \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2530     \gdef\presectionname{\appendixname}%
2531     \gdef\postsectionname{}%
2532 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
```

```

2533 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2534 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2535 \else
2536 \gdef\@secapp{\appendixname}%
2537 \gdef\@secpos{}%
2538 \gdef\thesection{\@Alph\c@section}%
2539 \fi}
2540 %<!/book&!report>
2541 %<*book|report>
2542 \newcommand{\appendix}{\par
2543 \setcounter{chapter}{0}%
2544 \setcounter{section}{0}%
2545 \gdef\@chapapp{\appendixname}%
2546 \gdef\@chappos{}%
2547 \gdef\thechapter{\@Alph\c@chapter}}
2548 %</book|report>

```

8.4 パラメータの設定

■array と tabular 環境

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

```
2549 \setlength\arraycolsep{5\p@?}
```

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

```
2550 \setlength\tabcolsep{6\p@?}
```

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

```
2551 \setlength\arrayrulewidth{.4\p@}
```

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

```
2552 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2553 \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。

```
2554 \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。

```
2555 \setlength\fboxsep{3\p@?}
2556 \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。

```
2557 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2558 %<*book|report>
2559 \@addtoreset{equation}{chapter}
2560 \renewcommand\theequation
2561 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
2562 %</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。

```
2563 % \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。

```
2564 % \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。

```
2565 % \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。

```
2566 %<*!book&!report>
2567 \newcounter{figure}
2568 \renewcommand \thefigure {\@arabic\c@figure}
```

```

2569 %</*!book&!report>
2570 %<*book|report>
2571 \newcounter{figure}[chapter]
2572 \renewcommand \thefigure
2573     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2574 %</book|report>

```

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

```

\ext@figure 2575 \def\fps@figure{tbp}
2576 \def\ftype@figure{1}
\fnun@figure 2577 \def\ext@figure{lof}
2578 \def\fnun@figure{\figurename\nobreak\thefigure}

```

figure * 形式は段抜きのフロートです。

```

figure* 2579 \newenvironment{figure}%
2580     {\@float{figure}}%
2581     {\end@float}
2582 \newenvironment{figure*}%
2583     {\@dblfloat{figure}}%
2584     {\end@dblfloat}

```

■table 環境

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

```

2585 %</*!book&!report>
2586 \newcounter{table}
2587 \renewcommand\thetable{\@arabic\c@table}
2588 %</*!book&!report>
2589 %<*book|report>
2590 \newcounter{table}[chapter]
2591 \renewcommand \thetable
2592     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2593 %</book|report>

```

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

```

\ext@table 2594 \def\fps@table{tbp}
2595 \def\ftype@table{2}
\fnun@table 2596 \def\ext@table{lot}
2597 \def\fnun@table{\tablename\nobreak\thetable}

```

table * は段抜きのフロートです。

```

table* 2598 \newenvironment{table}%
2599     {\@float{table}}%
2600     {\end@float}
2601 \newenvironment{table*}%

```

```

2602          {\@dblfloat{table}}}%
2603          {\end@dblfloat}

```

8.6 キャプション

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

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

```

2604 \newlength\abovecaptionskip
2605 \newlength\belowcaptionskip
2606 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2607 \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。

```

2608 %<*.jspf>
2609 % \long\def\@makecaption#1#2{{\small
2610 %   \advance\leftskip10\jsc@mmm
2611 %   \advance\rightskip10\jsc@mmm
2612 %   \vskip\abovecaptionskip
2613 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2614 %   \ifdim \wd\@tempboxa >\hsize
2615 %     #1\hskip1\jsZw\relax #2\par
2616 %   \else
2617 %     \global \@minipagefalse
2618 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2619 %   \fi
2620 %   \vskip\belowcaptionskip}}
2621 \long\def\@makecaption#1#2{{\small
2622   \advance\leftskip .0628\linewidth
2623   \advance\rightskip .0628\linewidth
2624   \vskip\abovecaptionskip
2625   \sbox\@tempboxa{#1\zwspace#2}%
2626   \ifdim \wd\@tempboxa <\hsize \centering \fi
2627   #1\zwspace#2\par
2628   \vskip\belowcaptionskip}}
2629 %</!.jspf>
2630 %<*.jspf>
2631 \long\def\@makecaption#1#2{%
2632   \vskip\abovecaptionskip
2633   \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2634   \ifdim \wd\@tempboxa >\hsize
2635     {\small\sffamily

```

```

2636 \list{#1}{%
2637 \renewcommand{\makelabel}[1]{##1\hfil}
2638 \itemsep \z@
2639 \itemindent \z@
2640 \labelsep \z@
2641 \labelwidth 11\jsc@mmm
2642 \listparindent\z@
2643 \leftmargin 11\jsc@mmm\item\relax #2\endlist}
2644 \else
2645 \global \@minipagefalse
2646 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2647 \fi
2648 \vskip\belowcaptionskip}
2649 %</jspf>

```

9 フォントコマンド

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

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

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

```

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

```

```

2650 \newif\if@jsc@warnoldfontcmd
2651 \@jsc@warnoldfontcmdtrue
2652 \newif\if@jsc@warnoldfontcmdexception
2653 \@jsc@warnoldfontcmdexceptionfalse

```

`\jsc@DeclareOldFontCommand`

```

2654 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2655 \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2656 \DeclareOldFontCommand{#1}{%
2657 \bxjs@oldfontcmd{#1}#2%
2658 }{%
2659 \bxjs@oldfontcmd{#1}#3%
2660 }%
2661 }
2662 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%

```

```

2663 \ClassInfo\bxjs@clsname
2664 {Old font command '\string#1' is used!!\MessageBreak
2665   The first occurrence is}%
2666 }

```

`\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。

```

2667 \newcommand*{\allowoldfontcommands}{%
2668   \@jsc@warnoldfontcmdexceptiontrue}
2669 \newcommand*{\disallowoldfontcommands}{%
2670   \@jsc@warnoldfontcmdexceptionfalse}

2671 \let\bxjs@oldfontcmd@list\@empty
2672 \def\bxjs@oldfontcmd#1{%
2673   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2674 \def\bxjs@oldfontcmd@a#1#2{%
2675   \if@jsc@warnoldfontcmdexception\else
2676     \global\@jsc@warnoldfontcmdfalse
2677     \ifx#1\relax
2678       \global\let#1=t%
2679       \jsc@warnoldfontcmd{#2}%
2680     \fi
2681   \fi}
2682 \def\bxjs@warnoldfontcmd@final{%
2683 % \par
2684 \global\let\bxjs@warnoldfontcmd@final\@empty
2685 \let\@tempa\@empty
2686 \def\do##1{%
2687   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2688     \edef\@tempa{\@tempa \space\string##1}\fi}
2689 \bxjs@oldfontcmd@list
2690 \ifx\@tempa\@empty\else
2691   \ClassWarningNoLine\bxjs@clsname
2692   {Some old font commands were used in text:\MessageBreak
2693     \space\@tempa\MessageBreak
2694     You should note, that since 1994 LaTeX2e provides a\MessageBreak
2695     new font selection scheme called NFSS2 with several\MessageBreak
2696     new, combinable font commands. The
2697     class provides\MessageBreak
2698     the old font commands only for compatibility}
2699 \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。


```

2700 \def\bxjs@warnoldfontcmd@kick@final{%
2701   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2702 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2703 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2704 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2705 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2706 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2707 \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。

```

2708 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2709 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2710 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2711 \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 2712 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2713 \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。

```
2714 \newcommand\@pnumwidth{1.55em}
2715 \newcommand\@tocrmarg{2.55em}
2716 \newcommand\@dotsep{4.5}
2717 %<!book&!report>\setcounter{tocdepth}{2}
2718 %<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)

```
2719 \newdimen\jsc@tocl@width
2720 \newcommand{\tableofcontents}{%
2721 %<*book|report>
2722   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2723   \settowidth\@tempdima{\headfont\appendixname}%
2724   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2725   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2726   \if@twocolumn
2727     \@restonecoltrue\onecolumn
2728   \else
2729     \@restonecolfalse
2730   \fi
2731   \chapter*{\contentsname}%
2732   \@mkboth{\contentsname}{}%
2733 %</book|report>
2734 %<*&!book&!report>
```

```

2735 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2736 \settowidth\@tempdima{\headfont\appendixname}%
2737 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2738 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2739 \section*{\contentsname}%
2740 \@mkboth{\contentsname}{\contentsname}%
2741 %<!/book&!report>
2742 \@starttoc{toc}%
2743 %<book|report> \if@restonecol\twocolumn\fi
2744 }

```

\l@part 部の目次です。

```

2745 \newcommand*{\l@part}[2]{%
2746 \ifnum \c@tocdepth >-2\relax
2747 %<!book&!report> \addpenalty\@secpenalty
2748 %<book|report> \addpenalty{-\@highpenalty}%
2749 \addvspace{2.25em \@plus\p@}%
2750 \begingroup
2751 \parindent \z@
2752 % \@pnumwidth should be \@tocrmarg
2753 % \rightskip \@pnumwidth
2754 \rightskip \@tocrmarg
2755 \parfillskip -\rightskip
2756 {\leavevmode
2757 \large \headfont
2758 \setlength\@lnumwidth{4\jsZw}%
2759 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2760 \nobreak
2761 %<book|report> \global\@nobreaktrue
2762 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2763 \endgroup
2764 \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)

```

2765 %<*book|report>
2766 \newcommand*{\l@chapter}[2]{%
2767 \ifnum \c@tocdepth >\m@ne
2768 \addpenalty{-\@highpenalty}%
2769 \addvspace{1.0em \@plus\p@}
2770 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2771 \begingroup
2772 \parindent\z@
2773 % \rightskip\@pnumwidth
2774 \rightskip\@tocrmarg
2775 \parfillskip-\rightskip
2776 \leavevmode\headfont
2777 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi

```

```

2778      \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2779      \advance\leftskip\@lnumwidth \hskip-\leftskip
2780      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2781      \penalty\@highpenalty
2782      \endgroup
2783      \fi}
2784 %</book|report>

```

\l@section 節の目次です。

```

2785 %<*&book&!report>
2786 \newcommand*{\l@section}[2]{%
2787   \ifnum \c@tocdepth >\z@
2788     \addpenalty{\@secpenalty}%
2789     \addvspace{1.0em \@plus\p@}%
2790     \begingroup
2791     \parindent\z@
2792 %     \rightskip\@pnumwidth
2793     \rightskip\@tocrmarg
2794     \parfillskip-\rightskip
2795     \leavevmode\headfont
2796 %     % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2797     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2798     \advance\leftskip\@lnumwidth \hskip-\leftskip
2799     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2800     \endgroup
2801     \fi}
2802 %</!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。

```

2803 %<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] こども \jsc@tocl@width から決めるようにしてみました。(by ts)

```

\l@subparagraph 2804 %<*&book&!report>
2805 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2806 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2807 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2808 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2809 %
2810 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2811 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2812 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2813 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2814 %
2815 \newcommand*{\l@subsection}{%
2816   \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw

```

```

2817 \dottedtocline{2}{\@tempdima}{3\jsZw}}
2818 \newcommand*\l@subsubsection}{%
2819 \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2820 \dottedtocline{3}{\@tempdima}{4\jsZw}}
2821 \newcommand*\l@paragraph}{%
2822 \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2823 \dottedtocline{4}{\@tempdima}{5\jsZw}}
2824 \newcommand*\l@subparagraph}{%
2825 \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2826 \dottedtocline{5}{\@tempdima}{6\jsZw}}
2827 %<!/book&!report>
2828 %<*book|report>
2829 % \newcommand*\l@section}{\@dottedtocline{2}{3.8em}{3.2em}}
2830 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2831 % \newcommand*\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
2832 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
2833 \newcommand*\l@section}{%
2834 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2835 \dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2836 \newcommand*\l@subsubsection}{%
2837 \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2838 \dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2839 \newcommand*\l@subsubsection}{%
2840 \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2841 \dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2842 \newcommand*\l@paragraph}{%
2843 \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2844 \dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2845 \newcommand*\l@subparagraph}{%
2846 \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2847 \dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2848 %</book|report>

```

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

```

2849 \newdimen\@lnumwidth
2850 \def\numberline#1{\hb@xt@\@lnumwidth{#1\hfil}\hspace{0pt}}

```

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

[2018-06-23] デフォルトでは のようにベースラインになります。

これを変更可能にするため、\jsTocLine というマクロに切り出しました。例えば、仮想
 ボディの中央..... に変更したい場合は

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

とします。

```
2851 \def\jsTocLine{\leaders\hbox{%
2852   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2853 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2854   \vskip \z@ \@plus.2\p@?
2855   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2856     \parindent #2\relax\@afterindenttrue
2857     \interlinepenalty\@M
2858     \leavevmode
2859     \@lnumwidth #3\relax
2860     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2861     {#4}\nobreak
2862     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2863       \hfil\normalfont \normalcolor #5}\par}\fi}
```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```
2864 \newcommand{\listoffigures}{%
2865   %<*book|report>
2866   \if@twocolumn\@restonecoltrue\onecolumn
2867   \else\@restonecolfalse\fi
2868   \chapter*{\listfigurename}%
2869   \@mkboth{\listfigurename}{}%
2870   %</book|report>
2871   %<!*book&!report>
2872   \section*{\listfigurename}%
2873   \@mkboth{\listfigurename}{\listfigurename}%
2874   %</!*book&!report>
2875   \@starttoc{lof}%
2876   %<book|report> \if@restonecol\twocolumn\fi
2877 }
```

`\l@figure` 図目次の項目を出力します。

```
2878 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```
2879 \newcommand{\listoftables}{%
2880   %<*book|report>
2881   \if@twocolumn\@restonecoltrue\onecolumn
2882   \else\@restonecolfalse\fi
2883   \chapter*{\listtablename}%
2884   \@mkboth{\listtablename}{}%
2885   %</book|report>
2886   %<!*book&!report>
2887   \section*{\listtablename}%
2888   \@mkboth{\listtablename}{\listtablename}%
2889   %</!*book&!report>
```

```

2890 \starttoc{lot}%
2891 %<book|report> \if@restonecol\twocolumn\fi
2892 }

```

`\l@table` 表目次は図目次と同じです。

```

2893 \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。

```

2894 \newdimen\bibindent
2895 \setlength\bibindent{2\jsZw}

```

`thebibliography` 参考文献リストを出力します。

[2016-07-16] L^AT_EX 2.09 で使われていたフォントコマンドの警告を、文献スタイル (.bst) ではよく `\bf` がいまだに用いられることが多いため、`thebibliography` 環境内では例外的に出さないようにしました。

```

2896 \newenvironment{thebibliography}[1]{%
2897 \jsc@warnoldfontcmdexceptiontrue
2898 \global\let\presectionname\relax
2899 \global\let\postsectionname\relax
2900 %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2901 %<*kiyou>
2902 \vspace{1.5\baselineskip}
2903 \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2904 \vspace{0.5\baselineskip}
2905 %</kiyou>
2906 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2907 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2908 \list{\@biblabel{\@arabic\c@enumiv}}%
2909 {\settowidth\labelwidth{\@biblabel{#1}}%
2910 \leftmargin\labelwidth
2911 \advance\leftmargin\labelsep
2912 \@openbib@code
2913 \usecounter{enumiv}%
2914 \let\p@enumiv\empty
2915 \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2916 %<kiyou> \small
2917 \sloppy
2918 \clubpenalty4000
2919 \@clubpenalty\clubpenalty
2920 \widowpenalty4000%
2921 \sfcode`.\@m}
2922 {\def\@noitemerr
2923 {\@latex@warning{Empty `thebibliography' environment}}}%
2924 \endlist}

```

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

```
2925 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}
```

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

```
2926 \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。

```
2927 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

`\cite` 文献の番号を出力する部分は ltbibl.dtx で定義されていますが、コンマとカッコを和文

`\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必要

`\@citex` に応じて生かしてください。かつこの前後に入るグルーを `\jsInhibitGlue` で取りますので、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。

```
2928 % \def\@citex[#1]#2{\leavevmode
2929 %   \let\@citea\empty
2930 %   \@cite{\@for\@citeb:=#2\do
2931 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2932 %     \edef\@citeb{\expandafter\@firstofone\@citeb\empty}%
2933 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2934 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2935 %       \G@refundefinedtrue
2936 %       \@latex@warning
2937 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2938 %       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2939 % \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。

```
2940 % \DeclareRobustCommand\cite{\unskip
2941 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2942 %   \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2943 %     , \jsInhibitGlue\ #2\fi}}}$}
```

10.3 索引

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

```
2944 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2945   \if@twocolumn
2946     \onecolumn\@restonecolfalse
2947   \else
2948     \clearpage\@restonecoltrue
2949   \fi}
```



```

2950 \columnseprule.4pt \columnsep 2\jsZw
2951 \ifx\multicols\@undefined
2952 %<book|report> \twocolumn[\@makeschapterhead{\indexname}%
2953 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]%
2954 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2955 %<!book&!report> \twocolumn[\section*{\indexname}]%
2956 \else
2957 \ifdim\textwidth<\fullwidth
2958 \setlength{\evensidemargin}{\oddsidemargin}
2959 \setlength{\textwidth}{\fullwidth}
2960 \setlength{\linewidth}{\fullwidth}
2961 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}%
2962 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]%
2963 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2964 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]%
2965 \else
2966 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}%
2967 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]%
2968 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2969 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]%
2970 \fi
2971 \fi
2972 %<book|report> \mkboth{\indexname}{}%
2973 %<!book&!report> \mkboth{\indexname}{\indexname}%
2974 \plainifnotempty % \thispagestyle{plain}
2975 \parindent\z@
2976 \parskip\z@ \@plus .3\p@?\relax
2977 \let\item\@idxitem
2978 \raggedright
2979 \footnotesize\narrowbaselines
2980 }{
2981 \ifx\multicols\@undefined
2982 \if@restonecol\onecolumn\fi
2983 \else
2984 \end{multicols}
2985 \fi
2986 \clearpage
2987 }

```

`\@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 2988 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2989 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2990 \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。

```

2991 \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。

```
2992 \newcommand\seename{\if@english see\else →\fi}  
2993 \newcommand\alsoname{\if@english see also\else →\fi}
```

10.4 脚注

`\footnote` 和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため、`\footnotemark` `\inhibitglue` を入れることにします。pL^AT_EX の日付が 2016/09/03 より新しい場合は、このパッチが不要なのであてません。

パッチの必要性は「`\pltx@foot@penalty` が未定義か」で行う。`\inhibitglue` の代わりに `\jsInhibitGlue` を使う。

```
2994 \ifx\pltx@foot@penalty\@undefined  
2995 \let\footnotes@ve=\footnote  
2996 \def\footnote{\jsInhibitGlue\footnotes@ve}  
2997 \let\footnotemarks@ve=\footnotemark  
2998 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}  
2999 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

pT_EX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

[2016-10-08] TODO: 脚注番号が `newttext` や `newpertext` 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)。

```
3000 \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。

```
3001 % \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。

```
3002 \renewcommand{\footnoterule}{%
3003   \kern-2.6\p@? \kern-.4\p@
3004   \hrule width .4\columnwidth
3005   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
3006 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

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

```
3007 \long\def\@footnotetext{%
3008   \insert\footins\bgroup
3009     \normalfont\footnotesize
3010     \interlinepenalty\interfootnotelinepenalty
3011     \splittopskip\footnotesep
3012     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
3013     \hsize\columnwidth \@parboxrestore
3014     \protected@edef\@currentlabel{%
3015       \csname p@footnote\endcsname\@thefnmark
3016     }%
3017     \color@begingroup
3018     \@makefnmark
3019     \rule{\z@\footnotesep}{\columnwidth}%
3020     \futurelet\jsc@next\jsc@fo@t}
3021 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@fo@t
3022               \else \let\jsc@next\jsc@fo@t\fi \jsc@next}
3023 \def\jsc@fo@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3024 \def\jsc@fo@t#1{#1\jsc@@foot}
3025 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
3026   \ifx\pltx@foot@penalty\@undefined\else
3027     \ifhmode\null\fi
3028     \ifnum\pltx@foot@penalty=\z@\else
3029       \penalty\pltx@foot@penalty
3030       \pltx@foot@penalty\z@
3031     \fi
3032   \fi}
```

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

```
3033 \newcommand\@makefntext[1]{%
3034   \advance\leftskip 3\jsZw
3035   \parindent 1\jsZw
3036   \noindent
3037   \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。

```
3038 % \def\@xfootnotenext[#1]{%
3039 %   \begingroup
3040 %     \ifnum#1>\z@
3041 %       \csname c@\@mpfn\endcsname #1\relax
3042 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
3043 %     \else
3044 %       \unrestored@protected@xdef\@thefnmark{}%
3045 %     \fi
3046 %   \endgroup
3047 %   \@footnotetext}
```

ここまでのコードは JS クラスを踏襲する。

11 段落の頭へのグルー挿入禁止

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

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

```
3048 \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。

```
3049 \def\everyparhook{\jsInhibitGlueAtParTop}
3050 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
3051 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3052 \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。

```
3053 \ifx j\jsEngine
3054 \def\@inhibitglue{%
3055   \futurelet\@let@token\@@inhibitglue}
3056 \begingroup
3057 \let\GDEF=\gdef
3058 \let\CATCODE=\catcode
3059 \let\ENDGROUP=\endgroup
3060 \CATCODE`k=12
3061 \CATCODE`a=12
3062 \CATCODE`n=12
3063 \CATCODE`j=12
3064 \CATCODE`i=12
3065 \CATCODE`c=12
3066 \CATCODE`h=12
3067 \CATCODE`r=12
3068 \CATCODE`t=12
3069 \CATCODE`e=12
3070 \GDEF\KANJI@CHARACTER{kanji character }
3071 \ENDGROUP
3072 \def\@@inhibitglue{%
3073   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3074   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
3075   \def\jsc@ig@temp{#1}%
3076   \ifx\jsc@ig@temp\@empty
3077     \ifnum\the\inhibitxspcode`#2=2\relax
3078       \inhibitglue
3079     \fi
3080   \fi}
3081 \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。

```
3082 \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。

```
3083 \def\@doendpe{%
3084   \@endpetrue
3085   \def\par{%
3086     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3087   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] `minipage` 環境にも対策します。

```
3088 \def\@setminipage{%
3089   \@minipagetrue
3090   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3091 }
```

`\item` 命令の直後です。

```
3092 \def\@item[#1]{%
3093   \if@noparitem
3094     \@donoparitem
3095   \else
3096     \if@inlabel
3097       \indent \par
3098     \fi
3099     \ifhmode
3100       \unskip\unskip \par
3101     \fi
3102     \if@newlist
3103       \if@nobreak
3104         \@nbitem
3105       \else
3106         \addpenalty\@beginparpenalty
3107         \addvspace\@topsep
3108         \addvspace{-\parskip}%
3109       \fi
3110     \else
3111       \addpenalty\@itempenalty
3112       \addvspace\itemsep
3113     \fi
3114     \global\@inlabeltrue
3115   \fi
3116   \everypar{%
3117     \@minipagefalse
3118     \global\@newlistfalse
3119     \if@inlabel
3120       \global\@inlabelfalse
3121     {\setbox\z@\lastbox
```

```

3122      \ifvoid\z@
3123      \kern-\itemindent
3124      \fi}%
3125      \box\@labels
3126      \penalty\z@
3127      \fi
3128      \if@nobreak
3129      \@nobreakfalse
3130      \clubpenalty \@M
3131      \else
3132      \clubpenalty \@clubpenalty
3133      \everypar{\everyparhook}%
3134      \fi
3135      \everyparhook}%
3136      \if@noitemarg
3137      \@noitemargfalse
3138      \if@nmbrlist
3139      \refstepcounter\@listctr
3140      \fi
3141      \fi
3142      \sbox\@tempboxa{\makelabel{#1}}}%
3143      \global\setbox\@labels\hbox{%
3144      \unhbox\@labels
3145      \hskip \itemindent
3146      \hskip -\labelwidth
3147      \hskip -\labelsep
3148      \ifdim \wd\@tempboxa >\labelwidth
3149      \box\@tempboxa
3150      \else
3151      \hbox to\labelwidth {\unhbox\@tempboxa}%
3152      \fi
3153      \hskip \labelsep}%
3154      \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。

```

3155 \def\@afterheading{%
3156   \@nobreaktrue
3157   \everypar{%
3158     \if@nobreak
3159     \@nobreakfalse
3160     \clubpenalty \@M
3161     \if@afterindent \else
3162     {\setbox\z@\lastbox}%
3163     \fi
3164     \else
3165     \clubpenalty \@clubpenalty
3166     \everypar{\everyparhook}%
3167     \fi\everyparhook}}

```

「\everyparhook 用のパッチ処理」はここまで。

3168 \fi

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

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

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

```
3169 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none\else
3170 \AtEndOfPackage{%
3171 \def\@gnewline #1{%
3172   \ifvmode
3173     \@nolnerr
3174   \else
3175     \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
3176     \jsInhibitGlue \ignorespaces
3177   \fi}
3178 }
3179 \fi
```

12 いろいろなロゴ

\LaTeX 関連のロゴを作り直します。

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

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

※\小、\上小 の制御綴は定義しない。

```
3180 \if@jslogo
3181   \IfFileExists{jslogo.sty}{%
3182     \RequirePackage{jslogo}%
3183   }{%
3184     \ClassWarningNoLine\bxjs@clsname
3185       {The package 'jslogo' is not installed.\MessageBreak
3186       It is included in the recent release of\MessageBreak
```



```

3187         the 'jsclasses' bundle}
3188     }
3189 \fi

```

13 amsmath との衝突の回避

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

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

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

```

3190 %\let\ltx@ifnextchar\@ifnextchar
3191 %\def\ProvidesFile#1{%
3192 %   \begingroup
3193 %       \catcode`\ 10 %
3194 %       \ifnum \endlinechar<256 %
3195 %           \ifnum \endlinechar>\m@ne
3196 %               \catcode\endlinechar 10 %
3197 %       \fi
3198 %       \fi
3199 %       \@makeother\/%
3200 %       \@makeother\&%
3201 %       \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 3202 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3203 \newcommand{\postpartname}{\if@english\else 部\fi}
\postchaptername 3204 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3205 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3206 \newcommand{\presectionname}{}% 第
\postsectionname 3207 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3208 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3209 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3210 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname
\indexname

```

```

3211 \newcommand{\refname}{\if@english References\else 参考文献\fi}
3212 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3213 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 3214 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3215 %<jspf>\newcommand{\figurename}{Fig.~}
3216 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3217 %<jspf>\newcommand{\tablename}{Table~}

\appendixname

\abstractname 3218 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3219 \newcommand{\appendixname}{\if@english \else 付録\fi}
3220 %<!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。

```

3221 \@onlypreamble\bxjs@decl@Seireki@cmds
3222 \@tempswafalse
3223 \if p\jsEngine \@tempswatruetrue \fi
3224 \if n\jsEngine \@tempswatruetrue \fi
3225 \bxjs@cond\if@tempswa\fi{%
3226 % 欧文 8bitTeX の場合
3227 \newif\ifjsSeireki \jsSeirekitruetrue
3228 \def\bxjs@decl@Seireki@cmds{%
3229   \def\西暦{\jsSeirekitruetrue}%
3230   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3231 \def\Seireki{\jsSeirekitruetrue}
3232 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3233 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3234 \def\bxjs@iai{\noexpand~}
3235 }{%
3236 \newif\if西暦 \西暦 true
3237 \def\bxjs@decl@Seireki@cmds{%
3238   \def\西暦{\西暦 true}%
3239   \def\和暦{\西暦 false\bxjs@wareki@used}}
3240 \def\Seireki{\西暦 true}
3241 \def\Wareki{\西暦 false\bxjs@wareki@used}
3242 \def\bxjs@if@use@seireki{\bxjs@cond\if西暦\fi}
3243 \let\bxjs@iai\@empty
3244 }

```

```

3245 \bxjs@decl@Seireki@cmds
3246 \let\bxjs@unxp@\firstofone \let\bxjs@onxp@\firstofone
3247 \bxjs@test@engine\unexpanded{%
3248   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}

```

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

```

3249 \newif\ifbxjs@bxwareki@avail
3250 \IfFileExists{bxwareki.sty}{%
3251   \RequirePackage{bxwareki}[]%
3252   \bxjs@bxwareki@availtrue}{%

```

\bxjs@wareki@used 和暦が非対応の場合に警告を出す。

```

3253 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\@empty
3254 \else
3255   \bxjs@robust@def\bxjs@wareki@used{%
3256     \global\let\bxjs@wareki@used\@empty
3257     \ClassWarning\bxjs@clsname
3258       {Wareki mode is not supported, since\MessageBreak
3259         'bxwareki' is unavailable, found}}
3260   \g@addto@macro\bxjs@begin@document@hook{%
3261     \let\bxjs@wareki@used\@empty}
3262 \fi

```

\jyear 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。

```

3263 \ifbxjs@bxwareki@avail
3264   \let\jyear\warekiyear
3265   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3266     \newcount\heisei \heisei=\value{warekiyear}
3267   \fi

```

bxwareki が使えない場合は西暦表示にする。

```

3268 \else
3269   \edef\jyear{\the\year \bxjs@iai}
3270 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3271 \let\bxjs@next\relax
3272 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3273   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3274 \fi\fi
3275 \edef\bxjs@today{%
3276   \if@english
3277     \ifcase\month\or
3278       January\or February\or March\or April\or May\or June\or
3279       July\or August\or September\or October\or November\or December\fi
3280     \space\number\day, \number\year
3281   \else

```

```

3282 \ifx\bxjs@next\relax \expandafter\@firstoftwo
3283 \else \noexpand\bxjs@if@use@seireki
3284 \fi {%
3285 \number\year\bxjs@iai\bxjs@unxp{年}%
3286 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3287 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3288 }{\bxjs@next}%
3289 \fi}
3290 \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。

```

3291 \g@addto@macro\bxjs@begin@document@hook{%
3292 \ifx\bbl@jpn@maybekansuji\@undefined\else
3293 \bxjs@decl@Seireki@cmds
3294 \g@addto@macro\datejapanese{%
3295 \let\today\bxjs@today}%
3296 \fi}

```

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

```

3297 \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。

```

3298 %<slide>\pagestyle{empty}%
3299 %<article|report>\pagestyle{plain}%
3300 %<book>\pagestyle{headings}%
3301 \pagenumbering{arabic}
3302 \if@twocolumn
3303 \twocolumn
3304 \sloppy
3305 \flushbottom
3306 \else
3307 \onecolumn
3308 \raggedbottom
3309 \fi
3310 %<*slide>
3311 \renewcommand\familydefault{\sfdefault}
3312 \raggedright
3313 %</slide>

```

■BXJS 独自の追加処理 

フックを実行する。

3314 \bxjs@pre@jadriver@hook

和文ドライバのファイルを読み込む。

3315 \input{bxjsja-\bxjs@jadriver.def}

おしまい。

3316 %</class>

以上です。

付録 A 和文ドライバの仕様

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

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

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

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

以下で標準で用意されている和文ドライバの実装を示す。

```
3317 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

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

B.1 補助マクロ

```
3318 %<*minimal>
```

```
3319 %% 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.
```

```
3320 \def\DeclareJaTextFontCommand#1#2{%
```

```
3321   \DeclareRobustCommand#1[1]{%
```

```
3322     \relax
```

```
3323     \ifmmode \expandafter\nfss@text \fi
```

```
3324     {#2##1}}%
```

```
3325 }
```

```
\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。
```

```
3326 \def\DeclareJaMathFontCommand#1#2{%
```

```
3327   \DeclareRobustCommand#1[1]{%
```

```
3328     \relax
```

```
3329     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3330     \nfss@text{\fontfamily\familydefault
```

```
3331       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3332       #2##1}}%
```

```

3333 }%
3334 }

```

`\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。

```

3335 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3336 \@onlypreamble\bxjs@if@sf@default
3337 \def\bxjs@if@sf@default#1{%
3338   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3339   \g@addto@macro\bxjs@begin@document@hook{%
3340     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3341 }

```

`\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。

```

3342 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3343 \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。

```

3344 \def\jsLetHeadChar#1#2{%
3345   \begingroup
3346     \escapechar=`\ %
3347     \let\bxjs@tmpa={% brace-match-hack
3348       \bxjs@let@hchar@exp#2}%
3349   \endgroup
3350   \let#1\bxjs@g@tmpa}
3351 \def\bxjs@let@hchar@exp{%
3352   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3353 \def\bxjs@let@hchar@exp@a{%
3354   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3355   \bxjs@let@hchar@out\let\relax
3356 }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3357   \bxjs@let@hchar@out\let\space%
3358 }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3359   \bxjs@let@hchar@out\let\@backslashchar
3360 }{\bxjs@let@hchar@exp@b}}}%
3361 \def\bxjs@let@hchar@exp@b#1{%
3362   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3363 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3364   %\message{<#1#2>}%
3365   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```



```

3366 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3367 \bxjs@let@hchar@out\let\relax
3368 }{%else
3369 \expandafter\bxjs@let@hchar@exp
3370 }%
3371 }{%else
3372 \bxjs@let@hchar@chr#1%
3373 }}
3374 \def\bxjs@let@hchar@chr#1{%
3375 \bxjs@let@hchar@out\def{{#1}}}}
3376 \def\bxjs@let@hchar@out#1#2{%
3377 \global#1\bxjs@g@tmpa#2\relax
3378 \toks@{bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3379 \chardef\bxjs@let@hchar@csta=128
3380 \chardef\bxjs@let@hchar@cstb=192
3381 \chardef\bxjs@let@hchar@cstc=224
3382 \chardef\bxjs@let@hchar@cstd=240
3383 \chardef\bxjs@let@hchar@cste=248
3384 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3385 \def\bxjs@let@hchar@chr@ue#1{%
3386 \@tempcnta=`#1\relax
3387 %\message{\the\@tempcnta}%
3388 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3389 \bxjs@let@hchar@chr@ue@a#1%
3390 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3391 \bxjs@let@hchar@out\let\relax
3392 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3393 \bxjs@let@hchar@chr@ue@b
3394 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3395 \bxjs@let@hchar@chr@ue@c
3396 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3397 \bxjs@let@hchar@chr@ue@d
3398 }{%else
3399 \bxjs@let@hchar@out\let\relax
3400 }}}}}
3401 \def\bxjs@let@hchar@chr@ue@a#1{%
3402 \bxjs@let@hchar@out\def{{#1}}}}
3403 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3404 \bxjs@let@hchar@out\def{{#1#2}}}}
3405 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3406 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3407 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3408 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
3409 \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。

```
3410 \def\bxjs@let@hchar@chr@pp#1#2{%
3411   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3412 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3413   %\message{(\meaning#3:\meaning#4)}%
3414   \bxjs@cond\if#1k\fi{%
3415     \bxjs@let@hchar@out\def{#{#4}}%
3416   }{%else
3417     \bxjs@let@hchar@chr@ue#3#4%
3418   }}
3419 \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。

```
3420 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3421 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3422 \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。

```
3423 \@onlypreamble\bxjs@declarefontshape
3424 \ifjsWithupTeX
3425 \def\bxjs@declarefontshape{%
3426   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3427   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3428   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3429   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3430 }
3431 \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。

```
3432 \else
3433 \def\bxjs@declarefontshape{%
3434   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3435   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3436   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3437 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3438 }
3439 \def\bxjs@sizereference{jis}
3440 \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。

```

3441 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3442   \def\bxjs@tmpb{#5}}
3443 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3444 \else
3445 \expandafter\expandafter\expandafter\bxjs@next
3446 \expandafter\string\the\jfont\relax
3447 \fi
3448 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3449                 \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3450   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3451     \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。

```

3452 \begingroup
3453 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3454 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3455 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3456 % 幅が丁度 10pt なら補正は不要
3457 \ifdim\wd\z@=10pt
3458   \global\let\bxjs@scale\jsScale
3459 \else
3460 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3461   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3462   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3463   \bxjs@invscale\@tempdima\bxjs@tmpa
3464   \xdef\bxjs@scale{\strip@pt\@tempdima}
3465 \fi
3466 \endgroup
3467 %\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。

```

3468 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3469 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3470 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3471 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3472 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
3473 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
3474 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
3475 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
3476 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
3477 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
3478 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
3479 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
3480 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
3481 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
3482 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
3483 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
3484 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{ }
3485 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
3486 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
3487 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
3488 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
3489 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
3490 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
3491 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
3492 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
3493 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
3494 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
3495 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
3496 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
3497 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
3498 \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。

```

3499 \@ifl@t@r\fmtversion{2020/10/01}
3500   {\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
3501 \ifjsc@needspace@tch          % --- for 2020-02-02 or older BEGIN
3502 \ifx\@rmfamilyhook\@undefined % old
3503 \DeclareRobustCommand\rmfamily
3504   {\not@math@alphabet\rmfamily\mathrm
3505    \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3506 \DeclareRobustCommand\sffamily
3507   {\not@math@alphabet\sffamily\mathsf
3508    \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3509 \DeclareRobustCommand\ttfamily
3510   {\not@math@alphabet\ttfamily\mathtt
3511    \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3512 \g@addto@macro\bxjs@begin@document@hook{%
3513   \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3514     % my definitions above should have been overwritten, recover it!
3515     % \selectfont is executed twice but I don't care about speed...
3516     \expandafter\g@addto@macro\csname rmfamily \endcsname

```

```

3517     {\kanjifamily\mcdefault\selectfont}%
3518     \expandafter\g@addto@macro\csname sffamily \endcsname
3519     {\kanjifamily\gtdefault\selectfont}%
3520     \expandafter\g@addto@macro\csname ttfamily \endcsname
3521     {\kanjifamily\gtdefault\selectfont}%
3522     \fi}
3523 \else                                     % 2020-02-02
3524 \g@addto@macro\@rmfamilyhook
3525     {\prepare@family@series@update@kanji{mc}\mcdefault}
3526 \g@addto@macro\@sffamilyhook
3527     {\prepare@family@series@update@kanji{gt}\gtdefault}
3528 \g@addto@macro\@ttfamilyhook
3529     {\prepare@family@series@update@kanji{gt}\gtdefault}
3530 \fi
3531 \else % --- for 2020-02-02 or older END & for 2020-10-01 BEGIN
3532 \AddToHook{rmfamily}%
3533     {\prepare@family@series@update@kanji{mc}\mcdefault}
3534 \AddToHook{sffamily}%
3535     {\prepare@family@series@update@kanji{gt}\gtdefault}
3536 \AddToHook{ttfamily}%
3537     {\prepare@family@series@update@kanji{gt}\gtdefault}
3538 \fi % --- for 2020-10-01 END
3539 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3540 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3541 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3542 \fi
3543 \bxjs@if@sf@default{%
3544     \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3545 \selectfont

```

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

```

3546 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3547 \let\bxjs@parse@qh@a\undefined
3548 \let\bxjs@parse@qh@b\undefined

```

■パラメタの設定

```

3549 \prebreakpenalty\jis"2147=10000
3550 \postbreakpenalty\jis"2148=10000
3551 \prebreakpenalty\jis"2149=10000
3552 \inhibitxspcode`!=1
3553 \inhibitxspcode`〒=2
3554 \xspcode`+=3
3555 \xspcode`\%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3556 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%

```

```
3557 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
```

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

```
3558 \let\jsInhibitGlueAtParTop\@inhibitglue
```

`\jsResetDimen` は空のままでよい。

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

```
3559 \begingroup
```

```
3560 \catcode`\!=0
```

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

```
3561 \gdef\bxjs@ptex@dir{%
```

```
3562 !iftdir t%
```

```
3563 !else!ifydir y%
```

```
3564 !else ?%
```

```
3565 !fi!fi}
```

新版の pTeX で脚注番号の周囲の空きが過大になる現象への対処。

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

```
3566 % 古い \@makefnmark の定義
```

```
3567 \long\def\bxjs@tmpa{\hbox{%
```

```
3568 !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
```

```
3569 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
```

```
3570 \ifx\@makefnmark\bxjs@tmpa
```

```
3571 \long\gdef\@makefnmark{%
```

```
3572 !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
```

```
3573 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
```

```
3574 \fi
```

エスケープ文字の変更はここまで。

```
3575 \endgroup
```

■minijs パッケージのブロック やっておく。

```
3576 \@namedef{ver@minijs.sty}{}
```

B.3 pdfTeX 用の処理

```
3577 \else\ifx p\jsEngine
```

```
3578 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

```
3579 \@onlypreamble\bxjs@cjk@loaded
```

```
3580 \def\bxjs@cjk@loaded{%
```

```
3581 \def\@footnotemark{%
```

```
3582 \leavevmode
```

```
3583 \ifhmode
```

```

3584 \edef\x@sf{\the\spacefactor}%
3585 \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3586 \unkern\unkern
3587 \ifdim\lastskip>\z@ \unskip \fi
3588 \fi\fi
3589 \nobreak
3590 \fi
3591 \@makefnmark
3592 \ifhmode \spacefactor\x@sf \fi
3593 \relax}%
3594 \let\bxjs@cjk@loaded\relax
3595 }
3596 \g@addto@macro\bxjs@begin@document@hook{%
3597 \ifpackageloaded{CJK}{%
3598 \bxjs@cjk@loaded
3599 }{}%
3600 }

```

B.4 XeTeX 用の処理

```

3601 \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。

```

3602 \def\bxjs@let@hchar@chr#1{%
3603 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3604 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3605 \bxjs@let@hchar@chr@xe
3606 }{\bxjs@let@hchar@out\def{{#1}}}}
3607 \def\bxjs@let@hchar@chr@xe#1{%
3608 \lccode`0=`#1\relax
3609 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3610 \onlypreamble\bxjs@do@precisetext
3611 \ifx\XeTeXgenerateactualtext\undefined\else
3612 \def\bxjs@do@precisetext{%
3613 \XeTeXgenerateactualtext=\@ne}
3614 \fi

```

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

```

3615 \onlypreamble\bxjs@do@simplejasetup
3616 \def\bxjs@do@simplejasetup{%
3617 \ifnum\XeTeXinterchartokenstate>\z@
3618 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3619 \jsSimpleJaSetup
3620 \ClassInfo\bxjs@clsname
3621 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3622 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```
3623 \newcommand*{\jsSimpleJaSetup}{%
3624   \XeTeXlinebreaklocale "ja"\relax
3625   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3626   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

```
3627 \fi\fi\fi
```

`simplejasetup` オプションの処理。

```
3628 \ifx\bxjs@do@simplejasetup\@undefined\else
3629   \g@addto@macro\bxjs@begin@document@hook{%
3630     \ifbxjs@simplejasetup
3631       \bxjs@do@simplejasetup
3632     \fi}
3633 \fi

3634 \ifbxjs@precisetext
3635   \ifx\bxjs@do@precisetext\@undefined
3636     \ClassWarning\bxjs@clsname
3637       {The current engine does not support the\MessageBreak
3638        'precise-text' option\@gobble}
3639   \else
3640     \bxjs@do@precisetext
3641   \fi
3642 \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。

```
3643 \g@addto@macro\bxjs@begin@document@hook{%
3644   \ifx\jsInhibitGlueAtParTop\@empty
3645     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3646     \ifx\everyparhook\bxjs@tmpa
3647       \let\everyparhook\@empty
3648     \fi
3649   \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。

```
3650 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@modern

3651   \let\bxjs@everypar\everypar
3652   \newtoks\everypar
3653   \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。


```

3654 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3655 \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。

```

3656 \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。

```

3657 \@onlypreamble\bxjs@adjust@fancyhdr
3658 \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。

```

3659 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3660 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3661 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3662 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3663 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3664 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3665 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3666 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3667 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3668 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3669 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3670 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3671 \def\bxjs@tmpa{\rm\thepage\strut}%
3672 \def\bxjs@tmpb{\thepage\strut}%
3673 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3674 \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。

```

3675 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth
3676 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3677 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3678 }\bxjs@tmpa
3679 \fi\fi
3680 \PackageInfo\bxjs@clsname
3681 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3682 \def\bxjs@pagestyle@hook{%
3683 \@ifpackageloaded{fancyhdr}{%

```

```

3684 \bxjs@adjust@fancyhdr
3685 \global\let\bxjs@adjust@fancyhdr\relax
3686 }{}

```

\pagestyle にフックを入れ込む。

```

3687 \let\bxjs@org@pagestyle\pagestyle
3688 \def\pagestyle{%
3689 \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。

```

3690 \g@addto@macro\bxjs@begin@document@hook{%
3691 \bxjs@pagestyle@hook
3692 \global\let\bxjs@pagestyle@hook\relax}

```

■和文空白命令

```

3693 \ifbxjs@jaspace@cmd

```

\jaenspace 半角幅の水平空き。

```

3694 \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。

```

3695 \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。

```

3696 \def\_ {\zwspace}

```

\jaspace jlreq クラスと互換の命令。

```

3697 \DeclareRobustCommand*{\jaspace}[1]{%
3698 \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3699 \ClassError\bxjs@clsname
3700 {Unknown jaspac: #1}{\@eha}%
3701 \else
3702 \csname bxjs@jaspace@@#1\endcsname
3703 \fi}
3704 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3705 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3706 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3707 \fi

```

終わり。

```

3708 \fi

```

以上で終わり。

```

3709 %</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/\gttfamily`
- `\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。

```
3710 %<*standard>
3711 %% このファイルは日本語文字を含みます
3712 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3713 \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。

```
3714 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3715 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3716 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3717 \define@key{bxjsStd}{jis2004}[true]{%
3718   \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。

```
3719 \newif\ifbxjs@jp@units

units オプションの処理。
```

```

3720 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3721 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3722 \define@key{bxjsStd}{units}[true]{%
3723   \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。

```

3724 \let\bxjs@jp@font\@empty

font オプションの処理。
3725 \define@key{bxjsStd}{font}{%
3726   \edef\bxjs@jp@font{#1}}

```

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

```

3727 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue

strong-cmd オプションの処理。
3728 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
3729 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
3730 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}

```

実際の `japaram` の値を適用する。

```

3731 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3732 \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。

```

3733 \onlypreamble\bxjs@apply@mmiv
3734 \def\bxjs@apply@mmiv{%
3735   \g@addto@macro\@classoptionslist{,jis2004}
3736 % \@ifpackagewith 判定への対策
3737   \PassOptionsToPackage{jis2004}{otf}
3738   \global\let\bxjs@apply@mmiv\relax}
3739 \ifbxjs@jp@jismmiv \bxjs@apply@mmiv \fi

```

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

```

3740 \if j\jsEngine
3741   \bxjs@jp@unitsfalse
3742 \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。

```

3743 \ifbxjs@jp@units
3744   \IfFileExists{bxcalc.sty}{%
3745     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3746     \ifx\usepTeXunits\@undefined

```

```

3747 \PackageWarningNoLine\bxjs@clsname
3748 {Cannot support pTeX units (zw etc.), since\MessageBreak
3749 the package 'bxcalc' is too old}%
3750 \bxjs@jp@unitsfalse
3751 \else \usepTeXunits
3752 \fi
3753 }{%else
3754 \PackageWarningNoLine\bxjs@clsname
3755 {Cannot support pTeX units (zw etc.), since\MessageBreak
3756 the package 'bxcalc' is unavailable}%
3757 \bxjs@jp@unitsfalse
3758 }
3759 \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。

```

3760 \ifbxjs@jp@units
3761 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3762 \let\bxjs@parse@qh@a\@undefined
3763 \let\bxjs@parse@qh@b\@undefined
3764 \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。

```

3765 \ifbxjs@jp@units
3766 \def\bxjs@let@lenexpr#1#2{%
3767 \edef#1{#2}%
3768 \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3769 \else
3770 \def\bxjs@let@lenexpr{\edef}
3771 \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。

※

```

3772 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3773 \ifx\strong\@undefined\ifx\strongenv\@undefined
3774 \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3775 \DeclareTextFontCommand{\strong}{\strongenv}%

```

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

※\strongfontdeclare は試験的機能とする。

```

3776 \@ifdefinable{\strongfontdeclare}{\bxjs@protected\def\strongfontdeclare#1{%
3777 \bxjs@strong@font@declare@a#1,\@nil}}%

```

```

3778 \def\bxjs@strong@font@declare@a#1,#2\@nil{\def\bxjs@strong@font{#1}}%
3779 \def\bxjs@strong@font{\bfseries}%
3780 \fi\fi
3781 }\fi

```

■共通命令の実装 `\jQ` 等の「単位」系の共通命令を実装する。まず ϵ -TeX 拡張が使えるか
 検査する。

```

3782 \ifjsWitheTeX

```

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

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

```

\jH 3783 \@tempdima=0.25mm
3784 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3785 \let\jH\jQ

```

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

```

\trueH 3786 \ifjsc@mag
3787 \@tempdimb=\jsBaseFontSize\relax
3788 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3789 \@tempdima=2.5mm
3790 \bxjs@invscale\@tempdima\bxjs@tmpa
3791 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3792 \@tempdima=10pt
3793 \bxjs@invscale\@tempdima\bxjs@tmpa
3794 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3795 \else \let\trueQ\jQ \let\bxjs@truept\p@
3796 \fi
3797 \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。

```

3798 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3799 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3800 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3801 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3802 \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。

```

3803 \def\jafontsize#1#2{%
3804 \begingroup
3805 \bxjs@jafontsize@a{#1}%
3806 \@tempdimb\jsInverseScale\@tempdima
3807 \bxjs@jafontsize@a{#2}%

```

```

3808 \xdef\bxjs@g@tmpa{%
3809 \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3810 \endgroup\bxjs@g@tmpa}
3811 \def\bxjs@jafontsize@a#1{%
3812 \bxjs@parse@qh{#1}%
3813 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3814 \@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。

```
3815 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```

3816 \newcommand*\setkanjiskip[1]{%
3817 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3818 \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3819 \newcommand*\getkanjiskip{%
3820 \bxjs@kanjiskip}

```

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

```
3821 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

`\bxjs@enable@kanjiskip` 和文間空白の挿入を有効／無効にする。(pTeX 以外)

```

\bxjs@disable@kanjiskip 3822 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3823 \bxjs@kanjiskip@enabledtrue
3824 \bxjs@reset@kanjiskip}
3825 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3826 \bxjs@kanjiskip@enabledfalse
3827 \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。

```

3828 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3829 \ifbxjs@kanjiskip@enabled
3830 \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3831 \else \@tempskipa\z@
3832 \fi
3833 \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 3834 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3835 \newcommand*\setxkanjiskip[1]{%
```

```

3836 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3837 \bxjs@reset@xkanjiskip}

```

```
\bxjs@enable@xkanjiskip 3838 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3839 \bxjs@xkanjiskip}
```

```
\bxjs@reset@xkanjiskip
```

```

3840 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
3841 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3842   \bxjs@xkanjiskip@enabledtrue
3843   \bxjs@reset@xkanjiskip}
3844 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3845   \bxjs@xkanjiskip@enabledfalse
3846   \bxjs@reset@xkanjiskip}
3847 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3848   \ifbxjs@xkanjiskip@enabled
3849     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3850   \else \@tempskipa\z@
3851   \fi
3852   \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。

```

3853 \g@addto@macro\jsResetDimen{%
3854   \bxjs@reset@kanjiskip
3855   \bxjs@reset@xkanjiskip}
3856 \let\bxjs@apply@kanjiskip\relax
3857 \let\bxjs@apply@xkanjiskip\relax

```

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

\bxjs@adjust@jafont \jsJaFont に入っている和文フォント設定の値を“調整”して、その結果を \bxjs@tmpa に返す。#1 が f の場合は“非埋込 (noEmbed)”の設定が禁止される。この禁止の場合も含め、何か異常がある場合は \bxjs@tmpa は空になる。

```

3858 \@onlypreamble\bxjs@adjust@jafont
3859 \def\bxjs@adjust@jafont#1{%
3860   \ifx\jsJaFont\bxjs@auto
3861     \bxjs@get@kanjiEmbed
3862     \ifx\bxjs@jaEmbed\relax
3863       \let\bxjs@tmpa\@empty
3864     \else
3865       \let\bxjs@tmpa\bxjs@jaEmbed
3866       \ifx\bxjs@jaVariant\bxjs@hziv
3867         \bxjs@apply@mmiv
3868       \fi
3869     \fi
3870   \else
3871     \let\bxjs@tmpa\jsJaFont
3872   \fi
3873   \if f#1\ifx\bxjs@tmpa\bxjs@noEmbed
3874     \ClassWarningNoLine\bxjs@clsname
3875     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak

```



```

3876      not available on the current situation}%
3877      \let\bxjs@tmpa\@empty
3878      \fi\fi
3879  }
3880  \def\bxjs@@auto{auto}
3881  \def\bxjs@@noEmbed{noEmbed}
3882  \def\bxjs@@hziv{-04}

```

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

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

```

3883 \let\bxjs@jaEmbed\relax
3884 \let\bxjs@jaVariant\relax

```

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

```

3885 \@onlypreamble\bxjs@get@kanjiEmbed
3886 \def\bxjs@get@kanjiEmbed{%
3887   \begingroup\setbox\z@=\hbox{%
3888     \global\let\bxjs@tmpdo\@empty
3889     \def\bxjs@next##1##2##3{%
3890       \def##1###1##3 ###2\@nil###3\@nnil{%
3891         \ifx$###1$\gdef##2{###2}\fi}%
3892       \g@addto@macro\bxjs@tmpdo{%
3893         \expandafter##1\bxjs@tmpa\@nil##3 \@nil\@nnil}}%
3894       \bxjs@next\bxjs@tmpdo@a\bxjs@g@tmpa{kanjiEmbed}%
3895       \bxjs@next\bxjs@tmpdo@b\bxjs@g@tmpa{jaEmbed}%
3896       \bxjs@next\bxjs@tmpdo@c\bxjs@g@tmpb{kanjiVariant}%
3897       \bxjs@next\bxjs@tmpdo@d\bxjs@g@tmpb{jaVariant}%
3898   }%
3899   \global\let\bxjs@g@tmpa\relax
3900   \global\let\bxjs@g@tmpb\relax
3901   \endlinechar\m@ne
3902   \let\do\@makeother\dospecials
3903   \catcode32=10 \catcode12=10 %form-feed
3904   \let\bxjs@tmpa\@empty
3905   \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3906   \ifeof\@inputcheck\else
3907     \read\@inputcheck to\bxjs@tmpa
3908     \closein\@inputcheck
3909   \fi
3910   \ifx\bxjs@tmpa\@empty\else
3911     \openin\@inputcheck="\bxjs@tmpa"\relax
3912     \@tempswatrue
3913     \loop\if@tempswa
3914       \read\@inputcheck to\bxjs@tmpa
3915       \bxjs@tmpdo
3916       \ifeof\@inputcheck \@tempswafalse \fi
3917     \repeat
3918   \fi

```

```

3919 } \endgroup
3920 \let\bxjs@jaEmbed\bxjs@g@tmpa
3921 \let\bxjs@jaVariant\bxjs@g@tmpb
3922 }

```

`\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。

```

3923 \@onlypreamble\bxjs@resolve@jafont@paren
3924 \def\bxjs@resolve@jafont@paren#1{%
3925   \def\bxjs@tmpb{\let#1}%
3926   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\@nnil#1}
3927 \@onlypreamble\bxjs@resolve@jafont@paren@a
3928 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\@nnil#5{%
3929   \ifx\relax#4\relax \bxjs@tmpb#5%
3930   \else
3931     \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3932     \bxjs@tmpb\bxjs@tmpa
3933   \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。

```

3934 \newcommand*\jachar[1]{%
3935   \begingroup
3936     \jsLetHeadChar\bxjs@tmpa{#1}%
3937     \ifx\bxjs@tmpa\relax
3938       \ClassWarningNoLine\bxjs@clsname
3939         {Illegal argument given to \string\jachar}%
3940     \else
3941       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3942     \fi
3943   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3944 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3945 \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。

```

3946 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3947 \@onlypreamble\bxjs@fix@hyperref@unicode
3948 \def\bxjs@fix@hyperref@unicode#1{%

```

```

3949 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3950 \@namedef{KV@Hyp@bxjs/hook}##1{%
3951   \KV@Hyp@unicode{##1}%
3952   \def\KV@Hyp@unicode####1{%
3953     \expandafter\ifx\csname if##1\expandafter\endcsname
3954       \csname if####1\endcsname\else
3955       \ClassWarningNoLine{bxjs@clsname
3956         {Blcoked hyperref option 'unicode=####1'}}%
3957     \fi
3958   }%
3959 }%
3960 }

```

`\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。

```

3961 \@onlypreamble\jsCheckHyperrefUnicode
3962 \let\jsCheckHyperrefUnicode\@empty
3963 \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。

```

3964 \@onlypreamble\bxjs@check@hyperref@unicode
3965 \def\bxjs@check@hyperref@unicode#1{%
3966   \g@addto@macro\jsCheckHyperrefUnicode{%
3967     \@tempwafalse
3968     \begingroup
3969       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3970       \aftergroup\@tempwattrue \fi
3971       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3972         \csname if#1\endcsname
3973       \aftergroup\@tempwattrue \fi
3974     \endgroup
3975     \if@tempwa\else
3976       \ClassError{bxjs@clsname
3977         {The value of hyperref 'unicode' key is not suitable\MessageBreak
3978           for the present engine (must be #1)}}%
3979       {\@ehc}%
3980     \fi}}

```

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

```

3981 \@onlypreamble\bxjs@urgent@special
3982 \@onlypreamble\bxjs@urgent@special@a

```

LaTeX 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。

```

3983 \ifbxjs@old@hook@system
3984 \def\bxjs@urgent@special#1{%
3985   \AtBeginDvi{\special{#1}}%
3986   \g@addto@macro\bxjs@begin@document@hook{%
3987     \ifpackageloaded{atbegshi}{%
3988       \begingroup
3989       \toks\z@{\special{#1}}%

```

```

3990      \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3991      \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw}%
3992      \endgroup
3993      }{}%
3994      }%
3995 }

```

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。

```

3996 \else
3997   \def\bxjs@urgent@special#1{%
3998     \bxjs@urgent@special@a
3999     \AddToHook{shipout/firstpage}[pxjahyper/enc]{\special{#1}}
4000   \def\bxjs@urgent@special@a{%
4001     \DeclareHookRule{shipout/firstpage}{pxjahyper/enc}{<}{hyperref}%
4002     \global\let\bxjs@urgent@special@a\relax}
4003 \fi

```

C.4 pTeX 用設定

```
4004 \if j\jsEngine
```

■ 共通命令の実装

```

4005 \def\bxjs@apply@kanjiskip{%
4006   \kanjiskip\@tempkipa}
4007 \def\bxjs@apply@xkanjiskip{%
4008   \xkanjiskip\@tempkipa}

```

\jaJaChar のサブマクロ。

```

4009 \def\bxjs@jachar#1{%
4010   \bxjs@jachar@a#1...\@nil}
4011 \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。

```
4012   \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。

```

4013   \else\ifx.#3%
4014     \@tempcnta`#1 \multiply\@tempcnta64
4015     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
4016     \bxjs@jachar@b
4017   \else\ifx.#4%
4018     \@tempcnta`#1 \multiply\@tempcnta64
4019     \advance\@tempcnta`#2 \multiply\@tempcnta64
4020     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
4021     \bxjs@jachar@b
4022   \else

```

```

4023 \tempcnta`#1 \multiply\tempcnta64
4024 \advance\tempcnta`#2 \multiply\tempcnta64
4025 \advance\tempcnta`#3 \multiply\tempcnta64
4026 \advance\tempcnta`#4 \advance\tempcnta-"3C82080
4027 \bxjs@jachar@b
4028 \fi\fi\fi}

```

符号値が \tempcnta の和文文字を出力する処理。

```

4029 \ifjsWithupTeX
4030 \def\bxjs@jachar@b{\kchar\tempcnta}
4031 \else
4032 \def\bxjs@jachar@b{%
4033 \ifx\bxUInt\@undefined\else
4034 \bxUInt{\tempcnta}%
4035 \fi}
4036 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

4037 \ifbxjs@jaspace@cmd
4038 \def\jathinspace{\hskip\xkanjiskip}
4039 \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。

```

4040 \ifbxjs@jp@jismmiv
4041 \PassOptionsToPackage{prefer2004jis}{pxchfon}
4042 \PassOptionsToPackage{prefer2004jis}{pxbabel}
4043 \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。

```

4044 \let\bxjs@jafont@paren\@firstofone
4045 \let\bxjs@tmpa\jsJaFont
4046 \ifx\bxjs@tmpa\bxjs@@auto
4047 \let\bxjs@tmpa\@empty
4048 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
4049 \def\bxjs@tmpa{noembed}
4050 \fi\fi
4051 \bxjs@resolve@jafont@paren\bxjs@tmpa
4052 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4053 \ifx\bxjs@tmpa\@empty\else
4054 \edef\bxjs@next{%
4055 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4056 }\bxjs@next
4057 \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……。)

```

4058 \begingroup
4059   \global\let\bxjs@g@tmpa\relax
4060   \catcode`\|=0 \catcode`\|=12
4061   |def|bxjs@tmpdo@a#1|@nil{%
4062     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4063   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4064     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4065   |catcode`\|=0 \catcode`\|=12
4066   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4067     \ifx$#2$\else
4068       \xdef\bxjs@g@tmpa{%
4069         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4070     \fi}
4071 \@firstofone{%
4072   \catcode10=12 \endlinechar\m@ne
4073   \let\do\@makeother \dospecials \catcode32=10
4074   \openin\@inputcheck=otf.sty\relax
4075   \@tempwattrue
4076   \loop\if@tempwa
4077     \ifeof\@inputcheck \@tempwafalse \fi
4078     \if@tempwa
4079       \read\@inputcheck to\bxjs@next
4080       \expandafter\bxjs@tmpdo\bxjs@next\@nil
4081     \fi
4082   \repeat
4083   \closein\@inputcheck
4084 \endgroup}
4085 \bxjs@g@tmpa

```

■hyperref 対策 unicode にしてはいけない。

```

4086 \ifbxjs@hyperref@enc
4087   \bxjs@check@hyperref@unicode{false}
4088 \fi

```

tounicode special 命令を出力する。

```

4089 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4090   \else\ifjsWithpTeXng T\else F\fi\fi T%
4091   \IfFileExists{pxjahyper-enc.sty}{\@tempwattrue}{\@tempwafalse}
4092   \if@tempwa
4093     \RequirePackage{pxjahyper-enc}[2020/10/05]%v0.6
4094     \ifbxjs@bigcode\else \suppressbigcode \fi
4095   \else

```

```

4096 \ifnum\jis"2121="A1A1 %euc
4097 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4098 \else\ifnum\jis"2121="8140 %sjis
4099 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4100 \else\ifnum\jis"2121="3000 %uptex
4101 \ifbxjs@bigcode
4102 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4103 \PassOptionsToPackage{bigcode}{pxjahyper}
4104 \else
4105 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4106 \PassOptionsToPackage{nobigcode}{pxjahyper}
4107 \fi
4108 \fi\fi\fi
4109 \let\bxToUnicodeSpecialDone=t
4110 \fi
4111 \fi

```

■和文数式ファミリー 和文数式ファミリーは既定で有効とする。すなわち `enablejfam=false` 以外の場合は `@enablejfam` を真にする。

```

4112 \ifx f\bxjs@enablejfam\else
4113 \@enablejfamtrue
4114 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

4115 \if@enablejfam
4116 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4117 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4118 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4119 \jfam\symmincho
4120 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4121 \g@addto@macro\bxjs@begin@document@hook{%
4122 \ifx\reDeclareMathAlphabet\undefined\else
4123 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4124 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4125 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4126 \fi}
4127 \fi

```

C.5 pdf_TE_X 用設定 : CJK + bxcjkatype

```

4128 \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。

```

4129 \bxjs@adjust@jafont{f}

```

```

4130 \let\bxjs@jafont@paren\@firstofone
4131 \bxjs@resolve@jafont@paren\bxjs@tmpa
4132 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4133 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4134 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4135   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4136 \fi
4137 \edef\bxjs@next{%
4138   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4139 }\bxjs@next
4140 \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。

```

4141 \ifbxjs@hyperref@enc
4142   \PassOptionsToPackage{unicode}{hyperref}
4143 \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。

```

4144 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
4145 \begingroup
4146   \CJK@input{UTF8.bdg}
4147 \endgroup
4148 \g@addto@macro\pdfstringdefPreHook{%
4149   \@nameuse{CJK@UTF8Binding}%
4150 }
4151 \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。

```

4152 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
4153 \g@addto@macro\pdfstringdefPreHook{%
4154   \ifx~\bxjs@@CJKtilde
4155     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4156     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4157     \let~\@empty
4158   \fi
4159 }
4160 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4161 \def\bxjs@@tildecmd{~}
4162 \def\bxjs@LetUnexpandableSpace#1{%
4163   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4164     \bxjs@org@LetUnexpandableSpace#1%
4165   \fi}
4166 \fi

```


■共通命令の実装

```
4167 \newskip\jsKanjiSkip
4168 \newskip\jsXKanjiSkip
4169 \ifx\CJKecglue\@undefined
4170   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4171 \fi
4172 \let\autospacing\bxjs@enable@kanjiskip
4173 \let\noautospacing\bxjs@disable@kanjiskip
4174 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4175 \def\bxjs@apply@kanjiskip{%
4176   \jsKanjiSkip\@tempskipa
4177   \let\CJKglue\bxjs@CJKglue}
4178 \let\autoxspacing\bxjs@enable@xkanjiskip
4179 \let\noautoxspacing\bxjs@disable@xkanjiskip
4180 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4181 \def\bxjs@apply@xkanjiskip{%
4182   \jsXKanjiSkip\@tempskipa
4183   \let\CJKecglue\bxjs@CJKecglue}
```

\jachar のサブマクロの実装。

```
4184 \def\bxjs@jachar#1{%
4185   \CJKforced{#1}}
```

和欧文間空白の命令 \jathinspace の実装。

```
4186 \ifbxjs@jaspace@cmd
4187   \protected\def\jathinspace{\CJKecglue}
4188 \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。

```
4189 \ifx t\bxjs@enablejfam
4190   \ClassWarningNoLine\bxjs@clsname
4191     {You cannot use 'enablejfam=true', since the\MessageBreak
4192     CJK package does not support Japanese math}
4193 \fi
```

C.6 Xe_ΛTeX 用設定：xeCJK + zxjatype

```
4194 \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。

```
4195 \RequirePackage{zxjatype}
4196 \PassOptionsToPackage{no-math}{fontspec}%!
4197 \PassOptionsToPackage{xetex}{graphicx}%!
4198 \PassOptionsToPackage{xetex}{graphics}%!
4199 \ifx\zxJaFamilyName\@undefined
4200   \ClassError\bxjs@clsname
4201     {xeCJK or zxjatype is too old}\@ehc
```

4202 \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。

```
4203 \bxjs@adjust@jafont{f}
4204 \let\bxjs@jafont@paren\gobble
4205 \bxjs@resolve@jafont@paren\bxjs@tmpa
4206 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4207 \ifx\bxjs@tmpa\@empty
4208   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4209   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4210 \else
4211   \edef\bxjs@next{%
4212     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4213   }\bxjs@next
4214 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X_ƎTeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、X_ƎTeX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「X_ƎTeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```
4215 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4216   \ifbxjs@hyperref@enc
4217     \PassOptionsToPackage{unicode}{hyperref}
4218   \fi
4219 \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。

```
4220 \AtEndOfPackage{%
4221 \def\@gnewline #1{%
```

```

4222 \ifvmode \@nolnerr
4223 \else
4224 \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
4225 \nobreak \hskip-1sp\hskip1sp\relax
4226 \ignorespaces
4227 \fi}
4228 }

```

■ 共通命令の実装

```

4229 \newskip\jsKanjiSkip
4230 \newskip\jsXKanjiSkip
4231 \ifx\CJKecglue\@undefined
4232 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4233 \fi
4234 \let\autospacing\bxjs@enable@kanjiskip
4235 \let\noautospacing\bxjs@disable@kanjiskip
4236 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4237 \def\bxjs@apply@kanjiskip{%
4238 \jsKanjiSkip\@tempskipa
4239 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4240 \let\autoxspacing\bxjs@enable@xkanjiskip
4241 \let\noautoxspacing\bxjs@disable@xkanjiskip
4242 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4243 \def\bxjs@apply@xkanjiskip{%
4244 \jsXKanjiSkip\@tempskipa
4245 \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。

```

4246 \ifx\mcfamily\@undefined
4247 \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4248 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4249 \fi

```

\jachar のサブマクロの実装。

```

4250 \def\bxjs@jachar#1{%
4251 \xeCJKDeclareCharClass{CJK}{`#1}\relax
4252 #1}

```

\jathinspace の実装。

```

4253 \ifbxjs@jaspace@cmd
4254 \protected\def\jathinspace{\CJKecglue}
4255 \fi

```

■ 和文数式ファミリ 和文数式ファミリは既定で無効とする。すなわち enablejfam=true の場合にのみ @enablejfam を真にする。

```

4256 \ifx t\bxjs@enablejfam
4257 \@enablejfamtrue
4258 \fi

```

実際に和文用の数式ファミリの設定を行う。

※ FIXME: 要検討。

```
4259 \if@enablejfam
4260   \xeCJKsetup{CJKmath=true}
4261 \fi
```

C.7 LuaTeX 用設定 : LuaTeX-ja

```
4262 \else\if 1\jsEngine
```

■ LuaTeX-ja パッケージの読込 `luatexja` とともに `luatexja-fontspec` パッケージを読み込む。

`luatexja` は自前の `\zw` (これは実際の現在和文フォントに基づく値を返す) を定義するので、`\zw` の定義を消しておく。なお、レイアウト定義の「全角幅」は「規定」に基づく `\jsZw` であることに注意が必要。

※ 1.0b 版から「graphics パッケージに `pdftex` オプションを渡す」処理を行っていたが、1.4 版で廃止された。

```
4263 \let\zw\@undefined
4264 \RequirePackage{luatexja}
4265 \edef\bxjs@next{%
4266   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4267 }\bxjs@next
```

フォント代替の明示的定義。

```
4268 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4269 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4270 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4271 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4272 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4273 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4274 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4275 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4276 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4277 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4278 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4279 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4280 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4281 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4282 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4283 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4284 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4285 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4286 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4287 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4288 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4289 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4290 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
```

```

4291 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4292 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4293 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4294 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4295 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4296 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4297 \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。

```

4298 \bxjs@adjust@jafont{t}
4299 \ifx\bxjs@tmpa\bxjs@@noEmbed
4300   \def\bxjs@tmpa{noembed}
4301 \fi
4302 \let\bxjs@jafont@paren\@gobble
4303 \bxjs@resolve@jafont@paren\bxjs@tmpa
4304 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4305 \ifx\bxjs@tmpa\@empty
4306   \defaultjfontfeatures{ Kerning=Off }
4307   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
Regular.otf}
4308   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
Medium.otf}
4309 \else
4310   \edef\bxjs@next{%
4311     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4312   }\bxjs@next
4313 \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。

```

4314 \@ifpackagelater{luatexja}{2016/03/31}{ }{%else
4315 \DeclareRobustCommand\rmfamily
4316   {\not@math@alphabet\rmfamily\mathrm
4317   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4318 \DeclareRobustCommand\sffamily
4319   {\not@math@alphabet\sffamily\mathsf
4320   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4321 \DeclareRobustCommand\ttfamily
4322   {\not@math@alphabet\ttfamily\mathtt
4323   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4324 }
4325 \long\def\jttdefault{\gtdefault}
4326 \unless\ifx\@ltj@match@familytrue\@undefined
4327   \@ltj@match@familytrue
4328 \fi

```

```

4329 \g@addto@macro\bxjs@begin@document@hook{%
4330   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4331   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4332   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4333 \bxjs@if@sf@default{%
4334   \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

4335 % 次の3つは既定値の通り
4336 %\ltjsetparameter{prebreakpenalty={`',10000}}
4337 %\ltjsetparameter{postbreakpenalty={`",10000}}
4338 %\ltjsetparameter{prebreakpenalty={`",10000}}
4339 \ltjsetparameter{jaxspmode={`!,1}}
4340 \ltjsetparameter{jaxspmode={`〒,2}}
4341 \ltjsetparameter{alxspmode={`+,3}}
4342 \ltjsetparameter{alxspmode={`\%,3}}

```

■段落頭でのグルー挿入禁止 基本的に現状の `ltjs*` クラスの処理に合わせる。

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の `LuaTEX-ja` で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合に備えて同等のものを用意する。

```

4343 \ifx\ltjfakeparbegin\undefined
4344   \protected\def\ltjfakeparbegin{%
4345     \ifhmode
4346       \relax\directlua{%
4347         luatexja.jfmglue.create_beginpar_node()}}
4348   \fi}
4349 \fi

ltjs* クラスの定義と同等になるようにパッチを当てる。

4350 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none
4351 \begingroup
4352   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4353   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4354   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
4355     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4356   \directlua{
4357     local function patchcmd(cs, code, from, to)
4358       tex.sprint(code:gsub(from:gsub("%W", "\\%\\%0"), "%0"..to)
4359         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4360     end
4361     patchcmd(\@@xsect, [[\meaning\xsect]],
4362       \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4363     patchcmd(\@@item, [[\meaning\@item]],
4364       \bxjs@tmpa, \@{\ltjfakeparbegin})}
4365 \endgroup
4366 \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。

```
4367 \ifbxjs@hyperref@enc
4368   \PassOptionsToPackage{unicode}{hyperref}
4369   \bxjs@check@hyperref@unicode{true}
4370 \fi
```

■共通命令の実装

```
4371 \protected\def\autospacing{%
4372   \ltjsetparameter{autospacing=true}}
4373 \protected\def\noautospacing{%
4374   \ltjsetparameter{autospacing=false}}
4375 \protected\def\autoxspacing{%
4376   \ltjsetparameter{autoxspacing=true}}
4377 \protected\def\noautoxspacing{%
4378   \ltjsetparameter{autoxspacing=false}}
4379 \def\bxjs@apply@kanjiskip{%
4380   \ltjsetparameter{kanjiskip={\@tempskipa}}}
4381 \def\bxjs@apply@xkanjiskip{%
4382   \ltjsetparameter{xkanjiskip={\@tempskipa}}}
```

\jachar のサブマクロの実装。

```
4383 \def\bxjs@jachar#1{%
4384   \ltjjachar`#1\relax}
```

\jathinspace の実装。

```
4385 \ifbxjs@jaspace@cmd
4386   \protected\def\jathinspace{%
4387     \hspace\ltjgetparameter{xkanjiskip}\relax}
4388 \fi
```

■和文数式ファミリー LuaTeX-jd では和文数式ファミリーは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablejfam は常に真になる。

```
4389 \ifx f\bxjs@enablejfam
4390   \ClassWarningNoLine\bxjs@clsname
4391     {You cannot use 'enablejfam=false', since the\MessageBreak
4392       LuaTeX-jd always provides Japanese math families}
4393 \fi
```

C.8 共通処理 (2)

```
4394 \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 4395 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4396   \DeclareJaTextFontCommand{\textmc}{\mcfamily}
```

```
4397 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4398 \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。

```
4399 \ifx\mathmc\@undefined
4400   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4401   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4402 \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)。

```
4403 \ifbxjs@jaspace@cmd
4404   \bxjs@protected\def\bxjs@choice@jathinspace{%
4405     \relax\ifmmode \mskip\medmuskip
4406     \else \jathinspace\ignorespaces
4407     \fi}
4408   \jsAtEndOfClass{%
4409     \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4410     \else \def\>{\protect\bxjs@choice@jathinspace}%
4411     \fi}
4412 \fi
```

■和文・和欧文間空白の初期値

```
4413 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4414 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4415 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4416 \fi
```

以上で終わり。

```
4417 %</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。

```
4418 %<*modern>
4419 \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。


```

4420 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4421 \def\encodingdefault{T1}%
4422 \input{t1enc.def}%
4423 \fontencoding\encodingdefault\selectfont
4424 \fi

```

基本フォントを Latin Modern フォントファミリに変更する。

※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。

```

4425 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4426 \renewcommand{\rmdefault}{lmr}
4427 \renewcommand{\sfdefault}{lmss}
4428 \renewcommand{\ttdefault}{lmtt}
4429 \fi

```

大型演算子用の数式フォントの設定。

※`amsmath` パッケージと同等にする。

```

4430 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4431   <-7.5>cmex7<7.5-8.5>cmex8%
4432   <8.5-9.5>cmex9<9.5->cmex10}{}%
4433 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

```

`amsmath` 読み込時に上書きされるのを防ぐ。

```

4434 \def\cmex@opt{10}

```

D.2 fixltx2e 読み込

※`fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```

4435 \ifx\@IncludeInRelease\@undefined
4436 \RequirePackage{fixltx2e}
4437 \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。

```

4438 \RequirePackage{bxjcsjkat}

```

D.4 完了

おしまい。

```

4439 %</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。

```
4440 %<*pandoc>
```

```
4441 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4442 \RequirePackage{bxjspandoc}
```

\bxjs@endpreamble@hook etoolbox の \AtEndPreamble で実行される BXJS クラス用のフック。

※ ε -TeX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```
4443 \@onlypreamble\bxjs@endpreamble@hook
```

```
4444 \let\bxjs@endpreamble@hook\empty
```

```
4445 \ifjsWiththeTeX
```

```
4446 \RequirePackage{etoolbox}[2010/08/21]% v2.0
```

```
4447 \AtEndPreamble{\bxjs@endpreamble@hook}
```

```
4448 \fi
```

E.2 和文ドライバパラメタ

keyval のファミリーは bxjsPan とする。

\ifbxjs@jp@fix@strong 重要要素を補正するか。

```
4449 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue
```

fix-strong オプションの処理。

```
4450 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
```

```
4451 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
```

```
4452 \define@key{bxjsPan}{fix-strong}[true]{%
```

```
4453 \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。

```
4454 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue
```

fix-code オプションの処理。

```
4455 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
```

```
4456 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
```

```
4457 \define@key{bxjsPan}{fix-code}[true]{%
```

```
4458 \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。

```
4459 \chardef\bxjs@jp@strong=0
```

strong オプションの処理。

```
4460 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
```

```
4461 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
```

```

4462 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }
4463 \define@key{bxjsPan}{strong}{%
4464   \bxjs@set@keyval{strong}{#1}{}}

```

\ifbxjs@jp@or@indent プレアンブルでのレイアウト上書きを許可するか。既定値は真。

```

\ifbxjs@jp@or@secnumdepth 4465 \newif\ifbxjs@jp@or@indent \bxjs@jp@or@indenttrue
\ifbxjs@jp@or@block@heading 4466 \newif\ifbxjs@jp@or@secnumdepth \bxjs@jp@or@secnumdepthtrue
4467 \newif\ifbxjs@jp@or@block@heading \bxjs@jp@or@block@headingtrue

```

クラスで pandoc+ が指定された場合、内部和文パラメタ `_plus` が和文ドライバに渡される。この場合、レイアウト上書きを禁止する。

※ `_plus` は必ずパラメタ列の先頭にあるので、個別のパラメタ設定の方が常に優先される。

```

4468 \define@key{bxjsPan}{_plus}[]{}
4469   \bxjs@jp@or@indentfalse
4470   \bxjs@jp@or@secnumdepthfalse
4471   \bxjs@jp@or@block@headingfalse}

```

レイアウト上書き許可オプション (`or-indent`・`or-secnumdepth`・`or-block-heading`) の処理。

```

4472 \let\bxjs@kv@orindent@true\bxjs@jp@or@indenttrue
4473 \let\bxjs@kv@orindent@false\bxjs@jp@or@indentfalse
4474 \define@key{bxjsPan}{or-indent}[true]{%
4475   \bxjs@set@keyval{orindent}{#1}{}}
4476 \let\bxjs@kv@orsecnumdepth@true\bxjs@jp@or@secnumdepthtrue
4477 \let\bxjs@kv@orsecnumdepth@false\bxjs@jp@or@secnumdepthfalse
4478 \define@key{bxjsPan}{or-secnumdepth}[true]{%
4479   \bxjs@set@keyval{orsecnumdepth}{#1}{}}
4480 \let\bxjs@kv@orblockheading@true\bxjs@jp@or@block@headingtrue
4481 \let\bxjs@kv@orblockheading@false\bxjs@jp@or@block@headingfalse
4482 \define@key{bxjsPan}{or-block-heading}[true]{%
4483   \bxjs@set@keyval{blockheading}{#1}{}}

```

実際の `japaram` の値を適用する。

```

4484 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}
4485 \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。

```

4486 \@onlypreamble\bxjs@set@dupload@proc
4487 \def\bxjs@set@dupload@proc#1{%

```

```

4488 \expandafter\bxjs@set@dupload@proc@a\csize bxjs@dlp/#1\endcsize}
4489 \@onlypreamble\bxjs@set@dupload@proc@a
4490 \def\bxjs@set@dupload@proc@a#1{%
4491   \@onlypreamble#1\def#1##1}
4492 \def\bxjs@unset@dupload@proc#1{%
4493   \expandafter\let\csize bxjs@dlp/#1\endcsize\@undefined}

```

`\@if@options` `\@if@options` の再定義。

```

4494 \@onlypreamble\bxjs@org@if@options
4495 \let\bxjs@org@if@options\@if@options
4496 \@onlypreamble\bxjs@org@reset@options
4497 \let\bxjs@org@reset@options\relax
4498 \def\@if@options#1#2#3{%
4499   \let\bxjs@next\@secondoftwo
4500   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4501   \ifx\bxjs@tmpa\bxjs@tmpb
4502     \expandafter\ifx\csize bxjs@dlp/#2.#1\endcsize\relax\else
4503       \let\bxjs@next\@firstoftwo \fi
4504   \fi
4505   \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4506 \g@addto@macro\bxjs@begin@document@hook{%
4507   \let\@if@options\bxjs@org@if@options}
4508 \@onlypreamble\bxjs@do@dupload@proc
4509 \def\bxjs@do@dupload@proc#1#2#3{%
4510   \ifx\bxjs@org@reset@options\relax
4511     \let\bxjs@org@reset@options\@reset@options
4512   \fi
4513   \bxjs@csize\bxjs@next\bxjs@dlp/#2.#1}%
4514 \def\@reset@options{%
4515   \let\@reset@options\bxjs@org@reset@options
4516   \@reset@options
4517   \bxjs@next{#3}}%
4518 \@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。

```

4519 \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。

```

4520 \let\bxjs@babel@options\relax

```

■Polyglossia について つまり $X_{\text{E}}\text{TeX}$ ・ LuaTeX の場合。

※この場合 etoolbox が使用可能になっている。

```
4521 \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。

```
4522 \pandocSkipLoadPackage{polyglossia}
4523 \bxjs@set@dupload@proc{polyglossia.sty}{%
4524   \bxjs@unset@dupload@proc{polyglossia.sty}%
4525   \ClassWarning\bxjs@clsname
4526     {Package polyglossia is required}%
4527   \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。

```
4528   \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。

```
4529     \ifblank{##2}{%
4530       \ClassWarning\bxjs@clsname
4531         {Main language is invalid, thus the fallback\MessageBreak
4532           definitions will be employed}%
4533       \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。

```
4534     }{%else
4535       \ClassWarning\bxjs@clsname
4536         {Main language is valid (##2),\MessageBreak
4537           thus polyglossia will be loaded}%
4538       \csundef{ver@polyglossia.sty}%
4539       \edef\bxjs@next{%
4540         \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4541       }\bxjs@next
4542       \setmainlanguage[##1]{##2}%
4543     }}}
```

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。

```
4544 \g@addto@macro\bxjs@endpreamble@hook{%
4545   \ifx\bxjs@polyglossia@options\relax
4546     \csundef{ver@polyglossia.sty}%
4547   \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。

```
4548 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4549 \def\bxjs@pandoc@polyglossia@ja{%
4550   \renewcommand*\setmainlanguage[2] []{%
```

```

4551 \newcommand*\setotherlanguage[2] [] {%
4552   \ifblank{##2}{-}{%else
4553     \cslet{##2}\@empty \cslet{end##2}\@empty
4554     \cslet{text##2}\@firstofone}}%
4555 \newcommand*\setotherlanguages[2] [] {%
4556   \@for\bxjs@tmpa:={##2}\do{%
4557     \setotherlangauge{\bxjs@tmpa}}}%

```

Polyglossia の読込済マークは外れるようにしておく。

```

4558 \let\bxjs@polyglossia@options\relax}%
4559 \fi

```

■Babel について Xe_{La}TeX 以外の場合。

※ Pandoc 2.6 版において、Lua_{TeX} で用いられる多言語パッケージが Polyglossia から Babel に変更された。

```

4560 \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。

```

4561 \pandocSkipLoadPackage{babel}
4562 \bxjs@set@dupload@proc{babel.sty}{%
4563   \bxjs@unset@dupload@proc{babel.sty}%
4564   \ClassWarning\bxjs@clsname
4565   {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。

```

4566 \@tempswafalse \let\bxjs@babel@options\@empty
4567 \def\bxjs@tmpb{main=}%
4568 \@for\bxjs@tmpa:=#1\do{%
4569   \ifx\bxjs@tmpa\bxjs@tmpb \@tempswatrue
4570   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4571   \fi}%
4572 \if@tempswa
4573   \ClassWarning\bxjs@clsname
4574   {Main language is invalid, so fallback\MessageBreak
4575     definitions will be employed}%
4576   \bxjs@pandoc@polyglossia@ja

```

ない場合は、本来の babel の処理を実行する。

```

4577 \else
4578   \ClassWarning\bxjs@clsname
4579   {Main language is valid,\MessageBreak
4580     thus babel will be loaded}%
4581   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4582   \RequirePackage[#1]{babel}[]%
4583 \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。

```

4584 \g@addto@macro\bxjs@endpreamble@hook{%
4585   \ifx\bxjs@babel@options\relax
4586     \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4587   \fi}

```

`\bxjs@pandoc@babel@ja` Pandoc 側で `lang=ja` が指定されていた場合の処理。

```

4588 \@onlypreamble\bxjs@pandoc@babel@ja
4589 \def\bxjs@pandoc@babel@ja{%
4590   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4591   \edef\bxjs@next{%
4592     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4593   }\bxjs@next
4594   \if j\jsEngine
4595     \RequirePackage[main=japanese]{pxbabel}[]%
4596   \else
4597     \RequirePackage{bxorigcapt}[]%
4598   \fi}

```

`lang` 対策はこれで終わり。

```

4599 \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。

```

4600 \bxjs@set@dupload@proc{geometry.sty}{%
4601   \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。

```

4602 \if l\jsEngine
4603   \pandocSkipLoadPackage{xeCJK}
4604   \providecommand*\setCJKmainfont{\setmainfont}
4605 \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。

```

4606 \if x\jsEngine
4607   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4608     ,space}
4609 \fi

```

E.8 レイアウト上書き禁止

レイアウト上書き禁止の実装は `etoolbox` の機能を使う。

```
4610 \ifjsWiththeTeX
4611 \@onlypreamble\bxjs@info@or@ban
4612 \def\bxjs@info@or@ban#1{%
4613   \PackageInfo\bxjs@clsname
4614   {Freeze layout on '#1',\MessageBreak detected}}
```

■`indent` について `indent` 変数を指定しない場合に「段落表現形式をインデント方式に変更する」動作を抑止する。

```
4615 \unless\ifbxjs@jp@or@indent
4616   \bxjs@info@or@ban{indent}
```

`parskip` がある場合はそれを読み込もうとするため、`parskip` の読込をブロックする。

```
4617 \IfFileExists{parskip.sty}{%
4618   \pandocSkipLoadPackage{parskip}%
```

`parskip` がない場合はパラメタを変更しようとするため、該当のパラメタを復帰させる。

```
4619 }{%else
4620   \eappto\bxjs@endpreamble@hook{%
4621     \parindent=\the\parindent\relax
4622     \parskip=\the\parskip\relax}}
4623 \fi
4624 %
4625 % \paragraph{secnumdepth について}
4626 % |secnumdepth| の値を決めるのは numbersections 変数
4627 % (|-N|/|--number-sections| オプションに連動する)
4628 % や secnumdepth 変数であるが、何れにしても |secnumdepth|
4629 % の値は書き換えられる。
4630 % そのため、|secnumdepth| を復帰させる。
4631 \ifbxjs@jp@or@secnumdepth\else
4632   \bxjs@info@or@ban{secnumdepth}
4633   \eappto\bxjs@endpreamble@hook{%
4634     \c@secnumdepth=\the\c@secnumdepth\relax}
4635 \fi
4636 %
4637 % \paragraph{block-heading について}
4638 \ifbxjs@jp@or@block@heading\else
4639   \let\bxjs@frozen@paragraph\paragraph
4640   \let\bxjs@frozen@subparagraph\subparagraph
4641   \bxjs@info@or@ban{block-heading}
4642   \appto\bxjs@endpreamble@hook{%
4643     \let\oldparagraph\@undefined
4644     \let\paragraph\bxjs@frozen@paragraph
4645     \let\subparagraph\bxjs@frozen@subparagraph}
4646 \fi
4647 \fi
```


E.9 paragraph のマーク

BXJS クラスでは `\paragraph` の見出しの前に `\jsParagraphMark` で指定したマークが付加され、既定ではこれは“■”である。しかし、この規定は `\paragraph` が本来のレイアウトを保っている、すなわち「行内見出しである」「節番号が付かない」ことが前提になっていると考えられる。Pandoc はこの規定を変更することがある（特に既定で `\paragraph` を別行見出しに再定義する）ため、変更された場合は `\jsParagraphMark` の既定値を空にする。

Pandoc がプレアンブルで行う再定義の結果を調べるため、begin-document フックを利用する。

```
4648 \g@addto@macro\bxjs@begin@document@hook{%
4649   \@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。

```
4650   \ifx\oldparagraph\@undefined\else
4651     \@tempswattrue
4652   \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4653   \ifnum\c@secnumdepth>3
4654     \@tempswattrue
4655   \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。

```
4656   \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4657     \let\jsParagraphMark\@empty
4658   \fi\fi}
```

E.10 全角空白文字

```
4659 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4660   \catcode"3000=\active
4661   \begingroup \catcode`\!=7
4662   \protected\gdef!!!3000{\zwspace}
4663   \endgroup
4664 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4665   \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4666   \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4667 \fi\fi
```

E.11 hyperref 対策

`hyperref` の `unicode` オプションの固定を行う。

```
4668 \if j\jsEngine
```

```

4669 \bxjs@fix@hyperref@unicode{false}
4670 \else
4671 \bxjs@fix@hyperref@unicode{true}
4672 \fi

```

E.12 Pandoc 要素に対する和文用の補正

■重要要素 重要 (Strong) 要素に対する IAT_EX 出力は `\textbf` となるが、代わりに `\strong` を使いたいため、`\textbf` を書き換えてしまう (うわぁ)。

```

4673 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
4674 \let\orgtextbf\textbf
4675 \DeclareRobustCommand\bxjstextbf[1]{%
4676 \begingroup
4677 \let\textbf\orgtextbf
4678 \strong{#1}%
4679 \endgroup}%
4680 \g@addto@macro\bxjs@begin@document@hook{%
4681 \let\textbf\bxjstextbf}
4682 \fi\fi

```

`\strong` の書体を設定する。

```

4683 \jsAtEndOfClass{%
4684 \ifx\strongfontdeclare\@undefined\else
4685 \ifcase\bxjs@jp@strong
4686 \or \strongfontdeclare{\sfamily}%
4687 \or \strongfontdeclare{\sfamily\bfseries}%
4688 \fi
4689 \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 IAT_EX 出力は `\texttt` となる。`\texttt` の両端に欧文ゴーストが入るようにする。

```

4690 \ifbxjs@jp@fix@code
4691 \chardef\bxjs@eghost@c=23
4692 \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4693 \else\ifx l\jsEngine \ltjsetParameter{alxspmode={\bxjs@eghost@c,3}}
4694 \else\ifx x\jsEngine %no-op
4695 \else \let\bxjs@eghost@c\@undefined
4696 \fi\fi\fi
4697 \ifx\bxjs@eghost@c\@undefined\else
4698 \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4699 \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4700 \let\orgtexttt\texttt
4701 \DeclareRobustCommand\bxjstexttt[1]{%
4702 \ifmmode \nfss@text{\ttfamily #1}%
4703 \else
4704 \ifvmode \leavevmode \fi
4705 \bxjs@pan@eghost\null
4706 \begingroup \ttfamily #1\endgroup

```

```

4707      \null\bxjs@pan@eghost
4708      \fi}
4709      \g@addto@macro\bxjs@begin@document@hook{%
4710      \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。

```

4711      \expandafter\def\expandafter\verb\expandafter{%
4712      \expandafter\bxjs@pan@eghost\verb}
4713      \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4714      \fi
4715 \fi

```

E.13 完了

おしまい。

```

4716 %</pandoc>

```

和文ドライバ実装はここまで。

```

4717 %</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。
- `bxjscjcat` : modern ドライバ用の和文カテゴリを適用する。
- `bxjspandoc` : Pandoc 用のナニカ。

```

4718 %<*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

```

4719 %<*compat>
4720 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4721 \let\bxac@engine=n
4722 \def\bxac@do#1#2{%
4723   \edef\bxac@tmpa{\string#1}%
4724   \edef\bxac@tmpb{\meaning#1}%
4725   \ifx\bxac@tmpa\bxac@tmpb #2\fi}

```

```

4726 \brac@do\XeTeXversion{\let\bxac@engine=x}
4727 \brac@do\luatexversion{\let\bxac@engine=l}

```

`\brac@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。

```

4728 \ifx\jsAtEndOfClass\@undefined
4729   \let\brac@delayed@if@bxjs\@firstofone
4730 \else \let\brac@delayed@if@bxjs\jsAtEndOfClass
4731 \fi

```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```

\RevokeOldLuaTeXBehavior 4732 \newif\ifbxac@in@old@behavior
4733 \let\ImposeOldLuaTeXBehavior\relax
4734 \let\RevokeOldLuaTeXBehavior\relax

```

G.2 XeTeX 部分

```

4735 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。
4736 \@onlypreamble\brac@adjust@charclass
4737 \brac@delayed@if@bxjs{%
4738   \@ifpackageloaded{xeCJK}{%}{%else
4739     \ifx\xe@alloc@intercharclass\@undefined\else
4740       \ifnum\xe@alloc@intercharclass=\z@
4741         \PackageInfo\bxac@pkgname
4742         {Setting up interchar class for CJK...\@gobble}%
4743         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4744           \xe@alloc@intercharclass=3
4745         }{%else
4746           \PackageWarning\bxac@pkgname
4747             {Cannot find file 'load-unicode-xetex-classes.tex'%
4748             \@gobble}%
4749         }%
4750       \fi\fi
4751     \ifnum\XeTeXcharclass"3041=\z@
4752       \PackageInfo\bxac@pkgname
4753       {Adjusting interchar class for CJK...\@gobble}%
4754       \@for\bxac@x={%
4755         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4756         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4757         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4758         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4759         31FF%
4760       }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4761     \fi
4762   }%
4763 }

```

以上。

```
4764 \fi
```

G.3 LuaTeX 部分

```
4765 \ifx l\bxac@engine
```

ムニヤムニヤ。

```
4766 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
```

```
4767 \chardef\pdfTeXversion=200
```

```
4768 \def\pdfTeXrevision{0}
```

```
4769 \let\pdfTeXbanner\luatexbanner
```

```
4770 \fi\fi
```

```
\ImposeOldLuaTeXBehavior ムニヤムニヤ。
```

```
\RevokeOldLuaTeXBehavior 4771 \begingroup\expandafter\expandafter\expandafter\endgroup
```

```
4772 \expandafter\ifx\cename outputmode\endcsname\relax\else
```

```
4773 \def\bxac@ob@list{%
```

```
4774 \do{\let}\pdfoutput{\outputmode}%
```

```
4775 \do{\let}\pdfpagewidth{\pagewidth}%
```

```
4776 \do{\let}\pdfpageheight{\pageheight}%
```

```
4777 \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
```

```
4778 \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
```

```
4779 \def\bxac@ob@do#1#2{\begingroup
```

```
4780 \expandafter\bxac@ob@do@a\cename bxac@\string#2\endcsname{#1}#2}
```

```
4781 \def\bxac@ob@do@a#1#2#3#4{\endgroup
```

```
4782 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
```

```
4783 \else \let#3#1\relax \let#1\@undefined
```

```
4784 \fi}
```

```
4785 \protected\def\ImposeOldLuaTeXBehavior{%
```

```
4786 \unless\ifbxac@in@old@behavior
```

```
4787 \bxac@in@old@behaviortrue
```

```
4788 \let\do\bxac@ob@do \bxac@ob@list
```

```
4789 \fi}
```

```
4790 \protected\def\RevokeOldLuaTeXBehavior{%
```

```
4791 \ifbxac@in@old@behavior
```

```
4792 \bxac@in@old@behaviorfalse
```

```
4793 \let\do\bxac@ob@do \bxac@ob@list
```

```
4794 \fi}
```

```
4795 \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。

```
4796 \ifnum\luatexversion>64 \directlua{
```

```
4797 local function range(cs, ce, cc, ff)
```

```
4798 if ff or not tex.getcatcode(cs) == cc then
```

```
4799 local setcc = tex.setcatcode
```

```
4800 for c = cs, ce do setcc(c, cc) end
```

```
4801 end
```

```
4802 end
```

```
4803 range(0x3400, 0x4DB5, 11, false)
```

```
4804 range(0x4DB5, 0x4DBF, 11, true)
```

```

4805     range(0x4E00, 0x9FCC, 11, false)
4806     range(0x9FCD, 0x9FFF, 11, true)
4807     range(0xAC00, 0xD7A3, 11, false)
4808     range(0x20000, 0x2A6D6, 11, false)
4809     range(0x2A6D7, 0x2A6FF, 11, true)
4810     range(0x2A700, 0x2B734, 11, false)
4811     range(0x2B735, 0x2B73F, 11, true)
4812     range(0x2B740, 0x2B81D, 11, false)
4813     range(0x2B81E, 0x2B81F, 11, true)
4814     range(0x2B820, 0x2CEA1, 11, false)
4815     range(0x2CEA2, 0x2FFFD, 11, true)
4816 } \fi

```

以上。

```
4817 \fi
```

G.4 完了

おしまい。

```
4818 %</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

```

4819 %<*cjkcat>
4820 \def\bxjx@pkgname{bxjscjkat}
4821 \newcount\bxjx@canta
4822 \@onlypreamble\bxjx@tmpdo
4823 \@onlypreamble\bxjx@tmpdo@a
4824 \@onlypreamble\bxjx@tmpdo@b

\bxjx@engine エンジンの種別。

4825 \let\bxjx@engine=n
4826 \def\bxjx@tmpdo#1#2{%
4827   \edef\bxjx@tmpa{\string#1}%
4828   \edef\bxjx@tmpb{\meaning#1}%
4829   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4830 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4831 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4832 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4833 \bxjx@tmpdo\pdfTeXversion{\let\bxjx@engine=p}
4834 \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。

```

4835 \def\bxjx@tmpdo#1#2{%
4836   \if#1\bxjx@engine
4837     \@ifpackageloaded{#2}{-}{%else
4838       \PackageError\bxjx@pkgname
4839         {Package '#2' must be loaded}%
4840         {Package loading is aborted.\MessageBreak\@ehc}%
4841     \endinput}
4842 \fi}
4843 \bxjx@tmpdo{p}{bxcjkjatyp}
4844 \bxjx@tmpdo{x}{xeCJK}
4845 \bxjx@tmpdo{l}{luatexja}

```

古い L^AT_EX の場合、`\TextOrMath` は `fixltx2e` パッケージで提供される。

```

4846 \ifx\TextOrMath\undefined
4847   \RequirePackage{fixltx2e}
4848 \fi

```

H.2 和文カテゴリコードの設定

upL^AT_EX の場合、和文カテゴリコードの設定を LuaT_EX-ja と（ほぼ）等価なものに変更する。

※ LuaT_EX-ja との相違点：A830、A960、1B000。

```

4849 \if u\bxjx@engine
4850 \@for\bxjx@tmpa:={%
4851 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4852 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4853 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4854 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4855 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4856 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4857 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4858 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4859 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4860 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4861 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4862 FE70,FFF0,%
4863 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4864 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4865 10480,104B0,10500,10530,10600,10800,10840,10860,%
4866 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4867 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4868 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4869 11200,11280,112B0,11300,11400,11480,11580,11600,%
4870 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4871 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4872 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4873 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4874 1EE00,1F000,1F030,1F0A0,1F300,1F600,1F650,1F680,%

```

```

4875 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4876 00C0%
4877 }\do{%
4878 \@tempcnta="\bxjx@tmpa\relax
4879 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4880 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4881 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4882 \fi

```

H.3 ギリシャ・キリル文字の扱い

「特定 CJK 曖昧文字」について、和文・欧文扱いを制御できるようにする。ここで「特定 CJK 曖昧文字」とは以下に該当する文字の集合を指す：

- Unicode と JIS X 0213 に共通して含まれるギリシャ文字・キリル文字。
- Latin-1 の上位部分と JIS X 0208 に共通して含まれる文字（LuaTeX-ja の定める“範囲 8”）。

`\bxjx@grkcyr@list` 「特定 CJK 曖昧文字」に関する情報をもつ `\do-` リスト。各項目の形式は以下の通り：

`\do{(Unicode 符号値)}{(対象 fontenc)}{(テキスト LICR)}{(数式 LICR)}`

※数式で使わない文字は〈数式 LICR〉を空にする。

```

4883 \@onlypreamble\bxjx@grkcyr@list
4884 \def\bxjx@grkcyr@list{%
4885 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4886 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4887 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4888 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4889 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4890 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4891 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
4892 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4893 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4894 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4895 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4896 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4897 \do{039D}{LGR}{\textNu}{N}%             % GR. C. L. NU
4898 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
4899 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
4900 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
4901 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
4902 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4903 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
4904 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4905 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
4906 \do{03A7}{LGR}{\textChi}{X}%           % GR. C. L. CHI
4907 \do{03A8}{LGR}{\textPsi}{\Psi}%         % GR. C. L. PSI
4908 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA

```


4909	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
4910	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
4911	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
4912	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4913	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4914	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4915	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4916	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4917	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4918	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4919	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4920	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4921	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4922	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4923	\do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
4924	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4925	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4926	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4927	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4928	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4929	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4930	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4931	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4932	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4933	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4934	\do{0401}{T2A}{\CYRYO}{\CYRYO}%	% CY. C. L. IO
4935	\do{0410}{T2A}{\CYRA}{\CYRA}%	% CY. C. L. A
4936	\do{0411}{T2A}{\CYRB}{\CYRB}%	% CY. C. L. BE
4937	\do{0412}{T2A}{\CYRV}{\CYRV}%	% CY. C. L. VE
4938	\do{0413}{T2A}{\CYRG}{\CYRG}%	% CY. C. L. GHE
4939	\do{0414}{T2A}{\CYRD}{\CYRD}%	% CY. C. L. DE
4940	\do{0415}{T2A}{\CYRE}{\CYRE}%	% CY. C. L. IE
4941	\do{0416}{T2A}{\CYRZH}{\CYRZH}%	% CY. C. L. ZHE
4942	\do{0417}{T2A}{\CYRZ}{\CYRZ}%	% CY. C. L. ZE
4943	\do{0418}{T2A}{\CYRI}{\CYRI}%	% CY. C. L. I
4944	\do{0419}{T2A}{\CYRISHRT}{\CYRISHRT}%	% CY. C. L. SHORT I
4945	\do{041A}{T2A}{\CYRK}{\CYRK}%	% CY. C. L. KA
4946	\do{041B}{T2A}{\CYRL}{\CYRL}%	% CY. C. L. EL
4947	\do{041C}{T2A}{\CYRM}{\CYRM}%	% CY. C. L. EM
4948	\do{041D}{T2A}{\CYRN}{\CYRN}%	% CY. C. L. EN
4949	\do{041E}{T2A}{\CYRO}{\CYRO}%	% CY. C. L. O
4950	\do{041F}{T2A}{\CYRP}{\CYRP}%	% CY. C. L. PE
4951	\do{0420}{T2A}{\CYRR}{\CYRR}%	% CY. C. L. ER
4952	\do{0421}{T2A}{\CYRS}{\CYRS}%	% CY. C. L. ES
4953	\do{0422}{T2A}{\CYRT}{\CYRT}%	% CY. C. L. TE
4954	\do{0423}{T2A}{\CYRU}{\CYRU}%	% CY. C. L. U
4955	\do{0424}{T2A}{\CYRF}{\CYRF}%	% CY. C. L. EF
4956	\do{0425}{T2A}{\CYRH}{\CYRH}%	% CY. C. L. HA
4957	\do{0426}{T2A}{\CYRC}{\CYRC}%	% CY. C. L. TSE

4958	\do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
4959	\do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
4960	\do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
4961	\do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
4962	\do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
4963	\do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
4964	\do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
4965	\do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
4966	\do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
4967	\do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
4968	\do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
4969	\do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
4970	\do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
4971	\do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
4972	\do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
4973	\do{0436}{T2A}{\cyrrh}{}%	% CY. S. L. ZHE
4974	\do{0437}{T2A}{\cyrrz}{}%	% CY. S. L. ZE
4975	\do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
4976	\do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
4977	\do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
4978	\do{043B}{T2A}{\cyrll}{}%	% CY. S. L. EL
4979	\do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
4980	\do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
4981	\do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
4982	\do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
4983	\do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
4984	\do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES
4985	\do{0442}{T2A}{\cyrt}{}%	% CY. S. L. TE
4986	\do{0443}{T2A}{\cyru}{}%	% CY. S. L. U
4987	\do{0444}{T2A}{\cyrf}{}%	% CY. S. L. EF
4988	\do{0445}{T2A}{\cyrrh}{}%	% CY. S. L. HA
4989	\do{0446}{T2A}{\cyrc}{}%	% CY. S. L. TSE
4990	\do{0447}{T2A}{\cyrch}{}%	% CY. S. L. CHE
4991	\do{0448}{T2A}{\cyrrsh}{}%	% CY. S. L. SHA
4992	\do{0449}{T2A}{\cyrshch}{}%	% CY. S. L. SHCHA
4993	\do{044A}{T2A}{\cyrhrdsn}{}%	% CY. S. L. HARD SIGN
4994	\do{044B}{T2A}{\cyrery}{}%	% CY. S. L. YERU
4995	\do{044C}{T2A}{\cyrsftsn}{}%	% CY. S. L. SOFT SIGN
4996	\do{044D}{T2A}{\cyrerev}{}%	% CY. S. L. E
4997	\do{044E}{T2A}{\cyryu}{}%	% CY. S. L. YU
4998	\do{044F}{T2A}{\cyrya}{}%	% CY. S. L. YA
4999	\do{0451}{T2A}{\cyryo}{}%	% CY. S. L. IO
5000	\do{00A7}{TS1}{\textsection}{\mathsection}%	SECTION SYMBOL
5001	\do{00A8}{TS1}{\textasciidieresis}{}%	% DIAERESIS
5002	\do{00B0}{TS1}{\textdegree}{\mathdegree}%	% DEGREE SIGN
5003	\do{00B1}{TS1}{\textpm}{\pm}%	% PLUS-MINUS SIGN
5004	\do{00B4}{TS1}{\textasciicircum}{}%	% ACUTE ACCENT
5005	\do{00B6}{TS1}{\textparagraph}{\mathparagraph}%	PILCROW SIGN
5006	\do{00D7}{TS1}{\texttimes}{\times}%	% MULTIPLICATION SIGN

```

5007 \do{00F7}{TS1}{\textdiv}{\div}%          % DIVISION SIGN
5008 }

\mathdegree 面倒なので補っておく。
5009 \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。
5010 \newif\ifbxjx@gcc@cjk

\greekasCJK 〔公開命令〕「特定 CJK 曖昧文字」を和文扱いにする。
5011 \newcommand*\greekasCJK{%
5012   \bxjx@gcc@cjktrue}

\nogreekasCJK 〔公開命令〕「特定 CJK 曖昧文字」を欧文扱いにする。
5013 \newcommand*\nogreekasCJK{%
5014   \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。

5015 \def\bxjx@tmpdo#1\relax{%
5016   \def\bxjx@fake@grk##1##2{%
5017     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
5018   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
5019     \ifx\##1\%
5020       \bxjx@canta##4\divide\bxjx@canta\@cclvi
5021       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
5022       \mathchar\bxjx@canta
5023     \else ##3\fi}
5024 }\expandafter\bxjx@tmpdo\string\mathchar\relax

```

■pdfTeX・upTeX の場合

```

5025 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0

```

- `\[bxjx@KC/〈符号値〉]` : その文字が「特定曖昧 CJK 文字」に該当する場合に定義済になる。

まず `inputenc` を読み込んで入力エンコーディングを `utf8` に変更する。

※「既定 UTF-8 化」後の L^AT_EX においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```

5026 \@ifpackageloaded{inputenc}{}{%else
5027   \RequirePackage[utf8]{inputenc}}

```

```

5028 \def\bxjx@tmpa{utf8}
5029 \ifx\bxjx@tmpa\inputencdoingname
5030   \PackageWarningNoLine\bxjx@pkgnam
5031     {Input encoding changed to utf8}%
5032   \inputencoding{utf8}%
5033 \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。

```

5034 \if u\bxjx@engine
5035 \kcatcode"0370=15
5036 \kcatcode"0400=15
5037 \kcatcode"0500=15
5038 \fi

```

各文字について \DeclareUnicodeCharacter を実行する。

```

5039 \def\bxjx@tmpdo#1{%
5040   \@tempcnta="#1\relax
5041   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
5042 \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。

```

5043   \ifx\\#5\\%
5044     \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。

```

5045   \else\ifcat A\noexpand#5%
5046     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5047       {\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。

```

5048   \else \def\bxjx@tmpa{#5}%
5049   \fi\fi
5050   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
5051   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

5052 \if u\bxjx@engine
5053 \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。

```

5054   \kchardef#1=\@tempcnta

```

```

5055 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
5056 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}

```

pdfTeX の場合も処理はほとんど同じ。ただし、和文用の定義として \UTF{⟨符号値⟩} を使う (\UTF は bxcjkjatypе の命令)。`\[bxjx@KC/...]` は使わないが定義済にする必要がある。

```

5057 \else\if p\bxjx@engine
5058 \def\bxjx@tmpdo@b#1#2#3#4#5{%
5059 \mathchardef#1=\@tempcnta
5060 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
5061 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
5062 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

5063 \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止したもの。

```

5064 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
5065 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
5066 \@onlypreamble\bxjx@DeclareUnicodeCharacter
5067 \def\bxjx@DeclareUnicodeCharacter#1#2{%
5068 \count@="#1\relax
5069 \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
5070 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
5071 \else
5072 \wlog{ \space\space skipped defining Unicode char U+#1}%
5073 \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。

```

5074 \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。

```

5075 \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。

```

5076 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
5077 \else \UseTextSymbol{#2}{#3}%
5078 \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。

```

5079 \begingroup
5080 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
5081 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
5082 \noexpand\bxjx@swap@DUC@cmd
5083 \the\toks@
5084 \noexpand\bxjx@swap@DUC@cmd}}

```

```

5085 \endgroup\next
5086 \def\bxjx@swap@DUC@cmd{%
5087   \let\bxjx@tmpa\DeclareUnicodeCharacter
5088   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
5089   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
5090   \let\bxjx@tmpa\relax}

```

以上。

■Xe_{La}TeX・Lua_{TeX} の場合

```

5091 \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。

```

5092 \def\bxjx@tmpdo#1{%
5093   \bxjx@cmta="#1\relax
5094   \begingroup
5095     \lccode`~=\bxjx@cmta
5096     \lowercase{\endgroup
5097       \bxjx@tmpdo@a{~}}{#1}}
5098 \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_{La}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。

```

5099   \ifx\\#5\\\let\bxjx@tmpa\relax
5100   \else\ifcat A\noexpand#5%
5101     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5102       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
5103   \else \def\bxjx@tmpa{#5}%
5104   \fi\fi
5105   \ifx\bxjx@tmpa\relax\else
5106     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
5107   \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。

```

5108 \mathchardef\bxjx@tmpa="119
5109 \ifx\bxjx@tmpa\pi \let\do\bxjx@tmpdo \bxjx@grkcyr@list \fi

```

次に、テキストにおいて「特定 CJK 曖昧文字」の扱いが `\[no]greekasCJK` で切り替わるようにする。

Lua_{TeX} の場合は、Lua_{TeX}-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。

```

5110 \if 1\bxjx@engine
5111   \protected\def\greekasCJK{%
5112     \bxjx@gcc@cjctrue
5113     \ltjsetparameter{jacharrange={+2, +8}}
5114   \protected\def\nogreekasCJK{%
5115     \bxjx@gcc@cjkfalse
5116     \ltjsetparameter{jacharrange={-2, -8}}

```

```
5117 \fi
```

X_ƎT_ƎX の場合、xeCJK は X_ƎT_ƎX 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。

```
5118 \if x\bxjx@engine
5119   \let\bxjx@gcc@CJK@list\@empty
5120   \def\do#1#2#3#4{%
5121     \edef\bxjx@gcc@CJK@list{\bxjx@gcc@CJK@list
5122       \noexpand\XeTeXcharclass"#1\bxjx@canta}}
5123   \bxjx@grkcyr@list
5124   \protected\def\greekasCJK{%
5125     \bxjx@gcc@CJKtrue
5126     \bxjx@canta=\@ne \bxjx@gcc@CJK@list}
5127   \protected\def\nogreekasCJK{%
5128     \bxjx@gcc@CJKfalse
5129     \bxjx@canta=\z@ \bxjx@gcc@CJK@list}
5130 \fi
```

以上。

```
5131 \fi\fi
```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5132 \nogreekasCJK
```

H.5 完了

おしまい。

```
5133 %</cjkat>
```

付録 I 補助パッケージ : bxjspandoc

Pandoc の L^AT_ƎX 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_ƎX 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

```
5134 %<*ancpandoc>
5135 %% このファイルは日本語文字を含みます。
5136 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```

5137 \let\bxjsp@engine=n
5138 \@onlypreamble\bxjsp@do
5139 \def\bxjsp@do#1#2{%
5140   \edef\bxjsp@tmpa{\string#1}%
5141   \edef\bxjsp@tmpb{\meaning#1}%
5142   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
5143 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
5144 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
5145 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
5146 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}

```

I.2 パッケージオプション

`english` オプションが指定されている場合、`\ldots` の調整を抑止する。
 ※つまり、「グローバルの `english` オプション」が指定されている場合も抑止の対象になる。
 BXJS クラスの英語モードを想定しているが、それ以外の場合でも、一般的な L^AT_EX の習慣として、グローバルの `english` は「その文書の基底言語が英語である」ことを示す。

```

5147 \newif\ifbxjsp@english
5148 \DeclareOption{english}{\bxjsp@englishttrue}

```

オプション定義はおしまい。

```

5149 \ProcessOptions*

```

I.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。

```

5150 \@onlypreamble\pandocSkipLoadFile
5151 \newcommand*\pandocSkipLoadFile[1]{%
5152   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5153 \def\bxjsp@skip@load@file@a#1#2{%
5154   \ifx#1\relax
5155     \def#1{2001/01/01}%
5156     \PackageInfo\bxjsp@pkgname
5157       {File '#2' marked as loaded\@gobble}%
5158   \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。

```

5159 \@onlypreamble\pandocSkipLoadPackage
5160 \newcommand*\pandocSkipLoadPackage[1]{%
5161   \pandocSkipLoadFile{#1.sty}}

```


I.4 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近（2015 年版以降）の \LaTeX ではこれで警告が出る。これを抑止する。

\LaTeX カーネルが新しい場合は `fixltx2e` を読込済にする。

```
5162 \ifx\@IncludeInRelease\undefined\else
5163   \pandocSkipLoadPackage{fixltx2e}
5164 \fi
```

I.5 cmap パッケージ

エンジンが (u)p \LaTeX のときに `cmap` パッケージが読み込まれるのを阻止する。（実際は警告が出るだけで無害であるが。）

```
5165 \if j\bxjisp@engine
5166   \pandocSkipLoadPackage{cmap}
5167 \fi
```

I.6 microtype パッケージ

警告が多すぎなので消す。

```
5168 \if j\bxjisp@engine \else
5169   \PassOptionsToPackage{verbose=silent}{microtype}
5170 \fi
```

エンジンが (u)p \LaTeX のときに `microtype` パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は `standard` ドライバでこの処理を行っていたが、元来は Pandoc 用の処理なので、1.5 版で `pandoc` に移動。

```
5171 \if j\bxjisp@engine
5172   \pandocSkipLoadPackage{microtype}
5173   \newcommand*\UseMicrotypeSet[2][]{\}
5174 \fi
```

I.7 Unicode 文字変換対策

Pandoc で \LaTeX 形式に書き出す場合は、元データ中の一部の Unicode 文字を「 \LaTeX の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

…→`\ldots{}` ‘→``` ’→`'` “→`“` ”→`”`

日本語 \LaTeX では「 \LaTeX の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「`\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。

```
5175 \DeclareRobustCommand{\pandocLdots}{%
5176   \relax\ifmmode \expandafter\bxjsp@org@ldots
5177   \else \expandafter\bxjsp@ja@ellipsis
5178   \fi}
5179 \def\bxjsp@ja@ellipsis{…}
5180 \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。

```
5181 \AtBeginDocument{%
5182   \let\bxjsp@org@ldots\ldots
5183   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}
```

`\ldots` の直後の文字が非英字の場合、Pandoc は「`\ldots。`」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが $X_{\text{T}}\text{T}_{\text{E}}\text{X}$ ・ $\text{LuaT}_{\text{E}}\text{X}$ は英字と見なす（または将来その可能性がある）」文字で、特に日本語文書に現れるものについて、非英字扱いにしておく。

※ Pandoc は「Unicode 7.0 で GC が Letter」な文字を英字と判定している。

```
5184 \chardef\bxjsp@cc@other=12
5185 \@onlypreamble\bxjsp@makeother@range
5186 \def\bxjsp@makeother@range#1#2{%
5187   \@tempcnta"#1\relax \@tempcntb"#2\relax
5188   \loop\ifnum\@tempcnta<\@tempcntb
5189     \catcode\@tempcnta\bxjsp@cc@other
5190     \advance\@tempcnta\@ne
5191   \repeat}
5192 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
5193   \catcode"1F23B=\bxjsp@cc@other
5194   \bxjsp@makeother@range{9FCD}{A000}
5195   \bxjsp@makeother@range{1B002}{1B170}
5196   \bxjsp@makeother@range{2B820}{2EBF0}
5197 \fi
```

1.8 PandoLa モジュール

インストール済であれば読み込む。

```
5198 \IfFileExists{bxpandola.sty}{%
5199   \RequirePackage{bxpandola}\relax
5200   \PackageInfo{bxjsp@pkgname
5201     {PandoLa module is loaded@gobble}
5202 }{}
```

1.9 完了

おしまい。

5203 %</ancpandoc>

補助パッケージ実装はここまで。

5204 %</anc>