

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”)

v2.7a [2022/04/10]

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

目次

1	はじめに	3
2	オプション	11
3	和文フォントの変更	41
4	フォントサイズ	42
5	レイアウト	48
5.1	ページレイアウト	49
6	改ページ (日本語 TeX 開発コミュニティ版のみ)	63
7	ページスタイル	65
8	文書のマークアップ	68
8.1	表題	68
8.2	章・節	73
8.3	リスト環境	85
8.4	パラメータの設定	93
8.5	フロート	94
8.6	キャプション	96
9	フォントコマンド	97

10	相互参照	99
10.1	目次の類	99
10.2	参考文献	105
10.3	索引	106
10.4	脚注	108
11	段落の頭へのグルー挿入禁止	110
12	いろいろなロゴ	114
13	amsmath との衝突の回避	115
14	初期設定	115
付録 A	和文ドライバの仕様 	120
付録 B	和文ドライバ：minimal 	121
B.1	補助マクロ	121
B.2	(u)pTeX 用の設定	124
B.3	pdfTeX 用の処理	128
B.4	X _Y TeX 用の処理	129
B.5	後処理（エンジン共通）	130
付録 C	和文ドライバ：standard 	133
C.1	準備	133
C.2	和文ドライバパラメタ	133
C.3	共通処理 (1)	134
C.4	pTeX 用設定	142
C.5	pdfTeX 用設定：CJK + bxcjkatype	146
C.6	X _Y TeX 用設定：xeCJK + zxjatype	148
C.7	LuaTeX 用設定：LuaTeX-ja	150
C.8	共通処理 (2)	154
付録 D	和文ドライバ：modern 	155
D.1	フォント設定	155
D.2	fixltx2e 読込	155
D.3	和文カテゴリコード	156
D.4	完了	156
付録 E	和文ドライバ：pandoc 	156
E.1	準備	156
E.2	和文ドライバパラメタ	157
E.3	dupload システム	158

E.4	lang 変数	159
E.5	geometry 変数	162
E.6	CJKmainfont 変数	162
E.7	Option clash 対策	162
E.8	レイアウト上書き禁止	163
E.9	paragraph のマーク	164
E.10	全角空白文字	164
E.11	hyperref 対策	165
E.12	Pandoc 要素に対する和文用の補正	165
E.13	ifPDFTeX スイッチ	167
E.14	完了	168
付録 F	補助パッケージ一覧 🐼	168
付録 G	補助パッケージ：bxjscompat 🐼	168
G.1	準備	168
G.2	X _q TeX 部分	169
G.3	LuaTeX 部分	170
G.4	完了	171
付録 H	補助パッケージ：bxjsjkat 🐼	171
H.1	準備	171
H.2	和文カテゴリコードの設定	172
H.3	ギリシャ・キリル文字の扱い	173
H.4	初期設定	180
H.5	完了	181
付録 I	補助パッケージ：bxjspandoc 🐼	181
I.1	準備	181
I.2	パッケージオプション	181
I.3	パッケージ読込の阻止	182
I.4	fixltx2e パッケージ	182
I.5	cmap パッケージ	182
I.6	microtype パッケージ	183
I.7	Unicode 文字変換対策	183
I.8	PandoLa モジュール	184
I.9	完了	184

1 はじめに

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

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

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「pL^AT_EX 2_ε 新ドキュメントクラス」に改変を加えたものである。本ドキュメントクラスに関する説明は全てこの形式の枠の中に記す。枠の外にあるものは原原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以下のパッケージが読み込まれる。

`geometry`、`ifpdf`

また状況によっては以下のパッケージが読み込まれる可能性がある。

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

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

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

互換性のための補助パッケージを読み込む。

```
22 \IfFileExists{bxjscompat.sty}{%
23   \RequirePackage{bxjscompat}%
24 }
```

`\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`。

```
25 \let\jsArticle=a
26 \let\jsBook=b
27 \let\jsReport=r
28 \let\jsSlide=s
29 %<article>\let\jsDocClass\jsArticle
```

```

30 %<book>\let\jsDocClass\jsBook
31 %<report>\let\jsDocClass\jsReport
32 %<slide>\let\jsDocClass\jsSlide

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

51 \newif\ifjsWitheTeX
52 \bxjs@test@engine\epsilonTeXversion{\jsWitheTeXtrue}

```

非サポートのエンジンの場合は強制終了させる。

※ NTT jT_EX と Omega 系。

```

53 \let\bxjs@tmpa\relax
54 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
55 \ifx 0\jsEngine \def\bxjs@tmpa{Omega}\fi
56 \ifx\bxjs@tmpa\relax \expandafter\@gobble
57 \else
58   \ClassError\bxjs@clsname
59   {The engine in use (\bxjs@tmpa) is not supported}
60   {It's a fatal error. I'll quit right now.}
61   \expandafter\@firstofone
62 \fi{\endinput\@@end}

```

LuaT_EX の場合、本クラス用の Lua モジュールを用意する。

```

63 \ifx l\jsEngine

```

```
64 \directlua{ bxjs = {} }
65 \fi
```

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

```
66 \ifjsWithTeX \let\bxjs@protected\protected
67 \else \let\bxjs@protected\empty
68 \fi
```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

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

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

TODO: `iftex` パッケージ (+自前実装) に置き換える。

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

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

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

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

※ 2017 年 1 月以降の L^AT_EX カーネルにおいて「Unicode を表す L^AT_EX 公式のフォントエンコーディング」である “TU” が導入され、これ以降の L^AT_EX を Xe_LTeX または LuaTeX で動かしている場合は、既定のエンコーディングが TU になる。それ以外の場合は、既定のエンコーディングは OT1 である。

```
84 \newif\ifbxjs@TUenc
85 \def\bxjs@tmpa{TU}\edef\bxjs@tmpb{\f@encoding}
86 \ifx\bxjs@tmpa\bxjs@tmpb
87 \bxjs@TUenctrue
88 \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。

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

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

```
91 \@ifl@t@r\fmtversion{2021/11/15}{%
92   \def\bxjs@CGHN#1{#1}%
93 }{%else
94   \def\bxjs@CGHN#1{\bxjs@CGHN@a#1//}%
95   \def\bxjs@CGHN@a#1/#2/#3//{#1/#3/#2}}
```

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

```
96 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
97   #1\expandafter\@firstoftwo
98   \else\expandafter\@secondoftwo
99   \fi}
```

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

```
100 \def\bxjs@cslet#1{%
101   \expandafter\let\csname#1\endcsname}
```

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

```
102 \def\bxjs@csletcs#1#2{%
103   \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。

```
104 \def\bxjs@catopt#1#2{%
105   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}
```

`\bxjs@ifplus` `\@ifstar` の + 版。

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

`\bxjs@trim` `\bxjs@trim\CS` で、`\CS` の内容のトークン列を先頭と末尾の空白トークン群を除去したものに置き換える。

```
107 \def\bxjs@trim#1{\expandafter\bxjs@trim@a#1\@nil#1}
108 \def\bxjs@trim@a{\futurelet\bxjs@tmpb\bxjs@trim@b}
109 \def\bxjs@trim@b{\bxjs@cond\ifx\bxjs@tmpb\@sptoken\fi
110   {\bxjs@trim@c\bxjs@trim@a}{\bxjs@trim@d}}
111 \def\bxjs@trim@c#1 {#1}
112 \def\bxjs@trim@d#1\@nil{\bxjs@trim@e#1\@nil: \@nil\@nnil}
113 \def\bxjs@trim@e#1 \@nil#2\@nnil{\bxjs@cond\ifx\@nnil#2\@nnil\fi
114   {\bxjs@trim@f#1\@nnil}{\bxjs@trim@e#1\@nil: \@nil\@nnil}}
115 \def\bxjs@trim@f#1\@nil#2\@nnil#3{\def#3{#1}}
```

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

※各要素について、先頭・末尾の空白トークン群は除去される。

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

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

```
122 \let\c@bxjs@tempcnta\@tempcnta
123 \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）。

```
124 \def\jsSetQHLLength#1#2{%
125   \begingroup
126   \bxjs@parse@qh{#2}%
127   \ifx\bxjs@tmpb\relax
128     \setlength\@tempdima{#2}%
129     \xdef\bxjs@g@tmpa{\the\@tempdima}%
130   \else \global\let\bxjs@g@tmpa\bxjs@tmpb
131   \fi
132 \endgroup
133 #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。

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

```

150 \ifx#3\relax
151 \ClassError\bxjs@clsname
152 {You cannot use '\bxjs@tmpa' here}{\@ehc}%
153 \def\bxjs@tmpb{Opt}%
154 \else
155 \@tempdimb#3\relax \@tempdimb#1\@tempdimb
156 \edef\bxjs@tmpb{\the\@tempdimb}%
157 \fi
158 \fi}

```

今の段階では Q/H だけが使用可能。

```

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

```

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

```

160 \newif\ifbxjs@after@preamble

```

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

```

161 \@onlypreamble\bxjs@begin@document@hook
162 \def\bxjs@begin@document@hook{\bxjs@after@preambletrue}
163 \AtBeginDocument{\bxjs@begin@document@hook}

```

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

```

164 \@onlypreamble\bxjs@post@option@hook
165 \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。

```

166 \@onlypreamble\bxjs@pre@jadriver@hook
167 \let\bxjs@pre@jadriver@hook\@empty

```

一時的な手続き用の制御綴。

```

168 \@onlypreamble\bxjs@tmpdo
169 \@onlypreamble\bxjs@tmpdo@a
170 \@onlypreamble\bxjs@tmpdo@b
171 \@onlypreamble\bxjs@tmpdo@c
172 \@onlypreamble\bxjs@tmpdo@d

```

`\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。

```

173 \bxjs@robust\def\jsInhibitGlue{%
174 \ifx\inhibitglue\undefined\else \inhibitglue \fi}

```

■環境検査 ☹ `TEX` 処理系のバージョンがサポート対象であるかを確認する。

※現状での処理系バージョン要件は「X_qT_EX は 0.997 版（2007 年頃）以上」という現実離れしたものになっている。

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

```

175 \@tempwattrue
176 \if x\jsEngine

```

```

177 \ifdim\the\XeTeXversion\XeTeXrevision\p<0.997\p@
178 \@tempwafalse \fi
179 \fi

```

非サポートのバージョン場合は強制終了させる。

```

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

```

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

```

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

```
197 \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。

```
198 \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。

```
199 %<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。

```
200 %<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。

```
201 %<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。

```
202 \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 1m^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 1.5m^2 ですが、ISO では B1 判の辺の長さが A0 判と A1 判の辺の長さの幾何平均です。したがって ISO の B0 判は $1000\text{mm} \times 1414\text{mm}$ です。このため、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$ の `b5paper` は $250\text{mm} \times 176\text{mm}$ ですが、 $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$ の `b5paper` は $257\text{mm} \times 182\text{mm}$ になっています。ここでは $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$ にならって JIS に従いました。

デフォルトは `a4paper` です。

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

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

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

```

222 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
223 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
224 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
225 \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。

```

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

```

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

```

237 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
238 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
239 \DeclareOption{screenpaper}{\bxjs@setpaper{screen}}
240 \DeclareOption{custompaper}{}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

241 \newif\if@landscape
242 \@landscapefalse
243 \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ため、単純のため article のみで使えるオプションとしました。

```

244 \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。

```

245 %<!slide>\@slidefalse
246 %<slide>\@slidetrue

```

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

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

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

```
247 \newcommand{\@ptsize}{0}
248 \def\bxjs@param@basefontsize{10pt}
249 \def\jsUnusualPtSize{-20}
```

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

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

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

※クラスオプションのトークン列の中に展開可能なトークンがある場合, L^AT_EX 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。

```
251 \jsSetQHLlength\@tempdima{#1}%
252 \edef\bxjs@param@basefontsize{\the\@tempdima}%
253 \ifdim\@tempdima=10pt \long\def\@ptsize{0}%
254 \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
255 \else\ifdim\@tempdima=12pt \long\def\@ptsize{2}%
256 \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。

```
257 \def\bxjs@setjbasefontsize#1{%
258 \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。

```
259 \newif\ifjsc@mag
260 \newif\ifjsc@mag@xreal
261 %\let\jsc@magscale\undefined
262 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
263 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
```

```

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

```

JS クラス互換の magstyle 設定オプション。

```

281 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@@usemag}
282 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@@nomag}
283 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@@xreal}

```

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

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

```

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

```

299 \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。

```
300 \if j\jsEngine
301 \DeclareOption{mentuke}{%
302   \tombowtrue \tombowdatefalse
303   \setlength{\@tombowwidth}{\z@}%
304   \maketombowbox}
305 \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。

```
306 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
307 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
308 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■**二段組** `twocolumn` で二段組になります。

```
309 \DeclareOption{onecolumn}{\@twocolumnfalse}
310 \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。

```
311 \DeclareOption{titlepage}{\@titlepagetrue}
312 \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。

```
313 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
314 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
315 %<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。

```
316 \def\eqnarray{%
317   \stepcounter{equation}%
318   \def\@currentlabel{\p@equation\theequation}%
319   \global\@eqnswtrue
```



```

320 \m@th
321 \global\@eqcnt\z@
322 \tabskip\@centering
323 \let\@@@eqncr
324 $$\everycr{}\halign to\displaywidth\bgroup
325     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
326     &\global\@eqcnt\@ne \hfil$\displaystyle{##}\$ \hfil
327     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
328     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
329         \tabskip\z@skip
330     \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。

```

331 \DeclareOption{leqno}{\input{leqno.clo}}
332 \DeclareOption{fleqn}{\input{fleqn.clo}}
333 % fleqn 用の eqnarray 環境の再定義
334 \def\eqnarray{%
335     \stepcounter{equation}%
336     \def\@currentlabel{\p@equation\theequation}%
337     \global\@eqnswtrue\m@th
338     \global\@eqcnt\z@
339     \tabskip\mathindent
340     \let\@@@eqncr
341     \setlength\abovedisplayskip{\topsep}%
342     \ifvmode
343         \addtolength\abovedisplayskip{\partopsep}%
344     \fi
345     \addtolength\abovedisplayskip{\parskip}%
346     \setlength\belowdisplayskip{\abovedisplayskip}%
347     \setlength\belowdisplayshortskip{\abovedisplayskip}%
348     \setlength\abovedisplayshortskip{\abovedisplayskip}%
349     $$\everycr{}\halign to\linewidth% $$
350     \bgroup
351     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
352     &\global\@eqcnt\@ne \hfil$\displaystyle{##}\$ \hfil
353     &\global\@eqcnt\tw@
354         $\displaystyle{##}$\hfil \tabskip\@centering
355     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
356     \tabskip\z@skip\cr
357 }

```

■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。

```

358 % \DeclareOption{openbib}{%
359 %     \AtEndOfPackage{%
360 %         \renewcommand\@openbib@code{%
361 %             \advance\leftmargin\bibindent

```

```

362 %      \itemindent -\bibindent
363 %      \listparindent \itemindent
364 %      \parsep \z@}%
365 %      \renewcommand\newblock{\par}}

```

■数式フォントとして和文フォントを登録しないオプション 数式中では 16 通りのフォントしか使えません。AMSFonTS や mathptmx パッケージを使って数式フォントをたくさん使うと “Too many math alphabets ...” というエラーが起こってしまいます。disablejfam オプションを付ければ、明朝・ゴシックを数式用フォントとして登録するのをやめますので、数式用フォントが二つ節約できます。いずれにしても \textmc や \mbox や amsmath パッケージの \text を使えば数式中で和文フォントが使えますので、この新ドキュメントクラスでは標準で和文フォントを数式用に登録しないことにしていたのですが、従来のドキュメントクラスの仕様に合わせることにしました。

\bxjs@enablejfam [暗黙文字トークン] enablejfam オプションの状態：

```

366 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。
367 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
368 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
369 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
370 \define@key{bxjs}{enablejfam}[true]{%
371   \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。

```

372 \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。

```

373 \newif\ifjsDraft
374 \DeclareOption{draft}{\jsDrafttrue \overfullrule=5pt }
375 \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, dviptdpmx など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。

```
376 \newif\ifbxjs@papersize
377 \bxjs@papersizetrue
378 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
379 \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。

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

BXJS クラスでは、nojslogo を既定とする。

```
383 \newif\if@jslogo \@jslogofalse
384 \DeclareOption{jslogo}{\@jslogotrue}
385 \DeclareOption{nojslogo}{\@jslogofalse}
```

■複合設定オプション

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

\bxjs@invscale \bxjs@invscale は T_EX における「長さのスケール」の逆関数を求めるもの。例えば \bxjs@invscale\dimX{1.3} は \dimX=1.3\dimX の逆の演算を行う。

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

※元の長さが 128 pt 以上の場合でも動作するように修正した。

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

複合設定オプションとは、「エンジンやドライバや和文ドライバの設定を含む、複数の設定

を一度に行うオプション」のことである。ある特定の設定を短く書く必要が高いと判断される場合に用意される。

`pandoc` オプションは、Pandoc で \LaTeX 用の既定テンプレートを用いて他形式から \LaTeX (および PDF) 形式に変換する用途に最適化した設定を与える。

```
411 \DeclareOption{pandoc}{%
412   \bxjs@apply@pandoc@opt}
413 \@onlypreamble\bxjs@apply@pandoc@opt
414 \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。

```
415   \g@addto@macro\bxjs@post@option@hook{%
416     \bxjs@oldfontcommandstrue
417     \setkeys{bxjs}{ja=pandoc}%
418     \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)。

```
419   \ifx\bxjs@driver@opt\undefined
420     \def\bxjs@driver@opt{dvipdfmx}%
421     \bxjs@dvi@opttrue
422   \fi
423   \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。

```
424 \DeclareOption{pandoc+}{%
425   \g@addto@macro\bxjs@post@option@hook{%
426     \edef\jsJaParam{\bxjs@catopt{+_plus}\jsJaParam}}%
427   \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。

```
428 %\let\bxjs@engine@given\undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
429 %\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 \LaTeX の慣習に従って「ドライバの指定」とみなすべきだから。

```
430 \DeclareOption{autodetect-engine}{%
431   \let\bxjs@engine@given=*}
432 \DeclareOption{latex}{%
```

```

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

```

457 %\let\bxjs@driver@given\undefined
458 \let\bxjs@driver@@dvimode=0
459 \let\bxjs@driver@@dvipdfmx=1
460 \let\bxjs@driver@@pdfmode=2
461 \let\bxjs@driver@@xetex=3
462 \let\bxjs@driver@@dvips=4
463 \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。

```

464 %\let\bxjs@driver@opt\undefined

```

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

```

465 \DeclareOption{dvips}{%
466 \def\bxjs@driver@opt{dvips}%
467 \let\bxjs@driver@given\bxjs@driver@@dvips}
468 \DeclareOption{dviout}{%
469 \def\bxjs@driver@opt{dviout}%
470 \let\bxjs@driver@given\bxjs@driver@@dvimode}
471 \DeclareOption{xdvi}{%
472 \def\bxjs@driver@opt{xdvi}%
473 \let\bxjs@driver@given\bxjs@driver@@dvimode}
474 \DeclareOption{dvipdfmx}{%

```

```

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

dvipdfmx-if-dvi は 2.0 版より非推奨となった。
492 \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 493 \@onlypreamble\bxjs@depre@opt
494 \def\bxjs@depre@opt#1#2{%
495 \ClassWarningNoLine\bxjs@clsname
496 {The old option '#1' is DEPRECATED\MessageBreak
497 and may be abolished in future!\MessageBreak
498 You should instead write:\MessageBreak
499 \space\space #2}}
500 \@onlypreamble\bxjs@depre@opt@do
501 \def\bxjs@depre@opt@do#1#2{%
502 \bxjs@depre@opt{#1}{#2}%
503 \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。

```

504 \newif\ifbxjs@bigcode \bxjs@bigcodetrue

nobigcode /bigcode オプションの定義。
505 \DeclareOption{nobigcode}{%

```

```

506 \bxjs@bigcodefalse}
507 \DeclareOption{bigcode}{%
508 \bxjs@bigcodetrue}

```

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

```

509 \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 `enabledeprecatfontcommands` であるがこれはチョットアレなので避けた。

```

510 \DeclareOption{nooldfontcommands}{%
511 \bxjs@oldfontcommandsfalse}
512 \DeclareOption{oldfontcommands}{%
513 \bxjs@oldfontcommandstrue}


```

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

```

514 \DeclareOption{winjis}{%
515 \ClassWarningNoLine\bxjs@clsname
516 {This class does not support `winjis' option}%
517 }
518 \DeclareOption{mingoth}{%
519 \ClassWarningNoLine\bxjs@clsname
520 {This class does not support `mingoth' option}%
521 }
522 \DeclareOption{jis}{%
523 \ClassWarningNoLine\bxjs@clsname
524 {This class does not support `jis' option}%
525 }

```

■`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。

```

526 \DeclareOption*{%
527 \def\bxjs@next{\bxjs@safe@setkeys{bxjs}}%
528 \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。

```

529 \def\bxjs@safe@setkeys#1#2{%
530 \let\bxjs@save@KV@errx\KV@errx \let\KV@errx\@gobble
531 \setkeys{#1}{#2}%
532 \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。


```

533 \@onlypreamble\bxjs@declare@enum@option
534 \def\bxjs@declare@enum@option#1#2#3{%
535   \bxjs@csletcs{bxjs@#2}{bxjs@#2@#3}%
536   \define@key{bxjs}{#1}{%
537     \expandafter\ifx\csname bxjs@#2@##1\endcsname\relax
538     \bxjs@error@keyval{#1}{##1}%
539   \else \bxjs@csletcs{bxjs@#2}{bxjs@#2@##1}%
540   \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。

```

541 \@onlypreamble\bxjs@declare@bool@option
542 \def\bxjs@declare@bool@option#1#2#3{%
543   \csname newif\expandafter\endcsname\csname ifbxjs@#2\endcsname
544   \@nameuse{bxjs@#2#3}%
545   \define@key{bxjs}{#1}[true]{%
546     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
547     \bxjs@error@keyval{#1}{##1}%
548   \else \@nameuse{bxjs@#2##1}%
549   \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。

```

550 \def\bxjs@set@keyval#1#2#3{%
551   \bxjs@csletcs{bxjs@next}{bxjs@kv@#1@#2}%
552   \ifx\bxjs@next\relax
553     \bxjs@error@keyval{#1}{#2}%
554     #3%
555   \else \bxjs@next
556   \fi}
557 \@onlypreamble\bxjs@error@keyval
558 \def\bxjs@error@keyval#1#2{%
559   \ClassError\bxjs@clsname
560   {Invalid value '#2' for option #1}\@ehc}

```

`\jsScale` [実数値マクロ] 和文スケール値。

```
561 \def\jsScale{0.924715}
```

`\bxjs@base@opt` 明示された base オプションの値。

```
562 %\let\bxjs@base@opt\undefined
```

base オプションの処理。

```

563 \define@key{bxjs}{base}{%
564   \edef\bxjs@base@opt{#1}%
565   \bxjs@setbasefontsize{#1}}
566 \define@key{bxjs}{fontsize}{\setkeys{bxjs}{base=#1}}

```

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

```
567 %\let\bxjs@jbase@opt\@undefined
```

jbase オプションの処理。

```
568 \define@key{bxjs}{jbase}{\edef\bxjs@jbase@opt{#1}}
```

```
569 \define@key{bxjs}{jafontsize}{\setkeys{bxjs}{jbase=#1}}
```

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

```
570 %\let\bxjs@scale@opt\@undefined
```

scale オプションの処理。

```
571 \define@key{bxjs}{scale}{%
```

```
572 \edef\bxjs@scale@opt{#1}%
```

```
573 \let\jsScale\bxjs@scale@opt}
```

```
574 \define@key{bxjs}{jafontscale}{\setkeys{bxjs}{scale=#1}}
```

noscale オプションの処理。

TODO: noscale は 3.0 版で廃止の予定。

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

`\bxjs@param@mag` mag オプションの値。

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

mag オプションの処理。

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

paper オプションの処理。

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

`\bxjs@jadriver` 和文ドライバの名前。

```
579 \let\bxjs@jadriver\relax
```

```
580 %\let\bxjs@jadriver@opt\@undefined
```

ja オプションの処理。

※jadriver は 0.9 版で用いられた旧称。

TODO: jadriver は 3.0 版で廃止の予定。

※単なる ja という指定は無視される (Pandoc 対策)。

```
581 \define@key{bxjs}{jadriver}{%
```

```
582 \bxjs@depre@opt{jadriver}{ja=#1}\edef\bxjs@jadriver@opt{#1}}
```

```
583 \define@key{bxjs}{ja}[\relax]{%
```

```
584 \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}
```

`\jsJaFont` 和文フォント設定の名前。

```
585 \let\jsJaFont\@empty
```

jafont オプションの処理。

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

`\jsJaParam` 和文ドライバパラメタの文字列。

```
587 \let\jsJaParam\@empty
```

japaram オプションの処理。

```
588 \define@key{bxjs}{japaram}{%
589   \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。

```
590 \define@key{bxjs}{pandoc}[]{%
591   \ExecuteOptions{pandoc}%
592   \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}
593 \define@key{bxjs}{pandoc+}[]{%
594   \ExecuteOptions{pandoc+}%
595   \edef\jsJaParam{\bxjs@catopt\jsJaParam{#1}}
```

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

```
596 \let\bxjs@magstyle@@mag=m
597 \let\bxjs@magstyle@@real=r
598 \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。

```
599 \let\bxjs@magstyle@@usemag\bxjs@magstyle@@mag
600 \let\bxjs@magstyle@@nomag\bxjs@magstyle@@real
601 \bxjs@cslet{bxjs@magstyle@@nomag*}\bxjs@magstyle@@xreal
```

`\bxjs@magstyle@@default` は既定の値を表す。

```
602 \let\bxjs@magstyle@@default\bxjs@magstyle@@usemag
603 \ifx l\jsEngine \ifnum\luatexversion>86
604   \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
605 \fi\fi
606 \ifjsWithpTeXng
607   \let\bxjs@magstyle@@default\bxjs@magstyle@@xreal
608 \fi
609 \let\bxjs@magstyle\bxjs@magstyle@@default
```

magstyle オプションの処理。

```
610 \define@key{bxjs}{magstyle}{%
611   \bxjs@csletcs{bxjs@magstyle}{bxjs@magstyle@@#1}%
612   \ifx\bxjs@magstyle\relax
613     \bxjs@error@keyval{magstyle}{#1}%
614     \let\bxjs@magstyle\bxjs@magstyle@@default
615   \fi}
```

`\bxjs@geometry` geometry オプションの指定値。

```
616 \let\bxjs@geometry@@class=c
617 \let\bxjs@geometry@@user=u
618 \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。

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

`\ifbxjs@dvi@opt` dvi オプションが指定されたか。

```
620 \newif\ifbxjs@dvi@opt
```

DVI モードのドライバとドライバ種別との対応。

```
621 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
```

```
622 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
```

```
623 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
```

```
624 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
```

```
625 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none
```

```
626 \bxjs@cslet{bxjs@dvidriver@@nodvidriver*}\bxjs@driver@@none
```

dvi オプションの処理。

```
627 \define@key{bxjs}{dvi}{%
```

```
628 \bxjs@csletcs{bxjs@tmpa}{bxjs@dvidriver@@#1}%
```

```
629 \ifx\bxjs@tmpa\relax
```

```
630 \bxjs@error@keyval{dvi}{#1}%
```

```
631 \else
```

`\bxjs@driver@given` を未定義にしていることに注意。

```
632 \def\bxjs@driver@opt{#1}%
```

```
633 \let\bxjs@driver@given\undefined
```

```
634 \bxjs@dvi@opttrue
```

```
635 \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。

```
636 \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。

```
637 \newif\ifbxjs@force@chapterabstract
```

```
638 %<book>\bxjs@force@chapterabstracttrue
```

`layout` オプションの処理。

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

```
640 %<book>\bxjs@layout@buggyhmargintrue
```

```
641 %<report>\bxjs@force@chapterabstracttrue
```

```
642 }
```

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

```
644 %<book>\bxjs@layout@buggyhmarginfalse
```

```
645 %<report>\bxjs@force@chapterabstractfalse
```

```
646 }
```

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

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

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

```
649 %\let\bxjs@textwidth@limit@opt\undefined
```

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

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

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

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

```
653 %\let\bxjs@textwidth@opt\@undefined
654 \define@key{bxjs}{textwidth}{\edef\bxjs@textwidth@opt{#1}}
655 \define@key{bxjs}{line_length}{\setkeys{bxjs}{textwidth=#1}}
```

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

```
656 %\let\bxjs@number@of@lines@opt\@undefined
657 \define@key{bxjs}{number-of-lines}{\edef\bxjs@number@of@lines@opt{#1}}
658 \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。

```
659 %\let\bxjs@paragraph@mark\@undefined
660 \define@key{bxjs}{paragraph-mark}{%
661 \edef\bxjs@paragraph@mark{#1}}
```

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

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

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

```
663 \bxjs@declare@bool@option{jaspace-cmd}{jaspace@cmd}{true}
664 \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。

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

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

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

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

```
667 \chardef\bxjs@everyparhook@@none=0
668 \chardef\bxjs@everyparhook@@compat=1
669 \chardef\bxjs@everyparhook@@modern=2
670 \bxjs@declare@enum@option{everyparhook}{everyparhook}{%
671 \if j\jsEngine compat\else modern\fi}
```

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

```
672 \chardef\bxjs@label@section@@none=0
673 \chardef\bxjs@label@section@@compat=1
674 \chardef\bxjs@label@section@@modern=2
675 \bxjs@declare@enum@option{label-section}{label@section}{compat}
```

`\ifbxjs@usezw` [スイッチ] `use-zw` の指定値。

TODO: `zw/nozw` は 3.0 版で廃止の予定。

```
676 \bxjs@declare@bool@option{use-zw}{usezw}{true}
677 \DeclareOption{nozw}{\bxjs@depre@opt@do{nozw}{use-zw=false}}
678 \DeclareOption{zw}{\bxjs@depre@opt@do{zw}{use-zw=true}}
```

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

TODO: js/nojs は 3.0 版で廃止の予定。

```
679 \bxjs@declare@bool@option{disguise-js}{disguise@js}{true}
680 \DeclareOption{nojs}{\bxjs@depre@opt@do{nojs}{disguise-js=false}}
681 \DeclareOption{js}{\bxjs@depre@opt@do{js}{disguise-js=true}}
```

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

```
682 \bxjs@declare@bool@option{precise-text}{precisetext}{false}
683 \DeclareOption{noprecisetext}{\bxjs@depre@opt@do{noprecisetext}{precise-
  text=false}}
684 \DeclareOption{precisetext}{\bxjs@depre@opt@do{precisetext}{precise-
  text=true}}
```

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

```
685 \bxjs@declare@bool@option{simple-ja-setup}{simplejasetup}{true}
686 \DeclareOption{nosimplejasetup}{\bxjs@depre@opt@do{nosimplejasetup}{simple-ja-
  setup=false}}
687 \DeclareOption{simplejasetup}{\bxjs@depre@opt@do{simplejasetup}{simple-ja-
  setup=true}}
```

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

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

■ オプションの実行

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

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

```
690 \let\bxjs@org@removeelement\@removeelement
691 \def\@removeelement#1#2#3{%
692   \def\reserved@a{#2}%
693   \ifx\reserved@a\@empty \let#3\@empty
694   \else \bxjs@org@removeelement{#1}{#2}{#3}%
695   \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。

```
696 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
697 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
698 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
699 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
700 \ProcessOptions\relax
701 \bxjs@post@option@hook
```

後処理

```
702 \if@slide
703 \def\maybeblue{\@ifundefined{ver@color.sty}{{\color{blue}}}
704 \fi
705 \if@landscape
706 \setlength\@tempdima {\paperheight}
707 \setlength\paperheight{\paperwidth}
708 \setlength\paperwidth {\@tempdima}
709 \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。

```
710 \def\bxjs@tmpdo{%
711 \def\bxjs@tmpa{\gobble}%
712 \expandafter\bxjs@tmpdo@a\@classoptionslist,\@nil,%
713 \let\@classoptionslist\bxjs@tmpa}
714 \def\bxjs@tmpdo@a#1,{%
715 \ifx\@nil#1\relax\else
716 \bxjs@tmpdo@b#1}\@nil
717 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
718 \expandafter\bxjs@tmpdo@a
719 \fi}
720 \def\bxjs@tmpdo@b#1#\{bxjs@tmpdo@c}
721 \def\bxjs@tmpdo@c#1\@nil{%
722 \ifx\@nil#1\@nil \@tempwatruel \else \@tempwafalse \fi}
723 \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。

```
724 \@expandtwoargs\@removeelement
725 {papersize}\@classoptionslist\@classoptionslist
726 \@expandtwoargs\@removeelement
727 {10pt}\@classoptionslist\@classoptionslist
728 \@expandtwoargs\@removeelement
729 {noscale}\@classoptionslist\@classoptionslist
```

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

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

このコードを削除。

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

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

```
730 \ifx\bxjs@jadriver@opt\@undefined\else
731   \let\bxjs@jadriver\bxjs@jadriver@opt
732 \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。

```
733 \let\bxjs@tmpb\jsEngine
734 \ifx j\bxjs@tmpb\ifjsWithpTeXng
735   \let\bxjs@tmpb=g
736 \fi\fi
737 \ifx j\bxjs@tmpb\ifjsWithupTeX
738   \let\bxjs@tmpb=u
739 \fi\fi
740 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
741   \let\bxjs@tmpb=n
742 \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。)

```
743 \ifx *\bxjs@engine@given
744   \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。

```
745 \ifx j\bxjs@engine@given
746   \g@addto@macro\@classoptionslist{,latex}
747 \else\ifx u\bxjs@engine@given
748   \g@addto@macro\@classoptionslist{,uplatex}
749 \fi\fi
750 \fi
751 \ifx\bxjs@engine@given\@undefined\else
752   \ifx\bxjs@engine@given\bxjs@tmpb\else
753     \ClassError\bxjs@clsname
754     {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
755   \fi
756 \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。

```
757 \ifjsWithpTeXng
758   \g@addto@macro\@classoptionslist{,uplatex}
```


759 \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。

```
760 \@tempswatruue
761 \ifx \bxjs@driver@given\@undefined\else
762   \ifjsInPdfMode
763     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
764       \@tempswafalse
765     \fi
766   \else\ifx x\jsEngine
767     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
768       \@tempswafalse
769     \fi
770   \else
771     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
772       \@tempswafalse
773     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
774       \@tempswafalse
775     \fi\fi
776     \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
777       \@tempswafalse
778     \fi\fi
779   \fi\fi
780 \fi
781 \if@tempswa\else
782   \ClassError\bxjs@clsname
783   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
784 \fi
```

DVI 出力のエンジンである場合の追加処理。

```
785 \ifjsInPdfMode \@tempswafalse
786 \else\ifx x\jsEngine \@tempswafalse
787 \else\ifjsWithpTeXng \@tempswafalse
788 \else \@tempswatruue
789 \fi\fi\fi
790 \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)。

```
791 \ifx\bxjs@driver@opt\@undefined
792   \if \ifbxjs@explIII T\else\ifx\bxjs@jadriver@opt\@undefined F\else T\fi\fi T%
793   \ClassWarningNoLine\bxjs@clsname
794   {A driver option is MISSING!!\MessageBreak
795   You should properly specify one of the valid\MessageBreak
796   driver options according to the DVI driver\MessageBreak
797   that is in use:\MessageBreak
798   \@spaces dvips, dvipdfmx, dviout, xdvi,\MessageBreak
799   \@spaces nodvidriver}
```

```
800 \fi
801 \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。)

```
802 \ifbxjs@dvi@opt
803 \edef\bxjs@next{%
804 \let\noexpand\bxjs@driver@given
805 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
806 \noexpand\g@addto@macro\noexpand\@classoptionslist
807 {,\bxjs@driver@opt}%
808 }\bxjs@next
809 \fi
810 \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。

```
811 \ifjsWithpTeXng
812 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
813 \let\bxjs@platexng@nodrv@\undefined
814 \else\ifx t\bxjs@platexng@nodrv\else
815 \g@addto@macro\@classoptionslist{,dvipdfmx}
816 \fi\fi
817 \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。

```
818 \ifx\bxjs@driver@given\bxjs@driver@@none
819 \bxjs@papersizefalse
820 \fi
```

■その他の BXJS 特有の後処理 ☹ 標準の和文ドライバの名前の定数。

```
821 \def\bxjs@@minimal{minimal}
822 \def\bxjs@@standard{standard}
823 \def\bxjs@@pandoc{pandoc}
824 \def\bxjs@@modern{modern}
```

\bxjs@jadriver の正規化。値が未指定の場合は minimal に変える。ただしエンジンが (u)pTeX である場合は standard に変える。

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

```
825 \ifx\bxjs@jadriver\relax
826 \ifx j\jsEngine
827 \let\bxjs@jadriver\bxjs@@standard
828 \else
829 \ClassWarningNoLine\bxjs@clsname
830 {The option 'ja' is MISSING!!\MessageBreak
831 So 'ja=minimal' is assumed as fallback, but\MessageBreak
832 such implicit setting is now DEPRECATED!\MessageBreak
833 You should write 'ja=minimal' explicitly,\MessageBreak
```

```

834     if it is intended}
835     \let\bxjs@jadriver\bxjs@@minimal
836     \fi
837 \fi

```

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

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

※Pandoc モードでは plautopatch の既定値を真とする。

```

838 \ifx\bxjs@jadriver\bxjs@@pandoc \ifx\bxjs@plautopatch@given\@undefined
839     \ifjsWitheTeX
840     \bxjs@plautopatchtrue
841 \fi\fi\fi
842 \ifx j\jsEngine \ifbxjs@plautopatch
843     \RequirePackage{plautopatch}[2018/08/22]%v0.3
844 \fi\fi

```

エンジンオプションがない場合はエラーを出す。

※ただし ja 非指定の場合はスキップする。

```

845 \ifx\bxjs@jadriver@opt\@undefined\else
846     \ifx\bxjs@engine@given\@undefined
847     \ClassError\bxjs@clsname
848     {An engine option must be explicitly given}%
849     {When you use a Japanese-driver you must specify a correct\MessageBreak
850     engine option.\MessageBreak \@ehc}
851 \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。)

```

852 \ifx\bxjs@magstyle@@default\bxjs@magstyle@@mag\else
853     \ifx\bxjs@magstyle\bxjs@magstyle@@mag
854     \let\bxjs@magstyle\bxjs@magstyle@@default
855     \ClassError\bxjs@clsname
856     {The engine does not support 'magstyle=usemag'}%
857     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
858     The default value 'nomag*' is used instead.\MessageBreak \@ehc}
859     \fi
860 \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。

```

861 \ifx\bxjs@jbase@opt\@undefined\else
862     \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。

```

863     \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
864     \bxjs@invscale\@tempdima\jsScale

```

```

865 \bxjs@setbasefontsize{\@tempdima}%
866 \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。

867 \ifx\bxjs@scale@opt\undefined\else
868 \ClassWarningNoLine\bxjs@clsname
869 {Redundant 'scale' option is ignored}%
870 \fi
871 \jsSetQHLength\@tempdima{\bxjs@jbase@opt}%
872 \@tempdimb=\bxjs@param@basefontsize\relax
873 \edef\jsScale{\strip@pt\@tempdimb}%
874 \bxjs@invscale\@tempdima\jsScale
875 \edef\jsScale{\strip@pt\@tempdima}%
876 \fi
877 \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。

```

878 \let\Cjascale\jsScale

```

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

```

879 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
880 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
881 \catcode\@tempcnta\active
882 \advance\@tempcnta\@ne
883 \repeat
884 \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。

```

885 \ifbxjs@disguise@js
886 %<book|report>\def\bxjs@js@clsname{jsbook}
887 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
888 \@namedef{ver\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
889 \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。

```

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

```

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

```

891 \ifbxjs@oldfontcommands
892 \AtEndOfClass{\allowoldfontcommands}
893 \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。

```
894 \newdimen\stockwidth \newdimen\stockheight
895 \begingroup\expandafter\expandafter\expandafter\endgroup
896 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
897 % \newdimen\stockwidth \newdimen\stockheight
898 \setlength{\stockwidth}{\paperwidth}
899 \setlength{\stockheight}{\paperheight}
900 \advance \stockwidth 2in
901 \advance \stockheight 2in
902 \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。

```
903 %<slide>\def\n@baseline{13}%
904 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
905 %<!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。

```
906 \ifx\bxjs@magstyle\bxjs@magstyle@@mag
907   \jsc@magtrue
908 \else\ifx\bxjs@magstyle\bxjs@magstyle@@xreal
909   \jsc@mag@xrealtrue
910 \fi\fi
```

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

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

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

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

```
911 \ifx\bxjs@param@mag\relax
912   \@tempdima=\bxjs@param@basefontsize
913   \advance\@tempdima.001pt \multiply\@tempdima25
914   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
915   \edef\bxjs@param@mag{\the\@tempcnta}
916 \else
917 % mag 値が直接指定された場合
918   \bxjs@gset@tempcnta{\bxjs@param@mag}
919   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
920 % 有効な mag 値の範囲は 1--32768
921   \edef\bxjs@param@mag{\the\@tempcnta}
922   \advance\@tempcnta100000
923   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
924   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
925   \edef\bxjs@param@basefontsize{\the\@tempdima}
926 \fi
927 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
928 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
929 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
930 \edef\jsc@magscale{\strip@pt\@tempdima}
931 \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。

```

932 \newdimen\jsc@mpt
933 \newdimen\jsc@mmm
934 \ifjsc@mag
935   \jsc@mpt=1\p@
936   \jsc@mmm=1mm
937   \catcode`\?=9 % \p@? read as \p@
938 \else
939   \jsc@mpt=\jsc@magscale\p@
940   \jsc@mmm=\jsc@magscale mm
941   \catcode`\?=11 \let\p@?\jsc@mpt
942 \fi
943 \chardef\bxjs@qmcc=\catcode`\?\relax
944 \g@addto@macro\bxjs@pre@jadriver@hook{\catcode`\?=12\relax}

```

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

use-zw が真の時は \zw を \jsZw と同義にする。

```

945 \newdimen\jsZw
946 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
947 \ifbxjs@usezw
948   \providecommand*\zw{\jsZw}
949 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

951 \ifjsc@mag@xreal
952   \RequirePackage{type1cm}
953   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

954 \ifbxjs@TUenc
955   \expandafter\let\csname TU/lmr/m/n/10\endcsname\relax
956 \else
957   \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
958 \fi
959 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
960 \let\jsc@get@external@font\get@external@font
961 \def\get@external@font{%
962   \jsc@preadjust@extract@font
963   \jsc@get@external@font}
964 \def\jsc@fstrunc#1{%

```

```

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

```

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

```

\jsc@smallskip
\jsc@medskip 998 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 999 %\def\jsc@medskip{\vspace\jsc@medskipamount}
1000 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 1001 \newskip\jsc@smallskipamount
\jsc@bigskipamount 1002 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
1003 %\newskip\jsc@medskipamount
1004 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
1005 %\newskip\jsc@bigskipamount

```



```
1006 %\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。

```
1007 % \ifpapersize
1008 %   \setlength{\@tempdima}{\paperwidth}
1009 %   \setlength{\@tempdimb}{\paperheight}
1010 %   \iftombow
1011 %     \advance \@tempdima 2truein
1012 %     \advance \@tempdimb 2truein
1013 %   \fi
1014 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
1015 % \fi
```

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

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

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

BXJS クラスでの変更点:

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

```
1016 \chardef\bxjs@periodchar=`\ .
1017 \bxjs@protected\def\bxjs@SE{%
1018   \ifnum\spacefactor<\@m \spacefactor\@m
1019   \else \spacefactor\sfcode\bxjs@periodchar
```

```

1020 \fi}
1021 \ifbxjs@fix@at@cmd
1022 \def\@{\bxjs@SE{}}
1023 \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>\@xiipt</code>	12	<code>\@xivpt</code>	14.4

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

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

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

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

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

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

```

1024 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
1025 \expandafter\bxjs@tmpa\expandafter{%
1026 \set@fontsize{#1}{#2}{#3}%
1027 % 末尾にコードを追加

```

```

1028 \expandafter\def\expandafter\size@update\expandafter{%
1029   \size@update
1030   \jsFontSizeChanged}%
1031 }

```

`\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。

```

1032 \newcommand*\jsFontSizeChanged{%
1033   \jsZw=\f@size\p@
1034   \jsZw=\jsScale \jsZw
1035   \ifdim\parindent>\z@
1036     \if@english \parindent=1em
1037     \else      \parindent=1\jsZw
1038   \fi
1039   \fi\relax
1040   \jsResetDimen}

```

`\jsResetDimen` ユーザ定義用のフック。

```

1041 \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。

```

1042 \ifjsc@mag
1043   \let\jsc@setfontsize\@setfontsize
1044 \else
1045   \def\jsc@setfontsize#1#2#3{%
1046     \@setfontsize#1{#2\jsc@mpt}{#3\jsc@mpt}}
1047 % microtype 対策
1048 \ifjsWithTeX\if j\jsEngine\else
1049   \def\jsc@setfontsize#1#2#3{%
1050     \edef\bxjs@sfs@next{%
1051       \unexpanded{\@setfontsize#1}%
1052       {\the\dimexpr#2\jsc@mpt\relax}{\the\dimexpr#3\jsc@mpt\relax}%
1053     }\bxjs@sfs@next}
1054   \fi\fi
1055 \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。

```

1056 \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`

`\widebaselines`

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

```
1057 \newif\ifnarrowbaselines
1058 \if@english
1059   \narrowbaselinestru
1060 \fi
1061 \def\narrowbaselines{%
1062   \narrowbaselinestru
1063   \skip0=\abovedisplayskip
1064   \skip2=\abovedisplayshortskip
1065   \skip4=\belowdisplayskip
1066   \skip6=\belowdisplayshortskip
1067 % 一時的に警告を無効化する
1068   \let\bxjs@save@nomath\@nomath
1069   \let\@nomath\@gobble
1070   \@currsize\selectfont
1071   \let\@nomath\bxjs@save@nomath
1072   \abovedisplayskip=\skip0
1073   \abovedisplayshortskip=\skip2
1074   \belowdisplayskip=\skip4
1075   \belowdisplayshortskip=\skip6\relax}
1076 \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。

```
1077 \def\bxjs@ifnarrowbaselines{%
1078   \ifnarrowbaselines\expandafter\@firstoftwo
1079   \else \expandafter\@secondoftwo
1080   \fi
1081 }
```

`\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 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。

```
1082 \renewcommand{\normalsize}{%
1083   \bxjs@if@narrowbaselines{%
1084     \jsc@setfontsize\normalsize\@xpt\@xiipt
1085   }{%else
1086     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1087   }%
```

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

[2003-02-16] ちょっと変えました。

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

```
1088 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1089 \abovedisplayshortskip \z@ \@plus3\p@?
1090 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1091 \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。

```
1092 \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。

```
1093 %</class>
1094 %<*class|minijs>
1095 %% initialize
1096 \normalsize
1097 %</class|minijs>
1098 %<*class>
```

`\Cht` 基準となる長さの設定をします。p_LA_TE_X 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 (`lzw`) です。

`\Cwd` [2017-08-31] 基準とする文字を「全角空白」（EUC コード 0xA1A1）から「漢」（JIS コード 0x3441）へ変更しました。

`\Chs`

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

```
1099 \ifx\Cht\@undefined \newdimen\Cht \fi
1100 \ifx\Cdp\@undefined \newdimen\Cdp \fi
```

```

1101 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1102 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1103 \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。

```

1104 \setlength\Cht{0.88\jsZw}
1105 \setlength\Cdp{0.12\jsZw}
1106 \setlength\Cwd{1\jsZw}
1107 \setlength\Cvs{\baselineskip}
1108 \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。

```

1109 \newcommand{\small}{%
1110   \bxjs@if@narrowbaselines{%
1111     <!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1112     <kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1113   }{%else
1114     <!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1115     <kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1116   }%
1117   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1118   \abovedisplayshortskip \z@ \@plus3\p@?
1119   \belowdisplayskip \abovedisplayskip
1120   \belowdisplayshortskip \belowdisplayskip
1121   \def\@listi{\leftmargin\leftmargini
1122             \topsep \z@
1123             \parsep \z@
1124             \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。

```

1125 \newcommand{\footnotesize}{%
1126   \bxjs@if@narrowbaselines{%
1127     <!kiyou>   \jsc@setfontsize\footnotesize\@viiipt{9.5}%
1128     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1129   }{%else
1130     <!kiyou>   \jsc@setfontsize\footnotesize\@viiipt{11}%
1131     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1132   }%

```

```

1133 \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1134 \abovedisplayskip \z@ \@plus2\p@?
1135 \belowdisplayskip \abovedisplayskip
1136 \belowdisplayskip \belowdisplayskip
1137 \def\@listi{\leftmargin\leftmargini
1138         \topsep \z@
1139         \parsep \z@
1140         \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` 行が揃うようになります。

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1141 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1142 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 1143 \if@twocolumn
\HUGE 1144 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
1145 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1146 \else
1147 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
1148 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1149 \fi
1150 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1151 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1152 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1153 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1154 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1155 \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。

```

1156 \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
あればお教えてください。

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

```

1157 % \newcommand{\headfont}{\bfseries}

```

```

1158 \newcommand{\headfont}{\sffamily}
1159 % \newcommand{\headfont}{\sffamily\fontseries{sbcs}\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。

```

1160 %<kiyou>\setlength\columnsep{2\Cwd}
1161 %<kiyou>\setlength\columnsep{28truebp}
1162 \setlength\columnseprule{\z@}

```

■段落

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

```

\lineskiplimit 1163 \setlength\lineskip{1\jsc@empt}
1164 \setlength\normallineskip{1\jsc@empt}
\normallineskiplimit 1165 \setlength\lineskiplimit{1\jsc@empt}
1166 \setlength\normallineskiplimit{1\jsc@empt}

```

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

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

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

```
1167 \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。

```

1168 \setlength\parskip{\z@}
1169 \if@slide
1170 \setlength\parindent{0\p@}
1171 \else
1172 \setlength\parindent{1\Cwd}
1173 \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 1174 \@lowpenalty 51
1175 \@medpenalty 151
1176 \@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。

```
1177 % \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。

```
1178 % \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。

```
1179 \@onlypreamble\bxjs@bd@pre@geometry@hook
1180 \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。

```
1181 \ifjsc@mag
1182 \mag=\bxjs@param@mag
1183 \fi
1184 \setlength{\topskip}{10\p0?}
```

`\jsSetQHLlength` のための和文単位の定義。

```
1185 \def\bxjs@unit@trueQ{0.25truemm}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1186 \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。

```
1187 \@tempwafalse
1188 \def\bxjs@tmpdo{\@ifnextchar\bgroup\bxjs@tmpdo@a\remove@to@nnil}
1189 \def\bxjs@tmpdo@a#1{\edef\bxjs@tmpa{#1}%
1190   \@ifnextchar\bgroup\bxjs@tmpdo@b\remove@to@nnil}
1191 \def\bxjs@tmpdo@b#1{\edef\bxjs@tmpa{\bxjs@tmpa,#1}%
1192   \@ifnextchar\@nnil\bxjs@tmpdo@c\remove@to@nnil}
1193 \def\bxjs@tmpdo@c\@nnil{\@tempwatrue
1194   \edef\bxjs@param@paper{papersize={\bxjs@tmpa}}}
1195 \expandafter\bxjs@tmpdo\bxjs@param@paper\@nnil
```

`W,H` の形式について。

```
1196 \if@tempwa\else
1197   \def\bxjs@tmpa{\@nil,\@nil}
1198   \def\bxjs@tmpdo#1,#2,#3\@nnil{%
1199     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1200       \@tempwatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1201   \expandafter\bxjs@tmpdo\bxjs@param@paper,\@nil,\@nil\@nnil
1202 \fi
```

W*H の形式について。

```
1203 \if@tempswa\else
1204   \def\bxjs@tmpa{\@nil*\@nil}
1205   \def\bxjs@tmpdo#1*#2*#3\@nnil{%
1206     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1207       \@tempswatrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1208   \expandafter\bxjs@tmpdo\bxjs@param@paper*\@nil*\@nil\@nnil
1209 \fi
```

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

```
1210 \edef\bxjs@layout@paper{%
1211   \ifjsc@mag truedimen,\fi
1212   \if@landscape landscape,\fi
1213   \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。

```
1214 %<*article|report>
1215 \def\bxjs@layout@base{%
1216   headheight=\topskip,footskip=0.03367\paperheight,%
1217   headsep=\footskip-\topskip,includeheadfoot,%
1218 }
1219 \edef\bxjs@layout{\bxjs@layout@base
1220   hscale=0.76,hmarginratio=1:1,%
1221   vscale=0.83,vmarginratio=1:1,%
1222 }
1223 %</article|report>
1224 %<*book>
1225 \def\bxjs@layout@base{%
1226   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1227 }
1228 \ifbxjs@layout@buggyhmargin    %---
1229 % アレ
1230 \edef\bxjs@layout{\bxjs@layout@base
1231   hmargin=36\jsc@mmm,hmarginratio=1:1,%
1232   vscale=0.83,vmarginratio=1:1,%
1233 }
1234 \else    %---
1235 % 非アレ
1236 \edef\bxjs@layout{\bxjs@layout@base
1237   hmargin=18\jsc@mmm,%
1238   vscale=0.83,vmarginratio=1:1,%
1239 }
1240 \fi    %---
1241 %</book>
1242 %<*slide>
1243 \def\bxjs@layout@base{%
1244   noheadfoot,%
1245 }
1246 \edef\bxjs@layout{\bxjs@layout@base
```

```

1247 hscale=0.9,hmarginratio=1:1,%
1248 vscale=0.944,vmarginratio=1:1,%
1249 }
1250 %</slide>

textwidth オプションの設定を反映する。
1251 %<!*book>
1252 \ifx\bxjs@textwidth@opt\@undefined\else
1253 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1254 \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1255 \fi
1256 %</!*book>
1257 \ifx\bxjs@number@of@lines@opt\@undefined\else
1258 \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1259 \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1260 \fi

```

`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1261 \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。

```

1262 %<!*book>
1263 \newcommand\jsTextWidthLimit{40}
1264 \@tempdima=\jsTextWidthLimit\Cwd
1265 \ifx\bxjs@textwidth@limit@opt\@undefined\else
1266 \bxjs@gset@tempcnta{\bxjs@textwidth@limit@opt}
1267 \@tempdima=\@tempcnta\Cwd
1268 \fi
1269 \ifx\bxjs@textwidth@opt\@undefined\else
1270 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1271 \fi
1272 \edef\bxjs@textwidth@limit{\the\@tempdima}
1273 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1274 \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1275 \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1276 \fi
1277 %</!*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。

```

1278 \def\bxjs@preproc@layout{%
1279 \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@empt}

```

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

```
1280 \def\bxjs@postproc@layout{%
```

geometry のドライバを再設定する。

```
1281 \ifx\bxjs@geometry@driver\relax\else
1282   \let\Gm@driver\bxjs@geometry@driver
1283 \fi
```

\ht\strutbox の値を元に戻す。

```
1284 \ht\strutbox=\bxjs@save@ht@strutbox\relax
```

\textwidth の値を補正する。

```
1285 \ifbxjs@whole@zw@lines
1286   \@tempdimb=\textwidth
1287   \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1288   \advance\textwidth.005pt\relax
1289   \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1290   \advance\@tempdimb-\textwidth
1291   \advance\oddsidemargin 0.5\@tempdimb
1292   \advance\evensidemargin 0.5\@tempdimb
1293 \fi
1294 \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。

```
1295 %<*book>
1296 \@tempdima=\bxjs@textwidth@limit\relax
1297 \ifbxjs@whole@zw@lines
1298   \advance\@tempdima.005pt\relax
1299   \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1300 \fi
1301 \ifdim\textwidth>\@tempdima
1302   \textwidth=\@tempdima
1303   \addtolength\evensidemargin{\fullwidth-\textwidth}
1304 \fi
1305 %</book>
```

\textheight 関連の調整。

```
1306 \@tempdimb=\textheight
1307 \advance\textheight-\topskip
1308 \advance\textheight.005pt\relax
1309 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1310 \advance\textheight\topskip
1311 \advance\@tempdimb-\textheight
1312 \advance\topmargin0.5\@tempdimb
```

\headheight 関連の調整。

```
1313 \@tempdima=\topskip
1314 \advance\headheight\@tempdima
1315 \advance\topmargin-\@tempdima
```

marginpar 関連の調整。

```
1316 \setlength\marginparsep{\columnsep}
```

```

1317 \setlength\marginparpush{\baselineskip}
1318 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1319   -\textwidth-10\jsc@mmm-\marginparsep}
1320 \ifbxjs@whole@zw@lines
1321   \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1322 \fi

```

連動する変数。

```

1323 \maxdepth=.5\topskip
1324 \stockwidth=\paperwidth
1325 \stockheight=\paperheight
1326 }

```

`\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。

```

1327 \edef\jsGeometryOptions{%
1328   \bxjs@layout@paper,\bxjs@layout}

```

■geometry パッケージを読み込む

ムニャムニャ。

```

1329 \def\bxjs@geometry@name{geometry}
1330 \ifbxjs@old@hook@system
1331   \let\bxjs@apply@bd@pre@geometry@hook\AtBeginDocument
1332 \else
1333   \def\bxjs@apply@bd@pre@geometry@hook{%
1334     \AddToHook{begindocument}{\bxjs@geometry@name}}
1335 \fi

```

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

```

1336 \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。

```

1337 \ifbxjs@papersize
1338   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1339     \PassOptionsToPackage{dvipdfm}{geometry}
1340   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1341     \PassOptionsToPackage{dvipdfm}{geometry}
1342   \fi\fi
1343   \let\bxPapersizeSpecialDone=t
1344 \else
1345   \PassOptionsToPackage{driver=none}{geometry}
1346 \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。

```
1347 \bxjs@apply@bd@pre@geometry@hook{\bxjs@bd@pre@geometry@hook}
1348 \bxjs@apply@bd@pre@geometry@hook{\ImposeOldLuaTeXBehavior}
1349 \bxjs@preproc@layout
1350 \edef\bxjs@next{%
1351   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1352 }\bxjs@next
1353 \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。

```
1354 \let\bxjs@geometry@driver\Gm@driver
1355 \bxjs@postproc@layout
```

geometry のドライバ自動判別に対する前処理。

```
1356 \g@addto@macro\bxjs@bd@pre@geometry@hook{%
  BXS の 2.0 版より、geometry の 4.x 版のサポートは廃止された。
1357   \ifpackagelater{geometry}{2010/02/12}{\else
1358     \PackageError\bxjs@clsname
1359       {Your 'geometry' package is too old (< v5.0)}%
1360       {\@ehc}%
1361     \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。

```
1362   \ifjsWithpTeXng
1363     \ifx\Gm@driver\@empty
1364       \def\Gm@driver{pdftex}%
1365     \fi
1366   \fi}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```
1367 \def\setpagelayout{%
1368   \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{\else
1369     \@ifstar{\bxjs@setpagelayout@a@one}{\bxjs@setpagelayout@a@z@}}
1370 \def\bxjs@setpagelayout@a#1#2{%
1371   \ifcase#1% modify
1372     \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1373   \or% reset(*)
1374     \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1375   \or% semireset(+)
1376     \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1377   \fi
1378   \bxjs@preproc@layout
1379   \edef\bxjs@next{%
1380     \noexpand\geometry{\bxjs@next}%
1381   }\bxjs@next
```

1382 \bxjs@postproc@layout}

■ geometry パッケージを読み込まない ☹

geometry=user の場合の処理。

1383 \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。

1384 \g@addto@macro\bxjs@begin@document@hook{%

1385 \ifdim\textwidth=.5\maxdimen

1386 \ClassError\bxjs@clsname

1387 {Page layout is not properly set}%

1388 {\@ehd}%

1389 \fi}

1390 \def\jsUseMinimalPageLayout{%

1391 \setlength{\textwidth}{6.5in}%

1392 \setlength{\textheight}{8in}}

\setpagelayout はとりあえず無効にしておく。

1393 \let\bxjs@geometry@driver\relax

1394 \def\setpagelayout{%

1395 \bxjs@ifplus{\bxjs@pagelayout@a}{%else

1396 \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}

1397 \def\bxjs@pagelayout@a#1{%

1398 \ClassError\bxjs@clsname

1399 {Command '\string\setpagelayout' is not supported,\MessageBreak

1400 because 'geometry' value is not 'class'}\@eha}

1401 %

1402 \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。

1403 %<*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。

```
1404 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1405 \if@slide
1406   \setlength\headheight{0\jsc@mpt}
1407 \else
1408   \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
      06-26)
1409 \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。

```
1410 %<*article|kiyou>
1411 \if@slide
1412   \setlength\footskip{0pt}
1413 \else
1414   \setlength\footskip{0.03367\paperheight}
1415   \ifdim\footskip<\baselineskip
1416     \setlength\footskip{\baselineskip}
1417   \fi
1418 \fi
1419 %</article|kiyou>
1420 %<jspf>\setlength\footskip{9\jsc@mmm}
1421 %<*book>
1422 \if@report
1423   \setlength\footskip{0.03367\paperheight}
1424   \ifdim\footskip<\baselineskip
1425     \setlength\footskip{\baselineskip}
1426   \fi
1427 \else
1428   \setlength\footskip{0pt}
1429 \fi
1430 %</book>
1431 %<*report>
1432 \setlength\footskip{0.03367\paperheight}
1433 \ifdim\footskip<\baselineskip
1434   \setlength\footskip{\baselineskip}
```



```
1435 \fi
1436 %</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)。

```
1437 %<*article>
1438 \if@slide
1439   \setlength\headsep{0\jsc@mpt}
1440   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1441   \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1442 \else
1443   \setlength\headsep{\footskip}
1444   \addtolength\headsep{-\topskip}
1445 \fi
1446 %</article>
1447 %<*book>
1448 \if@report
1449   \setlength\headsep{\footskip}
1450   \addtolength\headsep{-\topskip}
1451 \else
1452   \setlength\headsep{6\jsc@mmm}
1453   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1454   \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1455 \fi
1456 %</book>
1457 %<*report>
1458 \setlength\headsep{\footskip}
1459 \addtolength\headsep{-\topskip}
1460 %</report>
1461 %<*jspf>
1462 \setlength\headsep{9\jsc@mmm}
1463 \addtolength\headsep{-\topskip}
1464 %</jspf>
1465 %<*kiyou>
1466 \setlength\headheight{0\jsc@mpt}
1467 \setlength\headsep{0\jsc@mpt}
1468 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1469 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1470 %</kiyou>
```

`\maxdepth` `\maxdepth` は本文最下行の最大の深さで, plain $\text{T}_\text{E}\text{X}$ や $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ 2.09 では 4pt に固定でした。 $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}2\text{e}$ 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。

```
1471 \setlength\maxdepth{.5\topskip}
```

■本文の幅と高さ

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

```
1472 \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。

```
1473 %<*article>
1474 \if@slide
1475   \setlength\fullwidth{0.9\paperwidth}
1476 \else
1477   \setlength\fullwidth{0.76\paperwidth}
1478 \fi
1479 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1480 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1481 \setlength\textwidth{\fullwidth}
1482 %</article>
1483 %<*book>
1484 \if@report
1485   \setlength\fullwidth{0.76\paperwidth}
1486 \else
1487   \setlength\fullwidth{\paperwidth}
1488   \addtolength\fullwidth{-36\jsc@mmm}
1489 \fi
1490 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1491 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1492 \setlength\textwidth{\fullwidth}
1493 \if@report \else
1494   \if@twocolumn \else
1495     \ifdim \fullwidth>40zw
1496       \setlength\textwidth{40zw}
1497     \fi
1498   \fi
1499 \fi
1500 %</book>
1501 %<*report>
```

```

1502 \setlength\fullwidth{0.76\paperwidth}
1503 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1504 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1505 \setlength\textwidth{\fullwidth}
1506 %</report>
1507 %<*jspf>
1508 \setlength\fullwidth{50zw}
1509 \addtolength\fullwidth{8\jsc@mmm}
1510 \setlength\textwidth{\fullwidth}
1511 %</jspf>
1512 %<*kiyou>
1513 \setlength\fullwidth{48zw}
1514 \addtolength\fullwidth{\columnsep}
1515 \setlength\textwidth{\fullwidth}
1516 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

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

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

```

1517 %<*article|book|report>
1518 \if@slide
1519 \setlength{\textheight}{0.95\paperheight}
1520 \else
1521 \setlength{\textheight}{0.83\paperheight}
1522 \fi
1523 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1524 \addtolength{\textheight}{-\headsep}
1525 \addtolength{\textheight}{-\footskip}
1526 \addtolength{\textheight}{-\topskip}
1527 \divide\textheight\baselineskip
1528 \multiply\textheight\baselineskip
1529 %</article|book|report>
1530 %<jspf>\setlength{\textheight}{51\baselineskip}
1531 %<kiyou>\setlength{\textheight}{47\baselineskip}

```

```

1532 \addtolength{\textheight}{\topskip}
1533 \addtolength{\textheight}{0.1\jsc@mpt}
1534 %<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。

```

1535 \def\flushbottom{%
1536 \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1537 \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。

```

1538 \setlength\marginparsep{\columnsep}
1539 \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` 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。

```

1540 \setlength{\oddsidemargin}{\paperwidth}
1541 \addtolength{\oddsidemargin}{-\fullwidth}
1542 \setlength{\oddsidemargin}{.5\oddsidemargin}
1543 \iftombow
1544 \addtolength{\oddsidemargin}{-1in}
1545 \else
1546 \addtolength{\oddsidemargin}{-\inv@mag in}
1547 \fi
1548 \setlength{\evensidemargin}{\oddsidemargin}
1549 \if@mparswitch
1550 \addtolength{\evensidemargin}{\fullwidth}
1551 \addtolength{\evensidemargin}{-\textwidth}
1552 \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。

```

1553 \setlength\marginparwidth{\paperwidth}
1554 \addtolength\marginparwidth{-\oddsidemargin}
1555 \addtolength\marginparwidth{-\inv@mag in}
1556 \addtolength\marginparwidth{-\textwidth}
1557 \addtolength\marginparwidth{-10\jsc@mmm}
1558 \addtolength\marginparwidth{-\marginparsep}

```

```

1559 \@tempdima=1zw
1560 \divide\marginparwidth\@tempdima
1561 \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）。

```

1562 \setlength\topmargin{\paperheight}
1563 \addtolength\topmargin{-\textheight}
1564 \if@slide
1565   \addtolength\topmargin{-\headheight}
1566 \else
1567   \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
     \headheight (2003-06-26)
1568 \fi
1569 \addtolength\topmargin{-\headsep}
1570 \addtolength\topmargin{-\footskip}
1571 \setlength\topmargin{0.5\topmargin}
1572 %<kiyou>\setlength\topmargin{81truebp}
1573 \iftombow
1574   \addtolength\topmargin{-1in}
1575 \else
1576   \addtolength\topmargin{-\inv@mag in}
1577 \fi
1578 %</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。

```
1579 \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。

1580 `\setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}`

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

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

[2003-08-23] ちょっと増やしました。

1581 `\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。

1582 `\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。

1583 `\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。

1584 `\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。

1585 `\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。

1586 `\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。

1587 `\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。

1588 `\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。

1589 `\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。

1590 `\renewcommand{\dblfloatpagefraction}{.8}`

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

```
1591 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1592 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1593 \setlength\intertextsep {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 1594 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1595 \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 1596 \setlength\@fptop{0\p@? \@plus 1fil}
1597 \setlength\@fpsep{8\p@? \@plus 2fil}
1598 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpptop` 段抜きフロートについての値です。

```
\@dblfpsep 1599 \setlength\@dblfpptop{0\p@? \@plus 1fil}
\@dblfpbot 1600 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1601 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ (日本語 TeX 開発コミュニティ版のみ)

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

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

となっています。

```
1602 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1603 % \ifodd\c@page
1604 % \iftdir
1605 % \hbox{}\thispagestyle{empty}\newpage
1606 % \if@twocolumn\hbox{}\newpage\fi
1607 % \fi
1608 % \else
1609 % \ifydir
1610 % \hbox{}\thispagestyle{empty}\newpage
1611 % \if@twocolumn\hbox{}\newpage\fi
1612 % \fi
1613 % \fi\fi}
1614 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
```

```

1615 % \ifodd\c@page
1616 %   \ifdir
1617 %     \hbox{}\thispagestyle{empty}\newpage
1618 %     \if@twocolumn\hbox{}\newpage\fi
1619 %   \fi
1620 % \else
1621 %   \iftdir
1622 %     \hbox{}\thispagestyle{empty}\newpage
1623 %     \if@twocolumn\hbox{}\newpage\fi
1624 %   \fi
1625 % \fi\fi}
1626 \def\pltx@cleartooddpage{\clearpage\if@twoside
1627 \ifodd\c@page\else
1628   \hbox{}\thispagestyle{empty}\newpage
1629   \if@twocolumn\hbox{}\newpage\fi
1630 \fi\fi}
1631 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1632 \ifodd\c@page
1633   \hbox{}\thispagestyle{empty}\newpage
1634   \if@twocolumn\hbox{}\newpage\fi
1635 \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。

```

1636 \let\pltx@cleartorightpage\pltx@cleartooddpage
1637 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

\vsizer の値がアレな場合は本体開始まで \clearpage を無効にする。

1638 \ifdim\vsizer=\z@
1639 \begingroup
1640 \toks@\expandafter{\clearpage}
1641 \xdef\clearpage{\noexpand\ifbxjs@after@preamble\the\toks@\noexpand\fi}
1642 \endgroup
1643 \fi

```

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

```

1644 %<*book|report>
1645 \if@openleft
1646   \let\cleardoublepage\pltx@cleartoleftpage
1647 \else\if@openright
1648   \let\cleardoublepage\pltx@cleartorightpage
1649 \fi\fi
1650 %</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。

```
1651 % \def\ps@empty{%
1652 %   \let\@mkboth\@gobbletwo
1653 %   \let\@oddhead\@empty
1654 %   \let\@oddfoot\@empty
1655 %   \let\@evenhead\@empty
1656 %   \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。

```
1657 \def\ps@plainfoot{%
1658   \let\@mkboth\@gobbletwo
1659   \let\@oddhead\@empty
1660   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1661   \let\@evenhead\@empty
```

```

1662 \let\@evenfoot\@oddfoot}
1663 \def\ps@plainhead{%
1664 \let\@mkboth\@gobbletwo
1665 \let\@oddfoot\@empty
1666 \let\@evenfoot\@empty
1667 \def\@evenhead{%
1668 \if@mparswitch \hss \fi
1669 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1670 \if@mparswitch\else \hss \fi}%
1671 \def\@oddhead{%
1672 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1673 %<book>\let\ps@plain\ps@plainhead
1674 %<!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。

```

1675 %<*article|slide>
1676 \if@twoside
1677 \def\ps@headings{%
1678 \let\@oddfoot\@empty
1679 \let\@evenfoot\@empty
1680 \def\@evenhead{\if@mparswitch \hss \fi
1681 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1682 \if@mparswitch\else \hss \fi}%
1683 \def\@oddhead{%
1684 \underline{%
1685 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1686 \let\@mkboth\markboth
1687 \def\sectionmark##1{\markboth{%
1688 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1689 ##1}}}%
1690 \def\subsectionmark##1{\markright{%
1691 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1692 ##1}}}%
1693 }
1694 \else % if not twoside
1695 \def\ps@headings{%
1696 \let\@oddfoot\@empty
1697 \def\@oddhead{%
1698 \underline{%
1699 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1700 \let\@mkboth\markboth
1701 \def\sectionmark##1{\markright{%
1702 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1703 ##1}}}%
1704 \fi
1705 %</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。

```
1706 %<*book|report>
1707 \def\bxjs@maybe@autoxspacing{%
1708   \ifx\autoxspacing\undefined\else \autoxspacing \fi}
1709 \newif\ifomit@number
1710 \def\ps@headings{%
1711   \let\@oddfont\empty
1712   \let\@evenfont\empty
1713   \def\@evenhead{%
1714     \ifmparswitch \hss \fi
1715     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1716       \textbf{\thepage}\hfil\leftmark}}}%
1717     \ifmparswitch\else \hss \fi}%
1718   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1719     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1720   \let\@mkboth\markboth
1721   \def\chaptermark##1{\markboth{%
1722     \ifnum \c@secnumdepth >\m@ne
1723       \ifmainmatter
1724         \ifomit@number\else
1725           \@chapapp\thechapter\@chappos\hskip1\jsZw
1726         \fi
1727       \fi
1728     \fi
1729     ##1}{}}}%
1730   \def\sectionmark##1{\markright{%
1731     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1732     ##1}}}%
1733 %</book|report>
```

最後は学会誌の場合です。

```
1734 %<*jspf>
1735 \def\ps@headings{%
1736   \def\@oddfont{\normalfont\hfil\thepage\hfil}
1737   \def\@evenfont{\normalfont\hfil\thepage\hfil}
1738   \def\@oddhead{\normalfont\hfil \@title \hfil}
1739   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1740 %</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。

```

1741 \def\ps@myheadings{%
1742   \let\@oddfont\@empty\let\@evenfont\@empty
1743   \def\@evenhead{%
1744     \if@mparswitch \hss \fi%
1745     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1746     \if@mparswitch\else \hss \fi}%
1747   \def\@oddhead{%
1748     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1749   \let\@mkboth\@gobbletwo
1750 %<book|report> \let\chaptermark\@gobble
1751 \let\sectionmark\@gobble
1752 %<!book&!report> \let\subsectionmark\@gobble
1753 }

```

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 1754 % \newcommand*\title}[1]{\gdef\@title{#1}}
1755 % \newcommand*\author}[1]{\gdef\@author{#1}}
1756 % \newcommand*\date}[1]{\gdef\@date{#1}}
1757 % \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。

```

1758 \newcommand*\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1759 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。
1760 \renewcommand*\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1761 \g@addto@macro\bxjs@begin@document@hook{\bxjs@decl@subtitle}
1762 \def\bxjs@decl@subtitle{%
1763   \global\let\bxjs@decl@subtitle\relax
1764   \ifx\subtitle\@undefined
1765     \global\let\subtitle\jsSubtitle
1766   \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。

```

1767 \def\bxjs@annihilate@subtitle{%

```

```

1768 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1769 \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 1770 %<*\jspf>
\keywords 1771 \newcommand*\etitle}[1]{\gdef\@etitle{#1}}
1772 \newcommand*\author}[1]{\gdef\@author{#1}}
1773 \newcommand*\keywords}[1]{\gdef\@keywords{#1}}
1774 \newcommand*\email}[1]{\gdef\authors@mail{#1}}
1775 \newcommand*\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1776 %</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。

```

1777 \def\plainifnotempty{%
1778 \ifx \@oddhead \@empty
1779 \ifx \@oddfoot \@empty
1780 \else
1781 \thispagestyle{plainfoot}%
1782 \fi
1783 \else
1784 \thispagestyle{plainhead}%
1785 \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。

```

1786 %<*\article|book|report|slide>
1787 \if@titlepage
1788 \newcommand{\maketitle}{%
1789 \begin{titlepage}%
1790 \let\footnotesize\small
1791 \let\footnoterule\relax
1792 \let\footnote\thanks
1793 \null\vfil
1794 \if@slide
1795 {\footnotesize \@date}%
1796 \begin{center}
1797 \mbox{} \\\[1\jsZw]
1798 \large

```

```

1799         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1800         \jsc@smallskip
1801         \@title
1802         \ifx\bxjs@subtitle\@undefined\else
1803             \par\vskip\z@
1804             {\small \bxjs@subtitle\par}
1805         \fi
1806         \jsc@smallskip
1807         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1808         \vfill
1809         {\small \@author}%
1810     \end{center}
1811 \else
1812 \vskip 60\p@?
1813 \begin{center}%
1814     {\LARGE \@title \par}%
1815     \ifx\bxjs@subtitle\@undefined\else
1816         \vskip5\p@?
1817         {\normalsize \bxjs@subtitle\par}
1818     \fi
1819     \vskip 3em%
1820     {\large
1821         \lineskip .75em
1822         \begin{tabular}[t]{c}%
1823             \@author
1824         \end{tabular}\par}%
1825     \vskip 1.5em
1826     {\large \@date \par}%
1827 \end{center}%
1828 \fi
1829 \par
1830 \@thanks\vfil\null
1831 \end{titlepage}%
1832 \setcounter{footnote}{0}%
1833 \global\let\thanks\relax
1834 \global\let\maketitle\relax
1835 \global\let\@thanks\@empty
1836 \global\let\@author\@empty
1837 \global\let\@date\@empty
1838 \global\let\@title\@empty
1839 \global\let\title\relax
1840 \global\let\author\relax
1841 \global\let\date\relax
1842 \global\let\and\relax
1843 \bxjs@annihilate@subtitle
1844 }%
1845 \else
1846 \newcommand{\maketitle}{\par
1847 \begin{group

```

```

1848 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1849 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1850 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1851 \parindent 1\jsZw\noindent
1852 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1853 \if@twocolumn
1854 \ifnum \col@number=\@ne
1855 \maketitle
1856 \else
1857 \twocolumn[\maketitle]%
1858 \fi
1859 \else
1860 \newpage
1861 \global\@topnum\z@ % Prevents figures from going at top of page.
1862 \maketitle
1863 \fi
1864 \plainifnotempty
1865 \@thanks
1866 \endgroup
1867 \setcounter{footnote}{0}%
1868 \global\let\thanks\relax
1869 \global\let\maketitle\relax
1870 \global\let\@thanks\@empty
1871 \global\let\@author\@empty
1872 \global\let\@date\@empty
1873 \global\let\@title\@empty
1874 \global\let\title\relax
1875 \global\let\author\relax
1876 \global\let\date\relax
1877 \global\let\and\relax
1878 \bxjs@annihilate@subtitle
1879 }

```

\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。

```

1880 \def\maketitle{%
1881 \newpage\null
1882 \vskip 2em
1883 \begin{center}%
1884 \let\footnote\thanks
1885 {\LARGE \@title \par}%
1886 \ifx\bxjs@subtitle\undefined\else
1887 \vskip3\p@?
1888 {\normalsize \bxjs@subtitle\par}
1889 \fi
1890 \vskip 1.5em
1891 {\large
1892 \lineskip .5em
1893 \begin{tabular}[t]{c}%
1894 \author

```

```

1895     \end{tabular}\par}%
1896     \vskip 1em
1897     {\large \@date}%
1898     \end{center}%
1899     \par\vskip 1.5em
1900 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1901 }
1902 \fi
1903 %</article|book|report|slide>
1904 %<*jspf>
1905 \newcommand{\maketitle}{\par
1906   \begin{group}
1907     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1908     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1909     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1910       \parindent 1\jsZw\noindent
1911       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1912     \twocolumn[\@maketitle]%
1913     \plainifnotempty
1914     \@thanks
1915   \end{group}
1916   \setcounter{footnote}{0}%
1917   \global\let\thanks\relax
1918   \global\let\maketitle\relax
1919   \global\let\@thanks\@empty
1920   \global\let\@author\@empty
1921   \global\let\@date\@empty
1922 % \global\let\@title\@empty % \@title は柱に使う
1923   \global\let\title\relax
1924   \global\let\author\relax
1925   \global\let\date\relax
1926   \global\let\and\relax
1927   \ifx\authors@mail\@undefined\else{%
1928     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1929     \footnotetext[0]{\itshape\authors@mail}%
1930   }\fi
1931   \global\let\authors@mail\@undefined}
1932 \def\@maketitle{%
1933   \newpage\null
1934   \vskip 6em % used to be 2em
1935   \begin{center}
1936     \let\footnote\thanks
1937     \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1938     \lineskip .5em
1939     \ifx\@author\@undefined\else
1940       \vskip 1em
1941       \begin{tabular}[t]{c}%
1942         \@author
1943       \end{tabular}\par

```



```

1944 \fi
1945 \ifx\@etitle\@undefined\else
1946 \vskip 1em
1947 {\large \@etitle \par}%
1948 \fi
1949 \ifx\@eauthor\@undefined\else
1950 \vskip 1em
1951 \begin{tabular}[t]{c}%
1952 \@eauthor
1953 \end{tabular}\par
1954 \fi
1955 \vskip 1em
1956 \@date
1957 \end{center}
1958 \vskip 1.5em
1959 \centerline{\box\@abstractbox}
1960 \ifx\@keywords\@undefined\else
1961 \vskip 1.5em
1962 \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
1963 \fi
1964 \vskip 1.5em}
1965 %</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。

```

1966 \def\bxjs@label@sect#1{%
1967 \expandafter\ifx\csname label#1\endcsname\relax
1968 \csname the#1\endcsname
1969 \else \csname label#1\endcsname
1970 \fi}
1971 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1972 \ifnum\bxjs@label@section=\bxjs@label@section@@compat\else
1973 \def\@secapp{\presectionname}
1974 \def\@secpos{\postsectionname}
1975 \fi

```

`\labelsection` 節番号の出力書式。

```

1976 \ifnum\bxjs@label@section=\bxjs@label@section@@modern
1977 \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、見出しの下の空きです。負の場合は、絶対値が見出しの右の空きです (見出しと同じ行から本文を始めます)。

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```
1979 \def\startsection#1#2#3#4#5#6{%
1980   \if@noskipsec \leavevmode \fi
1981   \par
1982 % 見出し上の空きを \@tempskipa にセットする
1983   \@tempskipa #4\relax
1984 % \@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
1985   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1986 % 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
1987   \ifdim \@tempskipa <\z@
1988     \@tempskipa -\@tempskipa \@afterindentfalse
1989   \fi
1990   \if@nobreak
1991 %   \everypar{\everyparhook}% これは間違い
1992     \everypar{}%
1993   \else
1994     \addpenalty\@secpenalty
1995 % 次の行は削除
```

```

1996 % \addvspace\@tempskipa
1997 % 次の \noindent まで追加
1998 \ifdim \@tempskipa >\z@
1999 \if@slide\else
2000 \null
2001 \vspace*{-\baselineskip}%
2002 \fi
2003 \vskip\@tempskipa
2004 \fi
2005 \fi
2006 \noindent
2007 % 追加終わり
2008 \@ifstar
2009 {\@ssect{#3}{#4}{#5}{#6}}%
2010 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}

```

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

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

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

```

2011 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
2012 \let\bxjs@if@ceph\@firstofone
2013 \else \let\bxjs@if@ceph\@gobble
2014 \fi

```

```

2015 \def\@sect#1#2#3#4#5#6[#7]#8{%
2016 \ifnum #2>\c@secnumdepth
2017 \let\@svsec\@empty
2018 \else
2019 \refstepcounter{#1}%
2020 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2021 \fi
2022 % 見出し後の空きを \@tempskipa にセット
2023 \@tempskipa #5\relax
2024 % 条件判断の順序を入れ替えました
2025 \ifdim \@tempskipa<\z@
2026 \def\@svsechd{%
2027 #6{\hskip #3\relax
2028 \@svsec #8}%
2029 \c@name #1mark\endc@name{#7}%
2030 \addcontentsline{toc}{#1}{%
2031 \ifnum #2>\c@secnumdepth \else
2032 \protect\numberline{\bxjs@label@sect{#1}}%
2033 \fi
2034 #7}}% 目次にフルネームを載せるなら #8
2035 \else

```

```

2036 \begingroup
2037 \interlinepenalty \@M % 下から移動
2038 #6{%
2039 \hangfrom{\hskip #3\relax\@svsec}%
2040 % \interlinepenalty \@M % 上に移動
2041 #8\@par}%
2042 \endgroup
2043 \csname #1mark\endcsname{#7}%
2044 \addcontentsline{toc}{#1}{%
2045 \ifnum #2>\c@secnumdepth \else
2046 \protect\numberline{\bxjs@label@sect{#1}}%
2047 \fi
2048 #7}% 目次にフルネームを載せるならここは #8
2049 \fi
2050 \@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。

```

2051 \def\@xsect#1{%
2052 % 見出しの後ろの空きを \@tempskipa にセット
2053 \@tempskipa #1\relax
2054 % 条件判断の順序を変えました
2055 \ifdim \@tempskipa<\z@
2056 \@nbreakfalse
2057 \global\@noskipsectrue
2058 \everypar{%
2059 \if@noskipsec
2060 \global\@noskipsecfalse
2061 {\setbox\z@\lastbox}%
2062 \clubpenalty\@M
2063 \begingroup \@svsechd \endgroup
2064 \unskip
2065 \@tempskipa #1\relax
2066 \hskip -\@tempskipa
2067 \else
2068 \clubpenalty \@clubpenalty
2069 \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
2070 \fi\bxjs@if@ceph\everyparhook}%
2071 \else
2072 \par \nbreak
2073 \vskip \@tempskipa
2074 \@afterheading
2075 \fi
2076 \if@slide
2077 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
2078 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt

```

```

2079     \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
2080 \fi
2081 \par % 2000-12-18
2082 \ignorespaces}
2083 \def\@ssect#1#2#3#4#5{%
2084 \@tempkipa #3\relax
2085 \ifdim \@tempkipa<\z@
2086 \def\@svsechd{#4{\hskip #1\relax #5}}%
2087 \else
2088 \begingroup
2089 #4{%
2090 \@hangfrom{\hskip #1}%
2091 \interlinepenalty \@M #5\@par}%
2092 \endgroup
2093 \fi
2094 \@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 本体で定義済みです。

```

\subsectionmark 2095 \newcommand*\chaptermark[1]{}
\subsubsectionmark 2096 % \newcommand*{\sectionmark}[1]{}
2097 % \newcommand*{\subsectionmark}[1]{}
\paragraphmark 2098 % \newcommand*{\subsubsectionmark}[1]{}
\subparagraphmark 2099 % \newcommand*{\paragraphmark}[1]{}
2100 % \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。

```

2101 %<!book&!report>\setcounter{secnumdepth}{3}
2102 %<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。これは第

`\c@section` 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 2103 \newcounter{part}
\c@subsubsection 2104 %<book|report>\newcounter{chapter}
2105 %<book|report>\newcounter{section}[chapter]
\c@paragraph 2106 %<!book&!report>\newcounter{section}
\c@subparagraph 2107 \newcounter{subsection}[section]
2108 \newcounter{subsubsection}[subsection]
2109 \newcounter{paragraph}[subsubsection]
2110 \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。

```

2111 \renewcommand{\thepart}{\@Roman\c@part}
2112 %<!*book&!report>
2113 \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2114 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
2115 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
2116 \else
2117 \renewcommand{\thesection}{\@arabic\c@section}
2118 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2119 \fi
2120 %</!book&!report>
2121 %<*book|report>
2122 \renewcommand{\thechapter}{\@arabic\c@chapter}
2123 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
2124 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
2125 %</book|report>
2126 \renewcommand{\thesubsubsection}{%
2127   \thesubsection.\@arabic\c@subsubsection}
2128 \renewcommand{\theparagraph}{%
2129   \thesubsubsection.\@arabic\c@paragraph}
2130 \renewcommand{\thesubparagraph}{%
2131   \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。

```

2132 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2133 %<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 pL^AT_EX の標準クラス 2017/03/05 も参照)。

```
2134 %<*book|report>
2135 \newcommand\frontmatter{%
2136   \pltx@cleartooddpage
2137   \@mainmatterfalse
2138   \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。

```
2139 \newcommand\mainmatter{%
2140   \pltx@cleartooddpage
2141   \@mainmattertrue
2142   \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。

```
2143 \newcommand\backmatter{%
2144   \if@openleft
2145     \cleardoublepage
2146   \else\if@openright
2147     \cleardoublepage
2148   \else
2149     \clearpage
2150   \fi\fi
2151   \@mainmatterfalse}
2152 %</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。

```
2153 %<!*book&!report>
2154 \newcommand\part{%
2155   \if@noskipsec \leavevmode \fi
2156   \par
2157   \addvspace{4ex}%
2158   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2159   \secdef\@part\@spart}
2160 %</!*book&!report>
```

book および report クラスの場合は、少し複雑です。

```
2161 %<*book|report>
2162 \newcommand\part{%
2163   \if@openleft
2164     \cleardoublepage
2165   \else\if@openright
2166     \cleardoublepage
2167   \else
2168     \clearpage
2169   \fi\fi
2170   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2171   \if@twocolumn
2172     \onecolumn
2173     \@restonecoltrue
2174   \else
2175     \@restonecolfalse
2176   \fi
2177   \null\vfil
2178   \secdef\@part\@spart}
2179 %</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。

```
2180 %<!*book&!report>
2181 \def\@part[#1]#2{%
2182   \ifnum \c@secnumdepth >\m@ne
2183     \refstepcounter{part}%
2184     \addcontentsline{toc}{part}{%
2185       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2186   \else
2187     \addcontentsline{toc}{part}{#1}%
2188   \fi
2189   \markboth{}{}%
2190   {\parindent\z@
2191     \raggedright
2192     \interlinepenalty \@M
2193     \normalfont
2194     \ifnum \c@secnumdepth >\m@ne
2195       \Large\headfont\prepartname\thepart\postpartname
2196       \par\nobreak
2197     \fi
2198     \huge \headfont #2%
2199     \markboth{}{}\par}%
2200   \nobreak
2201   \vskip 3ex
2202   \@afterheading}
2203 %</!*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。

```
2204 %<*book|report>
2205 \def\@part[#1]#2{%
2206   \ifnum \c@secnumdepth >-2\relax
2207     \refstepcounter{part}%
2208     \addcontentsline{toc}{part}{%
2209       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2210   \else
2211     \addcontentsline{toc}{part}{#1}%
2212   \fi
2213   \markboth{}{}%
2214   {\centering
2215     \interlinepenalty \@M
2216     \normalfont
2217     \ifnum \c@secnumdepth >-2\relax
2218       \huge\headfont \prepartname\thepart\postpartname
2219       \par\vskip20\p@?
2220     \fi
2221     \Huge \headfont #2\par}%
2222   \@endpart}
2223 %</book|report>
```

\@spart 番号を付けない部です。

```
2224 %<!*book&!report>
2225 \def\@spart#1{%
2226   \parindent \z@ \raggedright
2227   \interlinepenalty \@M
2228   \normalfont
2229   \huge \headfont #1\par}%
2230   \nobreak
2231   \vskip 3ex
2232   \@afterheading}
2233 %</!*book&!report>
2234 %<*book|report>
2235 \def\@spart#1{%
2236   \centering
2237   \interlinepenalty \@M
2238   \normalfont
2239   \Huge \headfont #1\par}%
2240   \@endpart}
2241 %</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。

```
2242 %<*book|report>
```

```

2243 \def\@endpart{\vfil\newpage
2244 \if@twoside
2245 \if@openleft %% added (2017/02/24)
2246 \null\thispagestyle{empty}\newpage
2247 \else\if@openright %% added (2016/12/13)
2248 \null\thispagestyle{empty}\newpage
2249 \fi\fi %% added (2016/12/13, 2017/02/24)
2250 \fi
2251 \if@restonecol
2252 \twocolumn
2253 \fi}
2254 %</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。

```

2255 %<*book|report>
2256 \newcommand{\chapter}{%
2257 \if@openleft\cleardoublepage\else
2258 \if@openright\cleardoublepage\else\clearpage\fi\fi
2259 \plainifnotempty % 元: \thispagestyle{plain}
2260 \global\@topnum\z@
2261 \if@english \@afterindentfalse \else \@afterindenttrue \fi
2262 \secdef
2263 {\@omit@numberfalse\@chapter}%
2264 {\@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。

```

2265 \def\@chapter[#1]#2{%
2266 \ifnum \c@secnumdepth >\m@ne
2267 \if@mainmatter
2268 \refstepcounter{chapter}%
2269 \typeout{\@chapapp\thechapter\@chappos}%
2270 \addcontentsline{toc}{chapter}%
2271 {\protect\numberline
2272 % {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2273 {\@chapapp\thechapter\@chappos}%
2274 #1}%
2275 \else\addcontentsline{toc}{chapter}{#1}\fi
2276 \else
2277 \addcontentsline{toc}{chapter}{#1}%
2278 \fi
2279 \chaptermark{#1}%
2280 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
2281 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
2282 \if@twocolumn

```

```

2283 \topnewpage[\@makechapterhead{#2}]%
2284 \else
2285 \@makechapterhead{#2}%
2286 \@afterheading
2287 \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。

```

2288 \def\@makechapterhead#1{%
2289 \vspace*{2\Cvs}% 欧文は 50pt
2290 {\parindent \z@ \raggedright \normalfont
2291 \ifnum \c@secnumdepth >\m@ne
2292 \if@mainmatter
2293 \huge\headfont \@chapapp\thechapter\@chappos
2294 \par\nobreak
2295 \vskip \Cvs % 欧文は 20pt
2296 \fi
2297 \fi
2298 \interlinepenalty\@M
2299 \Huge \headfont #1\par\nobreak
2300 \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。

```

2301 \def\@chapter#1{%
2302 \chaptermark{#1}%
2303 \if@twocolumn
2304 \topnewpage[\@makeschapterhead{#1}]%
2305 \else
2306 \@makeschapterhead{#1}\@afterheading
2307 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2308 \def\@makeschapterhead#1{%
2309 \vspace*{2\Cvs}% 欧文は 50pt
2310 {\parindent \z@ \raggedright
2311 \normalfont
2312 \interlinepenalty\@M
2313 \Huge \headfont #1\par\nobreak
2314 \vskip 3\Cvs}} % 欧文は 40pt
2315 %</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。

```

2316 \if@twocolumn
2317 \newcommand{\section}{%
2318 %<jspf>\ifx\maketitle\relax\else\maketitle\fi

```

```

2319 \startsection{section}{1}{\z@}%
2320 %<kiyou> {0.6\Cvs}{0.4\Cvs}%
2321 %<kiyou> {\Cvs}{0.5\Cvs}%
2322 % {\normalfont\large\headfont\@secapp}}
2323 {\normalfont\large\headfont\raggedright}}
2324 \else
2325 \newcommand{\section}{%
2326 \if@slide\clearpage\fi
2327 \startsection{section}{1}{\z@}%
2328 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2329 {.5\Cvs \@plus.3\Cdp}% 後アキ
2330 % {\normalfont\Large\headfont\@secapp}}
2331 {\normalfont\Large\headfont\raggedright}}
2332 \fi

```

`\subsection` 同上です。

```

2333 \if@twocolumn
2334 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2335 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2336 {\normalfont\normalsize\headfont}}
2337 \else
2338 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2339 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2340 {.5\Cvs \@plus.3\Cdp}% 後アキ
2341 {\normalfont\large\headfont}}
2342 \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)。

```

2343 \if@twocolumn
2344 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2345 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2346 {\normalfont\normalsize\headfont}}
2347 \else
2348 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2349 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2350 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2351 {\normalfont\normalsize\headfont}}
2352 \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。

```

2353 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2354 \let\bxjs@org@paragraph@mark\jsParagraphMark
2355 \if\bxjs@paragraph@mark@empty
2356   \let\jsParagraphMark@empty
2357 \else\if\bxjs@paragraph@mark@undefined\else
2358   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2359 \fi\fi
2360 \let\jsJaChar@empty
2361 \if@twocolumn
2362   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z0}%
2363     {\z0}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2364     %<jspf>   {\normalfont\normalsize\headfont}}
2365     %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
2366   \else
2367     \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z0}%
2368       {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2369       {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2370       %<jspf>   {\normalfont\normalsize\headfont}}
2371       %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
2372   \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2373 \if@twocolumn
2374   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z0}%
2375     {\z0}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2376     {\normalfont\normalsize\headfont}}
2377   \else
2378     \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z0}%
2379       {\z0}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2380       {\normalfont\normalsize\headfont}}
2381   \fi

```

8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```
2382 \if@slide
2383   \setlength\leftmargini{1\jsZw}
2384 \else
2385   \if@twocolumn
2386     \setlength\leftmargini{2\jsZw}
2387   \else
2388     \setlength\leftmargini{3\jsZw}
2389   \fi
2390 \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 2391 \if@slide
\leftmarginv 2392   \setlength\leftmarginii {1\jsZw}
2393   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2394   \setlength\leftmarginiv {1\jsZw}
2395   \setlength\leftmarginv  {1\jsZw}
2396   \setlength\leftmarginvi {1\jsZw}
2397 \else
2398   \setlength\leftmarginii {2\jsZw}
2399   \setlength\leftmarginiii{2\jsZw}
2400   \setlength\leftmarginiv {2\jsZw}
2401   \setlength\leftmarginv  {1\jsZw}
2402   \setlength\leftmarginvi {1\jsZw}
2403 \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。

```
2404 \setlength \labelsep {0.5\jsZw} % .5em
2405 \setlength \labelwidth{\leftmargini}
2406 \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。

```
2407 \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 2408 \@beginparpenalty -\@lowpenalty
\@itempenalty 2409 \@endparpenalty -\@lowpenalty
2410 \@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 $\begin{smallmatrix} +0.2 \\ -0.1 \end{smallmatrix}$ `\baselineskip` を思い切って外しました。

```
2411 \def\@listi{\leftmargin\leftmargini
2412 \parsep \z@
2413 \topsep 0.5\baselineskip
2414 \itemsep \z@ \relax}
2415 \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)。

```
2416 \@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 2417 \def\@listii{\leftmargin\leftmarginii
\@listiv 2418 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2419 \topsep \z@
\@listv 2420 \parsep \z@
\@listvi 2421 \itemsep\parsep}
2422 \def\@listiii{\leftmargin\leftmarginiii
2423 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2424 \topsep \z@
2425 \parsep \z@
2426 \itemsep\parsep}
2427 \def\@listiv {\leftmargin\leftmarginiv
2428 \labelwidth\leftmarginiv
2429 \advance\labelwidth-\labelsep}
2430 \def\@listv {\leftmargin\leftmarginv
2431 \labelwidth\leftmarginv
2432 \advance\labelwidth-\labelsep}
2433 \def\@listvi {\leftmargin\leftmarginvi
2434 \labelwidth\leftmarginvi
2435 \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。

```
2436 \renewcommand{\theenumi}{\@arabic\c@enumi}
2437 \renewcommand{\theenumii}{\@alph\c@enumii}
2438 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2439 \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。

```
2440 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2441 \newcommand*\jsInJaParen}[1]{%
2442   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2443 \newcommand\labelenumi{\theenumi.}
2444 \newcommand\labelenumii{\jsInJaParen{\theenumii}}
2445 \newcommand\labelenumiii{\theenumiii.}
2446 \newcommand\labelenumiv{\theenumiv.}
```

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

```
\p@enumiv 2447 \renewcommand{\p@enumii}{\theenumi}
2448 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2449 \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 2450 \newcommand\labelitemi{\textbullet}
\labelitemiii 2451 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiv 2452 \newcommand\labelitemiii{\textasteriskcentered}
2453 \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。

```
2454 \newenvironment{description}{%
2455   \list{}{%
2456     \labelwidth=\leftmargin
2457     \labelsep=1\jsZw
2458     \advance \labelwidth by -\labelsep
2459     \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。

```
2460 \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。

```

2461 %<*book|report>
2462 \newenvironment{chapterabstract}{%
2463   \begin{list}{}{%
2464     \listparindent=1\jsZw
2465     \itemindent=\listparindent
2466     \rightmargin=0pt
2467     \leftmargin=5\jsZw}\item[]{\end{list}\vspace{\baselineskip}}
2468 %>/book|report>

```

“普通の” abstract 環境の定義。

```

2469 %<*article|report|slide>
2470 \newbox\@abstractbox
2471 \if@titlepage
2472   \newenvironment{abstract}{%
2473     \titlepage
2474     \null\vfil
2475     \@beginparpenalty\@lowpenalty
2476     \begin{center}%
2477       \headfont \abstractname
2478       \@endparpenalty\@M
2479     \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。

```

2480   \par}%
2481   {\par\vfil\null\endtitlepage}
2482 \else
2483   \newenvironment{abstract}{%
2484     \if@twocolumn
2485       \ifx\maketitle\relax
2486         \section*{\abstractname}%
2487       \else
2488         \global\setbox\@abstractbox\hbox\bgroup

```

```

2489     \begin{minipage}[b]{\textwidth}
2490         \small\parindent1\jsZw
2491         \begin{center}%
2492             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2493         \end{center}%
2494         \list{ }{%
2495             \listparindent\parindent
2496             \itemindent \listparindent
2497             \rightmargin \leftmargin}%
2498         \item\relax
2499     \fi
2500 \else
2501     \small
2502     \begin{center}%
2503         {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2504     \end{center}%
2505     \list{ }{%
2506         \listparindent\parindent
2507         \itemindent \listparindent
2508         \rightmargin \leftmargin}%
2509     \item\relax
2510 \fi}{\if@twocolumn
2511     \ifx\maketitle\relax
2512     \else
2513         \endlist\end{minipage}\egroup
2514     \fi
2515 \else
2516     \endlist
2517 \fi}
2518 \fi
2519 %</article|report|slide>
2520 %<*jspf>
2521 \newbox\@abstractbox
2522 \newenvironment{abstract}{%
2523     \global\setbox\@abstractbox\hbox\bgroup
2524     \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2525         \small
2526         \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2527     {\end{minipage}\egroup}
2528 %</jspf>

```

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

```

2529 %<*book|report>
2530 \ifbxjs@force@chapterabstract
2531     \let\abstract\chapterabstract
2532     \let\endabstract\endchapterabstract
2533 \fi
2534 %</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。

```
2535 %<*jspf>
2536 %\newbox\@keywordsbox
2537 %\newenvironment{keywords}{%
2538 % \global\setbox\@keywordsbox\hbox\bgroup
2539 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2540 % \small\parindent0\jsZw}%
2541 % {\end{minipage}\egroup}
2542 %</jspf>
```

■verse 環境

`verse` 詩のための `verse` 環境です。

```
2543 \newenvironment{verse}{%
2544 \let \=\@centercr
2545 \list{}{%
2546 \itemsep \z@
2547 \itemindent -2\jsZw % 元: -1.5em
2548 \listparindent\itemindent
2549 \rightmargin \z@
2550 \advance\leftmargin 2\jsZw}% 元: 1.5em
2551 \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。

```
2552 \newenvironment{quotation}{%
2553 \list{}{%
2554 \listparindent\parindent
2555 \itemindent\listparindent
2556 \rightmargin \z0}%
2557 \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。

```
2558 \newenvironment{quote}%
2559 {\list{}{\rightmargin\z0}\item\relax}{\endlist}
```

■定理など `lthm.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。

```
2560 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2561   \item[\hskip \labelsep{\headfont #1\ #2}]}
2562 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2563   \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。

```
2564 \newenvironment{titlepage}{%
2565 %<book>   \pltx@cleartooddpage %% 2017-02-24
2566   \if@twocolumn
2567     \@restonecoltrue\onecolumn
2568   \else
2569     \@restonecolfalse\newpage
2570   \fi
2571   \thispagestyle{empty}%
2572   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2573   24
2574 }%
2575 {\if@restonecol\twocolumn \else \newpage \fi
2576 \if@twoside\else
2577   \setcounter{page}\@ne
2578 \fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2578 %<!*book&!report>
2579 \newcommand{\appendix}{\par
2580   \setcounter{section}{0}%
2581   \setcounter{subsection}{0}%
2582   \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2583   \gdef\presectionname{\appendixname}%
2584   \gdef\postsectionname{}}%
2585 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]

```

```

2586 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2587 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2588 \else
2589 \gdef\@secapp{\appendixname}%
2590 \gdef\@secpos{}%
2591 \gdef\thesection{\@Alph\c@section}%
2592 \fi}
2593 %<!/book&!report>
2594 %<*book|report>
2595 \newcommand{\appendix}{\par
2596 \setcounter{chapter}{0}%
2597 \setcounter{section}{0}%
2598 \gdef\@chapapp{\appendixname}%
2599 \gdef\@chappos{}%
2600 \gdef\thechapter{\@Alph\c@chapter}}
2601 %</book|report>

```

8.4 パラメータの設定

■array と tabular 環境

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

```
2602 \setlength\arraycolsep{5\p@?}
```

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

```
2603 \setlength\tabcolsep{6\p@?}
```

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

```
2604 \setlength\arrayrulewidth{.4\p@}
```

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

```
2605 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2606 \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。

```
2607 \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。

```
2608 \setlength\fboxsep{3\p@?}
2609 \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。

```
2610 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2611 %<*book|report>
2612 \@addtoreset{equation}{chapter}
2613 \renewcommand\theequation
2614   {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
2615 %</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。

```
2616 % \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。

```
2617 % \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。

```
2618 % \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。

```
2619 %<*!book&!report>
2620 \newcounter{figure}
2621 \renewcommand \thefigure {\@arabic\c@figure}
```

```

2622 %</*!book&!report>
2623 %<*book|report>
2624 \newcounter{figure}[chapter]
2625 \renewcommand \thefigure
2626     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2627 %</book|report>

```

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

```

\ext@figure 2628 \def\fps@figure{tbp}
2629 \def\ftype@figure{1}
\fnun@figure 2630 \def\ext@figure{lof}
2631 \def\fnun@figure{\figurename\nobreak\thefigure}

```

`figure` * 形式は段抜きフロートです。

```

figure* 2632 \newenvironment{figure}%
2633     {\@float{figure}}%
2634     {\end@float}
2635 \newenvironment{figure*}%
2636     {\@dblfloat{figure}}%
2637     {\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。

```

2638 %</*!book&!report>
2639 \newcounter{table}
2640 \renewcommand\thetable{\@arabic\c@table}
2641 %</*!book&!report>
2642 %<*book|report>
2643 \newcounter{table}[chapter]
2644 \renewcommand \thetable
2645     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2646 %</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 2647 \def\fps@table{tbp}
2648 \def\ftype@table{2}
\fnun@table 2649 \def\ext@table{lot}
2650 \def\fnun@table{\tablename\nobreak\thetable}

```

`table` * は段抜きフロートです。

```

table* 2651 \newenvironment{table}%
2652     {\@float{table}}%
2653     {\end@float}
2654 \newenvironment{table*}%

```

```
2655          {\@dblfloat{table}}%
2656          {\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。

```
2657 \newlength\abovecaptionskip
2658 \newlength\belowcaptionskip
2659 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2660 \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。

```
2661 %<*\jspf>
2662 % \long\def\@makecaption#1#2{{\small
2663 %   \advance\leftskip10\jsc@mmm
2664 %   \advance\rightskip10\jsc@mmm
2665 %   \vskip\abovecaptionskip
2666 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2667 %   \ifdim \wd\@tempboxa >\hsize
2668 %     #1\hskip1\jsZw\relax #2\par
2669 %   \else
2670 %     \global \@minipagefalse
2671 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2672 %   \fi
2673 %   \vskip\belowcaptionskip}}
2674 \long\def\@makecaption#1#2{{\small
2675   \advance\leftskip .0628\linewidth
2676   \advance\rightskip .0628\linewidth
2677   \vskip\abovecaptionskip
2678   \sbox\@tempboxa{#1\zwspace#2}%
2679   \ifdim \wd\@tempboxa <\hsize \centering \fi
2680   #1\zwspace#2\par
2681   \vskip\belowcaptionskip}}
2682 %</\jspf>
2683 %<*\jspf>
2684 \long\def\@makecaption#1#2{%
2685   \vskip\abovecaptionskip
2686   \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2687   \ifdim \wd\@tempboxa >\hsize
2688     {\small\sffamily
```



```

2689     \list{#1}{%
2690         \renewcommand{\makelabel}[1]{##1\hfil}
2691         \itemsep \z@
2692         \itemindent \z@
2693         \labelsep \z@
2694         \labelwidth 11\jsc@mmm
2695         \listparindent\z@
2696         \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2697 \else
2698     \global \@minipagefalse
2699     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2700     \fi
2701     \vskip\belowcaptionskip}
2702 %</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。

```

2703 \newif\if@jsc@warnoldfontcmd
2704 \@jsc@warnoldfontcmdtrue
2705 \newif\if@jsc@warnoldfontcmdexception
2706 \@jsc@warnoldfontcmdexceptionfalse

```

```

\jsc@DeclareOldFontCommand
2707 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2708     \@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2709     \DeclareOldFontCommand{#1}{%
2710         \bxjs@oldfontcmd{#1}#2%
2711     }{%
2712         \bxjs@oldfontcmd{#1}#3%
2713     }%
2714 }
2715 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%

```

```

2716 \ClassInfo\bxjs@clsname
2717 {Old font command '\string#1' is used!!\MessageBreak
2718 The first occurrence is}%
2719 }

```

`\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。

```

2720 \newcommand*\allowoldfontcommands{%
2721 \@jsc@warnoldfontcmdexceptiontrue}
2722 \newcommand*\disallowoldfontcommands{%
2723 \@jsc@warnoldfontcmdexceptionfalse}

2724 \let\bxjs@oldfontcmd@list\@empty
2725 \def\bxjs@oldfontcmd#1{%
2726 \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2727 \def\bxjs@oldfontcmd@a#1#2{%
2728 \if@jsc@warnoldfontcmdexception\else
2729 \global\@jsc@warnoldfontcmdfalse
2730 \ifx#1\relax
2731 \global\let#1=t%
2732 \jsc@warnoldfontcmd{#2}%
2733 \fi
2734 \fi}
2735 \def\bxjs@warnoldfontcmd@final{%
2736 % \par
2737 \global\let\bxjs@warnoldfontcmd@final\@empty
2738 \let\@tempa\@empty
2739 \def\do##1{%
2740 \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2741 \edef\@tempa{\@tempa \space\string##1}\fi}
2742 \bxjs@oldfontcmd@list
2743 \ifx\@tempa\@empty\else
2744 \ClassWarningNoLine\bxjs@clsname
2745 {Some old font commands were used in text:\MessageBreak
2746 \space\@tempa\MessageBreak
2747 You should note, that since 1994 LaTeX2e provides a\MessageBreak
2748 new font selection scheme called NFSS2 with several\MessageBreak
2749 new, combinable font commands. The
2750 class provides\MessageBreak
2751 the old font commands only for compatibility}
2752 \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。

```

2753 \def\bxjs@warnoldfontcmd@kick@final{%
2754   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2755 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2756 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2757 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2758 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2759 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2760 \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。

```
2761 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

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

```

2762 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2763 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2764 \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 2765 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2766 \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。

```
2767 \newcommand\@pnumwidth{1.55em}
2768 \newcommand\@tocrmarg{2.55em}
2769 \newcommand\@dotsep{4.5}
2770 %<!book&!report>\setcounter{tocdepth}{2}
2771 %<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)

```
2772 \newdimen\jsc@tocl@width
2773 \newcommand{\tableofcontents}{%
2774 %<*book|report>
2775 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2776 \settowidth\@tempdima{\headfont\appendixname}%
2777 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2778 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2779 \if@twocolumn
2780 \@restonecoltrue\onecolumn
2781 \else
2782 \@restonecolfalse
2783 \fi
2784 \chapter*{\contentsname}%
2785 \mkboth{\contentsname}{}%
2786 %</book|report>
2787 %<*!book&!report>
2788 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2789 \settowidth\@tempdima{\headfont\appendixname}%
2790 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
```

```

2791 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2792 \section*{\contentsname}%
2793 \@mkboth{\contentsname}{\contentsname}%
2794 %<!/book&!report>
2795 \@starttoc{toc}%
2796 %<book|report> \if@restonecol\twocolumn\fi
2797 }

```

\l@part 部の目次です。

```

2798 \newcommand*{\l@part}[2]{%
2799 \ifnum \c@tocdepth >-2\relax
2800 %<!/book&!report> \addpenalty\@secpenalty
2801 %<book|report> \addpenalty{-\@highpenalty}%
2802 \addvspace{2.25em \@plus\p@?}%
2803 \begingroup
2804 \parindent \z@
2805 % \@pnumwidth should be \@tocrmarg
2806 % \rightskip \@pnumwidth
2807 \rightskip \@tocrmarg
2808 \parfillskip -\rightskip
2809 {\leavevmode
2810 \large \headfont
2811 \setlength\@lnumwidth{4\jsZw}%
2812 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2813 \nobreak
2814 %<book|report> \global\@nobreaktrue
2815 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2816 \endgroup
2817 \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)

```

2818 %<*book|report>
2819 \newcommand*{\l@chapter}[2]{%
2820 \ifnum \c@tocdepth >\m@ne
2821 \addpenalty{-\@highpenalty}%
2822 \addvspace{1.0em \@plus\p@?}
2823 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2824 \begingroup
2825 \parindent\z@
2826 % \rightskip\@pnumwidth
2827 \rightskip\@tocrmarg
2828 \parfillskip-\rightskip
2829 \leavevmode\headfont
2830 % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2831 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2832 \advance\leftskip\@lnumwidth \hskip-\leftskip
2833 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par

```

```

2834     \penalty\@highpenalty
2835     \endgroup
2836     \fi}
2837 %</book|report>

```

\l@section 節の目次です。

```

2838 %<!*book&!report>
2839 \newcommand*{\l@section}[2]{%
2840     \ifnum \c@tocdepth >\z@
2841         \addpenalty{\@secpenalty}%
2842         \addvspace{1.0em \@plus\p@?}%
2843         \begingroup
2844             \parindent\z@
2845 %         \rightskip\@pnumwidth
2846         \rightskip\@tocrmarg
2847         \parfillskip-\rightskip
2848         \leavevmode\headfont
2849 %         % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2850         \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2851         \advance\leftskip\@lnumwidth \hskip-\leftskip
2852         #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2853     \endgroup
2854     \fi}
2855 %</!*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。

```

2856 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

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

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

```

\l@subparagraph 2857 %<!*book&!report>
2858 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2859 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2860 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2861 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2862 %
2863 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1zw}{3zw}}
2864 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2865 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2866 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2867 %
2868 \newcommand*{\l@subsection}{%
2869     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2870     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2871 \newcommand*{\l@subsubsection}{%
2872     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw

```

```

2873      \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2874 \newcommand*\l@paragraph}{%
2875      \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2876      \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2877 \newcommand*\l@subparagraph}{%
2878      \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2879      \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2880 %</!book&!report>
2881 %<*book|report>
2882 % \newcommand*\l@subsection} {\@dottedtocline{2}{3.8em}{3.2em}}
2883 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2884 % \newcommand*\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2885 % \newcommand*\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2886 \newcommand*\l@section}{%
2887      \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2888      \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2889 \newcommand*\l@subsection}{%
2890      \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2891      \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2892 \newcommand*\l@subsubsection}{%
2893      \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2894      \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2895 \newcommand*\l@paragraph}{%
2896      \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2897      \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2898 \newcommand*\l@subparagraph}{%
2899      \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2900      \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2901 %</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。

```

2902 \newdimen\@lnumwidth
2903 \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。

```

2904 \def\jsTocLine{\leaders\hbox{%
2905   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}

```

```

2906 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2907   \vskip \z@ \@plus.2\p@?
2908   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2909     \parindent #2\relax\@afterindenttrue
2910     \interlinepenalty\@M
2911     \leavevmode
2912     \@lnumwidth #3\relax
2913     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2914     {#4}\nobreak
2915     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2916       \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2917 \newcommand{\listoffigures}{%
2918 %<*book|report>
2919   \if@twocolumn\@restonecoltrue\onecolumn
2920   \else\@restonecolfalse\fi
2921   \chapter*{\listfigurename}%
2922   \@mkboth{\listfigurename}{}%
2923 %</book|report>
2924 %<!*book&!report>
2925   \section*{\listfigurename}%
2926   \@mkboth{\listfigurename}{\listfigurename}%
2927 %</!book&!report>
2928   \@starttoc{lof}%
2929 %<book|report> \if@restonecol\twocolumn\fi
2930 }

```

`\l@figure` 図目次の項目を出力します。

```

2931 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2932 \newcommand{\listoftables}{%
2933 %<*book|report>
2934   \if@twocolumn\@restonecoltrue\onecolumn
2935   \else\@restonecolfalse\fi
2936   \chapter*{\listtablename}%
2937   \@mkboth{\listtablename}{}%
2938 %</book|report>
2939 %<!*book&!report>
2940   \section*{\listtablename}%
2941   \@mkboth{\listtablename}{\listtablename}%
2942 %</!book&!report>
2943   \@starttoc{lot}%
2944 %<book|report> \if@restonecol\twocolumn\fi
2945 }

```


`\l@table` 表目次は図目次と同じです。

```
2946 \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。

```
2947 \newdimen\bibindent
```

```
2948 \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。

```
2949 \newenvironment{thebibliography}[1]{%
2950   \@jsc@warnoldfontcmdexceptiontrue
2951   \global\let\presectionname\relax
2952   \global\let\postsectionname\relax
2953   %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2954   %<*kiyou>
2955   \vspace{1.5\baselineskip}
2956   \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2957   \vspace{0.5\baselineskip}
2958   %</kiyou>
2959   %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2960   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2961   \list{\@biblabel{\@arabic{c@enumiv}}}%
2962     {\settowidth\labelwidth{\@biblabel{#1}}%
2963     \leftmargin\labelwidth
2964     \advance\leftmargin\labelsep
2965     \@openbib@code
2966     \usecounter{enumiv}%
2967     \let\p@enumiv\@empty
2968     \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2969   %<kiyou> \small
2970   \sloppy
2971   \clubpenalty4000
2972   \@clubpenalty\clubpenalty
2973   \widowpenalty4000%
2974   \sfcode`.\@m}
2975   {\def\@noitemerr
2976     {\@latex@warning{Empty `thebibliography' environment}}}%
2977   \endlist}
```

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

```
2978 \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。

```
2979 \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。

```
2980 % \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
`\@citex` 要に応じて生かしてください。かっこの前後に入るグルーを `\jsInhibitGlue` で取っていますので、オリジナル同様、Knuth-`\cite{knu}`□ のように半角空白で囲んでください。

```
2981 % \def\@citex[#1]#2{\leavevmode
2982 %   \let\@citea\@empty
2983 %   \@cite{\@for\@citeb:=#2\do
2984 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m }%
2985 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2986 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2987 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2988 %     \G@refundefinedtrue
2989 %     \@latex@warning
2990 %       {Citation `@\@citeb' on page \thepage \space undefined}}%
2991 %     {\@citeofmt{\csname b@\@citeb\endcsname}}}{#1}}
2992 % \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。

```
2993 % \DeclareRobustCommand\cite{\unskip
2994 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2995 % \def\@cite#1#2{\hbox{\scriptsize{#1\if@tempswa
2996 %   , \jsInhibitGlue\ #2\fi}) }}$}
```

10.3 索引

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

```
2997 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2998   \if@twocolumn
2999     \onecolumn\@restonecolfalse
3000   \else
3001     \clearpage\@restonecoltrue
3002   \fi
3003   \columnseprule.4pt \columnsep 2\jsZw
3004   \ifx\multicols\@undefined
3005 %<book|report>   \twocolumn[\@makeschapterhead{\indexname}]%
3006 %<book|report>   \addcontentsline{toc}{chapter}{\indexname}}%
```

```

3007 %<!book&!report>      \def\presectionname{}\def\postsectionname{%
3008 %<!book&!report>      \twocolumn[\section*{\indexname}]%
3009      \else
3010          \ifdim\textwidth<\fullwidth
3011              \setlength{\evensidemargin}{\oddsidemargin}
3012              \setlength{\textwidth}{\fullwidth}
3013              \setlength{\linewidth}{\fullwidth}
3014 %<book|report>          \begin{multicols}{3}[\chapter*{\indexname}]%
3015 %<book|report>          \addcontentsline{toc}{chapter}{\indexname}]%
3016 %<!book&!report>      \def\presectionname{}\def\postsectionname{%
3017 %<!book&!report>      \begin{multicols}{3}[\section*{\indexname}]%
3018      \else
3019 %<book|report>          \begin{multicols}{2}[\chapter*{\indexname}]%
3020 %<book|report>          \addcontentsline{toc}{chapter}{\indexname}]%
3021 %<!book&!report>      \def\presectionname{}\def\postsectionname{%
3022 %<!book&!report>      \begin{multicols}{2}[\section*{\indexname}]%
3023      \fi
3024      \fi
3025 %<book|report>      \@mkboth{\indexname}{}%
3026 %<!book&!report>      \@mkboth{\indexname}{\indexname}]%
3027      \plainifnotempty % \thispagestyle{plain}
3028      \parindent\z@
3029      \parskip\z@ \@plus .3\p@\relax
3030      \let\item\@idxitem
3031      \raggedright
3032      \footnotesize\narrowbaselines
3033  }{
3034      \ifx\multicols\@undefined
3035          \if@restonecol\onecolumn\fi
3036      \else
3037          \end{multicols}
3038      \fi
3039      \clearpage
3040  }

```

`\@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 3041 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 3042 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
3043 \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。

```

3044 \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。

```

3045 \newcommand\seename{\if@english see\else →\fi}
3046 \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。

```
3047 \ifx\pltx@foot@penalty\undefined
3048 \let\footnotes@ve=\footnote
3049 \def\footnote{\jsInhibitGlue\footnotes@ve}
3050 \let\footnotemarks@ve=\footnotemark
3051 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
3052 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

pT_EX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
3053 \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。

```
3054 % \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。

```

3055 \renewcommand{\footnoterule}{%
3056   \kern-2.6\p@? \kern-.4\p@
3057   \hrule width .4\columnwidth
3058   \kern 2.6\p@?}

```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```

3059 %<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, *TEX and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9)

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

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

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

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

```

3060 \long\def\@footnotetext{%
3061   \insert\footins\bgroup
3062     \normalfont\footnotesize
3063     \interlinepenalty\interfootnotelinepenalty
3064     \splittopskip\footnotesep
3065     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
3066     \hsize\columnwidth \@parboxrestore
3067     \protected@edef\@currentlabel{%
3068       \csname p@footnote\endcsname\@thefnmark
3069     }%
3070     \color@begingroup
3071     \@makefnmark{%
3072       \rule{\z@\footnotesep}{\ignorespaces}}%
3073     \futurelet\jsc@next\jsc@fo@t}
3074 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@f@t
3075               \else \let\jsc@next\jsc@f@t\fi \jsc@next}
3076 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3077 \def\jsc@f@t#1{#1\jsc@@foot}
3078 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
3079   \ifx\pltx@foot@penalty\@undefined\else
3080     \ifhmode\null\fi
3081     \ifnum\pltx@foot@penalty=\z@\else
3082       \penalty\pltx@foot@penalty
3083       \pltx@foot@penalty\z@
3084     \fi
3085   \fi}

```

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

```

3086 \newcommand\@makefnmark[1]{%
3087   \advance\leftskip 3\jsZw

```

```

3088 \parindent 1\jsZw
3089 \noindent
3090 \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。

```

3091 % \def\@xfootnotenext[#1]{%
3092 %   \begingroup
3093 %     \ifnum#1>\z@
3094 %       \csname c@\@mpfn\endcsname #1\relax
3095 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
3096 %     \else
3097 %       \unrestored@protected@xdef\@thefnmark{}%
3098 %     \fi
3099 %   \endgroup
3100 %   \@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

段落頭のかぎかっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

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

```
3101 \let\jsInhibitGlueAtParTop\@empty
```

`\everyparhook` 全ての段落の冒頭で実行されるフック。これの初期値を先述の `\jsInhibitGlueAtParTop` とする。

```

3102 \def\everyparhook{\jsInhibitGlueAtParTop}
3103 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
3104 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3105 \fi

```

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

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

[2017-02-13] `\jsc@tempa` は実はテンポラリではなく「この処理専用のユニーク制御綴」である必要があります。間違っって別の箇所で使う危険性が高いため、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。

```
3106 \ifx j\jsEngine
3107 \def\@inhibitglue{%
3108   \futurelet\@let@token\@inhibitglue}
3109 \begingroup
3110 \let\GDEF=\gdef
3111 \let\CATCODE=\catcode
3112 \let\ENDGROUP=\endgroup
3113 \CATCODE`k=12
3114 \CATCODE`a=12
3115 \CATCODE`n=12
3116 \CATCODE`j=12
3117 \CATCODE`i=12
3118 \CATCODE`c=12
3119 \CATCODE`h=12
3120 \CATCODE`r=12
3121 \CATCODE`t=12
3122 \CATCODE`e=12
3123 \GDEF\KANJI@CHARACTER{kanji character }
3124 \ENDGROUP
3125 \def\@inhibitglue{%
3126   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
3127   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3128   \def\jsc@ig@temp{#1}%
3129   \ifx\jsc@ig@temp@empty
3130     \ifnum\the\inhibitxspcode`#2=2\relax
3131       \inhibitglue
3132     \fi
3133   \fi}
3134 \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。

```
3135 \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。

```
3136 \def\@doendpe{%
3137   \@endpetrue
3138   \def\par{%
3139     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3140   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] minipage 環境にも対策します。

```
3141 \def\@setminipage{%
3142   \@minipagetrue
3143   \everypar{\@minipagefalse\everypar{\everyparhook}}%
3144 }
```

\item 命令の直後です。

```
3145 \def\@item[#1]{%
3146   \if@noparitem
3147     \@donoparitem
3148   \else
3149     \if@inlabel
3150       \indent \par
3151     \fi
3152     \ifhmode
3153       \unskip\unskip \par
3154     \fi
3155     \if@newlist
3156       \if@nobreak
3157         \@nbitem
3158       \else
3159         \addpenalty\@beginparpenalty
3160         \addvspace\@topsep
3161         \addvspace{-\parskip}%
3162       \fi
3163     \else
3164       \addpenalty\@itempenalty
3165       \addvspace\itemsep
3166     \fi
3167     \global\@inlabeltrue
3168   \fi
3169   \everypar{%
3170     \@minipagefalse
3171     \global\@newlistfalse
3172     \if@inlabel
3173       \global\@inlabelfalse
3174       {\setbox\z@\lastbox
3175        \ifvoid\z@
3176          \kern-\itemindent
3177        \fi}%
3178     \box\@labels
3179     \penalty\z@
```



```

3180 \fi
3181 \if@nobreak
3182 \@nobreakfalse
3183 \clubpenalty \@M
3184 \else
3185 \clubpenalty \@clubpenalty
3186 \everypar{\everyparhook}%
3187 \fi
3188 \everyparhook}%
3189 \if@noitemarg
3190 \@noitemargfalse
3191 \if@nmbrrlist
3192 \refstepcounter\@listctr
3193 \fi
3194 \fi
3195 \sbox\@tempboxa{\makelabel{#1}}%
3196 \global\setbox\@labels\hbox{%
3197 \unhbox\@labels
3198 \hskip \itemindent
3199 \hskip -\labelwidth
3200 \hskip -\labelsep
3201 \ifdim \wd\@tempboxa >\labelwidth
3202 \box\@tempboxa
3203 \else
3204 \hbox to\labelwidth {\unhbox\@tempboxa}%
3205 \fi
3206 \hskip \labelsep}%
3207 \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。

```

3208 \def\@afterheading{%
3209 \@nobreaktrue
3210 \everypar{%
3211 \if@nobreak
3212 \@nobreakfalse
3213 \clubpenalty \@M
3214 \if@afterindent \else
3215 {\setbox\z@\lastbox}%
3216 \fi
3217 \else
3218 \clubpenalty \@clubpenalty
3219 \everypar{\everyparhook}%
3220 \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

```
3221 \fi
```

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

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

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

```
3222 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none\else
3223 \AtEndOfPackage{%
3224 \def\@gnewline #1{%
3225   \ifvmode
3226     \@nolnerr
3227   \else
3228     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3229     \jsInhibitGlue \ignorespaces
3230   \fi}
3231 }
3232 \fi
```

12 いろいろなロゴ

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 関連のロゴを作り直します。

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

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

※`\小`、`\上小` の制御綴は定義しない。

```
3233 \if@jslogo
3234   \IfFileExists{jslogo.sty}{%
3235     \RequirePackage{jslogo}%
3236   }{%
3237     \ClassWarningNoLine\bxjs@clsname
3238     {The package 'jslogo' is not installed.\MessageBreak
3239     It is included in the recent release of\MessageBreak
3240     the 'jsclasses' bundle}
3241   }
3242 \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: しっぽ愛好家さん)。

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

```
3243 %\let\ltx@ifnextchar\@ifnextchar
3244 %\def\ProvidesFile#1{%
3245 % \begingroup
3246 % \catcode\ 10 %
3247 % \ifnum \endlinechar<256 %
3248 % \ifnum \endlinechar>\m@ne
3249 % \catcode\endlinechar 10 %
3250 % \fi
3251 % \fi
3252 % \@makeother\%
3253 % \@makeother\&%
3254 % \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]
```

14 初期設定

■いろいろな語

```
\prepartname
\postpartname 3255 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3256 \newcommand{\postpartname}{\if@english\else 部\fi}
3257 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3258 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3259 \newcommand{\presectionname}{}% 第
\postsectionname 3260 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 3261 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3262 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3263 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibName 3264 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3265 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3266 \newcommand{\indexname}{\if@english Index\else 索引\fi}
```

```

\figurename
\tablename 3267 %<!jspf>\newcommand{\figurename}{\if@english Fig.-\else 図\fi}
3268 %<jspf>\newcommand{\figurename}{Fig.-}
3269 %<!jspf>\newcommand{\tablename}{\if@english Table-\else 表\fi}
3270 %<jspf>\newcommand{\tablename}{Table~}

```

\appendixname

```

\abstractname 3271 % \newcommand{\appendixname}{\if@english Appendix-\else 付録\fi}
3272 \newcommand{\appendixname}{\if@english \else 付録\fi}
3273 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

■今日の日付 L^AT_EX 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。

```

3274 \@onlypreamble\bxjs@decl@Seireki@cmds
3275 \@tempwafalse
3276 \if p\jsEngine \@tempwatruetrue \fi
3277 \if n\jsEngine \@tempwatruetrue \fi
3278 \bxjs@cond@if@tempwa\fi{%
3279 % 欧文 8bitTeX の場合
3280 \newif\ifjsSeireki \jsSeirekitrue
3281 \def\bxjs@decl@Seireki@cmds{%
3282   \def\西暦{\jsSeirekitrue}%
3283   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3284 \def\Seireki{\jsSeirekitrue}
3285 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3286 \def\bxjs@if@use@seireki{\bxjs@cond@ifjsSeireki\fi}
3287 \def\bxjs@iai{\noexpand~}
3288 }{%
3289 \newif\if西暦 \西暦 true
3290 \def\bxjs@decl@Seireki@cmds{%
3291   \def\西暦{\西暦 true}%
3292   \def\和暦{\西暦 false\bxjs@wareki@used}}
3293 \def\Seireki{\西暦 true}
3294 \def\Wareki{\西暦 false\bxjs@wareki@used}
3295 \def\bxjs@if@use@seireki{\bxjs@cond@if西暦\fi}
3296 \let\bxjs@iai@empty
3297 }
3298 \bxjs@decl@Seireki@cmds
3299 \let\bxjs@unxp@firstofone \let\bxjs@onxp@firstofone
3300 \bxjs@test@engine\unexpanded{%
3301   \let\bxjs@unxp\unexpanded \def\bxjs@onxp{\unexpanded\expandafter}}

```

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

```
3302 \newif\ifbxjs@bxwareki@avail
3303 \IfFileExists{bxwareki.sty}{%
3304   \RequirePackage{bxwareki}[]%
3305   \bxjs@bxwareki@availtrue}{}
```

`\bxjs@wareki@used` 和暦が非対応の場合に警告を出す。

```
3306 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\@empty
3307 \else
3308   \bxjs@robust@def\bxjs@wareki@used{%
3309     \global\let\bxjs@wareki@used\@empty
3310     \ClassWarning\bxjs@clsname
3311       {Wareki mode is not supported, since\MessageBreak
3312         'bxwareki' is unavailable, reported}}
3313     \g@addto@macro\bxjs@begin@document@hook{%
3314       \let\bxjs@wareki@used\@empty}
3315 \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。

```
3316 \ifbxjs@bxwareki@avail
3317   \let\jyear\warekiyear
3318   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3319     \newcount\heisei \heisei=\value{warekiyear}
3320 \fi
```

`bxwareki` が使えない場合は西暦表示にする。

```
3321 \else
3322   \edef\jyear{\the\year \bxjs@iai}
3323 \fi
```

`\today` 英語、西暦、和暦で場合分けをする。

```
3324 \let\bxjs@next\relax
3325 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3326   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3327 \fi\fi
3328 \edef\bxjs@today{%
3329   \if@english
3330     \ifcase\month\or
3331       January\or February\or March\or April\or May\or June\or
3332       July\or August\or September\or October\or November\or December\fi
3333     \space\number\day, \number\year
3334   \else
3335     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3336     \else \noexpand\bxjs@if@use@seireki
3337     \fi {%
3338       \number\year\bxjs@iai\bxjs@unxp{年}%
3339       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
```

```

3340     \bxjs@iaai\number\day\bxjs@iaai\bxjs@unxp{日}%
3341     }\bxjs@next}%
3342     \fi}
3343 \let\today\bxjs@today

```

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

```

3344 \g@addto@macro\bxjs@begin@document@hook{%
3345     \ifx\bb1@jpn@maybekansuji\@undefined\else
3346     \bxjs@decl@Seireki@cmds
3347     \g@addto@macro\datejapanese{%
3348     \let\today\bxjs@today}%
3349     \fi}

```

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

```

3350 \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。

```

3351 %<slide>\pagestyle{empty}%
3352 %<article|report>\pagestyle{plain}%
3353 %<book>\pagestyle{headings}%
3354 \pagenumbering{arabic}
3355 \if@twocolumn
3356     \twocolumn
3357     \sloppy
3358     \flushbottom
3359 \else
3360     \onecolumn
3361     \raggedbottom
3362 \fi
3363 %<*/slide>
3364 \renewcommand\familydefault{\sfdefault}
3365 \raggedright
3366 %</slide>

```

■BXJS 独自の追加処理 

フックを実行する。

```

3367 \bxjs@pre@jadriver@hook

```

和文ドライバのファイルを読み込む。

```

3368 \input{bxjsja-\bxjs@jadriver.def}

```

おしまい。

3369 %</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` `XqTeX`
 - `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 (呼び出すべき) マクロ。
- `\jsResetDimen` [マクロ] ★ 上記 `\jsFontSizeChanged` の中で呼び出される、ユーザ (和文モジュール) 用のフック。フォントサイズに依存するパラメタをここで設定することができる。既定の定義は空。

以下で標準で用意されている和文ドライバの実装を示す。

```
3370 %<*drv>
```

付録 B 和文ドライバ：minimal

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

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

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

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

B.1 補助マクロ

```
3371 %<*minimal>
```

```
3372 %% 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.
```

```
3373 \def\DeclareJaTextFontCommand#1#2{%
```

```
3374   \DeclareRobustCommand#1[1]{%
```

```
3375     \relax
```

```
3376     \ifmmode \expandafter\nfss@text \fi
```

```
3377     {#2##1}}%
```

```
3378 }
```

```
\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。
```

```
3379 \def\DeclareJaMathFontCommand#1#2{%
```

```
3380   \DeclareRobustCommand#1[1]{%
```

```
3381     \relax
```

```
3382     \ifmmode\else \non@alpherr{#1\space}\fi
```

```

3383 \nfss@text{\fontfamily\familydefault
3384 \fontseries{m}\fontshape{n}\selectfont\relax
3385 #2##1}%
3386 }%
3387 }

```

`\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。

```

3388 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3389 \onlypreamble\bxjs@if@sf@default
3390 \def\bxjs@if@sf@default#1{%
3391 \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3392 \g@addto@macro\bxjs@begin@document@hook{%
3393 \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3394 }

```

`\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。

```

3395 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3396 \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。

```

3397 \def\jsLetHeadChar#1#2{%
3398 \begingroup
3399 \escapechar=`\ %
3400 \let\bxjs@tmpa={% brace-match-hack
3401 \bxjs@let@hchar@exp#2}%
3402 \endgroup
3403 \let#1\bxjs@g@tmpa}
3404 \def\bxjs@let@hchar@exp{%
3405 \futurelet\@let@token\bxjs@let@hchar@exp@a}
3406 \def\bxjs@let@hchar@exp@a{%
3407 \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
3408 \bxjs@let@hchar@out\let\relax
3409 }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi{% 空白
3410 \bxjs@let@hchar@out\let\space%
3411 }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
3412 \bxjs@let@hchar@out\let\@backslashchar
3413 }{\bxjs@let@hchar@exp@b}}}}
3414 \def\bxjs@let@hchar@exp@b#1{%
3415 \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}

```

```

3416 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3417 %\message{<#1#2>}%
3418 \bxjs@cond\if#1\@backslashchar\fi{% 制御綴
3419 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3420 \bxjs@let@hchar@out\let\relax
3421 }{%else
3422 \expandafter\bxjs@let@hchar@exp
3423 }%
3424 }{%else
3425 \bxjs@let@hchar@chr#1%
3426 }}
3427 \def\bxjs@let@hchar@chr#1{%
3428 \bxjs@let@hchar@out\def{#{1}}}}
3429 \def\bxjs@let@hchar@out#1#2{%
3430 \global#1\bxjs@g@tmpa#2\relax
3431 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3432 \chardef\bxjs@let@hchar@csta=128
3433 \chardef\bxjs@let@hchar@cstb=192
3434 \chardef\bxjs@let@hchar@cstc=224
3435 \chardef\bxjs@let@hchar@cstd=240
3436 \chardef\bxjs@let@hchar@cste=248
3437 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3438 \def\bxjs@let@hchar@chr@ue#1{%
3439 \@tempcnta=#1\relax
3440 %\message{\the\@tempcnta}%
3441 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3442 \bxjs@let@hchar@chr@ue@a#1%
3443 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3444 \bxjs@let@hchar@out\let\relax
3445 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3446 \bxjs@let@hchar@chr@ue@b
3447 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3448 \bxjs@let@hchar@chr@ue@c
3449 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3450 \bxjs@let@hchar@chr@ue@d
3451 }{%else
3452 \bxjs@let@hchar@out\let\relax
3453 }}}}
3454 \def\bxjs@let@hchar@chr@ue@a#1{%
3455 \bxjs@let@hchar@out\def{#{1}}}}
3456 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3457 \bxjs@let@hchar@out\def{#{1#2}}}}
3458 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3459 \bxjs@let@hchar@out\def{#{1#2#3}}}}
3460 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3461 \bxjs@let@hchar@out\def{#{1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
3462 \ifx j\jsEngine
```

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

■補助マクロ `\jsLetHeadChar` を UTF-8 バイト列と和文文字トークンに対応させる。

```
3463 \def\bxjs@let@hchar@chr@pp#1#2{%
3464   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3465 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3466   %\message{(\meaning#3:\meaning#4)}%
3467   \bxjs@cond\if#1k\fi{%
3468     \bxjs@let@hchar@out\def{#4}}%
3469   }{%else
3470     \bxjs@let@hchar@chr@ue#3#4%
3471   }}
3472 \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。

```
3473 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3474 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3475 \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。

```
3476 \@onlypreamble\bxjs@declarefontshape
3477 \ifjsWithupTeX
3478 \def\bxjs@declarefontshape{%
3479   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3480   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3481   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3482   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3483 }
3484 \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。

```
3485 \else
3486 \def\bxjs@declarefontshape{%
3487   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%

```

```

3488 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3489 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
3490 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3491 }
3492 \def\bxjs@sizereference{jis}
3493 \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。

```

3494 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3495   \def\bxjs@tmpb{#5}}
3496 \ifjswithptexng \def\bxjs@tmpb{10}%
3497 \else
3498 \expandafter\expandafter\expandafter\bxjs@next
3499 \expandafter\string\the\jfont\relax
3500 \fi
3501 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3502                 \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3503   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3504    \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。

```

3505 \begingroup
3506 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3507 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3508 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3509 % 幅が丁度 10pt なら補正は不要
3510 \ifdim\wd\z@=10pt
3511   \global\let\bxjs@scale\jsScale
3512 \else
3513 %  $(10*s)/(10*f)$  として計算、\bxjs@invscale は BXJS で定義
3514   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3515   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3516   \bxjs@invscale\@tempdima\bxjs@tmpa
3517   \xdef\bxjs@scale{\strip@pt\@tempdima}
3518 \fi
3519 \endgroup
3520 %\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。

```

3521 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3522 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}

```

```

3523 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{
3524 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{
3525 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{
3526 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{
3527 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{
3528 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{
3529 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{
3530 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{
3531 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{
3532 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{
3533 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{
3534 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{
3535 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{
3536 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{
3537 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{
3538 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{
3539 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{
3540 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{
3541 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{
3542 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{
3543 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{
3544 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{
3545 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{
3546 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{
3547 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{
3548 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{
3549 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{
3550 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{
3551 \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。

```

3552 \@ifl@t@r\fmtversion{2020/10/01}
3553   {\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
3554 \ifjsc@needspace@tch          % --- for 2020-02-02 or older BEGIN
3555 \ifx\@rmfamilyhook\@undefined % old
3556 \DeclareRobustCommand\rmfamily
3557   {\not@math@alphabet\rmfamily\mathrm
3558   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3559 \DeclareRobustCommand\sffamily
3560   {\not@math@alphabet\sffamily\mathsf
3561   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3562 \DeclareRobustCommand\ttfamily
3563   {\not@math@alphabet\ttfamily\mathtt
3564   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3565 \g@addto@macro\bxjs@begin@document@hook{%
3566   \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3567     % my definitions above should have been overwritten, recover it!

```

```

3568 % \selectfont is executed twice but I don't care about speed...
3569 \expandafter\g@addto@macro\csname rmfamily \endcsname
3570   {\kanjifamily\mcdefault\selectfont}%
3571 \expandafter\g@addto@macro\csname sffamily \endcsname
3572   {\kanjifamily\gtdefault\selectfont}%
3573 \expandafter\g@addto@macro\csname ttfamily \endcsname
3574   {\kanjifamily\gtdefault\selectfont}%
3575 \fi}
3576 \else % 2020-02-02
3577 \g@addto@macro\@rmfamilyhook
3578   {\prepare@family@series@update@kanji{mc}\mcdefault}
3579 \g@addto@macro\@sffamilyhook
3580   {\prepare@family@series@update@kanji{gt}\gtdefault}
3581 \g@addto@macro\@ttfamilyhook
3582   {\prepare@family@series@update@kanji{gt}\gtdefault}
3583 \fi
3584 \else % --- for 2020-02-02 or older END & for 2020-10-01 BEGIN
3585 \AddToHook{rmfamily}%
3586   {\prepare@family@series@update@kanji{mc}\mcdefault}
3587 \AddToHook{sffamily}%
3588   {\prepare@family@series@update@kanji{gt}\gtdefault}
3589 \AddToHook{ttfamily}%
3590   {\prepare@family@series@update@kanji{gt}\gtdefault}
3591 \fi % --- for 2020-10-01 END
3592 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3593 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3594 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3595 \fi
3596 \bxjs@if@sf@default{%
3597   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```
3598 \selectfont
```

これ以降では、`\bxjs@parse@qh` の処理は p_TE_X 系では不要になるので無効化する（つまり `\jsSetQHLength` は `\setlength` と等価になる）。

```

3599 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3600 \let\bxjs@parse@qh@a\undefined
3601 \let\bxjs@parse@qh@b\undefined

```

■パラメタの設定

```

3602 \prebreakpenalty\jis"2147=10000
3603 \postbreakpenalty\jis"2148=10000
3604 \prebreakpenalty\jis"2149=10000
3605 \inhibitxspcode`!=1
3606 \inhibitxspcode`¯=2
3607 \xspcode`+=3
3608 \xspcode`%=3

```

"80~"FF の範囲の \spcode を 3 に変更。

```
3609 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3610   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
```

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

```
3611 \let\jsInhibitGlueAtParTop\@inhibitglue
```

\jsResetDimen は空のままよい。

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

```
3612 \begingroup
3613 \catcode`\!=0
```

\bxjs@ptex@dir 現在の組方向：t=縦、y=横、?=その他。

```
3614 \gdef\bxjs@ptex@dir{%
3615   !iftdir t%
3616   !else!ifydir y%
3617   !else ?%
3618   !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。

```
3619 % 古い \@makefnmark の定義
3620 \long\def\bxjs@tmpa{\hbox{%
3621   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3622   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3623 \ifx\@makefnmark\bxjs@tmpa
3624 \long\gdef\@makefnmark{%
3625   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3626   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3627 \fi
```

エスケープ文字の変更はここまで。

```
3628 \endgroup
```

■minijs パッケージのブロック やっておく。

```
3629 \@namedef{ver@minijs.sty}{}
```

B.3 pdfTeX 用の処理

```
3630 \else\if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
```

\jsLetHeadChar を UTF-8 バイト列に対応させる。

```
3631 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

ムニャムニャ。


```

3632 \@onlypreamble\bxjs@CJK@loaded
3633 \def\bxjs@CJK@loaded{%
3634   \def\@footnotemark{%
3635     \leavevmode
3636     \ifhmode
3637       \edef\@x@sf{\the\spacefactor}%
3638       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3639         \unkern\unkern
3640         \ifdim\lastskip>\z@ \unskip \fi
3641       \fi\fi
3642     \nobreak
3643   \fi
3644   \@makefnmark
3645   \ifhmode \spacefactor\@x@sf \fi
3646   \relax}%
3647 \let\bxjs@CJK@loaded\relax
3648 }
3649 \g@addto@macro\bxjs@begin@document@hook{%
3650   \@ifpackageloaded{CJK}{%
3651     \bxjs@CJK@loaded
3652   }{}%
3653 }

```

B.4 X₃TEX 用の処理

```
3654 \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。

```

3655 \def\bxjs@let@hchar@chr#1{%
3656   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3657   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3658     \bxjs@let@hchar@chr@xe
3659   }{\bxjs@let@hchar@out\def{#{1}}}}
3660 \def\bxjs@let@hchar@chr@xe#1{%
3661   \lccode`0=`#1\relax
3662   \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3663 \@onlypreamble\bxjs@do@precisetext
3664 \ifx\XeTeXgenerateactualtext\@undefined\else
3665   \def\bxjs@do@precisetext{%
3666     \XeTeXgenerateactualtext=\@ne}
3667 \fi

```

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

TODO: バージョン要件を見直して暫定措置を解除する。

```

3668 \@onlypreamble\bxjs@do@simplejasetup
3669 \def\bxjs@do@simplejasetup{%
3670   \@namedef{bxjs@zeroglue/0.0pt}{T}%

```

```

3671 \ifnum\XeTeXinterchartokenstate>\z@
3672 \else\expandafter\ifx\csname bxjs@zeroglue/\the\XeTeXlinebreakskip\endcsname\relax\else
3673 \jsSimpleJaSetup
3674 \ClassInfo\bxjs@clsname
3675 {\string\jsSimpleJaSetup' is applied@gobble}%
3676 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```

3677 \newcommand*{\jsSimpleJaSetup}{%
3678 \XeTeXlinebreaklocale "ja"\relax
3679 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3680 \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理 (エンジン共通)

```

3681 \fi\fi\fi

```

`simplejasetup` オプションの処理。

```

3682 \ifx\bxjs@do@simplejasetup\@undefined\else
3683 \g@addto@macro\bxjs@begin@document@hook{%
3684 \ifbxjs@simplejasetup
3685 \bxjs@do@simplejasetup
3686 \fi}
3687 \fi

```

`precisetext` オプションの処理。

```

3688 \ifbxjs@precisetext
3689 \ifx\bxjs@do@precisetext\@undefined
3690 \ClassWarning\bxjs@clsname
3691 {The current engine does not support the\MessageBreak
3692 'precise-text' option@gobble}
3693 \else
3694 \bxjs@do@precisetext
3695 \fi
3696 \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。

```

3697 \g@addto@macro\bxjs@begin@document@hook{%
3698 \ifx\jsInhibitGlueAtParTop\@empty
3699 \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3700 \ifx\everyparhook\bxjs@tmpa
3701 \let\everyparhook\@empty
3702 \fi
3703 \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。

```

3704 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@modern

```

まず `\everypar` を “乗っ取る” 処理を行う。

```
3705 \let\bxjs@everypar\everypar
3706 \newtoks\everypar
3707 \everypar\bxjs@everypar
```

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

```
3708 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3709 \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。

```
3710 \ifbxjs@fancyhdr
```

`\bxjs@adjust@fancyhdr` `fancyhdr` の初期設定に関する変更の処理。 `fancyhdr` 読み完了と `\pagestyle{fancy}` 実行の間で実行されるべき。

```
3711 \@onlypreamble\bxjs@adjust@fancyhdr
3712 \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 `\s1` は無い方がよいはず。

```
3713 \def\bxjs@tmpa{\fancyplain{}{\s1\rightmark}\strut}%
3714 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3715 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3716 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3717 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3718 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3719 \def\bxjs@tmpa{\fancyplain{}{\s1\leftmark}\strut}%
3720 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3721 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3722 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3723 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3724 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3725 \def\bxjs@tmpa{\rm\thepage\strut}%
3726 \def\bxjs@tmpb{\thepage\strut}%
3727 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3728 \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。

```
3729 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth
3730 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3731 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3732 }\bxjs@tmpa
3733 \fi\fi
```

```

3734 \PackageInfo\bxjs@clsname
3735 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3736 \def\bxjs@pagestyle@hook{%
3737 \ifpackageloaded{fancyhdr}{%
3738 \bxjs@adjust@fancyhdr
3739 \global\let\bxjs@adjust@fancyhdr\relax
3740 }{}}

```

`\pagestyle` にフックを入れ込む。

```

3741 \let\bxjs@org@pagestyle\pagestyle
3742 \def\pagestyle{%
3743 \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。

```

3744 \g@addto@macro\bxjs@begin@document@hook{%
3745 \bxjs@pagestyle@hook
3746 \global\let\bxjs@pagestyle@hook\relax}

```

終わり。

```

3747 \fi

```

■和文空白命令

```

3748 \ifbxjs@jaspace@cmd

```

`\jaenspace` 半角幅の水平空き。

```

3749 \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。

```

3750 \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。

```

3751 \def\_ {\zwspace}

```

`\jaspace` `jlreq` クラスと互換の命令。

```

3752 \DeclareRobustCommand*{\jaspace}[1]{%
3753 \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3754 \ClassError\bxjs@clsname
3755 {Unknown jaspac: #1}{\@eha}%
3756 \else
3757 \csname bxjs@jaspace@@#1\endcsname
3758 \fi}
3759 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3760 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3761 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}

```

終わり。

```
3762 \fi
```

以上で終わり。

```
3763 %</minimal>
```

付録 C 和文ドライバ：standard 🍡

標準のドライバ。

- `\rmfamily/\sffamily/\ttfamily` での和文ファミリー連動
- `\mcfamily/\gtfamily`
- `\textmc/\textgt`
- `\zw`
- `\jQ/\jH`
- `\trueQ/\trueH/\ascQ`
- `\setkanjiskip/\getkanjiskip`
- `\setxkanjiskip/\getxkanjiskip`
- `\autospacing/\noautospacing`
- `\autoxspacing/\noautoxspacing`

■和文フォント指定の扱い

C.1 準備

まず `minimal` ドライバを読み込む。

```
3764 %<*standard>
```

```
3765 %% このファイルは日本語文字を含みます
```

```
3766 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3767 \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。

```
3768 \newif\ifbxjs@jp@jismmiv
```

`jis2004` オプションの処理。

```
3769 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
```

```
3770 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
```

```
3771 \define@key{bxjsStd}{jis2004}[true]{%
```

```
3772 \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。

```
3773 \newif\ifbxjs@jp@units
```

units オプションの処理。

```
3774 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
```

```
3775 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
```

```
3776 \define@key{bxjsStd}{units}[true]{%
```

```
3777 \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。

```
3778 \let\bxjs@jp@font\@empty
```

font オプションの処理。

```
3779 \define@key{bxjsStd}{font}{%
```

```
3780 \edef\bxjs@jp@font{#1}}
```

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

```
3781 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue
```

strong-cmd オプションの処理。

```
3782 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
```

```
3783 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
```

```
3784 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}
```

実際の `japaram` の値を適用する。

```
3785 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
```

```
3786 \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。

```
3787 \@onlypreamble\bxjs@apply@mmiv
```

```
3788 \def\bxjs@apply@mmiv{%
```

```
3789 \g@addto@macro\@classoptionslist{,jis2004}
```

```
3790 % \ifpackagewith 判定への対策
```

```
3791 \PassOptionsToPackage{jis2004}{otf}
```

```
3792 \global\let\bxjs@apply@mmiv\relax}
```

```
3793 \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。

```
3794 \if j\jsEngine
```

```
3795 \bxjs@jp@unitsfalse
```

```
3796 \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。

```

3797 \ifbxjs@jp@units
3798   \IfFileExists{bxcalc.sty}{%
3799     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3800     \ifx\usepTeXunits\undefined
3801       \PackageWarningNoLine{bxjs@clsname
3802         {Cannot support pTeX units (zw etc.), since\MessageBreak
3803           the package 'bxcalc' is too old}}%
3804       \bxjs@jp@unitsfalse
3805     \else \usepTeXunits
3806     \fi
3807   }{%else
3808     \PackageWarningNoLine{bxjs@clsname
3809       {Cannot support pTeX units (zw etc.), since\MessageBreak
3810         the package 'bxcalc' is unavailable}}%
3811     \bxjs@jp@unitsfalse
3812   }
3813 \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。

```

3814 \ifbxjs@jp@units
3815 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3816 \let\bxjs@parse@qh@a\undefined
3817 \let\bxjs@parse@qh@b\undefined
3818 \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。

```

3819 \ifbxjs@jp@units
3820 \def\bxjs@let@lenexpr#1#2{%
3821   \edef#1{#2}%
3822   \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3823 \else
3824 \def\bxjs@let@lenexpr{\edef}
3825 \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 する。

※既に利用可能である場合は何もしない。

```

3826 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3827   \ifx\strong\undefined\ifx\strongenv\undefined
3828     \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3829     \DeclareTextFontCommand{\strong}{\strongenv}%

```

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

※`\strongfontdeclare` は試験的機能とする。

```
3830 \newcommand*{\strongfontdeclare}{\bxjs@strongfontdeclare}%
3831 \newcount\bxjs@strong@level
3832 \bxjs@protected\def\bxjs@strongfontdeclare#1{%
3833   \bxjs@set@array@from@clist{bxjs@strong}{#1}%
3834   \bxjs@strong@level\z0}%
3835 \bxjs@strongfontdeclare{\bfseries}%
3836 \def\bxjs@strong@font{%
3837   \bxjs@csletcs{bxjs@tmpa}{bxjs@strong/\the\bxjs@strong@level}%
3838   \ifx\bxjs@tmpa\relax
3839     \advance\bxjs@strong@level\m@ne \bxjs@strong@font
3840   \else \advance\bxjs@strong@level\@ne \bxjs@tmpa
3841   \fi}%
3842 \fi\fi
3843 }\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。

```
3844 \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 3845 \@tempdima=0.25mm
3846 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3847 \let\jH\jQ
```

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

```
\trueH 3848 \ifjsc@mag
3849   \@tempdimb=\jsBaseFontSize\relax
3850   \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3851   \@tempdima=2.5mm
3852   \bxjs@invscale\@tempdima\bxjs@tmpa
3853   \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3854   \@tempdima=10pt
3855   \bxjs@invscale\@tempdima\bxjs@tmpa
3856   \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3857   \else \let\trueQ\jQ \let\bxjs@truept\p@
3858   \fi
3859   \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。

```
3860 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3861 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3862 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
```



```

3863 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3864 \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。

```

3865 \def\jafontsize#1#2{%
3866   \begingroup
3867     \bxjs@jafontsize@a{#1}%
3868     \@tempdimb\jsInverseScale\@tempdima
3869     \bxjs@jafontsize@a{#2}%
3870     \xdef\bxjs@g@tmpa{%
3871       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3872   \endgroup\bxjs@g@tmpa}
3873 \def\bxjs@jafontsize@a#1{%
3874   \bxjs@parse@qh{#1}%
3875   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3876   \@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。

```

3877 \def\bxjs@kanjiskip{Opt}

```

`\setkanjiskip` 和文間空白の量を設定する。

```

3878 \newcommand*\setkanjiskip[1]{%
3879   \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3880   \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3881 \newcommand*\getkanjiskip{%
3882   \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。

```

3883 \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 3884 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3885   \bxjs@kanjiskip@enabledtrue
3886   \bxjs@reset@kanjiskip}
3887 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3888   \bxjs@kanjiskip@enabledfalse
3889   \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。

```

3890 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3891   \ifbxjs@kanjiskip@enabled
3892     \setlength{\@tempskipa}{\bxjs@kanjiskip}%

```

```

3893 \else \@tempskipa\z@
3894 \fi
3895 \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 3896 \def\bxjs@xkanjiskip{Opt}
\getxkanjiskip 3897 \newcommand*\setxkanjiskip[1]{%
\ifbxjs@xkanjiskip@enabled 3898 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
3899 \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3900 \newcommand*\getxkanjiskip{%
\bxjs@disable@xkanjiskip 3901 \bxjs@xkanjiskip}
3902 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3903 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3904 \bxjs@xkanjiskip@enabledtrue
3905 \bxjs@reset@xkanjiskip}
3906 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3907 \bxjs@xkanjiskip@enabledfalse
3908 \bxjs@reset@xkanjiskip}
3909 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3910 \ifbxjs@xkanjiskip@enabled
3911 \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3912 \else \@tempskipa\z@
3913 \fi
3914 \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。

```

3915 \g@addto@macro\jsResetDimen{%
3916 \bxjs@reset@kanjiskip
3917 \bxjs@reset@xkanjiskip}
3918 \let\bxjs@apply@kanjiskip\relax
3919 \let\bxjs@apply@xkanjiskip\relax

```

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

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

```

3920 \@onlypreamble\bxjs@adjust@jafont
3921 \def\bxjs@adjust@jafont#1{%
3922 \ifx\jsJaFont\bxjs@@auto
3923 \bxjs@get@kanjiEmbed
3924 \ifx\bxjs@jaEmbed\relax
3925 \let\bxjs@tmpa\@empty
3926 \else

```

```

3927     \let\bxjs@tmpa\bxjs@jaEmbed
3928     \ifx\bxjs@jaVariant\bxjs@zhziv
3929         \bxjs@apply@mmiv
3930     \fi
3931 \fi
3932 \else
3933     \let\bxjs@tmpa\jsJaFont
3934 \fi
3935 \if f#1\ifx\bxjs@tmpa\bxjs@noEmbed
3936     \ClassWarningNoLine\bxjs@clsname
3937     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3938     not available on the current situation}%
3939     \let\bxjs@tmpa@empty
3940 \fi\fi
3941 }
3942 \def\bxjs@@auto{auto}
3943 \def\bxjs@@noEmbed{noEmbed}
3944 \def\bxjs@@zhziv{-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。

```

3945 \let\bxjs@jaEmbed\relax
3946 \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。

```

3947 \@onlypreamble\bxjs@get@kanjiEmbed
3948 \def\bxjs@get@kanjiEmbed{%
3949     \begingroup\setbox\z@=\hbox{%
3950         \global\let\bxjs@tmpdo@empty
3951         \def\bxjs@next##1##2##3{%
3952             \def##1###1##3 ###2@nil###3@nnil{%
3953                 \ifx$###1$\gdef##2{###2}\fi}%
3954             \g@addto@macro\bxjs@tmpdo{%
3955                 \expandafter##1\bxjs@tmpa@nil##3 \nil\@nnil}}%
3956         \bxjs@next\bxjs@tmpdo@a\bxjs@g@tmpa{kanjiEmbed}%
3957         \bxjs@next\bxjs@tmpdo@b\bxjs@g@tmpa{jaEmbed}%
3958         \bxjs@next\bxjs@tmpdo@c\bxjs@g@tmpb{kanjiVariant}%
3959         \bxjs@next\bxjs@tmpdo@d\bxjs@g@tmpb{jaVariant}%
3960 %
3961         \global\let\bxjs@g@tmpa\relax
3962         \global\let\bxjs@g@tmpb\relax
3963         \endlinechar\m@ne
3964         \let\do@makeother\dospecials
3965         \catcode32=10 \catcode12=10 %form-feed
3966         \let\bxjs@tmpa@empty
3967         \openin@inputcheck="|kpsewhich updmap.cfg"\relax
3968         \ifeof@inputcheck\else
3969             \read@inputcheck to\bxjs@tmpa

```

```

3970     \closein\@inputcheck
3971     \fi
3972     \ifx\bxjs@tmpa\@empty\else
3973         \openin\@inputcheck="\bxjs@tmpa"\relax
3974         \@tempwattrue
3975         \loop\if@tempswa
3976             \read\@inputcheck to\bxjs@tmpa
3977             \bxjs@tmpdo
3978             \ifeof\@inputcheck \@tempwafalse \fi
3979         \repeat
3980     \fi
3981 } \endgroup
3982 \let\bxjs@jaEmbed\bxjs@g@tmpa
3983 \let\bxjs@jaVariant\bxjs@g@tmpb
3984 }

```

`\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。

```

3985 \@onlypreamble\bxjs@resolve@jafont@paren
3986 \def\bxjs@resolve@jafont@paren#1{%
3987     \def\bxjs@tmpb{\let#1}%
3988     \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\@nnil#1}
3989 \@onlypreamble\bxjs@resolve@jafont@paren@a
3990 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\@nnil#5{%
3991     \ifx\relax#4\relax \bxjs@tmpb#5%
3992     \else
3993         \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3994         \bxjs@tmpb\bxjs@tmpa
3995     \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。

```

3996 \newcommand*\jachar[1]{%
3997     \begingroup
3998         \jsLetHeadChar\bxjs@tmpa{#1}%
3999         \ifx\bxjs@tmpa\relax
4000             \ClassWarningNoLine\bxjs@clsname
4001             {Illegal argument given to \string\jachar}%
4002         \else
4003             \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
4004         \fi
4005     \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

4006 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
4007 \let\bxjs@jachar\@firstofone
```

■**hyperref 対策** 出力ページサイズに飽する処理は `geometry` パッケージが行うので、`hyperref` 側の処理は無効にしておく。

```
4008 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
4009 \@onlypreamble\bxjs@fix@hyperref@unicode
4010 \def\bxjs@fix@hyperref@unicode#1{%
4011   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
4012   \@namedef{KV@Hyp@bxjs/hook}##1{%
4013     \KV@Hyp@unicode{##1}%
4014     \def\KV@Hyp@unicode###1{%
4015       \expandafter\ifx\csname if##1\expandafter\endcsname
4016         \csname if###1\endcsname\else
4017         \ClassWarningNoLine\bxjs@clsname
4018         {Blcked hyperref option 'unicode=###1'}%
4019       \fi
4020     }%
4021   }%
4022 }
```

`\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。

```
4023 \@onlypreamble\jsCheckHyperrefUnicode
4024 \let\jsCheckHyperrefUnicode\@empty
4025 \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。

```
4026 \@onlypreamble\bxjs@check@hyperref@unicode
4027 \def\bxjs@check@hyperref@unicode#1{%
4028   \g@addto@macro\jsCheckHyperrefUnicode{%
4029     \@tempwafalse
4030     \begingroup
4031       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
4032         \aftergroup\@tempwattrue \fi
4033       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
4034         \csname if#1\endcsname
4035         \aftergroup\@tempwattrue \fi
4036     \endgroup
4037     \if@tempswa\else
4038       \ClassError\bxjs@clsname
4039       {The value of hyperref 'unicode' key is not suitable\MessageBreak
4040       for the present engine (must be #1)}%
4041       {\@ehc}%
4042     \fi}}}
```

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

```
4043 \@onlypreamble\bxjs@urgent@special
4044 \@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。

```
4045 \ifbxjs@old@hook@system
4046 \def\bxjs@urgent@special#1{%
4047   \AtBeginDvi{\special{#1}}%
4048   \g@addto@macro\bxjs@begin@document@hook{%
4049     \@ifpackageloaded{atbegshi}{%
4050       \begingroup
4051         \toks\z@\special{#1}}%
4052         \toks\tw@\expandafter{\AtBegShi@HookFirst}%
4053         \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
4054       \endgroup
4055     }{}%
4056   }%
4057 }
```

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。

```
4058 \else
4059   \def\bxjs@urgent@special#1{%
4060     \bxjs@urgent@special@a
4061     \AddToHook{shipout/firstpage}[pxjahyper/enc]{\special{#1}}
4062   \def\bxjs@urgent@special@a{%
4063     \DeclareHookRule{shipout/firstpage}{pxjahyper/enc}{<}{hyperref}%
4064     \global\let\bxjs@urgent@special@a\relax}
4065 \fi
```

C.4 pTeX 用設定

```
4066 \if j\jsEngine
```

■共通命令の実装

```
4067 \def\bxjs@apply@kanjiskip{%
4068   \kanjiskip\@tempkipa}
4069 \def\bxjs@apply@xkanjiskip{%
4070   \xkanjiskip\@tempkipa}
```

\jaJaChar のサブマクロ。

```
4071 \def\bxjs@jachar#1{%
4072   \bxjs@jachar@a#1...\@nil}
4073 \def\bxjs@jachar@a#1#2#3#4#5\@nil{%
```

引数が単一トークンなら和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。

```
4074   \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。

```

4075 \else\ifx.#3%
4076 \@tempcнта`#1 \multiply\@tempcнта64
4077 \advance\@tempcнта`#2 \advance\@tempcнта-"3080
4078 \bxjs@jachar@b
4079 \else\ifx.#4%
4080 \@tempcнта`#1 \multiply\@tempcнта64
4081 \advance\@tempcнта`#2 \multiply\@tempcнта64
4082 \advance\@tempcнта`#3 \advance\@tempcнта-"E2080
4083 \bxjs@jachar@b
4084 \else
4085 \@tempcнта`#1 \multiply\@tempcнта64
4086 \advance\@tempcнта`#2 \multiply\@tempcнта64
4087 \advance\@tempcнта`#3 \multiply\@tempcнта64
4088 \advance\@tempcнта`#4 \advance\@tempcнта-"3C82080
4089 \bxjs@jachar@b
4090 \fi\fi\fi}

```

符号値が \@tempcнта の和文文字を出力する処理。

```

4091 \ifjsWithupTeX
4092 \def\bxjs@jachar@b{\kchar\@tempcнта}
4093 \else
4094 \def\bxjs@jachar@b{%
4095 \ifx\bxUInt\@undefined\else
4096 \bxUInt{\@tempcнта}%
4097 \fi}
4098 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

4099 \ifbxjs@jaspace@cmd
4100 \def\jathinspace{\hskip\xkanjiskip}
4101 \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。

```

4102 \ifbxjs@jp@jismmiv
4103 \PassOptionsToPackage{prefer2004jis}{pxchfon}
4104 \PassOptionsToPackage{prefer2004jis}{pxbabel}
4105 \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。

```

4106 \let\bxjs@jafont@paren\@firstofone
4107 \let\bxjs@tmpa\jsJaFont
4108 \ifx\bxjs@tmpa\bxjs@@auto
4109 \let\bxjs@tmpa\@empty
4110 \else\ifx\bxjs@tmpa\bxjs@@noEmbed

```

```

4111 \def\bxjs@tmpa{noembed}
4112 \fi\fi
4113 \bxjs@resolve@jafont@paren\bxjs@tmpa
4114 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4115 \ifx\bxjs@tmpa\@empty\else
4116 \edef\bxjs@next{%
4117 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4118 }\bxjs@next
4119 \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……。)

```

4120 \begingroup
4121 \global\let\bxjs@g@tmpa\relax
4122 \catcode`\|=0 \catcode`\|=12
4123 |def|bxjs@tmpdo#1|@nil{%
4124 |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4125 |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4126 |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4127 |catcode`\|=0 \catcode`\|=12
4128 \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4129 \ifx$#2$\else
4130 \xdef\bxjs@g@tmpa{%
4131 \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4132 \fi}
4133 \@firstofone{%
4134 \catcode10=12 \endlinechar\m@ne
4135 \let\do@makeoother \dospecials \catcode32=10
4136 \openin\@inputcheck=otf.sty\relax
4137 \@tempwattrue
4138 \loop\if@tempswa
4139 \ifeof\@inputcheck \@tempwafalse \fi
4140 \if@tempswa
4141 \read\@inputcheck to\bxjs@next
4142 \expandafter\bxjs@tmpdo\bxjs@next\@nil
4143 \fi
4144 \repeat
4145 \closein\@inputcheck
4146 \endgroup}
4147 \bxjs@g@tmpa

```

■**hyperref 対策** unicode にしてはいけない。

```

4148 \ifbxjs@hyperref@enc
4149 \bxjs@check@hyperref@unicode{false}

```


暫定的なナニカ。

```
4150 \ifjswithupteX\ifbxjs@old@hook@system\else
4151   \IfFileExists{pxjahyper-uni.def}{%
4152     \AddToHook{\bxjs@CGHN{package/hyperref/after}}{\input{pxjahyper-
      uni.def}}
4153   }{}
4154 \fi\fi
4155 \fi
```

tounicode special 命令を出力する。

```
4156 \if \ifx\bxjs@driver@given\bxjs@driver@dvipdfmx T%
4157   \else\ifjswithptexng T\else F\fi\fi T%
4158 \IfFileExists{pxjahyper-enc.sty}{\@tempwattrue}{\@tempwafalse}
4159 \if@tempswa
4160   \RequirePackage{pxjahyper-enc}[2020/10/05]%v0.6
4161   \ifbxjs@bigcode\else \suppressbigcode \fi
4162 \else
4163 \ifnum\jis"2121"=A1A1 %euc
4164   \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4165 \else\ifnum\jis"2121"=8140 %sjis
4166   \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4167 \else\ifnum\jis"2121"=3000 %uptex
4168   \ifbxjs@bigcode
4169     \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4170     \PassOptionsToPackage{bigcode}{pxjahyper}
4171   \else
4172     \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4173     \PassOptionsToPackage{nobigcode}{pxjahyper}
4174   \fi
4175 \fi\fi\fi
4176 \let\bxToUnicodeSpecialDone=t
4177 \fi
4178 \fi
```

■和文数式ファミリ 和文数式ファミリは既定で有効とする。すなわち `enablejfam=false` 以外の場合は `@enablejfam` を真にする。

```
4179 \ifx f\bxjs@enablejfam\else
4180   \@enablejfamtrue
4181 \fi
```

実際に和文用の数式ファミリの設定を行う。

```
4182 \if@enablejfam
4183   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4184   \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4185   \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4186   \jfam\symmincho
4187   \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4188   \@addto@macro\bxjs@begin@document@hook{%
4189     \ifx\reDeclareMathAlphabet\@undefined\else
```

```

4190 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4191 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4192 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4193 \fi}
4194 \fi

```

C.5 pdfTeX 用設定：CJK + bxcjkjatype

```

4195 \else\if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T

```

■**bxcjkjatype** パッケージの読込 `\jsJaFont` が指定されている場合は、その値を `bxcjkjatype` のオプション（プリセット指定）に渡す。ここで値が `auto` である場合は `\bxjs@get@kanjiEmbed` を実行する。スケール値 (`\jsScale`) の反映は `bxcjkjatype` の側で行われる。

※ Pandoc モードでは `autotilde` を指定しない。

```

4196 \bxjs@adjust@jafont{f}
4197 \let\bxjs@jafont@paren\@firstofone
4198 \bxjs@resolve@jafont@paren\bxjs@tmpa
4199 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4200 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4201 \ifx\bxjs@jadriver\bxjs@@pandoc\else
4202 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4203 \fi
4204 \edef\bxjs@next{%
4205 \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkjatype}[2013/10/15]% v0.2c
4206 }\bxjs@next
4207 \bxjs@cjk@loaded

```

■**hyperref 対策** `bxcjkjatype` 使用時は `unicode` にするべき。

```

4208 \ifbxjs@hyperref@enc
4209 \PassOptionsToPackage{unicode}{hyperref}
4210 \fi

```

`\hypersetup` 命令で (CJK* 環境に入れなくても) 日本語文字を含む文書情報を設定できるようにするための細工。

※ `bxcjkjatype` を `whole` 付きで使っていることが前提。

※ パッケージオプションでの指定に対応するのは、「アクティブな高位バイトトークンがその場で展開されてしまう」ため困難である。

```

4211 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
4212 \begingroup
4213 \CJK@input{UTF8.bdg}
4214 \endgroup
4215 \g@addto@macro\pdfstringdefPreHook{%
4216 \@nameuse{CJK@UTF8Binding}%
4217 }
4218 \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。

```

4219 \ifx\bxckjatypeHyperrefPatchDone\@undefined
4220 \g@addto@macro\pdfstringdefPreHook{%
4221   \ifx~\bxjs@@CJKtilde
4222     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4223     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4224     \let~\@empty
4225   \fi
4226 }
4227 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
4228 \def\bxjs@tildecmd{~}
4229 \def\bxjs@LetUnexpandableSpace#1{%
4230   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@tildecmd\else
4231     \bxjs@org@LetUnexpandableSpace#1%
4232   \fi}
4233 \fi

```

■共通命令の実装

```

4234 \newskip\jskanjiskip
4235 \newskip\jsXkanjiskip
4236 \ifx\CJKecglue\@undefined
4237   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4238 \fi
4239 \let\autospacing\bxjs@enable@kanjiskip
4240 \let\noautospacing\bxjs@disable@kanjiskip
4241 \protected\def\bxjs@CJKglue{\hskip\jskanjiskip}
4242 \def\bxjs@apply@kanjiskip{%
4243   \jskanjiskip\@tempskipa
4244   \let\CJKglue\bxjs@CJKglue}
4245 \let\autoxspacing\bxjs@enable@xkanjiskip
4246 \let\noautoxspacing\bxjs@disable@xkanjiskip
4247 \protected\def\bxjs@CJKecglue{\hskip\jsXkanjiskip}
4248 \def\bxjs@apply@xkanjiskip{%
4249   \jsXkanjiskip\@tempskipa
4250   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

4251 \def\bxjs@jachar#1{%
4252   \CJKforced{#1}}

```

和欧文間空白の命令 \jathinspace の実装。

```

4253 \ifbxjs@jaspace@cmd
4254   \protected\def\jathinspace{\CJKecglue}
4255 \fi

```

■和文数式ファミリ CJK パッケージは（恐らく）数式文字として CJK 文字をサポートしていない。従って @enablejfam は常に偽になる。

```

4256 \ifx t\bxjs@enablejfam
4257   \ClassWarningNoLine\bxjs@clsname
4258   {You cannot use 'enablejfam=true', since the\MessageBreak

```

```
4259 CJK package does not support Japanese math}
4260 \fi
```

C.6 X_ƎTeX 用設定：xeCJK + zxjatype

```
4261 \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。

```
4262 \RequirePackage{zxjatype}
4263 \PassOptionsToPackage{no-math}{fontspec}%!
4264 \PassOptionsToPackage{xetex}{graphicx}%!
4265 \PassOptionsToPackage{xetex}{graphics}%!
4266 \ifx\zxJaFamilyName\undefined
4267 \ClassError\bxjs@clsname
4268 {xeCJK or zxjatype is too old}\@ehc
4269 \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。

```
4270 \bxjs@adjust@jafont{f}
4271 \let\bxjs@jafont@paren@gobble
4272 \bxjs@resolve@jafont@paren\bxjs@tmpa
4273 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4274 \ifx\bxjs@tmpa\@empty
4275 \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
Regular.otf}
4276 \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
Medium.otf}
4277 \else
4278 \edef\bxjs@next{%
4279 \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4280 }\bxjs@next
4281 \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。

※ TeX の小数の精度は十進で 4 桁までしか保証されないので、\stricmp を利用して文字列で比較している。(整数部が多桁になっても大丈夫。) しかし実は、\stricmp プリミティブが

追加されたのは 0.9994 版 (2009 年 6 月) かららしい。

TODO: バージョン要件を見直して暫定措置を解除する。

```
4282 \ifx\strcmp\@undefined\else % 未定義なら条件を満たさない
4283 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4284 \ifbxjs@hyperref@enc
4285 \PassOptionsToPackage{unicode}{hyperref}
4286 \fi
4287 \fi
4288 \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。

```
4289 \AtEndOfPackage{%
4290 \def\@gnewline #1{%
4291 \ifvmode \@nolnerr
4292 \else
4293 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
4294 \nobreak \hskip-1sp\hskip1sp\relax
4295 \ignorespaces
4296 \fi}
4297 }
```

■共通命令の実装

```
4298 \newskip\jsKanjiSkip
4299 \newskip\jsXKanjiSkip
4300 \ifx\CJKecglue\@undefined
4301 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4302 \fi
4303 \let\autospacing\bxjs@enable@kanjiskip
4304 \let\noautospacing\bxjs@disable@kanjiskip
4305 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4306 \def\bxjs@apply@kanjiskip{%
4307 \jsKanjiSkip\@tempkipa
4308 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4309 \let\autoxspacing\bxjs@enable@xkanjiskip
4310 \let\noautoxspacing\bxjs@disable@xkanjiskip
4311 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4312 \def\bxjs@apply@xkanjiskip{%
4313 \jsXKanjiSkip\@tempkipa
4314 \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。

```

4315 \ifx\mcfamily\@undefined
4316 \protected\def\mcfamily{\CJKfamily{\CJKrmddefault}}
4317 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4318 \fi

```

`\jachar` のサブマクロの実装。

```

4319 \def\bxjs@jachar#1{%
4320 \xeCJKDeclareCharClass{CJK}{`#1}\relax
4321 #1}

```

`\jathinspace` の実装。

```

4322 \ifbxjs@jaspace@cmd
4323 \protected\def\jathinspace{\CJKecglue}
4324 \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。

```

4325 \ifx t\bxjs@enablejfam
4326 \@enablejfamtrue
4327 \fi

```

実際に和文用の数式ファミリの設定を行う。

※ FIXME: 要検討。

```

4328 \if@enablejfam
4329 \xeCJKsetup{CJKmath=true}
4330 \fi

```

C.7 LuaTeX 用設定：LuaTeX-ja

```

4331 \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。

```

4332 \let\zw\@undefined
4333 \RequirePackage{luatexja}
4334 \edef\bxjs@next{%
4335 \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4336 }\bxjs@next

```

フォント代替の明示的定義。

```

4337 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{-}
4338 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
4339 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{-}

```

```

4340 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4341 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4342 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4343 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4344 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4345 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4346 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4347 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4348 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4349 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4350 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4351 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4352 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4353 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4354 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4355 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4356 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4357 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4358 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4359 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4360 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4361 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4362 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4363 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4364 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4365 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4366 \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。

```

4367 \bxjs@adjust@jafont{t}
4368 \ifx\bxjs@tmpa\bxjs@@noEmbed
4369   \def\bxjs@tmpa{noembed}
4370 \fi
4371 \let\bxjs@jafont@paren@gobble
4372 \bxjs@resolve@jafont@paren\bxjs@tmpa
4373 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4374 \ifx\bxjs@tmpa@empty
4375   \defaultjfontfeatures{ Kerning=Off }
4376   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4377   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4378 \else
4379   \edef\bxjs@next{%
4380     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%

```

```
4381 } \bxjs@next
4382 \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。

```
4383 \@ifpackagelater{luatexja}{2016/03/31}{\%else
4384 \DeclareRobustCommand\rmfamily
4385   {\not@math@alphabet\rmfamily\mathrm
4386     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4387 \DeclareRobustCommand\sffamily
4388   {\not@math@alphabet\sffamily\mathsf
4389     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4390 \DeclareRobustCommand\ttfamily
4391   {\not@math@alphabet\ttfamily\mathtt
4392     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4393 }
4394 \long\def\jttdefault{\gtdefault}
4395 \unless\ifx\@ltj@match@familytrue\undefined
4396   \@ltj@match@familytrue
4397 \fi
4398 \g@addto@macro\bxjs@begin@document@hook{%
4399   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4400   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4401   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4402 \bxjs@if@sf@default{%
4403   \renewcommand\kanjifamilydefault{\gtdefault}}
```

■和文パラメタの設定

```
4404 % 次の3つは既定値の通り
4405 \ltjsetparameter{prebreakpenalty={`',10000}}
4406 \ltjsetparameter{postbreakpenalty={`",10000}}
4407 \ltjsetparameter{prebreakpenalty={`",10000}}
4408 \ltjsetparameter{jaxspmode={`!,1}}
4409 \ltjsetparameter{jaxspmode={`¯,2}}
4410 \ltjsetparameter{alxspmode={`+,3}}
4411 \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。

```
4412 \ifx\ltjfakeparbegin\undefined
4413   \protected\def\ltjfakeparbegin{%
4414     \ifhmode
4415       \relax\directlua{%
4416         luatexja.jfmglue.create_beginpar_node()}}
4417   \fi}
```


4418 \fi

ltjs* クラスの定義と同等になるようにパッチを当てる。

```
4419 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none
4420 \begingroup
4421   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4422   \@gobble\if\def\bxjs@tmpa{\@{\everypar}\fi}
4423   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
4424     \@gobble\if\def\bxjs@tmpa{\@{\everypar\everyparhook}\fi}\fi
4425   \directlua{
4426     local function patchcmd(cs, code, from, to)
4427       tex.sprint(code:gsub(from:gsub("%w", "%\\%\\%0"), "%0"..to)
4428         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1)..")
4429     end
4430     patchcmd(\@xsect, [[\meaning\xsect]],
4431       \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4432     patchcmd(\@item, [[\meaning\@item]],
4433       \bxjs@tmpa, \@{\ltjfakeparbegin})
4434   \endgroup
4435 \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。

```
4436 \ifbxjs@hyperref@enc
4437   \PassOptionsToPackage{unicode}{hyperref}
4438   \bxjs@check@hyperref@unicode{true}
4439 \fi
```

■共通命令の実装

```
4440 \protected\def\autospacing{%
4441   \ltjsetparameter{autospacing=true}}
4442 \protected\def\noautospacing{%
4443   \ltjsetparameter{autospacing=false}}
4444 \protected\def\autoxspacing{%
4445   \ltjsetparameter{autoxspacing=true}}
4446 \protected\def\noautoxspacing{%
4447   \ltjsetparameter{autoxspacing=false}}
4448 \def\bxjs@apply@kanjiskip{%
4449   \ltjsetparameter{kanjiskip={\@tempskipa}}}
4450 \def\bxjs@apply@xkanjiskip{%
4451   \ltjsetparameter{xkanjiskip={\@tempskipa}}}
```

\jachar のサブマクロの実装。

```
4452 \def\bxjs@jachar#1{%
4453   \ltjjachar`#1\relax}
```

\jathinspace の実装。

```
4454 \ifbxjs@jaspace@cmd
4455   \protected\def\jathinspace{%
```

```
4456 \hskip\ltjgetparameter{xkanjiskip}\relax}
4457 \fi
```

■和文数式ファミリー LuaTeX-ja 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。

```
4458 \ifx f\bxjs@enablejfam
4459 \ClassWarningNoLine\bxjs@clsname
4460 {You cannot use 'enablejfam=false', since the\MessageBreak
4461 LuaTeX-ja always provides Japanese math families}
4462 \fi
```

C.8 共通処理 (2)

```
4463 \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 4464 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4465 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4466 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4467 \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。

```
4468 \ifx\mathmc\undefined
4469 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4470 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4471 \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)。

```
4472 \ifbxjs@jaspace@cmd
4473 \bxjs@protected\def\bxjs@choice@jathinspace{%
4474 \relax\ifmmode \mskip\medmuskip
4475 \else \jathinspace\ignorespaces
4476 \fi}
4477 \jsAtEndOfClass{%
4478 \ifjsWitheTeX \let\>\bxjs@choice@jathinspace
4479 \else \def\>{\protect\bxjs@choice@jathinspace}%
4480 \fi}
4481 \fi
```

■和文・和欧文間空白の初期値

```
4482 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
```

```

4483 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4484 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4485 \fi

```

以上で終わり。

```
4486 %</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。

```

4487 %<*modern>
4488 \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。

```

4489 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4490 \def\encodingdefault{T1}%
4491 \input{t1enc.def}%
4492 \fontencoding\encodingdefault\selectfont
4493 \fi

```

基本フォントを Latin Modern フォントファミリーに変更する。

※以下は `\usepackage[noamth]{lmodern}` と同じ。ユーザは後で `lmodern` を好きなオプションを付けて読み込むことができる。

```

4494 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4495 \renewcommand{\rmdefault}{lmr}
4496 \renewcommand{\sfdefault}{lmss}
4497 \renewcommand{\ttdefault}{lmtt}
4498 \fi

```

大型演算子用の数式フォントの設定。

※`amsmaths` パッケージと同等にする。

```

4499 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4500 <-7.5>cmex7<7.5-8.5>cmex8%
4501 <8.5-9.5>cmex9<9.5->cmex10}{}%
4502 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

```

`amsmath` 読込時に上書きされるのを防ぐ。

```
4503 \def\cmex@opt{10}
```

D.2 fixltx2e 読込

※`fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```

4504 \ifx\@IncludeInRelease\@undefined
4505 \RequirePackage{fixltx2e}
4506 \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。

```

4507 \RequirePackage{bxjscjkat}

```

D.4 完了

おしまい。

```

4508 %</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。

```

4509 %<*pandoc>
4510 \input{bxjsja-standard.def}

bxjspandoc パッケージを読み込む。
4511 \RequirePackage{bxjspandoc}

 $\epsilon$ -TeX ではない場合に警告を出す。
※近い将来に  $\epsilon$ -TeX 拡張を必須にする予定。
4512 \ifjswitheTeX\else
4513   \ClassWarningNoLine{bxjs@clsname}
4514   {!!!!!!!!!! WARNING !!!!!!!!!!\MessageBreak}
4515   This engine does not support  $\epsilon$ -TeX extension!\MessageBreak
4516   Some feature might not work properly}
4517 \fi

```

`\ifbxjs@bxghost@available` [スイッチ] `bxghost` パッケージが利用できるか。

```

4518 \newif\ifbxjs@bxghost@available
4519 \ifjswitheTeX
4520   \RequirePackage{pdftexcmds}[2009/09/22]% v0.5
4521   \IfFileExists{bxghost.sty}{%
4522     \bxjs@bxghost@availabletrue
4523     \@namedef{bxjs@bgbv/79E70A0991967E27981832C84DB5DF99}{1}%v0.2.0
4524     \ifx\pdf@filemddfivesum\@undefined\else

```

```

4525     \expandafter\ifx\csname bxjs@bgbv/\pdf@filemdfivesum{bxghost.sty}%
4526         \endcsname\relax\else \bxjs@bxghost@availablefalse \fi
4527     \fi
4528 }{}
4529 \fi

```

`\bxjs@endpreamble@hook` etoolbox の `\AtEndPreamble` で実行される BXJS クラス用のフック。

※ ε -TeX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```

4530 \@onlypreamble\bxjs@endpreamble@hook
4531 \let\bxjs@endpreamble@hook\empty

```

パッケージ読込。

```

4532 \RequirePackage{iftex}[2013/04/04]% v0.2
4533 \ifjswitheTeX
4534     \RequirePackage{etoolbox}[2010/08/21]% v2.0
4535     \AtEndPreamble{\bxjs@endpreamble@hook}
4536     \RequirePackage{filehook}[2011/10/12]% v0.5d
4537 \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。

```

4538 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue

```

`fix-strong` オプションの処理。

```

4539 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
4540 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
4541 \define@key{bxjsPan}{fix-strong}[true]{%
4542     \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。

```

4543 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue

```

`fix-code` オプションの処理。

```

4544 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
4545 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
4546 \define@key{bxjsPan}{fix-code}[true]{%
4547     \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。

```

4548 \chardef\bxjs@jp@strong=0

```

`strong` オプションの処理。

```

4549 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
4550 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
4551 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }

```

```
4552 \define@key{bxjsPan}{strong}{%
4553   \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 4554 \newif\ifbxjs@jp@or@indent \bxjs@jp@or@indenttrue
\ifbxjs@jp@or@block@heading 4555 \newif\ifbxjs@jp@or@secnumdepth \bxjs@jp@or@secnumdepthtrue
4556 \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。

```
4557 \define@key{bxjsPan}{_plus}[][%
4558   \bxjs@jp@or@indentfalse
4559   \bxjs@jp@or@secnumdepthfalse
4560   \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。

```
4561 \let\bxjs@kv@orindent@true\bxjs@jp@or@indenttrue
4562 \let\bxjs@kv@orindent@false\bxjs@jp@or@indentfalse
4563 \define@key{bxjsPan}{or-indent}[true]{%
4564   \bxjs@set@keyval{orindent}{#1}{}}
4565 \let\bxjs@kv@orsecnumdepth@true\bxjs@jp@or@secnumdepthtrue
4566 \let\bxjs@kv@orsecnumdepth@false\bxjs@jp@or@secnumdepthfalse
4567 \define@key{bxjsPan}{or-secnumdepth}[true]{%
4568   \bxjs@set@keyval{orsecnumdepth}{#1}{}}
4569 \let\bxjs@kv@orblockheading@true\bxjs@jp@or@block@headingtrue
4570 \let\bxjs@kv@orblockheading@false\bxjs@jp@or@block@headingfalse
4571 \define@key{bxjsPan}{or-block-heading}[true]{%
4572   \bxjs@set@keyval{blockheading}{#1}{}}
```

実際の `japaram` の値を適用する。

```
4573 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}
4574 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

E.3 dupload システム

パッケージが重複して読み込まれたときに “option clash” の検査をスキップする。この時に何らかのコードを実行させることができる。

`\bxjs@set@dupload@proc` `\bxjs@set@dupload@proc{〈ファイル名〉}{〈定義本体〉}`： 指定の名前の特定のファイルの読み込みが `\@filewithoptions` で指示されて、しかもそのファイルが読み込み済である場合に、オプション重複検査をスキップして、代わりに `〈定義本体〉` のコードを実行する。このコード中で `#1` は渡されたオプション列のテキストに置換される。

```
4575 \@onlypreamble\bxjs@set@dupload@proc
4576 \def\bxjs@set@dupload@proc#1{%
4577   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```

4578 \@onlypreamble\bxjs@set@dupload@proc@a
4579 \def\bxjs@set@dupload@proc@a#1{%
4580   \@onlypreamble#1\def#1##1}
4581 \def\bxjs@unset@dupload@proc#1{%
4582   \bxjs@cslet{bxjs@dlp/#1}\@undefined}

```

\@if@options \@if@options の再定義。

```

4583 \@onlypreamble\bxjs@org@if@options
4584 \let\bxjs@org@if@options\@if@options
4585 \@onlypreamble\bxjs@org@reset@options
4586 \let\bxjs@org@reset@options\relax
4587 \def\@if@options#1#2#3{%
4588   \let\bxjs@next\@secondoftwo
4589   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4590   \ifx\bxjs@tmpa\bxjs@tmpb
4591     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4592     \let\bxjs@next\@firstoftwo \fi
4593   \fi
4594   \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4595 \g@addto@macro\bxjs@begin@document@hook{%
4596   \let\@if@options\bxjs@org@if@options}
4597 \@onlypreamble\bxjs@do@dupload@proc
4598 \def\bxjs@do@dupload@proc#1#2#3{%
4599   \ifx\bxjs@org@reset@options\relax
4600     \let\bxjs@org@reset@options\@reset@options
4601   \fi
4602   \bxjs@csletcs{bxjs@next}{bxjs@dlp/#2.#1}%
4603   \def\@reset@options{%
4604     \let\@reset@options\bxjs@org@reset@options
4605     \@reset@options
4606     \bxjs@next{#3}}%
4607   \@firstoftwo}

```

E.4 lang 変数

lang=ja という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐための対策を行う。

※ Pandoc 2.12 版で lang=ja 指定に対応し、正しく L^AT_EX 側の言語名 `japanese` に変換されるようになった。しかし、日本語指定の場合は相変わらず調整処理が必要である。

\bxjs@polyglossia@options Polyglossia のオプション列のテキスト。“実際には読み込まれていない”場合は \relax になる。

```
4608 \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。

```
4609 \let\bxjs@babel@options\relax
```

■Polyglossia について つまり Xe_{La}TeX および Lua_{TeX} (古い Pandoc で) の場合。

※この場合 etoolbox が使用可能になっている。

```
4610 \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。

```
4611 \pandocSkipLoadPackage{polyglossia}
4612 \bxjs@set@dupload@proc{polyglossia.sty}{%
4613   \bxjs@unset@dupload@proc{polyglossia.sty}%
4614   \ClassWarning\bxjs@clsname
4615   {Package polyglossia is requested}%
4616   \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。

```
4617   \newcommand*\setmainlanguage[2] []{%
```

もし、言語名が空の \setmainlanguage{} が実行された場合は、lang=ja が指定されたと見なす。言語名が japanese だった場合も同様。

```
4618     \ifboolexpr{test{\ifblank{##2}}or test{\ifstrequal{##2}{japanese}}}{%
4619       \ClassWarning\bxjs@clsname
4620       {Main language is 'japanese', thus fallback\MessageBreak
4621         definitions will be employed}%
4622       \bxjs@pandoc@polyglossia@ja
```

それ以外は、改めて polyglossia を読み込んで、本来の処理を実行する。

```
4623     }{%else
4624       \ClassWarning\bxjs@clsname
4625       {Main language is '##2',\MessageBreak
4626         thus polyglossia will be loaded}%
4627       \csundef{ver@polyglossia.sty}%
4628       \edef\bxjs@next{%
4629         \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4630       }\bxjs@next
4631       \setmainlanguage[##1]{##2}%
4632     }}}
```

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。

```
4633 \g@addto@macro\bxjs@endpreamble@hook{%
4634   \ifx\bxjs@polyglossia@options\relax
4635     \csundef{ver@polyglossia.sty}%
4636   \fi}
```

\bxjs@pandoc@polyglossia@ja Pandoc 側で lang=ja が指定されていた場合の処理。この場合は Polyglossia の処理を無効化するためにダミーの定義を行う。その時点でダミーの \setotherlanguage(s) を定義する。

※現在では Polyglossia の日本語用の定義ファイル (gloss-japanese.1df) が存在するので、本来なら普通に処理できるはずであるが、現状の定義ファイルはアレなので回避したい。

```
4637 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4638 \def\bxjs@pandoc@polyglossia@ja{%
4639   \renewcommand*\setmainlanguage[2] []{%
4640     \newcommand*\setotherlanguage[2] []{%
4641       \ifblank{##2}{}{%else
4642         \cslet{##2}\@empty \cslet{end##2}\@empty
4643         \cslet{text##2}\@firstofone}}%
4644     \newcommand*\setotherlanguages[2] []{%
4645       \@for\bxjs@tmpa:={##2}\do{%
4646         \setotherlangauge{\bxjs@tmpa}}}%
```

Polyglossia の読込済マークは外れるようにしておく。

```
4647 \let\bxjs@polyglossia@options\relax}%
4648 \fi
```

■Babel について Xe_{La}TeX 以外の場合。

※ Pandoc 2.15 版から、テンプレートで用いられる多言語パッケージが Babel に統一された。(Lua_{TeX} は 2.6 版で Polyglossia から Babel に変更されている。)

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。

```
4649 \pandocSkipLoadPackage{babel}
4650 \bxjs@set@dupload@proc{babel.sty}{%
4651   \bxjs@unset@dupload@proc{babel.sty}%
4652   \ClassWarning\bxjs@clsname
4653   {Package babel is requested}}%
```

パッケージオプションに言語名が空の main= があるかを調べる。ある場合は lang=ja 対策を実行する。

※\bxjs@babel@options には main= を除いたオプション列を格納する。

```
4654 \@tempwafalse \let\bxjs@babel@options\@empty
4655 \def\bxjs@tmpb{main=}
4656 \def\bxjs@next{main=japanese}%
4657 \@for\bxjs@tmpa:=#1\do{%
4658   \ifx\bxjs@tmpa\bxjs@tmpb \@tempwatru
4659   \else\ifx\bxjs@tmpa\bxjs@next \@tempwatru
4660   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4661   \fi\fi}%
4662 \if@tempswa
4663   \ClassWarning\bxjs@clsname
4664   {Main language is 'japanese', thus fallback\MessageBreak
4665     definitions will be employed}%
4666   \bxjs@pandoc@babel@ja
```

ない場合は、本来の babel の処理を実行する。

```
4667 \else
4668   \ClassWarning\bxjs@clsname
```

```

4669     {Main language is not 'japanese',\MessageBreak
4670     thus babel will be loaded}%
4671     \bxjs@cslet{ver@babel.sty}\@undefined
4672     \RequirePackage[#1]{babel}[]%
4673     \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。

```

4674 \g@addto@macro\bxjs@endpreamble@hook{%
4675     \ifx\bxjs@babel@options\relax
4676     \bxjs@cslet{ver@babel.sty}\@undefined
4677     \fi}

```

`\bxjs@pandoc@babel@ja` Pandoc 側で `lang=ja` が指定されていた場合の処理。

```

4678 \@onlypreamble\bxjs@pandoc@babel@ja
4679 \def\bxjs@pandoc@babel@ja{%
4680     \bxjs@cslet{ver@babel.sty}\@undefined
4681     \edef\bxjs@next{%
4682         \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4683     }\bxjs@next
4684     \if j\jsEngine
4685         \RequirePackage[main=japanese]{pxbabel}[]%
4686     \else
4687         \RequirePackage{bxorigcapt}[]%
4688     \fi}

```

`lang` 対策はこれで終わり。

E.5 geometry 変数

`geometry` を “再度読み込んだ” 場合に、そのパラメタで `\setpagelayout*` が呼ばれるようにする。

```

4689 \bxjs@set@dupload@proc{geometry.sty}{%
4690     \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 `\setmainjfont` の指定にまわす。

```

4691 \if l\jsEngine
4692     \pandocSkipLoadPackage{xeCJK}
4693     \providecommand*\setCJKmainfont{\setmainjfont}
4694 \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。

```
4695 \if x\jsEngine
4696   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4697     ,space}
4698 \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。

```
4699 \ifjsWitheTeX
4700 \@onlypreamble\bxjs@info@or@ban
4701 \def\bxjs@info@or@ban#1{%
4702   \PackageInfo\bxjs@clsname
4703   {Freeze layout on '#1',\MessageBreak reported}}
```

■**indent** について indent 変数を指定しない場合に「段落表現形式をインデント方式に変更する」動作を抑止する。

```
4704 \unless\ifbxjs@jp@or@indent
4705   \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。

```
4706 \IfFileExists{parskip.sty}{%
4707   \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。

```
4708 }{%else
4709   \eappto\bxjs@endpreamble@hook{%
4710     \parindent=\the\parindent\relax
4711     \parskip=\the\parskip\relax}}
4712 \fi
```

■**secnumdepth** について secnumdepth の値を決めるのは numbersections 変数 (-N/--number-sections オプションに連動する) や secnumdepth 変数であるが、何れにしても secnumdepth の値は書き換えられる。そのため、secnumdepth を復帰させる。

```
4713 \ifbxjs@jp@or@secnumdepth\else
4714   \bxjs@info@or@ban{secnumdepth}
4715   \eappto\bxjs@endpreamble@hook{%
4716     \c@secnumdepth=\the\c@secnumdepth\relax}
4717 \fi
```

■**block-heading** について \paragraph、\subparagraph を別行見出しに変える処理を抑止する。

※ 2.7.1 版以前では別行見出し変更が既定で有効であった。

```
4718 \ifbxjs@jp@or@block@heading\else
4719   \let\bxjs@frozen@paragraph\paragraph
4720   \let\bxjs@frozen@subparagraph\subparagraph
```

```

4721 \bxjs@info@or@ban{block-heading}
4722 \appto\bxjs@endpreamble@hook{%
4723 \let\oldparagraph\undefined
4724 \let\paragraph\bxjs@frozen@paragraph
4725 \let\subparagraph\bxjs@frozen@subparagraph}
4726 \fi

```

以上。

```
4727 \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。

```

4728 \g@addto@macro\bxjs@begin@document@hook{%
4729 \@tempwafalse

```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```

4730 \ifx\oldparagraph\undefined\else
4731 \@tempwatrue
4732 \fi

```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```

4733 \ifnum\c@secnumdepth>3
4734 \@tempwatrue
4735 \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。

```

4736 \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4737 \let\jsParagraphMark\empty
4738 \fi\fi}

```

E.10 全角空白文字

L^AT_EX でない入力では、全角空きを入れるために全角空白文字 (U+3000) が使われる可能性があるので、全角空白文字を和文文字でなく空きとして扱うようにしておく。

※ (u)pL^AT_EX では対応できないので対象外。

`\pandocZWSpace` 全角空白文字の入力で実行されるコード。

```
4739 \def\pandocZWSpace{\zwspace}
```

全角空白文字の入力で `\pandocZWSpace` が実行されるようにする。

```
4740 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4741 \catcode"3000=\active
4742 \begingroup \catcode`\!=7
4743 \protected\gdef!!!3000{\pandocZWSpace}
4744 \endgroup
4745 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4746 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4747 \bxjs@protected\def\bxjs@zsp@char{\pandocZWSpace}
4748 \fi\fi
```

E.11 hyperref 対策

`hyperref` の `unicode` オプションの固定を行う。

```
4749 \if j\jsEngine
4750 \bxjs@fix@hyperref@unicode{false}
4751 \else
4752 \bxjs@fix@hyperref@unicode{true}
4753 \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 $\text{L}^{\text{T}}\text{E}^{\text{X}}$ 出力は `\textbf` となるが、代わりに `\strong` を使いたいため、`\textbf` を書き換えてしまう (うわあ)。

```
4754 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
4755 \let\orgtextbf\textbf
4756 \DeclareRobustCommand\pandocTextbf[1]{%
4757 \begingroup
4758 \let\textbf\orgtextbf
4759 \strong{#1}%
4760 \endgroup}%
4761 \g@addto@macro\bxjs@begin@document@hook{%
4762 \let\textbf\pandocTextbf}
4763 \fi\fi
```

`\strong` の書体を設定する。

```
4764 \jsAtEndOfClass{%
4765 \ifx\strongfontdeclare\@undefined\else
4766 \ifcase\bxjs@jp@strong
4767 \or \strongfontdeclare{\sffamily}%
4768 \or \strongfontdeclare{\sffamily\bfseries}%
4769 \fi
4770 \fi}
```

■**インラインコード要素** インラインコード (Code) 要素に対する L^AT_EX 出力は `\texttt` となる。`\texttt` の両端に欧文ゴーストが入るようにする。さらに `\verb` の外側にも欧文ゴーストが入るようにする。

```
4771 \ifbxjs@jp@fix@code
```

`bxghost` パッケージが利用できる場合はその機能を利用する。使えない場合は自前実装を用いる。

```
4772 \ifbxjs@bxghost@available
4773   \RequirePackage[verb]{bxghost}[2020/01/31]% v0.3.0
4774   \let\bxjs@eghostguarded\eghostguarded
4775   \else
4776   \chardef\bxjs@eghost@c=23
4777   \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4778   \else\ifx l\jsEngine \ltjsetparameter{alxspmode={\bxjs@eghost@c,3}}
4779   \else\ifx x\jsEngine %no-op
4780   \else \let\bxjs@eghost@c\undefined
4781   \fi\fi\fi
4782   \ifx\bxjs@eghost@c\undefined\else
4783     \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4784     \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4785     \def\bxjs@eghostguarded#1{%
4786       \bxjs@pan@eghost\null#1\null\bxjs@pan@eghost}
4787     \fi
4788     \fi
4789     \ifx\bxjs@eghostguarded\@undefined\else
4790       \let\orgtexttt\texttt
4791       \DeclareRobustCommand\pandocTexttt[1]{%
4792         \ifmmode \nfss@text{\ttfamily #1}%
4793         \else
4794           \ifvmode \leavevmode \fi
4795           \bxjs@eghostguarded{\begingroup\ttfamily#1\endgroup}%
4796         \fi}
4797       \g@addto@macro\bxjs@begin@document@hook{%
4798         \let\texttt\pandocTexttt}
```

`bxghost` を使わない場合の `\verb` の処理。

※`bxghost` の実装を参考にした。

```
4799   \ifbxjs@bxghost@available\else
4800   \expandafter\def\expandafter\verb\expandafter{%
4801     \expandafter\bxjs@pan@eghost\verb}
4802   \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4803   \fi
4804   \fi
4805 \fi
```

E.13 ifPDFTeX スイッチ

Pandoc モードでは Pandoc の既定テンプレートを (無理やり) (u)pTeX に対応させることを目的にしている。

旧版のテンプレートでは `ifxetex` と `ifluatex` パッケージを読み込んだ上で「XeTeX でも LuaTeX でもないものは pdfTeX」という前提の動作をしていた。よって、(u)pTeX に対応させる際には「pdfTeX 用の処理が実行される」ことを前提にすればよかった。

ところが、Pandoc の 2.12 版では `iftex` パッケージが導入されて「pdfTeX の判定を直接 `\ifPDFTeX` で行う」ように改修された。このため、(u)pTeX での実行でどのコードが実行されるかを予測することが困難になってしまった。

これに対処するため、「文書ファイルのプレアンブル実行中に限って `\ifPDFTeX` が (実際とは異なり) 真になるようにする」という細工を施すことで、従来通り「pdfTeX 用の処理が実行される」前提が維持されるようにする。

```
4806 \if j\jsEngine
```

```
\bxjs@check@frontier \bxjs@check@frontier\CS は現在のパッケージ読込ネストレベルが丁度 1 であるときにのみ \CS を実行する。
```

```
4807 \def\bxjs@check@frontier{%
4808   \expandafter\bxjs@check@frontier@a\@currnamestack\noindent...\@nil}
4809 \def\bxjs@check@frontier@a#1#2#3#4#5\@nil#6{%
4810   \ifx\noindent#4#6\fi}
```

```
\bxjs@unforge@ifPDFTeX \ifPDFTeX を偽 (正しい値) にする。
```

```
4811 \@onlypreamble\bxjs@unforge@ifPDFTeX
4812 \def\bxjs@unforge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iffalse}}
```

```
\bxjs@forge@ifPDFTeX \ifPDFTeX を真 (偽装した値) にする。
```

```
4813 \@onlypreamble\bxjs@forge@ifPDFTeX
4814 \def\bxjs@forge@ifPDFTeX{\global\bxjs@csletcs{ifPDFTeX}{iftrue}}
```

```
\bxjs@unload@forge@ifPDFTeX \ifPDFTeX に対する細工を無効化する。
```

```
4815 \def\bxjs@unload@forge@ifPDFTeX{%
4816   \bxjs@unforge@ifPDFTeX
4817   \global\let\bxjs@check@frontier\@gobble}
```

プレアンブル開始時は `\ifPDFTeX` は真で、終了時に偽装を無効化する。filehook のフックで「パッケージ読込中は偽装を解除する」ことを実現している。

```
4818 \jsAtEndOfClass{\bxjs@forge@ifPDFTeX}
4819 \ifjswitheTeX
4820   \AtBeginOfEveryFile{\bxjs@check@frontier\bxjs@unforge@ifPDFTeX}%
4821   \AtEndOfEveryFile{\bxjs@check@frontier\bxjs@forge@ifPDFTeX}%
4822   \g@addto@macro\bxjs@endpreamble@hook{\bxjs@unload@forge@ifPDFTeX}
4823 \else
4824   \g@addto@macro\bxjs@begin@document@hook{\bxjs@unload@forge@ifPDFTeX}
4825 \fi
4826 \fi
```

E.14 完了

おしまい。

```
4827 %</pandoc>
```

和文ドライバ実装はここまで。

```
4828 %</drv>
```

付録 F 補助パッケージ一覧 🍷

BXJS クラスの機能を実現するために用意されたものだが、他のクラスの文書で読み込んで利用することもできる。

- `bxjscompat` : ムニャムニャムニャ。
- `bxjscjkat` : modern ドライバ用の和文カテゴリを適用する。
- `bxjspandoc` : Pandoc 用のナニカ。

```
4829 %<*anc>
```

付録 G 補助パッケージ : `bxjscompat` 🍷

古いやつをどうにかするためのムニャムニャ。

G.1 準備

```
4830 %<*compat>
```

```
4831 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
4832 \let\bxac@engine=n
```

```
4833 \def\bxac@do#1#2{%
```

```
4834   \edef\bxac@tmpa{\string#1}%
```

```
4835   \edef\bxac@tmpb{\meaning#1}%
```

```
4836   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
4837 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
4838 \bxac@do\luatexversion{\let\bxac@engine=l}
```

`\bxac@delayed@if@bxjs` もし BXJS クラスの読込中でこのパッケージが読み込まれているならば、BXJS のクラスの終わりまで実行を遅延する。

```
4839 \ifx\jsAtEndOfClass\undefined
```

```
4840   \let\bxac@delayed@if@bxjs\@firstofone
```

```
4841 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
4842 \fi
```

`\ImposeOldLuaTeXBehavior` `\ImposeOldLuaTeXBehavior` は 0.85 版以降の LuaTeX を一時的に pdfTeX と互換である `\RevokeOldLuaTeXBehavior` ように見せかける。`\RevokeOldLuaTeXBehavior` で元に戻すことができる。

※エンジンが LuaTeX 以外の場合は何もしない。

```
4843 \newif\ifbxac@in@old@behavior
4844 \let\ImposeOldLuaTeXBehavior\relax
4845 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 XeTeX 部分

```
4846 \ifx x\bxac@engine
```

■文字クラスの設定 XeTeX の文字クラス (`\XeTeXcharclass`) の Unicode 規定に基づく設定は、初期の版ではフォーマットに組み込まれていたが、2016/02/01 以降の L^AT_EX カーネルでは「必要に応じて後から設定用のファイルを読み込む」方式に変更された。ここでは「設定されている状態」を担保する。

※ちなみに、XeTeX に「文字間トークン挿入」の機能が導入されたのは 0.997 版 (2007 年頃) からのようだ。

ただし xeCJK が読込済ならば (そちらが適切に設定しているはずなので) 何もしない。

```
4847 \ifx\XeTeXcharclass\@undefined\else
4848 \bxac@delayed@if@bxjs{%
4849   \@ifpackageloaded{xeCJK}{-}{%else
      設定が未実行の状態ならば、設定用のファイルを読む。
4850   \ifx\XeTeXcharclass\@undefined\else
4851     \ifnum\XeTeXcharclass=\z@
4852     \PackageInfo\bxac@pkgname
4853       {Setting up interchar class for CJK...\@gobble}%
4854     \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4855       \XeTeXcharclass=3
4856     }{%else
4857     \PackageWarning\bxac@pkgname
4858       {Cannot find file 'load-unicode-xetex-classes.tex'%
4859       \@gobble}%
4860     }%
4861   \fi\fi
```

フォーマット組込だった時代の設定は不完全なところがあるので補正する。

```
4862   \ifnum\XeTeXcharclass"3041=\z@
4863   \PackageInfo\bxac@pkgname
4864     {Adjusting interchar class for CJK...\@gobble}%
4865   \@for\bxac@tmpb:={%
4866     3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4867     3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4868     30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4869     31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4870     31FF%
4871   }\do{\XeTeXcharclass"\bxac@tmpb=\@ne}%
4872 \fi
4873 }%
4874 }
```

```
4875 \fi
```

漢字および完成形ハングルのカテゴリコードが確実に 11 になっているようにする。

```
4876 \chardef\bxac@tmpb=11
4877 \def\bxac@do#1#2{%
4878   \@tempcnta=#1\relax
4879   \unless\ifnum\catcode\@tempcnta=\bxac@tmpb
4880     \chardef\bxac@tmpa=#2\relax
4881     \@whilenum{\@tempcnta<\bxac@tmpa}\do{%
4882       \catcode\@tempcnta\bxac@tmpb \advance\@tempcnta@ne}%
4883   \fi}
4884 \bxac@do{"4E00}{"9FCD}
```

以上。

```
4885 \fi
```

G.3 LuaTeX 部分

```
4886 \ifx l\bxac@engine
```

0.82~0.84 版の LuaTeX を (0.81 版以前と同様に) 「pdfTeX の拡張である」 ように見せかける処理。

※恐らく必要な場面はなかったと思われるので、外しておく。

```
4887 %\unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4888 % \chardef\pdftexversion=200
4889 % \def\pdftexrevision{0}
4890 % \let\pdftexbanner\luatexbanner
4891 %\fi\fi
```

`\ImposeOldLuaTeXBehavior` 0.85 版以降であるかを検査する。

```
\RevokeOldLuaTeXBehavior 4892 \begingroup\expandafter\expandafter\expandafter\endgroup
4893 \expandafter\ifx\csname outputmode\endcsname\relax\else
```

該当する場合、以下の 5 つの pdfTeX 拡張プリミティブを復帰させることになる。

```
4894 \def\bxac@ob@list{%
4895   \do{\let}\pdfoutput{\outputmode}%
4896   \do{\let}\pdfpagewidth{\pagewidth}%
4897   \do{\let}\pdfpageheight{\pageheight}%
4898   \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
4899   \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}%
4900 \def\bxac@ob@do#1#2{\begingroup
4901   \expandafter\bxac@ob@do@a\csname bxac@string#2\endcsname{#1}#2}
4902 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4903   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4904   \else \let#3#1\relax \let#1\@undefined
4905   \fi}
4906 \protected\def\ImposeOldLuaTeXBehavior{%
4907   \unless\ifbxac@in@old@behavior
4908     \bxac@in@old@behaviortrue
4909     \let\do\bxac@ob@do \bxac@ob@list
```

```

4910 \fi}
4911 \protected\def\RevokeOldLuaTeXBehavior{%
4912 \ifbxac@in@old@behavior
4913 \bxac@in@old@behaviorfalse
4914 \let\do\bxac@ob@do \bxac@ob@list
4915 \fi}
4916 \fi

```

漢字および完成形ハングルのカテゴリコードが確実に 11 になっているようにする。

```

4917 \directlua{
4918   local function range(cs, ce, cc, ff)
4919     if ff or not tex.getcatcode(cs) == cc then
4920       local setcc = tex.setcatcode
4921       for c = cs, ce do setcc(c, cc) end
4922     end
4923   end
4924   range(0x3400, 0x4DB5, 11, false)
4925   \ifnum\luatexversion>64
4926     range(0x4DB5, 0x4DBF, 11, true)
4927     range(0x4E00, 0x9FCC, 11, false)
4928     range(0x9FCD, 0x9FFF, 11, true)
4929     range(0xAC00, 0xD7A3, 11, false)
4930     range(0x20000, 0x2A6D6, 11, false)
4931     range(0x2A6D7, 0x2A6FF, 11, true)
4932     range(0x2A700, 0x2B734, 11, false)
4933     range(0x2B735, 0x2B73F, 11, true)
4934     range(0x2B740, 0x2B81D, 11, false)
4935     range(0x2B81E, 0x2B81F, 11, true)
4936     range(0x2B820, 0x2CEA1, 11, false)
4937     range(0x2CEA2, 0x2FFFD, 11, true)
4938   \fi
4939 }

```

以上。

```
4940 \fi
```

G.4 完了

おしまい。

```
4941 %</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

```

4942 %<*cjkcat>
4943 \def\bxjx@pkgname{bxjscjkcat}
4944 \newcount\bxjx@canta
4945 \@onlypreamble\bxjx@tmpdo
4946 \@onlypreamble\bxjx@tmpdo@a
4947 \@onlypreamble\bxjx@tmpdo@b

```

`\bxjx@engine` エンジンの種別。

```

4948 \let\bxjx@engine=n
4949 \def\bxjx@tmpdo#1#2{%
4950   \edef\bxjx@tmpa{\string#1}%
4951   \edef\bxjx@tmpb{\meaning#1}%
4952   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4953 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4954 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4955 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4956 \bxjx@tmpdo\pdftexversion{\let\bxjx@engine=p}
4957 \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。

```

4958 \def\bxjx@tmpdo#1#2{%
4959   \if#1\bxjx@engine
4960     \@ifpackageloaded{#2}{-}{%else
4961       \PackageError\bxjx@pkgname
4962         {Package '#2' must be loaded}%
4963         {Package loading is aborted.\MessageBreak\@ehc}%
4964       \endinput}
4965   \fi}
4966 \bxjx@tmpdo{p}{bxcjkatype}
4967 \bxjx@tmpdo{x}{xeCJK}
4968 \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。

```

4969 \ifx\TextOrMath\@undefined
4970   \RequirePackage{fixltx2e}
4971 \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。

```

4972 \if u\bxjx@engine
4973 \@for\bxjx@tmpa:={%
4974 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4975 0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4976 0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%

```

```

4977 0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4978 1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4979 1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4980 1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4981 2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4982 A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4983 A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4984 AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%
4985 FE70,FF00,%
4986 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4987 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4988 10480,104B0,10500,10530,10600,10800,10840,10860,%
4989 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4990 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4991 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4992 11200,11280,112B0,11300,11400,11480,11580,11600,%
4993 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4994 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4995 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4996 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4997 1EE00,1F000,1F030,1F0A0,1F300,1F600,1F650,1F680,%
4998 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4999 00C0%
5000 }\do{%
5001 \@tempcnta="\bxjx@tmpa\relax
5002 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
5003 \chardef\bxjx@tmpb\kcatcode\@tempcntb
5004 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
5005 \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。

```

5006 \@onlypreamble\bxjx@grkcyr@list
5007 \def\bxjx@grkcyr@list{%
5008 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
5009 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
5010 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA

```

5011	$\backslash\text{do}\{0394\}\{\text{LGR}\}\{\backslash\text{textDelta}\}\{\Delta\}\%$	% GR. C. L. DELTA
5012	$\backslash\text{do}\{0395\}\{\text{LGR}\}\{\backslash\text{textEpsilon}\}\{\text{E}\}\%$	% GR. C. L. EPSILON
5013	$\backslash\text{do}\{0396\}\{\text{LGR}\}\{\backslash\text{textZeta}\}\{\text{Z}\}\%$	% GR. C. L. ZETA
5014	$\backslash\text{do}\{0397\}\{\text{LGR}\}\{\backslash\text{textEta}\}\{\text{H}\}\%$	% GR. C. L. ETA
5015	$\backslash\text{do}\{0398\}\{\text{LGR}\}\{\backslash\text{textTheta}\}\{\Theta\}\%$	% GR. C. L. THETA
5016	$\backslash\text{do}\{0399\}\{\text{LGR}\}\{\backslash\text{textIota}\}\{\text{I}\}\%$	% GR. C. L. IOTA
5017	$\backslash\text{do}\{039A\}\{\text{LGR}\}\{\backslash\text{textKappa}\}\{\text{K}\}\%$	% GR. C. L. KAPPA
5018	$\backslash\text{do}\{039B\}\{\text{LGR}\}\{\backslash\text{textLambda}\}\{\Lambda\}\%$	% GR. C. L. LAMDA
5019	$\backslash\text{do}\{039C\}\{\text{LGR}\}\{\backslash\text{textMu}\}\{\text{M}\}\%$	% GR. C. L. MU
5020	$\backslash\text{do}\{039D\}\{\text{LGR}\}\{\backslash\text{textNu}\}\{\text{N}\}\%$	% GR. C. L. NU
5021	$\backslash\text{do}\{039E\}\{\text{LGR}\}\{\backslash\text{textXi}\}\{\Xi\}\%$	% GR. C. L. XI
5022	$\backslash\text{do}\{039F\}\{\text{LGR}\}\{\backslash\text{textOmicron}\}\{\text{O}\}\%$	% GR. C. L. OMICRON
5023	$\backslash\text{do}\{03A0\}\{\text{LGR}\}\{\backslash\text{textPi}\}\{\Pi\}\%$	% GR. C. L. PI
5024	$\backslash\text{do}\{03A1\}\{\text{LGR}\}\{\backslash\text{textRho}\}\{\text{P}\}\%$	% GR. C. L. RHO
5025	$\backslash\text{do}\{03A3\}\{\text{LGR}\}\{\backslash\text{textSigma}\}\{\Sigma\}\%$	% GR. C. L. SIGMA
5026	$\backslash\text{do}\{03A4\}\{\text{LGR}\}\{\backslash\text{textTau}\}\{\text{T}\}\%$	% GR. C. L. TAU
5027	$\backslash\text{do}\{03A5\}\{\text{LGR}\}\{\backslash\text{textUpsilon}\}\{\Upsilon\}\%$	% GR. C. L. UPSILON
5028	$\backslash\text{do}\{03A6\}\{\text{LGR}\}\{\backslash\text{textPhi}\}\{\Phi\}\%$	% GR. C. L. PHI
5029	$\backslash\text{do}\{03A7\}\{\text{LGR}\}\{\backslash\text{textChi}\}\{\text{X}\}\%$	% GR. C. L. CHI
5030	$\backslash\text{do}\{03A8\}\{\text{LGR}\}\{\backslash\text{textPsi}\}\{\Psi\}\%$	% GR. C. L. PSI
5031	$\backslash\text{do}\{03A9\}\{\text{LGR}\}\{\backslash\text{textOmega}\}\{\Omega\}\%$	% GR. C. L. OMEGA
5032	$\backslash\text{do}\{03B1\}\{\text{LGR}\}\{\backslash\text{textalpha}\}\{\alpha\}\%$	% GR. S. L. ALPHA
5033	$\backslash\text{do}\{03B2\}\{\text{LGR}\}\{\backslash\text{textbeta}\}\{\beta\}\%$	% GR. S. L. BETA
5034	$\backslash\text{do}\{03B3\}\{\text{LGR}\}\{\backslash\text{textgamma}\}\{\gamma\}\%$	% GR. S. L. GAMMA
5035	$\backslash\text{do}\{03B4\}\{\text{LGR}\}\{\backslash\text{textdelta}\}\{\delta\}\%$	% GR. S. L. DELTA
5036	$\backslash\text{do}\{03B5\}\{\text{LGR}\}\{\backslash\text{textepsilon}\}\{\epsilon\}\%$	% GR. S. L. EPSILON
5037	$\backslash\text{do}\{03B6\}\{\text{LGR}\}\{\backslash\text{textzeta}\}\{\zeta\}\%$	% GR. S. L. ZETA
5038	$\backslash\text{do}\{03B7\}\{\text{LGR}\}\{\backslash\text{texteta}\}\{\eta\}\%$	% GR. S. L. ETA
5039	$\backslash\text{do}\{03B8\}\{\text{LGR}\}\{\backslash\text{texttheta}\}\{\theta\}\%$	% GR. S. L. THETA
5040	$\backslash\text{do}\{03B9\}\{\text{LGR}\}\{\backslash\text{textiota}\}\{\iota\}\%$	% GR. S. L. IOTA
5041	$\backslash\text{do}\{03BA\}\{\text{LGR}\}\{\backslash\text{textkappa}\}\{\kappa\}\%$	% GR. S. L. KAPPA
5042	$\backslash\text{do}\{03BB\}\{\text{LGR}\}\{\backslash\text{textlambda}\}\{\lambda\}\%$	% GR. S. L. LAMDA
5043	$\backslash\text{do}\{03BC\}\{\text{LGR}\}\{\backslash\text{textmu}\}\{\mu\}\%$	% GR. S. L. MU
5044	$\backslash\text{do}\{03BD\}\{\text{LGR}\}\{\backslash\text{textnu}\}\{\nu\}\%$	% GR. S. L. NU
5045	$\backslash\text{do}\{03BE\}\{\text{LGR}\}\{\backslash\text{textxi}\}\{\xi\}\%$	% GR. S. L. XI
5046	$\backslash\text{do}\{03BF\}\{\text{LGR}\}\{\backslash\text{textomicron}\}\{\text{o}\}\%$	% GR. S. L. OMICRON
5047	$\backslash\text{do}\{03C0\}\{\text{LGR}\}\{\backslash\text{textpi}\}\{\pi\}\%$	% GR. S. L. PI
5048	$\backslash\text{do}\{03C1\}\{\text{LGR}\}\{\backslash\text{textrho}\}\{\rho\}\%$	% GR. S. L. RHO
5049	$\backslash\text{do}\{03C2\}\{\text{LGR}\}\{\backslash\text{textvarsigma}\}\{\text{varsigma}\}\%$	% GR. S. L. FINAL SIGMA
5050	$\backslash\text{do}\{03C3\}\{\text{LGR}\}\{\backslash\text{textsigma}\}\{\sigma\}\%$	% GR. S. L. SIGMA
5051	$\backslash\text{do}\{03C4\}\{\text{LGR}\}\{\backslash\text{texttau}\}\{\tau\}\%$	% GR. S. L. TAU
5052	$\backslash\text{do}\{03C5\}\{\text{LGR}\}\{\backslash\text{textupsilon}\}\{\upsilon\}\%$	% GR. S. L. UPSILON
5053	$\backslash\text{do}\{03C6\}\{\text{LGR}\}\{\backslash\text{textphi}\}\{\phi\}\%$	% GR. S. L. PHI
5054	$\backslash\text{do}\{03C7\}\{\text{LGR}\}\{\backslash\text{textchi}\}\{\chi\}\%$	% GR. S. L. CHI
5055	$\backslash\text{do}\{03C8\}\{\text{LGR}\}\{\backslash\text{textpsi}\}\{\psi\}\%$	% GR. S. L. PSI
5056	$\backslash\text{do}\{03C9\}\{\text{LGR}\}\{\backslash\text{textomega}\}\{\omega\}\%$	% GR. S. L. OMEGA
5057	$\backslash\text{do}\{0401\}\{\text{T2A}\}\{\backslash\text{CYRYO}\}\{\}\%$	% CY. C. L. IO
5058	$\backslash\text{do}\{0410\}\{\text{T2A}\}\{\backslash\text{CYRA}\}\{\}\%$	% CY. C. L. A
5059	$\backslash\text{do}\{0411\}\{\text{T2A}\}\{\backslash\text{CYRB}\}\{\}\%$	% CY. C. L. BE

5060 \do{0412}{T2A}{\CYRV}{}	% CY. C. L. VE
5061 \do{0413}{T2A}{\CYRG}{}	% CY. C. L. GHE
5062 \do{0414}{T2A}{\CYRD}{}	% CY. C. L. DE
5063 \do{0415}{T2A}{\CYRE}{}	% CY. C. L. IE
5064 \do{0416}{T2A}{\CYRZH}{}	% CY. C. L. ZHE
5065 \do{0417}{T2A}{\CYRZ}{}	% CY. C. L. ZE
5066 \do{0418}{T2A}{\CYRI}{}	% CY. C. L. I
5067 \do{0419}{T2A}{\CYRISHRT}{}	% CY. C. L. SHORT I
5068 \do{041A}{T2A}{\CYRK}{}	% CY. C. L. KA
5069 \do{041B}{T2A}{\CYRL}{}	% CY. C. L. EL
5070 \do{041C}{T2A}{\CYRM}{}	% CY. C. L. EM
5071 \do{041D}{T2A}{\CYRN}{}	% CY. C. L. EN
5072 \do{041E}{T2A}{\CYRO}{}	% CY. C. L. O
5073 \do{041F}{T2A}{\CYRP}{}	% CY. C. L. PE
5074 \do{0420}{T2A}{\CYRR}{}	% CY. C. L. ER
5075 \do{0421}{T2A}{\CYRS}{}	% CY. C. L. ES
5076 \do{0422}{T2A}{\CYRT}{}	% CY. C. L. TE
5077 \do{0423}{T2A}{\CYRU}{}	% CY. C. L. U
5078 \do{0424}{T2A}{\CYRF}{}	% CY. C. L. EF
5079 \do{0425}{T2A}{\CYRH}{}	% CY. C. L. HA
5080 \do{0426}{T2A}{\CYRC}{}	% CY. C. L. TSE
5081 \do{0427}{T2A}{\CYRCH}{}	% CY. C. L. CHE
5082 \do{0428}{T2A}{\CYRSH}{}	% CY. C. L. SHA
5083 \do{0429}{T2A}{\CYRSHCH}{}	% CY. C. L. SHCHA
5084 \do{042A}{T2A}{\CYRHRDSN}{}	% CY. C. L. HARD SIGN
5085 \do{042B}{T2A}{\CYRERY}{}	% CY. C. L. YERU
5086 \do{042C}{T2A}{\CYRSFTSN}{}	% CY. C. L. SOFT SIGN
5087 \do{042D}{T2A}{\CYREREV}{}	% CY. C. L. E
5088 \do{042E}{T2A}{\CYRYU}{}	% CY. C. L. YU
5089 \do{042F}{T2A}{\CYRYA}{}	% CY. C. L. YA
5090 \do{0430}{T2A}{\cyra}{}	% CY. S. L. A
5091 \do{0431}{T2A}{\cyrb}{}	% CY. S. L. BE
5092 \do{0432}{T2A}{\cyrv}{}	% CY. S. L. VE
5093 \do{0433}{T2A}{\cyrg}{}	% CY. S. L. GHE
5094 \do{0434}{T2A}{\cyrd}{}	% CY. S. L. DE
5095 \do{0435}{T2A}{\cyre}{}	% CY. S. L. IE
5096 \do{0436}{T2A}{\cyrzh}{}	% CY. S. L. ZHE
5097 \do{0437}{T2A}{\cyrz}{}	% CY. S. L. ZE
5098 \do{0438}{T2A}{\cyri}{}	% CY. S. L. I
5099 \do{0439}{T2A}{\cyrishrt}{}	% CY. S. L. SHORT I
5100 \do{043A}{T2A}{\cyrk}{}	% CY. S. L. KA
5101 \do{043B}{T2A}{\cyrl}{}	% CY. S. L. EL
5102 \do{043C}{T2A}{\cyrm}{}	% CY. S. L. EM
5103 \do{043D}{T2A}{\cyrn}{}	% CY. S. L. EN
5104 \do{043E}{T2A}{\cyro}{}	% CY. S. L. O
5105 \do{043F}{T2A}{\cyrp}{}	% CY. S. L. PE
5106 \do{0440}{T2A}{\cyrr}{}	% CY. S. L. ER
5107 \do{0441}{T2A}{\cyrs}{}	% CY. S. L. ES
5108 \do{0442}{T2A}{\cyrt}{}	% CY. S. L. TE

```

5109 \do{0443}{T2A}{\cyru}{}%           % CY. S. L. U
5110 \do{0444}{T2A}{\cyrf}{}%           % CY. S. L. EF
5111 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
5112 \do{0446}{T2A}{\cyrc}{}%           % CY. S. L. TSE
5113 \do{0447}{T2A}{\cyrch}{}%          % CY. S. L. CHE
5114 \do{0448}{T2A}{\cyrrh}{}%          % CY. S. L. SHA
5115 \do{0449}{T2A}{\cyrrhch}{}%        % CY. S. L. SHCHA
5116 \do{044A}{T2A}{\cyrrhdsn}{}%       % CY. S. L. HARD SIGN
5117 \do{044B}{T2A}{\cyrrery}{}%        % CY. S. L. YERU
5118 \do{044C}{T2A}{\cyrsftsn}{}%       % CY. S. L. SOFT SIGN
5119 \do{044D}{T2A}{\cyrrerev}{}%       % CY. S. L. E
5120 \do{044E}{T2A}{\cyryu}{}%          % CY. S. L. YU
5121 \do{044F}{T2A}{\cyrya}{}%          % CY. S. L. YA
5122 \do{0451}{T2A}{\cyryo}{}%          % CY. S. L. IO
5123 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
5124 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
5125 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
5126 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
5127 \do{00B4}{TS1}{\textasciicute}{}%  % ACUTE ACCENT
5128 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
5129 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
5130 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
5131 }

```

`\mathdegree` 面倒なので補っておく。

```
5132 \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。

```
5133 \newif\ifbxjx@gcc@cjk
```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```
5134 \newcommand*\greekasCJK{%
```

```
5135   \bxjx@gcc@cjktrue}
```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```
5136 \newcommand*\nogreekasCJK{%
```

```
5137   \bxjx@gcc@cjkfalse}
```

`\bxjx@fake@grk` `\bxjx@fake@grk{〈出力文字〉}{〈基準文字〉}`： ラテン文字で代用される数式ギリシャ文字の出力を行う。〈基準文字〉 (`mathchardef` の制御綴) の数式クラスと数式ファミリーを引き継いで、〈出力文字〉 (ASCII 文字トークン) の文字コードの数式文字を出力する。例えば、`\Pi` の意味が `\mathchar"7005` である場合、`\bxjx@fake@grk{B}{\Pi}` は `\mathchar"7042` を実行する。

※フォントパッケージ使用時の再定義を考慮して、〈基準文字〉が `mathchardef` であるかを検査し、そうでない場合はフォールバックとして単に〈出力文字〉を実行する。

```
5138 \def\bxjx@tmpdo#1\relax{%
```

```
5139   \def\bxjx@fake@grk##1##2{%
```

```
5140     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
```



```

5141 \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
5142   \ifx\\##1\\%
5143     \bxjx@canta##4\divide\bxjx@canta@cclvi
5144     \multiply\bxjx@canta@cclvi \advance\bxjx@canta`##3\relax
5145     \mathchar\bxjx@canta
5146   \else ##3\fi}
5147 }\expandafter\bxjx@tmpdo\string\mathchar\relax

```

■pdfTeX・upTeX の場合

```
5148 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0
```

- \backslash [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。

```

5149 \@ifpackageloaded{inputenc}{}{%else
5150   \RequirePackage[utf8]{inputenc}}
5151 \def\bxjx@tmpa{utf8}
5152 \ifx\bxjx@tmpa\inputencdoingname
5153   \PackageWarningNoLine\bxjx@pkgnome
5154   {Input encoding changed to utf8}%
5155   \inputencoding{utf8}%
5156 \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。

```

5157 \if u\bxjx@engine
5158 \kcatcode"0370=15
5159 \kcatcode"0400=15
5160 \kcatcode"0500=15
5161 \fi

```

各文字について \DeclareUnicodeCharacter を実行する。

```

5162 \def\bxjx@tmpdo#1{%
5163   \@tempcnta="#1\relax
5164   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
5165 \def\bxjx@tmpdo@a#1#2#3#4#5{%

```

引数 = \backslash [bxjx@KC/<符号値>]{<符号値>}{<fontenc>}{<LICR>}{<数式 LICR>}

“数式中の動作”を決定する。<数式 LICR> が空（数式非対応）なら警告を出す。

```

5166   \ifx\\#5\\%
5167     \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 \backslash Pi、小文字なら \backslash pi を基準文字にする。

```
5168   \else\ifcat A\@noexpand#5%
```

```

5169 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5170 {\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。

```

5171 \else \def\bxjx@tmpa{#5}%
5172 \fi\fi
5173 \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
5174 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}

```

以降はエンジン種別で分岐する。upTeX の場合。

```

5175 \if u\bxjx@engine
5176 \def\bxjx@tmpdo@b#1#2#3#4#5{%

```

引数 = $\langle \text{bxjx@KC} / \langle \text{符号値} \rangle \{ \langle \text{符号値} \rangle \{ \langle \text{fontenc} \rangle \{ \langle \text{LICR} \rangle \{ \langle \text{数式中の動作} \rangle \}$

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 $\langle \text{bxjx@ja@or@not} \rangle$ 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 $\langle \text{bxjx@KC} / \dots \rangle$ を流用している。

```

5177 \kchardef#1=\@tempcnta
5178 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
5179 \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 $\langle \text{UTF} \{ \langle \text{符号値} \rangle \}$ を使う ($\langle \text{UTF} \rangle$ は `bxCJKatype` の命令)。 $\langle \text{bxjx@KC} / \dots \rangle$ は使わないが定義済にする必要がある。

```

5180 \else\if p\bxjx@engine
5181 \def\bxjx@tmpdo@b#1#2#3#4#5{%
5182 \mathchardef#1=\@tempcnta
5183 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
5184 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
5185 \fi\fi

```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```

5186 \let\do\bxjx@tmpdo \bxjx@grkcyr@list

```

$\langle \text{bxjx@DeclareUnicodeCharacter} \rangle$ $\langle \text{bxjx@DeclareUnicodeCharacter} \rangle$ を変更して、「特定 CJK 曖昧文字」の場合に再定義を抑制したもの。

```

5187 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
5188 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
5189 \@onlypreamble\bxjx@DeclareUnicodeCharacter
5190 \def\bxjx@DeclareUnicodeCharacter#1#2{%
5191 \count@=#1\relax
5192 \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
5193 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
5194 \else
5195 \wlog{ \space\space skipped defining Unicode char U+#1}%
5196 \fi}

```

$\langle \text{bxjx@ja@or@not} \rangle$ $\langle \text{bxjx@ja@or@not} \{ \langle \text{和文用定義} \rangle \{ \langle \text{対象 fontenc} \rangle \{ \langle \text{LICR} \rangle \}$: $\langle \text{no} \rangle \text{greekasCJK}$ の状態

に応じて和文または欧文で文字を出力する。

```
5197 \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。

```
5198 \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。

```
5199 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
```

```
5200 \else \UseTextSymbol{#2}{#3}%
```

```
5201 \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。

```
5202 \begingroup
```

```
5203 \toks@\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}
```

```
5204 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
```

```
5205 \noexpand\bxjx@swap@DUC@cmd
```

```
5206 \the\toks@
```

```
5207 \noexpand\bxjx@swap@DUC@cmd}}}
```

```
5208 \endgroup\next
```

```
5209 \def\bxjx@swap@DUC@cmd{%
```

```
5210 \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```
5211 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
```

```
5212 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
```

```
5213 \let\bxjx@tmpa\relax}
```

以上。

■X_YTeX・LuaTeX の場合

```
5214 \else\ifnum0\if x\bxjx@engine1\fi\if l\bxjx@engine1\fi>0
```

各文字について、数式中の動作を定義する。

```
5215 \def\bxjx@tmpdo#1{%
```

```
5216 \bxjx@canta="#1\relax
```

```
5217 \begingroup
```

```
5218 \lccode`~=\bxjx@canta
```

```
5219 \lowercase{\endgroup
```

```
5220 \bxjx@tmpdo@a{~}}{#1}}
```

```
5221 \def\bxjx@tmpdo@a#1#2#3#4#5{%
```

〈数式 LICR〉が空なら何もしない。空でない場合、upL^AT_EX 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。

```
5222 \ifx\#5\\\let\bxjx@tmpa\relax
```

```
5223 \else\ifcat A\noexpand#5%
```

```
5224 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
```

```
5225 {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
```

```
5226 \else \def\bxjx@tmpa{#5}%
```

```
5227 \fi\fi
```

```

5228 \ifx\bxjx@tmpa\relax\else
5229   \mathcode\bxjx@cnta"8000 \let#1\bxjx@tmpa
5230 \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。

```

5231 \mathchardef\bxjx@tmpa="119
5232 \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_T_EX の場合は、Lua_T_EX-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。

```

5233 \if 1\bxjx@engine
5234   \protected\def\greekasCJK{%
5235     \bxjx@gcc@cjktrue
5236     \ltjsetparameter{jacharrange={+2, +8}}
5237   \protected\def\nogreekasCJK{%
5238     \bxjx@gcc@cjkfalse
5239     \ltjsetparameter{jacharrange={-2, -8}}
5240 \fi

```

X_Ǝ_T_EX の場合、`xeCJK` は X_Ǝ_T_EX 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。

```

5241 \if x\bxjx@engine
5242   \let\bxjx@gcc@cjk@list\@empty
5243   \def\do#1#2#3#4{%
5244     \edef\bxjx@gcc@cjk@list{\bxjx@gcc@cjk@list
5245       \noexpand\XeTeXcharclass"#1\bxjx@cnta}}
5246   \bxjx@grkcyr@list
5247   \protected\def\greekasCJK{%
5248     \bxjx@gcc@cjktrue
5249     \bxjx@cnta=\@ne \bxjx@gcc@cjk@list}
5250   \protected\def\nogreekasCJK{%
5251     \bxjx@gcc@cjkfalse
5252     \bxjx@cnta=\z@ \bxjx@gcc@cjk@list}
5253 \fi

```

以上。

```
5254 \fi\fi
```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5255 \nogreekasCJK
```

H.5 完了

おしまい。

```
5256 %</cjkat>
```

付録 I 補助パッケージ：bxjspandoc 🐼

Pandoc の L^AT_EX 用標準テンプレートをより幸せに使うための設定。BXJS クラスの pandoc ドライバのコードの中の、“汎用的”に使える部分を切り出したもの。つまり現在の pandoc ドライバはこのパッケージを読みこむ。

※テンプレートの T_EX コードより前に読み込む必要があるため、専ら文書クラス内での読込に限られる。

I.1 準備

```
5257 %<*ancpandoc>
```

```
5258 %% このファイルは日本語文字を含みます。
```

```
5259 \def\bxjsp@pkgname{bxjscjkat}
```

\bxjsp@engine エンジンの種別。

```
5260 \let\bxjsp@engine=n
```

```
5261 \@onlypreamble\bxjsp@do
```

```
5262 \def\bxjsp@do#1#2{%
```

```
5263   \edef\bxjsp@tmpa{\string#1}%
```

```
5264   \edef\bxjsp@tmpb{\meaning#1}%
```

```
5265   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
```

```
5266 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
```

```
5267 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
```

```
5268 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
```

```
5269 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

\bxjsp@begin@document@hook 文書本体開始時フック。

```
5270 \@onlypreamble\bxjsp@begin@document@hook
```

```
5271 \let\bxjsp@begin@document@hook\@empty
```

```
5272 \AtBeginDocument{\bxjsp@begin@document@hook}
```

\ifbxjsp@babel@used [スイッチ] Babel が読み込まれたか。

```
5273 \newif\ifbxjsp@babel@used
```

```
5274 \g@addto@macro\bxjsp@begin@document@hook{%
```

```
5275   \@ifpackageloaded{babel}{\bxjsp@babel@usedtrue}{}}
```

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。

```
5276 \newif\ifbxjsp@english
5277 \DeclareOption{english}{\bxjsp@englishtrue}

    オプション定義はおしまい。
5278 \ProcessOptions*
```

1.3 パッケージ読込の阻止

`\pandocSkipLoadFile` `\pandocSkipLoadFile{〈ファイル名〉}`: 特定のファイルを (`\@filewithoptions` の処理
 に関して) 読込済であるとマークする。

```
5279 \@onlypreamble\pandocSkipLoadFile
5280 \newcommand*\pandocSkipLoadFile[1]{%
5281   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5282 \def\bxjsp@skip@load@file@a#1#2{%
5283   \ifx#1\relax
5284     \def#1{2001/01/01}%
5285     \PackageInfo\bxjsp@pkgname
5286       {File '#2' marked as loaded\@gobble}%
5287   \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。

```
5288 \@onlypreamble\pandocSkipLoadPackage
5289 \newcommand*\pandocSkipLoadPackage[1]{%
5290   \pandocSkipLoadFile{#1.sty}}
```

1.4 `fixltx2e` パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近 (2015 年版以降) の L^AT_EX
 ではこれで警告が出る。これを抑止する。

L^AT_EX カーネルが新しい場合は `fixltx2e` を読込済にする。

```
5291 \ifx\@IncludeInRelease\undefined\else
5292   \pandocSkipLoadPackage{fixltx2e}
5293 \fi
```

1.5 `cmap` パッケージ

エンジンが (u)pL^AT_EX のときに `cmap` パッケージが読み込まれるのを阻止する。(実際は
 警告が出るだけで無害であるが。)

```
5294 \if j\bxjsp@engine
5295   \pandocSkipLoadPackage{cmap}
5296 \fi
```

1.6 microtype パッケージ

警告が多すぎなので消す。

```
5297 \if j\bxjisp@engine \else
5298   \PassOptionsToPackage{verbose=silent}{microtype}
5299 \fi
```

エンジンが (u)pL^AT_EX のときに microtype パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は standard ドライバでこの処理を行っていたが、元来は Pandoc 用の処理なので、1.5 版で pandoc に移動。

```
5300 \if j\bxjisp@engine
5301   \pandocSkipLoadPackage{microtype}
5302   \newcommand*\UseMicrotypeSet [2] [] {}
5303 \fi
```

1.7 Unicode 文字変換対策

Pandoc で L^AT_EX 形式に書き出す場合は、元データ中の一部の Unicode 文字を「L^AT_EX の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

```
…→\ldots{} ‘→` ’→' “→` ”→''
```

日本語 L^AT_EX では「L^AT_EX の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「\ldots を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の `\ldots` の実装。非数式である場合は代わりに … を実行する。

※以前は「Pandoc が必ず `\ldots{}` の形で書き出す」ことを利用して後続に `{}` があるかで「元が … であるか」を判断していた。ところが、Pandoc 2.7 版で `{}` を必ずしも付けなくなったため、1.9f 版で非数式の `\ldots` を全て … に戻す動作に変更した。

```
5304 \DeclareRobustCommand{\pandocLdots}{%
5305   \let\bxjisp@do\bxjisp@ja@ellipsis
5306   \ifmmode \let\bxjisp@do\bxjisp@org@ldots
5307   \else\ifbxjisp@babel@used
5308     \expandafter\ifx\csname bxjisp@ld/\language\endcsname\relax
5309     \let\bxjisp@do\bxjisp@org@ldots \fi
5310   \fi\fi \bxjisp@do}
5311 \@namedef{bxjisp@ld/japanese}{1}
5312 \def\bxjisp@ja@ellipsis{…}
5313 \let\bxjisp@org@ldots\ldots
```

`\ldots` の実装を `\pandocLdots` に置き換える。

```
5314 \g@addto@macro\bxjsp@begin@document@hook{%
5315   \let\bxjsp@org@ldots\ldots
```

もしここで `\newcommand\pandocLdots{\ldots}` という定義である場合は置き換えない。

```
5316   \long\def\bxjsp@tmpa{\ldots}%
5317   \ifx\pandocLdots\bxjsp@tmpa\else
```

english オプションが指定されていてかつ Babel が読み込まれていない場合も置き換えない。

```
5318     \ifnum0\ifbxjsp@english\ifbxjsp@babel@used\else1\fi\fi=0
5319       \let\ldots\pandocLdots
5320     \fi
5321   \fi}
```

`\ldots` の直後の文字が非英字の場合、Pandoc は「`\ldots。`」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが $X_{\text{T}}\text{E}_X \cdot \text{Lua}\text{T}_E\text{X}$ 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。

```
5322 \chardef\bxjsp@cc@other=12
5323 \@onlypreamble\bxjsp@makeother@range
5324 \def\bxjsp@makeother@range#1#2{%
5325   \@tempcnta"#1\relax \@tempcntb"#2\relax
5326   \loop\ifnum\@tempcnta<\@tempcntb
5327     \catcode\@tempcnta\bxjsp@cc@other
5328     \advance\@tempcnta\@ne
5329   \repeat}
5330 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
5331   \catcode"1F23B=\bxjsp@cc@other
5332   \bxjsp@makeother@range{9FCD}{A000}
5333   \bxjsp@makeother@range{1B002}{1B170}
5334   \bxjsp@makeother@range{2B820}{2EBF0}
5335 \fi
```

1.8 PandoLa モジュール

インストール済であれば読み込む。

```
5336 \IfFileExists{bxpandola.sty}{%
5337   \RequirePackage{bxpandola}\relax
5338   \PackageInfo\bxjsp@pkgname
5339   {PandoLa module is loaded@gobble}
5340 }{}
```

1.9 完了

おしまい。

```
5341 %</ancpandoc>
```


補助パッケージ実装はここまで。

5342 %</anc>