

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”)

v2.5 [2021/02/02]

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

目次

1	はじめに	3
2	オプション	10
3	和文フォントの変更	40
4	フォントサイズ	40
5	レイアウト	46
5.1	ページレイアウト	47
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	62
7	ページスタイル	63
8	文書のマークアップ	66
8.1	表題	66
8.2	章・節	71
8.3	リスト環境	84
8.4	パラメータの設定	91
8.5	フロート	93
8.6	キャプション	94
9	フォントコマンド	95

10	相互参照	98
10.1	目次の類	98
10.2	参考文献	103
10.3	索引	105
10.4	脚注	106
11	段落の頭へのグルー挿入禁止	109
12	いろいろなロゴ	113
13	amsmath との衝突の回避	113
14	初期設定	114
付録 A	和文ドライバの仕様 	118
付録 B	和文ドライバ : minimal 	119
B.1	補助マクロ	119
B.2	(u)pTeX 用の設定	121
B.3	pdfTeX 用の処理	126
B.4	X _Y TeX 用の処理	127
B.5	後処理 (エンジン共通)	128
付録 C	和文ドライバ : standard 	131
C.1	準備	131
C.2	和文ドライバパラメタ	131
C.3	共通処理 (1)	132
C.4	pTeX 用設定	140
C.5	pdfTeX 用設定 : CJK + bxcjkatype	143
C.6	X _Y TeX 用設定 : xeCJK + zxjatype	145
C.7	LuaTeX 用設定 : LuaTeX-ja	148
C.8	共通処理 (2)	151
付録 D	和文ドライバ : modern 	152
D.1	フォント設定	152
D.2	fixltx2e 読込	153
D.3	和文カテゴリコード	153
D.4	完了	153
付録 E	和文ドライバ : pandoc 	153
E.1	準備	154
E.2	和文ドライバパラメタ	154
E.3	dupload システム	155

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

1 はじめに

この文書は「BXJS ドキュメントクラス」の DocStrip 形式のソースである。インストー

ル時のモジュール指定は以下のである。

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

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

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

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

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

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

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

ここでは次のドキュメントクラス（スタイルファイル）を作ります。

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

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

以下では実際のコードに即して説明します。

minijs は、jsclasses に似た設定を行うパッケージです。

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

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

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

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

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

■BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため `calc` を読み込む。

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

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

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

クラスの本体ではこの他に `geometry` パッケージが読み込まれる。

TODO: 依存パッケージの情報。

互換性のための補助パッケージを読み込む。

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

非サポートのエンジンの場合は強制終了させる。

※ NTT `jTeX` と `Omega` 系。

```

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

```

`LuaTeX` の場合、本クラス用の `Lua` モジュールを用意する。

```

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

```

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

```

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

```

`\bxjs@robust@def` 無引数の頑強な命令を定義する。

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

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

※ Lua_{TEX} 0.8x 版でのプリミティブ名変更に対応。

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

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

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

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

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

※カーネルの 2020/10/01 版で導入された。

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

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

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

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

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

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

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

```
94 \def\bxjs@csletcs#1#2{%
95   \expandafter\let\csname#1\expandafter\endcsname\csname#2\endcsname}
```

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

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

`\bxjs@ifplus` `\@ifstar` の + 版。

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

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

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

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

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

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

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

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



```

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

```

今の段階では Q/H だけが使用可能。

```

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

```

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

```

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

```

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

```

140 \@onlypreamble\bxjs@post@option@hook
141 \let\bxjs@post@option@hook\@empty

```

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

```

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

```

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

```

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

```

一時的な手続き用の制御綴。

```

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

```

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

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

```

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

```

```

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

```

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

```

160 \bxjs@robust@def\jsInhibitGlue{%
161   \ifx\inhibitglue\@undefined\else \inhibitglue \fi}

```

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

```

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

```

2 オプション

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

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

`\if@restonecol` 段組のときに真になる論理変数です。

```
172 \newif\if@restonecol
```

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

```
173 \newif\if@titlepage
```

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

```
174 %<book|report>\newif\if@openright
```

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

```
175 %<book|report>\newif\if@openleft
```

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

BXJS では report 系でも定義されることに注意。

```
176 %<book|report>\newif\if@mainmatter \@mainmattertrue
```

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

JS クラスと異なり、初期値は偽とする。

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

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

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

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

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

geometry の用紙サイズのオプション名を全てサポートする。

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

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

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

Pandoc では用紙サイズ指定について「後ろに paper を付けた名前のオプション」を指定する。これに対処するため、後ろに paper をつけた形を用意する。さらに、「用紙サイズを custom とすると何も設定しない」ようにするため custompaper というオプションを用意する。

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

■横置き 用紙の縦と横の長さを入れ換えます。

```
216 \newif\if@landscape
217 \@landscapefalse
218 \DeclareOption{landscape}{\@landscapetrue}
```

■slide オプション slide を新設しました。

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

```
219 \newif\if@slide
```

BXJS ではスライド用のクラス bxjsslide を用意しているので、本来はこのスイッチは不要なはずである。しかし、JS クラスの一部のコードをそのまま使うために保持している。※この \if@slide という制御綴は、ユニークでないにも関わらず、衝突した場合に正常動作が保たれない、という問題を抱えている。

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

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

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

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

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

[2016-07-24] オプティカルサイズを調整するために NFSS 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。

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

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

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

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

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

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

TODO: 恐らく 14pt と `base=14.4pt` 等の関係も全く等価であるべき。

```
232 \def\bxjs@setjbasefontsize#1{%
233 \setkeys{bxjs}{jbase=#1}}
```

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

`\ifjsc@mag@xreal` は「NFSS にパッチを当てるか」を表すスイッチ。

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

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

```

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

```

JS クラス互換の magstyle 設定オプション。

```

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

```

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

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

```

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

```

■面付け オプション `mentuke` で幅ゼロのトンボを出力します。面付けに便利です。これもアスキー版のままです。

```

275 \if j\jsEngine

```

```

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

```

■両面，片面オプション `twoside` で奇数ページ・偶数ページのレイアウトが変わります。

[2003-04-29] `vartwoside` でどちらのページも傍注が右側になります。

```

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

```

■二段組 `twocolumn` で二段組になります。

```

284 \DeclareOption{onecolumn}{\@twocolumnfalse}
285 \DeclareOption{twocolumn}{\@twocolumntrue}

```

■表題ページ `titlepage` で表題・概要を独立したページに出力します。

```

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

```

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

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

```

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

```

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

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

```

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

```

```

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

```

leqno で数式番号が左側になります。fleqn で数式が本文左端から一定距離のところに出
力されます。森本さんにしたがって訂正しました。

```

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

```

■文献リスト 文献リストを open 形式（著者名や書名の後に改行が入る）で出力します。
これは使われることはないのでコメントアウトしてあります。

```

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

```


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

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

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

enablejfam オプションの処理。
342 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
343 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
344 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
345 \define@key{bxjs}{enablejfam}[true]{%
346   \bxjs@set@keyval{enablejfam}{#1}{}}
```

JS クラスとの互換のため disablejfam オプションを定義する。

```
347 \DeclareOption{disablejfam}{\let\bxjs@enablejfam=f}
```

※実際に何らかの設定を行うのは和文ドライバである。和文ドライバとエンジンの組合せにより、enablejfam が default である場合に「数式和文ファミリ」が有効と無効の選択は異なるし、またそもそも有効と無効の一方しか選択できない場合もある。

■ドラフト draft で overfull box の起きた行末に 5pt の罫線を引きます。

[2016-07-13] \ifdraft を定義するのをやめました。

\ifjsDraft draft オプションが指定されているか。

※ JS クラスの \ifdraft が廃止されたので、BXJS クラスでも \ifdraft を 2.0 版で廃止した。

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

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

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

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

BXJS クラスではここは和文ドライバの管轄。

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

BXJS クラスでは geometry パッケージがこの処理を行う。

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

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

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

■英語化 オプション english を新設しました。

※`\if@english` は非ユニークで衝突耐性がない。

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

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

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

BXJS では当初から bxjsreport クラスが用意されている。

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

BXJS クラスでは、nojslogo を既定とする。

```

358 \newif\if@jslogo \@jslogofalse
359 \DeclareOption{jslogo}{\@jslogotrue}
360 \DeclareOption{nojslogo}{\@jslogofalse}

```

■複合設定オプション

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

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

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

※元の長さが 128pt 以上の場合でも動作するように修正した。

```

361 \mathchardef\bxjs@isc@ll=128
362 \mathchardef\bxjs@isc@sl=259
363 \def\bxjs@isc@sl@h{65539 }
364 \def\bxjs@invscale#1#2{%
365   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
366   \ifdim\@tempdima<\bxjs@isc@ll\p@
367     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
368     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
369   \else
370     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
371     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
372   \fi
373   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
374   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
375   \@tempdimb\@tempcnta\@ne
376   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
377   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
378   \@whiledim\@tempdimb<\@tempdimc\do{%
379     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
380     \advance\@tempcntb\@ne \divide\@tempcntb\@tw@
381     \ifdim #2\@tempcntb>\@tempdima
382       \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
383     \else \@tempdimb=\@tempcntb\@ne \fi}%
384   \xdef\bxjs@gtmpa{\the\@tempdimb}%
385   \endgroup #1=\bxjs@gtmpa\relax}

```

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

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

```

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

```

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

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

※実際の和文ドライバ・エンジン設定より優先される。

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

ドライバオプションを `dvi=dvipdfmx` 相当に変更する。

※これは実際のドライバ設定で上書きできる（オプション宣言順に注意）。

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

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

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

■エンジン・ドライバオプション

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

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

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

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

エンジン明示指定のオプションの処理。

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

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

```

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

```

`\bxjs@driver@given` オプションで明示されたドライバの種別。

```

432 %\let\bxjs@driver@given\@undefined
433 \let\bxjs@driver@@dvimode=0
434 \let\bxjs@driver@@dvipdfmx=1
435 \let\bxjs@driver@@pdfmode=2
436 \let\bxjs@driver@@xetex=3
437 \let\bxjs@driver@@dvips=4
438 \let\bxjs@driver@@none=5

```

`\bxjs@driver@opt` 明示された「ドライバ指定」のオプション名。

```

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

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

```

```

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

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

```

■その他の BXJS 独自オプション

TODO: 互換用オプションを分離する (2.0 版で?)。

`\bxjs@depre@opt` 非推奨のオプションについて警告を出す。

```

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

```

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

※ 2.0 版より、既定値を常に真とする。

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

`nobigcode` / `bigcode` オプションの定義。

```

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

```

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

```
481 \newif\ifbxjs@oldfontcommands
```

`nooldfontcommands`、`oldfontcommands` オプションの定義。

※`oldfontcommands` オプションの名前は memoir クラスに倣った。ちなみに KOMA-Script では `enabledeprecatedfontcommands` であるがこれはチョットアレなので避けた。

```

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

```

```

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


```

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

```

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

```

■`keyval` 型のオプション 

その他のオプションは `keyval` の機構を用いて処理する。

```

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

```

`\bxjs@safe@setkeys` 未知のキーに対してエラー無しで無視する `\setkeys`。

※ネスト不可。

```

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

```

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

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

```

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

```

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

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

```

513 \@onlypreamble\bxjs@declare@bool@option
514 \def\bxjs@declare@bool@option#1#2#3{%
515   \csname newif\expandafter\endcsname\csname ifbxjs@#2\endcsname
516   \@nameuse{bxjs@#2#3}%
517   \define@key{bxjs}{#1}[true]{%
518     \expandafter\ifx\csname bxjs@#2##1\endcsname\relax
519       \bxjs@error@keyval{#1}{##1}%
520     \else \@nameuse{bxjs@#2##1}%
521     \fi}}

\bxjs@set@keyval \bxjs@set@keyval{<key>}{<value>}{<error>}
    \bxjs@kv@<key>@<value> が定義済ならそれを実行し、未定義ならエラーを出す。
522 \def\bxjs@set@keyval#1#2#3{%
523   \bxjs@csletcs{bxjs@next}{bxjs@kv@#1@#2}%
524   \ifx\bxjs@next\relax
525     \bxjs@error@keyval{#1}{#2}%
526     #3%
527   \else \bxjs@next
528   \fi}
529 \@onlypreamble\bxjs@error@keyval
530 \def\bxjs@error@keyval#1#2{%
531   \ClassError\bxjs@clsname
532   {Invalid value '#2' for option #1}\@ehc}

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

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

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

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

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

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

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

```



```

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

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

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

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

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

\bxjs@jadriver 和文ドライバの名前。
551 \let\bxjs@jadriver\relax
552 %\let\bxjs@jadriver@opt\@undefined

    ja オプションの処理。
    ※jadriver は 0.9 版で用いられた旧称。
    ※単なる ja という指定は無視される (Pandoc 対策)。
553 \define@key{bxjs}{jadriver}{%
554 \bxjs@depre@opt{jadriver}{ja=#1}\edef\bxjs@jadriver@opt{#1}}
555 \define@key{bxjs}{ja}{\relax}%
556 \ifx\relax#1\else\edef\bxjs@jadriver@opt{#1}\fi}

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

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

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

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

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

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

```

```

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

```

(新しい素敵な名前。)

※ただし制御綴としては、*付の名前は扱い難いので、\bxjs@magstyle@@xreal の方を優先させる。

```

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

```

\bxjs@magstyle@@default は既定の値を表す。

```

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

```

magstyle オプションの処理。

```

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

```

\bxjs@geometry geometry オプションの指定値。

```

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

```

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

```

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

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

592 \newif\ifbxjs@dvi@opt

```

DVI モードのドライバとドライバ種別との対応。

```

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

```

dvi オプションの処理。

```

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

```

```

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

\bxjs@driver@given を未定義にしていることに注意。

603 \def\bxjs@driver@opt{#1}%
604 \let\bxjs@driver@given\undefined
605 \bxjs@dvi@opttrue
606 \fi}

\ifbxjs@layout@buggyhmargin [スイッチ] bxjsbook の左右マージンがアレか。
※layout が v1 の場合はアレになる。
607 \newif\ifbxjs@layout@buggyhmargin

\ifbxjs@force@chapterabstract [スイッチ] abstract 環境を chapterabstract にするか。
※bxjsbook では常に真。bxjsreport では layout が v1 の場合に真になる。
608 \newif\ifbxjs@force@chapterabstract
609 %<book>\bxjs@force@chapterabstracttrue

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

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

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

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

\bxjs@paragraph@mark paragraph-mark の指定値。パラグラフのマーク。

```

```

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

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

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

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

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

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

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

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

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

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

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

```

```

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

```

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

```

659 \bxjs@declare@bool@option{plautopatch}{plautopatch}{false}
660 \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。

```

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

```

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

```

後処理

```

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

```

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

```

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

```

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

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

ここのコードを削除。

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

正規化前の和文ドライバの値を \bxjs@jadriver に設定する。

```

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

```
704 \let\bxjs@tmpb\jsEngine
705 \ifx j\bxjs@tmpb\ifjsWithpTeXng
706   \let\bxjs@tmpb=g
707 \fi\fi
708 \ifx j\bxjs@tmpb\ifjsWithupTeX
709   \let\bxjs@tmpb=u
710 \fi\fi
711 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
712   \let\bxjs@tmpb=n
713 \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。)


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

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

```
728 \ifjsWithpTeXng
729   \g@addto@macro\@classoptionslist{,uplatex}
730 \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。

```
731 \@tempwattrue
732 \ifx \bxjs@driver@given\@undefined\else
733   \ifjsInPdfMode
734     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
735       \@tempwafalse
736     \fi
737   \else\ifx x\jsEngine
738     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
```

```

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

```

DVI 出力のエンジンである場合の追加処理。

```

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

```

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

```

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

```

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

```

775 \ifjsWithpTeXng

```



```

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

```

782 \ifx\bxjs@driver@given\bxjs@driver@@none
783 \bxjs@papersizefalse
784 \fi

```

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

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

```

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

```

799 \def\bxjs@@pandoc{pandoc}
800 \ifx\bxjs@jadriver\bxjs@@pandoc \ifx\bxjs@plautopatch@given\@undefined
801 \bxjs@plautopatchtrue
802 \fi\fi
803 \ifx j\jsEngine \ifbxjs@plautopatch
804 \RequirePackage{plautopatch}[2018/08/22]%v0.3
805 \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。

```

806 \ifx\bxjs@jadriver@opt\@undefined\else
807 \ifx\bxjs@engine@given\@undefined
808 \ClassError\bxjs@clsname
809 {An engine option must be explicitly given}%
810 {When you use a Japanese-driver you must specify a correct\MessageBreak

```

```

811     engine option.\MessageBreak\@ehc}
812 \fi\fi

```

新しい Lua_T_EX (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。)

```

813 \ifx\bxjs@magstyle@@default\bxjs@magstyle@@mag\else
814   \ifx\bxjs@magstyle\bxjs@magstyle@@mag
815     \let\bxjs@magstyle\bxjs@magstyle@@default
816     \ClassError\bxjs@clsname
817       {The engine does not support 'magstyle=usemag'}%
818       {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
819         The default value 'nomag*' is used instead.\MessageBreak \@ehc}
820   \fi
821 \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。

```

822 \ifx\bxjs@jbase@opt\@undefined\else
823   \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。

```

824     \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
825     \bxjs@invscale\@tempdima\jsScale
826     \bxjs@setbasefontsize{\@tempdima}%
827   \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。

```

828     \ifx\bxjs@scale@opt\@undefined\else
829       \ClassWarningNoLine\bxjs@clsname
830         {Redundant 'scale' option is ignored}%
831     \fi
832     \jsSetQHLlength\@tempdima{\bxjs@jbase@opt}%
833     \@tempdimb=\bxjs@param@basefontsize\relax
834     \edef\jsScale{\strip@pt\@tempdimb}%
835     \bxjs@invscale\@tempdima\jsScale
836     \edef\jsScale{\strip@pt\@tempdima}%
837   \fi
838 \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。

```

839 \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。)

```

840 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
841 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
842 \catcode\@tempcnta\active
843 \advance\@tempcnta\@ne
844 \repeat
845 \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。

```

846 \ifbxjs@disguise@js
847 %<book|report>\def\bxjs@js@clsname{jsbook}
848 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
849 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
850 \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。

```

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

```

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

```

852 \ifbxjs@oldfontcommands
853 \AtEndOfClass{\allowoldfontcommands}
854 \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。

```
855 \newdimen\stockwidth \newdimen\stockheight
856 \begingroup\expandafter\expandafter\expandafter\endgroup
857 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
858 % \newdimen\stockwidth \newdimen\stockheight
859 \setlength{\stockwidth}{\paperwidth}
860 \setlength{\stockheight}{\paperheight}
861 \advance \stockwidth 2in
862 \advance \stockheight 2in
863 \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。

```
864 %<slide>\def\n@baseline{13}%
865 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
866 %<!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。

```
867 \ifx\bxjs@magstyle\bxjs@magstyle@@mag
868 \jsc@magtrue
869 \else\ifx\bxjs@magstyle\bxjs@magstyle@@xreal
870 \jsc@mag@xrealtrue
871 \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` に直したので、`geometry` パッケージと共存できると思います。
なお、新ドキュメントクラス側で 10pt 以外にする場合の注意：

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

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

```
872 \ifx\bxjs@param@mag\relax
```

```

873 \@tempdima=\bxjs@param@basefontsize
874 \advance\@tempdima.001pt \multiply\@tempdima25
875 \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
876 \edef\bxjs@param@mag{\the\@tempcmta}
877 \else
878 % mag 値が直接指定された場合
879 \bxjs@gset@tempcmta{\bxjs@param@mag}
880 \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
881 % 有効な mag 値の範囲は 1--32768
882 \edef\bxjs@param@mag{\the\@tempcmta}
883 \advance\@tempcmta100000
884 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
885 \expandafter\bxjs@tmpa\the\@tempcmta\@nil
886 \edef\bxjs@param@basefontsize{\the\@tempdima}
887 \fi
888 \@tempcmta\bxjs@param@mag \advance\@tempcmta100000
889 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
890 \expandafter\bxjs@tmpa\the\@tempcmta\@nil
891 \edef\jsc@magscale{\strip@pt\@tempdima}
892 \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。

```

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

```

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

noz 非指定時は \zw を \jsZw と同義にする。

```

906 \newdimen\jsZw

```

```

907 \jsZw=10\jsc@empt \jsZw=\jsScale\jsZw
908 \ifbxjs@usezw
909 \providecommand*\zw{\jsZw}
910 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

912 \ifjsc@mag@xreal
913 \RequirePackage{type1cm}
914 \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

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

```

```

949 \edef\mandatory@arg{\mandatory@arg\jsc@ceref@size}%
950 \empty@sfcnt}
951 \def\genb@sfcnt{%
952 \edef\mandatory@arg{%
953 \mandatory@arg\expandafter\genb@x\jsc@ceref@size..\@@}%
954 \empty@sfcnt}
955 \ifbxjs@TUenc\else
956 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
957 \fi
958 \fi

```

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

`\jsc@smallskip`

```

\jsc@medskip 959 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 960 %\def\jsc@medskip{\vspace\jsc@medskipamount}
961 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

`\jsc@smallskipamount`

```

\jsc@medskipamount 962 \newskip\jsc@smallskipamount
\jsc@bigskipamount 963 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
964 %\newskip\jsc@medskipamount
965 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
966 %\newskip\jsc@bigskipamount
967 %\jsc@bigskipamount =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

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

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

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

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

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

```

968 % \ifpapersize
969 % \setlength{\@tempdima}{\paperwidth}
970 % \setlength{\@tempdimb}{\paperheight}
971 % \iftombow
972 % \advance \@tempdima 2truein
973 % \advance \@tempdimb 2truein
974 % \fi
975 % \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
976 % \fi

```

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

\@ 欧文といえば、 \LaTeX の $\text{\def\@{\spacefactor\@m}}$ 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 \LaTeX で、auxiliary files に書き出されたときにスペースが食われないようにする修正が入りました。これに合わせて $\{ \}$ を補いました。

BXJS クラスでの変更点：

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

```
977 \chardef\bxjs@periodchar=`\.  
978 \bxjs@protected\def\bxjs@SE{%  
979   \ifnum\spacefactor<\@m \spacefactor\@m  
980   \else \spacefactor\sfcode\bxjs@periodchar  
981   \fi}  
982 \ifbxjs@fix@at@cmd  
983   \def\@{\bxjs@SE{}}  
984 \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

$$\text{\@setfontsize{\normalsize}{10}{16}}$$

のようにして行います。これは

$$\text{\normalsize}$$
 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 \LaTeX の内部命令 \@xpt を使っています。この \@xpt の類は次のものがあり、 \LaTeX 本体で定義されています。

\@vpt	5	\@vipt	6	\@viipt	7
\@viipt	8	\@ixpt	9	\@xpt	10

`\@xipt 10.95 \@xipt 12 \@xivpt 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 $\mathrm{pLATEX}2_{\epsilon}$ で `0pt plus .4pt minus .5pt` に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

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

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

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

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

```
985 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
986 \expandafter\bxjs@tmpa\expandafter{%
987   \set@fontsize{#1}{#2}{#3}%
988 % 末尾にコードを追加
989   \expandafter\def\expandafter\size@update\expandafter{%
990     \size@update
991     \jsFontSizeChanged}%
992 }
```

`\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。

```
993 \newcommand*\jsFontSizeChanged{%
994   \jsZw=\f@size\p@
995   \jsZw=\jsScale \jsZw
996   \ifdim\parindent>\z@
997     \if@english \parindent=1em
998     \else       \parindent=1\jsZw
999     \fi
1000   \fi\relax
1001   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
1002 \newcommand*\jsResetDimen{}
```

`\jsc@setfontsize` クラスファイルの内部では、拡大率も考慮した `\jsc@setfontsize` を `\setfontsize` の

変わりに用いることにします。

```
1003 \ifjsc@mag
1004   \let\jsc@setfontsize\@setfontsize
1005 \else
1006   \def\jsc@setfontsize#1#2#3{%
1007     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
1008 % microtype 対策
1009 \ifjsWithTeX\if j\jsEngine\else
1010   \def\jsc@setfontsize#1#2#3{%
1011     \edef\bxjs@sfs@next{%
1012       \unexpanded{\@setfontsize#1}%
1013       {\the\dimexpr#2\jsc@empt\relax}{\the\dimexpr#3\jsc@empt\relax}%
1014     }\bxjs@sfs@next}
1015   \fi\fi
1016 \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。

```
1017 \emergencystretch 3\jsZw
```

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

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

`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対し

て、しっぱ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。

```
1018 \newif\ifnarrowbaselines
1019 \if@english
1020   \narrowbaselinestrue
1021 \fi
1022 \def\narrowbaselines{%
1023   \narrowbaselinestrue
1024   \skip0=\abovedisplayskip
1025   \skip2=\abovedisplayshortskip
```

```

1026 \skip4=\belowdisplayskip
1027 \skip6=\belowdisplayshortskip
1028 % 一時的に警告を無効化する
1029 \let\bxjs@save@nomath\@nomath
1030 \let\@nomath\@gobble
1031 \@currsizelselectfont
1032 \let\@nomath\bxjs@save@nomath
1033 \abovedisplayskip=\skip0
1034 \abovedisplayshortskip=\skip2
1035 \belowdisplayskip=\skip4
1036 \belowdisplayshortskip=\skip6\relax}
1037 \def\widebaselines{\narrowbaselinesfalse\@currsizelselectfont}

```

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。

```

1038 \def\bxjs@if@narrowbaselines{%
1039   \ifnarrowbaselines\expandafter\@firstoftwo
1040   \else \expandafter\@secondoftwo
1041   \fi
1042 }

```

\normalsize 標準のフォントサイズと行送りを選ぶコマンドです。

本文 10 ポイントのときの行送りは、欧文の標準クラスファイルでは 12 ポイント、アスキーの和文クラスファイルでは 15 ポイントになっていますが、ここでは 16 ポイントにしました。ただし \narrowbaselines で欧文用の 12 ポイントになります。

公称 10 ポイントの和文フォントが約 9.25 ポイント（アスキーのものの 0.961 倍）であることもあり、行送りがかなりゆったりとしたと思います。実際、 $16/9.25 \approx 1.73$ であり、和文の推奨値の一つ「二分四分」（1.75）に近づきました。

microtype 対策のため if 文を避ける。

```

1043 \renewcommand{\normalsize}{%
1044   \bxjs@if@narrowbaselines{%
1045     \jsc@setfontsize\normalsize\@xpt\@xiipt
1046   }{%else
1047     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
1048   }%

```

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。

```
1049 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
1050 \abovedisplayshortskip \z@ \@plus3\p@?
1051 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1052 \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` にコピーしてきます。`\@listI` の設定は後で出てきます。

```
1053 \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。

```
1054 %</class>
1055 %<*class|minijs>
1056 %% initialize
1057 \normalsize
1058 %</class|minijs>
1059 %<*class>
```

`\Cht` 基準となる長さの設定をします。p_{La}T_EX 2_ε カーネル (`plfonts.dtx`) で宣言されているパラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (`1zw`) です。
`\Cwd` [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。

```
1060 \ifx\Cht\@undefined \newdimen\Cht \fi
1061 \ifx\Cdp\@undefined \newdimen\Cdp \fi
1062 \ifx\Cwd\@undefined \newdimen\Cwd \fi
1063 \ifx\Cvs\@undefined \newdimen\Cvs \fi
1064 \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。

```
1065 \setlength\Cht{0.88\jsZw}
1066 \setlength\Cdp{0.12\jsZw}
1067 \setlength\Cwd{1\jsZw}
1068 \setlength\Cvs{\baselineskip}
1069 \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。

```

1070 \newcommand{\small}{%
1071   \bxjs@if@narrowbaselines{%
1072     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
1073     %<kiyou>    \jsc@setfontsize\small{8.8888}{11}%
1074   }{%else
1075     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
1076     %<kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
1077   }%
1078   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
1079   \abovedisplayshortskip \z@ \@plus3\p@?
1080   \belowdisplayskip \abovedisplayskip
1081   \belowdisplayshortskip \belowdisplayskip
1082   \def\@listi{\leftmargin\leftmargini
1083             \topsep \z@
1084             \parsep \z@
1085             \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。

```

1086 \newcommand{\footnotesize}{%
1087   \bxjs@if@narrowbaselines{%
1088     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
1089     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
1090   }{%else
1091     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
1092     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
1093   }%
1094   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
1095   \abovedisplayshortskip \z@ \@plus2\p@?
1096   \belowdisplayskip \abovedisplayskip
1097   \belowdisplayshortskip \belowdisplayskip
1098   \def\@listi{\leftmargin\leftmargini
1099             \topsep \z@
1100             \parsep \z@
1101             \itemsep \parsep}}

```

`\scriptsize` それ以外のサイズは、本文に使うことがないので、単にフォントサイズと行送りだけ変更します。特に注意すべきは `\large` で、これは二段組のときに節見出しのフォントとして使い、`\large` 行送りを `\normalsize` と同じにすることによって、節見出しが複数行にわたっても段間で
`\Large` 行が揃うようにします。

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1102 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1103 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 1104 \if@twocolumn
\HUGE 1105 %<!kiyou>   \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
1106 %<kiyou>    \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1107 \else
1108 %<!kiyou>   \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}

```

```

1109 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1110 \fi
1111 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{@xivpt{21}}
1112 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1113 \newcommand{\LARGE}{\jsc@setfontsize\LARGE{@xvipt{25}}
1114 \newcommand{\huge}{\jsc@setfontsize\huge{@xxpt{28}}
1115 \newcommand{\Huge}{\jsc@setfontsize\Huge{@xxvpt{33}}
1116 \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。

```

1117 \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教えてください。

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

```

1118 % \newcommand{\headfont}{\bfseries}
1119 \newcommand{\headfont}{\sffamily}
1120 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

5 レイアウト

■二段組

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

```

1121 %<kiyou>\setlength\columnsep{2\Cwd}
1122 %<kiyou>\setlength\columnsep{28truebp}
1123 \setlength\columnseprule{\z@}

```

■段落

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

```

\lineskiplimit 1124 \setlength\lineskip{1\jsc@empt}
\normallineskiplimit 1125 \setlength\normallineskip{1\jsc@empt}
1126 \setlength\lineskiplimit{1\jsc@empt}

```

```
1127 \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。

```
1128 \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。

```
1129 \setlength\parskip{\z@}
```

```
1130 \if@slide
```

```
1131 \setlength\parindent{0\p@}
```

```
1132 \else
```

```
1133 \setlength\parindent{1\Cwd}
```

```
1134 \fi
```

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

```
\@highpenalty 1135 \@lowpenalty 51
```

```
1136 \@medpenalty 151
```

```
1137 \@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。

```
1138 % \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。

```
1139 % \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。

```
1140 \onlypreamble\bxjs@bd@pre@geometry@hook
```

```
1141 \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。

```
1142 \ifjsc@mag
1143 \mag=\bxjs@param@mag
1144 \fi
1145 \setlength{\topskip}{10\p@?}
```

\jsSetQHLength のための和文単位の定義。

```
1146 \def\bxjs@unit@trueQ{0.25trueem}\let\bxjs@unit@trueH\bxjs@unit@trueQ
1147 \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。

```
1148 \@tempwafalse
1149 \def\bxjs@tmpdo{\@ifnextchar\bgroup\bxjs@tmpdo@a\remove@to@nnil}
1150 \def\bxjs@tmpdo@a#1{\edef\bxjs@tmpa{#1}%
1151   \@ifnextchar\bgroup\bxjs@tmpdo@b\remove@to@nnil}
1152 \def\bxjs@tmpdo@b#1{\edef\bxjs@tmpa{\bxjs@tmpa,#1}%
1153   \@ifnextchar\@nnil\bxjs@tmpdo@c\remove@to@nnil}
1154 \def\bxjs@tmpdo@c\@nnil{\@tempwattrue
1155   \edef\bxjs@param@paper{papersize={\bxjs@tmpa}}}%
1156 \expandafter\bxjs@tmpdo\bxjs@param@paper\@nnil
```

W,H の形式について。

```
1157 \if@tempwa\else
1158   \def\bxjs@tmpa{\@nil,\@nil}
1159   \def\bxjs@tmpdo#1,#2,#3\@nnil{%
1160     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1161       \@tempwattrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1162   \expandafter\bxjs@tmpdo\bxjs@param@paper,\@nil,\@nil\@nnil
1163 \fi
```

W*H の形式について。

```
1164 \if@tempwa\else
1165   \def\bxjs@tmpa{\@nil*\@nil}
1166   \def\bxjs@tmpdo#1*#2*#3\@nnil{%
1167     \def\bxjs@tmpb{#3}\ifx\bxjs@tmpa\bxjs@tmpb
1168       \@tempwattrue\edef\bxjs@param@paper{papersize={#1,#2}}\fi}
1169   \expandafter\bxjs@tmpdo\bxjs@param@paper*\@nil*\@nil\@nnil
1170 \fi
```

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

```
1171 \edef\bxjs@layout@paper{%
1172   \ifjsc@mag truedimen,\fi
1173   \if@landscape landscape,\fi
1174   \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。

```
1175 %<*article|report>
1176 \def\bxjs@layout@base{%
1177   headheight=\topskip,footskip=0.03367\paperheight,%
```



```

1178 headsep=\footskip-\topskip,includeheadfoot,%
1179 }
1180 \edef\bxjs@layout{\bxjs@layout@base
1181 hscale=0.76,hmarginratio=1:1,%
1182 vscale=0.83,vmarginratio=1:1,%
1183 }
1184 %</article|report>
1185 %<*book>
1186 \def\bxjs@layout@base{%
1187 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1188 }
1189 \ifbxjs@layout@buggyhmargin %---
1190 % アレ
1191 \edef\bxjs@layout{\bxjs@layout@base
1192 hmargin=36\jsc@mmm,hmarginratio=1:1,%
1193 vscale=0.83,vmarginratio=1:1,%
1194 }
1195 \else %---
1196 % 非アレ
1197 \edef\bxjs@layout{\bxjs@layout@base
1198 hmargin=18\jsc@mmm,%
1199 vscale=0.83,vmarginratio=1:1,%
1200 }
1201 \fi %---
1202 %</book>
1203 %<*slide>
1204 \def\bxjs@layout@base{%
1205 noheadfoot,%
1206 }
1207 \edef\bxjs@layout{\bxjs@layout@base
1208 hscale=0.9,hmarginratio=1:1,%
1209 vscale=0.944,vmarginratio=1:1,%
1210 }
1211 %</slide>

```

textwidth オプションの設定を反映する。

```

1212 %<*!book>
1213 \ifx\bxjs@textwidth@opt\undefined\else
1214 \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1215 \edef\bxjs@layout{\bxjs@layout width=\the\@tempdima,}
1216 \fi
1217 %</!book>
1218 \ifx\bxjs@number@of@lines@opt\undefined\else
1219 \bxjs@gset@tempcnta{\bxjs@number@of@lines@opt}
1220 \edef\bxjs@layout{\bxjs@layout lines=\the\@tempcnta,}
1221 \fi

```

\fullwidth 〔寸法レジスタ〕 ヘッダ・フッタ領域の横幅。

```

1222 \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。

```
1223 %<*book>
1224 \newcommand\jsTextWidthLimit{40}
1225 \@tempdima=\jsTextWidthLimit\Cwd
1226 \ifx\bxjs@textwidth@limit@opt\@undefined\else
1227   \bxjs@egset@tempcnta{\bxjs@textwidth@limit@opt}
1228   \@tempdima=\@tempcnta\Cwd
1229 \fi
1230 \ifx\bxjs@textwidth@opt\@undefined\else
1231   \jsSetQHLength\@tempdima{\bxjs@textwidth@opt}
1232 \fi
1233 \edef\bxjs@textwidth@limit{\the\@tempdima}
1234 \ifdim\@tempdima=\jsTextWidthLimit\Cwd\else
1235   \bxjs@invscale\@tempdima{\strip@pt\Cwd}
1236   \long\edef\jsTextWidthLimit{\strip@pt\@tempdima}
1237 \fi
1238 %</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。

```
1239 \def\bxjs@preproc@layout{%
1240   \edef\bxjs@save@ht@strutbox{\the\ht\strutbox}\ht\strutbox=10\jsc@mppt}
```

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

```
1241 \def\bxjs@postproc@layout{%
  geometry のドライバを再設定する。
1242   \ifx\bxjs@geometry@driver\relax\else
1243     \let\Gm@driver\bxjs@geometry@driver
1244   \fi
  \ht\strutbox の値を元に戻す。
1245   \ht\strutbox=\bxjs@save@ht@strutbox\relax
  \textwidth の値を補正する。
1246   \ifbxjs@whole@zw@lines
1247     \@tempdimb=\textwidth
1248     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1249     \advance\textwidth.005pt\relax
1250     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1251     \advance\@tempdimb-\textwidth
1252     \advance\oddsidemargin 0.5\@tempdimb
1253     \advance\evensidemargin 0.5\@tempdimb
1254   \fi
1255   \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。

```

1256 %<*book>
1257 \@tempdima=\bxjs@textwidth@limit\relax
1258 \ifbxjs@whole@zw@lines
1259 \advance\@tempdima.005pt\relax
1260 \divide\@tempdima\Cwd \multiply\@tempdima\Cwd
1261 \fi
1262 \ifdim\textwidth>\@tempdima
1263 \textwidth=\@tempdima
1264 \addtolength\evensidemargin{\fullwidth-\textwidth}
1265 \fi
1266 %</book>

```

\textheight 関連の調整。

```

1267 \@tempdimb=\textheight
1268 \advance\textheight-\topskip
1269 \advance\textheight.005pt\relax
1270 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1271 \advance\textheight\topskip
1272 \advance\@tempdimb-\textheight
1273 \advance\topmargin0.5\@tempdimb

```

\headheight 関連の調整。

```

1274 \@tempdima=\topskip
1275 \advance\headheight\@tempdima
1276 \advance\topmargin-\@tempdima

```

marginpar 関連の調整。

```

1277 \setlength\marginparsep{\columnsep}
1278 \setlength\marginparpush{\baselineskip}
1279 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1280 -\textwidth-10\jsc@mmm-\marginparsep}
1281 \ifbxjs@whole@zw@lines
1282 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1283 \fi

```

連動する変数。

```

1284 \maxdepth=.5\topskip
1285 \stockwidth=\paperwidth
1286 \stockheight=\paperheight
1287 }

```

\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。

```

1288 \edef\jsGeometryOptions{%
1289 \bxjs@layout@paper,\bxjs@layout}

```

■geometry パッケージを読み込む

ムニャムニャ。

```
1290 \def\bxjs@geometry@name{geometry}
1291 \ifbxjs@old@hook@system
1292   \let\bxjs@apply@bd@pre@geometry@hook\AtBeginDocument
1293 \else
1294   \def\bxjs@apply@bd@pre@geometry@hook{%
1295     \AddToHook{begindocument}{\bxjs@geometry@name}}
1296 \fi
```

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

```
1297 \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。

```
1298 \ifbxjs@papersize
1299   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1300     \PassOptionsToPackage{dvipdfm}{geometry}
1301   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1302     \PassOptionsToPackage{dvipdfm}{geometry}
1303   \fi\fi
1304   \let\bxPapersizeSpecialDone=t
1305 \else
1306   \PassOptionsToPackage{driver=none}{geometry}
1307 \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。

```
1308 \bxjs@apply@bd@pre@geometry@hook
1309 \bxjs@apply@bd@pre@geometry@hook{\bxjs@bd@pre@geometry@hook}
1310 \bxjs@apply@bd@pre@geometry@hook{\ImposeOldLuaTeXBehavior}
1311 \bxjs@preproc@layout
1312 \edef\bxjs@next{%
1313   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}%
1314 }\bxjs@next
1315 \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。

```
1316 \let\bxjs@geometry@driver\Gm@driver
1317 \bxjs@postproc@layout
```

geometry のドライバ自動判別に対する前処理。

```
1318 \g@addto@macro\bxjs@bd@pre@geometry@hook{%
```

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

```
1319 \ifpackageater{geometry}{2010/02/12}{\if%else
1320 \PackageError{bxjs@clsname
1321 {Your 'geometry' package is too old (< v5.0)}%
1322 {\@ehc}%
1323 \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。

```
1324 \ifjsWithpTeXng
1325 \ifx\Gm@driver\@empty
1326 \def\Gm@driver{pdftex}%
1327 \fi
1328 \fi}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

```
1329 \def\setpagelayout{%
1330 \bxjs@ifplus{\bxjs@setpagelayout@a\tw@}{\if%else
1331 \ifstar{\bxjs@setpagelayout@a\@ne}{\bxjs@setpagelayout@a\z@}}
1332 \def\bxjs@setpagelayout@a#1#2{%
1333 \ifcase#1% modify
1334 \def\bxjs@next{\ifjsc@mag truedimen,\fi #2}%
1335 \or% reset(*)
1336 \def\bxjs@next{reset,\bxjs@layout@paper,#2}%
1337 \or% semireset(+)
1338 \def\bxjs@next{reset,\bxjs@layout@paper,\bxjs@layout@base,#2}%
1339 \fi
1340 \bxjs@preproc@layout
1341 \edef\bxjs@next{%
1342 \noexpand\geometry{\bxjs@next}%
1343 }\bxjs@next
1344 \bxjs@postproc@layout}
```

■ geometry パッケージを読み込まない

geometry=user の場合の処理。

```
1345 \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。

```
1346 \g@addto@macro\bxjs@begin@document@hook{%
1347 \ifdim\textwidth=.5\maxdimen
1348 \ClassError{bxjs@clsname
1349 {Page layout is not properly set}%
1350 {\@ehd}%
```

```

1351 \fi}
1352 \def\jsUseMinimalPageLayout{%
1353 \setlength{\textwidth}{6.5in}%
1354 \setlength{\textheight}{8in}}

\setpagelayout はとりあえず無効にしておく。

1355 \let\bxjs@geometry@driver\relax
1356 \def\setpagelayout{%
1357 \bxjs@ifplus{\bxjs@pagelayout@a}{%else
1358 \@ifstar{\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1359 \def\bxjs@pagelayout@a#1{%
1360 \ClassError\bxjs@clsname
1361 {Command '\string\setpagelayout' is not supported,\MessageBreak
1362 because 'geometry' value is not 'class'}\@eha}
1363 %
1364 \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。

```
1365 %<*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。

```

1366 \setlength\topskip{1.38zw}%% from 10\jsc@empt (2016-08-17)
1367 \if@slide
1368 \setlength\headheight{0\jsc@empt}
1369 \else
1370 \setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-
    06-26)

```

1371 \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) になっていましたが、ここでは A4 判のときちょうど 1cm となるように、\paperheight の 0.03367 倍 (最小 \baselineskip) としました。書籍については、フッタは使わないことにして、ゼロにしました。

```
1372 %<*article|kiyou>
1373 \if@slide
1374   \setlength\footskip{0pt}
1375 \else
1376   \setlength\footskip{0.03367\paperheight}
1377   \ifdim\footskip<\baselineskip
1378     \setlength\footskip{\baselineskip}
1379   \fi
1380 \fi
1381 %</article|kiyou>
1382 %<jspf>\setlength\footskip{9\jsc@mmm}
1383 %<*book>
1384 \if@report
1385   \setlength\footskip{0.03367\paperheight}
1386   \ifdim\footskip<\baselineskip
1387     \setlength\footskip{\baselineskip}
1388   \fi
1389 \else
1390   \setlength\footskip{0pt}
1391 \fi
1392 %</book>
1393 %<*report>
1394 \setlength\footskip{0.03367\paperheight}
1395 \ifdim\footskip<\baselineskip
1396   \setlength\footskip{\baselineskip}
1397 \fi
1398 %</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)。

```
1399 %<*article>
1400 \if@slide
1401   \setlength\headsep{0\jsc@empt}
1402   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1403   \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1404 \else
1405   \setlength\headsep{\footskip}
1406   \addtolength\headsep{-\topskip}
```

```

1407 \fi
1408 %</article>
1409 %<*book>
1410 \if@report
1411   \setlength\headsep{\footskip}
1412   \addtolength\headsep{-\topskip}
1413 \else
1414   \setlength\headsep{6\jsc@mmm}
1415   \addtolength\headsep{-\topskip}%% added (2016-10-08)
1416   \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1417 \fi
1418 %</book>
1419 %<*report>
1420 \setlength\headsep{\footskip}
1421 \addtolength\headsep{-\topskip}
1422 %</report>
1423 %<*jspf>
1424 \setlength\headsep{9\jsc@mmm}
1425 \addtolength\headsep{-\topskip}
1426 %</jspf>
1427 %<*kiyou>
1428 \setlength\headheight{0\jsc@mpt}
1429 \setlength\headsep{0\jsc@mpt}
1430 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1431 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1432 %</kiyou>

```

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

```

1433 \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。

```

1434 \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。

```

1435 %<*article>
1436 \if@slide
1437   \setlength\fullwidth{0.9\paperwidth}
1438 \else
1439   \setlength\fullwidth{0.76\paperwidth}
1440 \fi
1441 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1442 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1443 \setlength\textwidth{\fullwidth}
1444 %</article>
1445 %<*book>
1446 \if@report
1447   \setlength\fullwidth{0.76\paperwidth}
1448 \else
1449   \setlength\fullwidth{\paperwidth}
1450   \addtolength\fullwidth{-36\jsc@mmm}
1451 \fi
1452 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1453 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1454 \setlength\textwidth{\fullwidth}
1455 \if@report \else
1456   \if@twocolumn \else
1457     \ifdim \fullwidth>40zw
1458       \setlength\textwidth{40zw}
1459     \fi
1460   \fi
1461 \fi
1462 %</book>
1463 %<*report>
1464 \setlength\fullwidth{0.76\paperwidth}
1465 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1466 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1467 \setlength\textwidth{\fullwidth}
1468 %</report>
1469 %<*jspf>
1470 \setlength\fullwidth{50zw}
1471 \addtolength\fullwidth{8\jsc@mmm}
1472 \setlength\textwidth{\fullwidth}
1473 %</jspf>
1474 %<*kiyou>
1475 \setlength\fullwidth{48zw}
1476 \addtolength\fullwidth{\columnsep}
1477 \setlength\textwidth{\fullwidth}
1478 %</kiyou>

```

\textheight 紙の高さ \paperheight は、1 インチと \topmargin と \headheight と \headsep と

`\textheight` と `\footskip` とページ下部の余白を加えたものです。

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

某学会誌スタイルでは 44 行にします。

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

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

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

```
1479 %<*article|book|report>
1480 \if@slide
1481   \setlength{\textheight}{0.95\paperheight}
1482 \else
1483   \setlength{\textheight}{0.83\paperheight}
1484 \fi
1485 \addtolength{\textheight}{-10\jsc@mppt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1486 \addtolength{\textheight}{-\headsep}
1487 \addtolength{\textheight}{-\footskip}
1488 \addtolength{\textheight}{-\topskip}
1489 \divide\textheight\baselineskip
1490 \multiply\textheight\baselineskip
1491 %</article|book|report>
1492 %<jspf>\setlength{\textheight}{51\baselineskip}
1493 %<kiyou>\setlength{\textheight}{47\baselineskip}
1494 \addtolength{\textheight}{\topskip}
1495 \addtolength{\textheight}{0.1\jsc@mppt}
1496 %<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 \LaTeX 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。

```
1497 \def\flushbottom{%
1498   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mppt}%
1499   \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 `\marginparpush`

みどうしの最小の間隔です。

```
1500 \setlength\marginparsep{\columnsep}
1501 \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。TEX は上・左マージンに `1truein` を挿入しますが，トンボ関係のオプションが指定されると pLATEX 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。

```
1502 \setlength{\oddsidemargin}{\paperwidth}
1503 \addtolength{\oddsidemargin}{-\fullwidth}
1504 \setlength{\oddsidemargin}{.5\oddsidemargin}
1505 \iftombow
1506   \addtolength{\oddsidemargin}{-1in}
1507 \else
1508   \addtolength{\oddsidemargin}{-\inv@mag in}
1509 \fi
1510 \setlength{\evensidemargin}{\oddsidemargin}
1511 \if@mparswitch
1512   \addtolength{\evensidemargin}{\fullwidth}
1513   \addtolength{\evensidemargin}{-\textwidth}
1514 \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。

```
1515 \setlength\marginparwidth{\paperwidth}
1516 \addtolength\marginparwidth{-\oddsidemargin}
1517 \addtolength\marginparwidth{-\inv@mag in}
1518 \addtolength\marginparwidth{-\textwidth}
1519 \addtolength\marginparwidth{-10\jsc@mmm}
1520 \addtolength\marginparwidth{-\marginparsep}
1521 \@tempdima=1zw
1522 \divide\marginparwidth\@tempdima
1523 \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つたので，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)。

```
1524 \setlength\topmargin{\paperheight}
1525 \addtolength\topmargin{-\textheight}
1526 \if@slide
1527   \addtolength\topmargin{-\headheight}
1528 \else
```

```

1529 \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1530 \fi
1531 \addtolength\topmargin{-\headsep}
1532 \addtolength\topmargin{-\footskip}
1533 \setlength\topmargin{0.5\topmargin}
1534 %<kiyou>\setlength\topmargin{81truebp}
1535 \iftombow
1536 \addtolength\topmargin{-1in}
1537 \else
1538 \addtolength\topmargin{-\inv@mag in}
1539 \fi
1540 %</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。

```
1541 \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。

```
1542 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

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

[2003-08-23] ちょっと増やしました。

```
1543 \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。

```
1544 \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。
 1545 `\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。
 1546 `\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。
 1547 `\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。
 1548 `\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。
 1549 `\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。
 1550 `\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。
 1551 `\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。
 1552 `\renewcommand{\dblfloatpagefraction}{.8}`

`\floatsep` `\floatsep` はページ上部・下部のフロート間の距離です。`\textfloatsep` はページ上部・
`\textfloatsep` 下部のフロートと本文との距離です。`\intextsep` は本文の途中に出力されるフロートと本
`\intextsep` 文との距離です。
 1553 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 1554 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`
 1555 `\setlength\intextsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`

`\dblfloatsep` 二段組のときの段抜きのフロートについての値です。
`\dbltextfloatsep` 1556 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 1557 `\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` 1558 `\setlength\@fptop{0\p@? \@plus 1fil}`
 1559 `\setlength\@fpsep{8\p@? \@plus 2fil}`
 1560 `\setlength\@fpbot{0\p@? \@plus 1fil}`

`\@dblftop` 段抜きフロートについての値です。

`\@dblfpsep` 1561 `\setlength\@dblftop{0\p@? \@plus 1fil}`

`\@dblfpbot` 1562 `\setlength\@dblfpsep{8\p@? \@plus 2fil}`

1563 `\setlength\@dblfpbot{0\p@? \@plus 1fil}`

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

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

```
1564 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1565 % \ifodd\c@page
1566 % \iftdir
1567 % \hbox{}\thispagestyle{empty}\newpage
1568 % \if@twocolumn\hbox{}\newpage\fi
1569 % \fi
1570 % \else
1571 % \ifydir
1572 % \hbox{}\thispagestyle{empty}\newpage
1573 % \if@twocolumn\hbox{}\newpage\fi
1574 % \fi
1575 % \fi\fi}
1576 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1577 % \ifodd\c@page
1578 % \ifydir
1579 % \hbox{}\thispagestyle{empty}\newpage
1580 % \if@twocolumn\hbox{}\newpage\fi
1581 % \fi
1582 % \else
1583 % \iftdir
1584 % \hbox{}\thispagestyle{empty}\newpage
1585 % \if@twocolumn\hbox{}\newpage\fi
1586 % \fi
1587 % \fi\fi}
1588 %\def\pltx@cleartooddpage{\clearpage\if@twoside
1589 % \ifodd\c@page\else
1590 % \hbox{}\thispagestyle{empty}\newpage
1591 % \if@twocolumn\hbox{}\newpage\fi
1592 % \fi\fi}
1593 %\def\pltx@cleartoevenpage{\clearpage\if@twoside
1594 % \ifodd\c@page
```

```

1595 \hbox{}\thispagestyle{empty}\newpage
1596 \if@twocolumn\hbox{}\newpage\fi
1597 \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。

```

1598 \let\pltx@cleartorightpage\pltx@cleartooddpage
1599 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

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

```

1600 %<*book|report>
1601 \if@openleft
1602 \let\cleardoublepage\pltx@cleartoleftpage
1603 \else\if@openright
1604 \let\cleardoublepage\pltx@cleartorightpage
1605 \fi\fi
1606 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

```

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

```

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

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

`\ps@empty` `empty` ページスタイルの定義です。L^AT_EX 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。

```
1607 % \def\ps@empty{%
1608 %   \let\mkboth\gobbletwo
1609 %   \let\oddhead\empty
1610 %   \let\oddfoot\empty
1611 %   \let\evenhead\empty
1612 %   \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。

```
1613 \def\ps@plainfoot{%
1614   \let\mkboth\gobbletwo
1615   \let\oddhead\empty
1616   \def\oddfoot{\normalfont\hfil\thepage\hfil}%
1617   \let\evenhead\empty
1618   \let\evenfoot\oddfoot}
1619 \def\ps@plainhead{%
1620   \let\mkboth\gobbletwo
1621   \let\oddfoot\empty
1622   \let\evenfoot\empty
1623   \def\evenhead{%
1624     \ifmparswitch \hss \fi
1625     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1626     \ifmparswitch\else \hss \fi}%
1627   \def\oddhead{%
1628     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}%
1629 %<book>\let\ps@plain\ps@plainhead
1630 %<!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。

```
1631 %<*article|slide>
1632 \if@twoside
1633   \def\ps@headings{%
1634     \let\oddfoot\empty
1635     \let\evenfoot\empty
1636     \def\evenhead{\ifmparswitch \hss \fi
1637       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1638     \ifmparswitch\else \hss \fi}%
1639   \def\oddhead{%
1640     \underline{%
1641       \hbox to \fullwidth{\rightmark\hfil\textbf{\thepage}}\hss}%
1642   }
```



```

1642 \let\mkboth\markboth
1643 \def\sectionmark##1{\markboth{%
1644 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1645 ##1}{}}%
1646 \def\subsectionmark##1{\markright{%
1647 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1648 ##1}}%
1649 }
1650 \else % if not twoside
1651 \def\ps@headings{%
1652 \let\@oddfoot\@empty
1653 \def\@oddhead{%
1654 \underline{%
1655 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1656 \let\mkboth\markboth
1657 \def\sectionmark##1{\markright{%
1658 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1659 ##1}}%
1660 \fi
1661 %</article|slide>

```

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

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

```

1662 %<*book|report>
1663 \def\bxjs@maybe@autoxspacing{%
1664 \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1665 \newif\ifomit@number
1666 \def\ps@headings{%
1667 \let\@oddfoot\@empty
1668 \let\@evenfoot\@empty
1669 \def\@evenhead{%
1670 \ifmparswitch \hss \fi
1671 \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1672 \textbf{\thepage}\hfil\leftmark}}}%
1673 \ifmparswitch\else \hss \fi}%
1674 \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1675 {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1676 \let\mkboth\markboth
1677 \def\chaptermark##1{\markboth{%
1678 \ifnum \c@secnumdepth >\m@ne
1679 \ifmainmatter
1680 \ifomit@number\else
1681 \@chapapp\thechapter\@chappos\hskip1\jsZw
1682 \fi

```

```

1683     \fi
1684     \fi
1685     ##1}{}}}%
1686     \def\sectionmark##1{\markright{%
1687         \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1688         ##1}}}%
1689 %</book|report>

    最後は学会誌の場合です。

1690 %<*jspf>
1691 \def\ps@headings{%
1692     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1693     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1694     \def\@oddhead{\normalfont\hfil \@title \hfil}
1695     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1696 %</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。

```

1697 \def\ps@myheadings{%
1698     \let\@oddfoot\@empty\let\@evenfoot\@empty
1699     \def\@evenhead{%
1700         \if@mparswitch \hss \fi%
1701         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1702         \if@mparswitch\else \hss \fi}%
1703     \def\@oddhead{%
1704         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1705     \let\@mkboth\@gobbletwo
1706 %<book|report> \let\chaptermark\@gobble
1707 \let\sectionmark\@gobble
1708 %<!book&!report> \let\subsectionmark\@gobble
1709 }

```

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 1710 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1711 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1712 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1713 % \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。

```
1714 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1715 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1716 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1717 \g@addto@macro\bxjs@begin@document@hook{\bxjs@decl@subtitle}
1718 \def\bxjs@decl@subtitle{%
1719   \global\let\bxjs@decl@subtitle\relax
1720   \ifx\subtitle\@undefined
1721     \global\let\subtitle\jsSubtitle
1722   \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。

```
1723 \def\bxjs@annihilate@subtitle{%
1724   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1725   \global\let\jsSubtitle\relax}
```

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

```
\eauthor 1726 %<*jspf>
\keywords 1727 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1728 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
1729 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1730 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1731 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1732 %</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。

```
1733 \def\plainifnotempty{%
1734   \ifx \@oddhead \@empty
1735     \ifx \@oddfoot \@empty
1736       \else
1737         \thispagestyle{plainfoot}%
1738       \fi
1739     \else
1740       \thispagestyle{plainhead}%
1741     \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。

```
1742 %<*article|book|report|slide>
1743 \if@titlepage
1744   \newcommand{\maketitle}{%
1745     \begin{titlepage}%
1746       \let\footnotesize\small
1747       \let\footnoterule\relax
1748       \let\footnote\thanks
1749       \null\vfil
1750       \if@slide
1751         {\footnotesize \@date}%
1752         \begin{center}
1753           \mbox{} \[1\jsZw]
1754           \large
1755           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1756           \jsc@smallskip
1757           \@title
1758           \ifx\bxjs@subtitle\@undefined\else
1759             \par\vskip\z@
1760             {\small \bxjs@subtitle\par}
1761           \fi
1762           \jsc@smallskip
1763           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1764           \vfill
1765           {\small \@author}%
1766         \end{center}
1767       \else
1768       \vskip 60\p@?
1769       \begin{center}%
1770         {\LARGE \@title \par}%
1771         \ifx\bxjs@subtitle\@undefined\else
1772           \vskip5\p@?
1773           {\normalsize \bxjs@subtitle\par}
1774         \fi
1775         \vskip 3em%
1776         {\large
1777           \lineskip .75em
1778           \begin{tabular}[t]{c}%
1779             \@author
1780           \end{tabular}\par}%
1781         \vskip 1.5em
1782         {\large \@date \par}%
1783       \end{center}%
1784     }
```

```

1784     \fi
1785     \par
1786     \@thanks\vfil\null
1787 \end{titlepage}%
1788 \setcounter{footnote}{0}%
1789 \global\let\thanks\relax
1790 \global\let\maketitle\relax
1791 \global\let\@thanks\@empty
1792 \global\let\@author\@empty
1793 \global\let\@date\@empty
1794 \global\let\@title\@empty
1795 \global\let\title\relax
1796 \global\let\author\relax
1797 \global\let\date\relax
1798 \global\let\and\relax
1799 \bxjs@annihilate@subtitle
1800 }%
1801 \else
1802 \newcommand{\maketitle}{\par
1803 \beginingroup
1804 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1805 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1806 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1807 \parindent 1\jsZw\noindent
1808 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1809 \if@twocolumn
1810 \ifnum \col@number=\@ne
1811 \@maketitle
1812 \else
1813 \twocolumn[\@maketitle]%
1814 \fi
1815 \else
1816 \newpage
1817 \global\@topnum\z@ % Prevents figures from going at top of page.
1818 \@maketitle
1819 \fi
1820 \plainifnotempty
1821 \@thanks
1822 \endgroup
1823 \setcounter{footnote}{0}%
1824 \global\let\thanks\relax
1825 \global\let\maketitle\relax
1826 \global\let\@thanks\@empty
1827 \global\let\@author\@empty
1828 \global\let\@date\@empty
1829 \global\let\@title\@empty
1830 \global\let\title\relax
1831 \global\let\author\relax
1832 \global\let\date\relax

```

```

1833 \global\let\and\relax
1834 \bxjs@annihilate@subtitle
1835 }

```

`\@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。

```

1836 \def\@maketitle{%
1837 \newpage\null
1838 \vskip 2em
1839 \begin{center}%
1840 \let\footnote\thanks
1841 {\LARGE \@title \par}%
1842 \ifx\bxjs@subtitle\@undefined\else
1843 \vskip3\p@?
1844 {\normalsize \bxjs@subtitle\par}
1845 \fi
1846 \vskip 1.5em
1847 {\large
1848 \lineskip .5em
1849 \begin{tabular}[t]{c}%
1850 \@author
1851 \end{tabular}\par}%
1852 \vskip 1em
1853 {\large \@date}%
1854 \end{center}%
1855 \par\vskip 1.5em
1856 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1857 }
1858 \fi
1859 %</article|book|report|slide>
1860 %<*jspf>
1861 \newcommand{\maketitle}{\par
1862 \begingroup
1863 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1864 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1865 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1866 \parindent 1\jsZw\noindent
1867 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1868 \twocolumn[\@maketitle]%
1869 \plainifnotempty
1870 \@thanks
1871 \endgroup
1872 \setcounter{footnote}{0}%
1873 \global\let\thanks\relax
1874 \global\let\maketitle\relax
1875 \global\let\@thanks\@empty
1876 \global\let\@author\@empty
1877 \global\let\@date\@empty
1878 % \global\let\@title\@empty % \@title は柱に使う
1879 \global\let\title\relax

```

```

1880 \global\let\author\relax
1881 \global\let\date\relax
1882 \global\let\and\relax
1883 \ifx\authors@mail\@undefined\else{%
1884     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1885     \footnotetext[0]{\itshape\authors@mail}%
1886 } \fi
1887 \global\let\authors@mail\@undefined}
1888 \def\@maketitle{%
1889     \newpage\null
1890     \vskip 6em % used to be 2em
1891     \begin{center}
1892         \let\footnote\thanks
1893         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par} \fi
1894         \lineskip .5em
1895         \ifx\@author\@undefined\else
1896             \vskip 1em
1897             \begin{tabular}[t]{c}%
1898                 \@author
1899             \end{tabular}\par
1900         \fi
1901         \ifx\@etitle\@undefined\else
1902             \vskip 1em
1903             {\large \@etitle \par}%
1904         \fi
1905         \ifx\@eauthor\@undefined\else
1906             \vskip 1em
1907             \begin{tabular}[t]{c}%
1908                 \@eauthor
1909             \end{tabular}\par
1910         \fi
1911         \vskip 1em
1912         \@date
1913     \end{center}
1914     \vskip 1.5em
1915     \centerline{\box\@abstractbox}
1916     \ifx\@keywords\@undefined\else
1917         \vskip 1.5em
1918         \centerline{\parbox{157\jscmmm}{\textsf{Keywords:}}\ \small\@keywords}}
1919     \fi
1920     \vskip 1.5em}
1921 %</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。

```
1922 \def\bxjs@label@sect#1{%
1923   \expandafter\ifx\csname label#1\endcsname\relax
1924     \csname the#1\endcsname
1925   \else \csname label#1\endcsname
1926   \fi}
1927 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}
```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```
1928 \ifnum\bxjs@label@section=\bxjs@label@section@@compat\else
1929 \def\@secapp{\presectionname}
1930 \def\@secpos{\postsectionname}
1931 \fi
```

`\labelsection` 節番号の出力書式。

```
1932 \ifnum\bxjs@label@section=\bxjs@label@section@@modern
1933 \def\labelsection{\@secapp\thesection\@secpos}
1934 \fi
```

■構成要素 `\@startsection` マクロは 6 個の必須引数と、オプションとして * と 1 個のオプション引数と 1 個の必須引数をとります。

`\@startsection{名}{レベル}{字下げ}{前アキ}{後アキ}{スタイル}`
`*[別見出し]{見出し}`

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

レベル 見出しの深さを示す数値です (chapter=1, section=2, ...)。この数値が `secnumdepth` 以下のとき見出し番号を出力します。

字下げ 見出しの字下げ量です。

前アキ この値の絶対値が見出し上側の空きです。負の場合は、見出し直後の段落をインデントしません。

後アキ 正の場合は、見出しの下側の空きです。負の場合は、絶対値が見出しの右の空きです (見出しと同じ行から本文を始めます)。

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```

1935 \def\@startsection#1#2#3#4#5#6{%
1936   \if@noskipsec \leavevmode \fi
1937   \par
1938 % 見出し上の空きを \@tempskipa にセットする
1939   \@tempskipa #4\relax
1940 % \@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
1941   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1942 % 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
1943   \ifdim \@tempskipa <\z@
1944     \@tempskipa -\@tempskipa \@afterindentfalse
1945   \fi
1946   \if@nobreak
1947 %   \everypar{\everyparhook}% これは間違い
1948     \everypar{}%
1949   \else
1950     \addpenalty\@secpenalty
1951 % 次の行は削除
1952 %   \addvspace\@tempskipa
1953 % 次の \noindent まで追加
1954     \ifdim \@tempskipa >\z@
1955       \if@slide\else
1956         \null
1957         \vspace*{-\baselineskip}%
1958       \fi
1959       \vskip\@tempskipa
1960     \fi
1961   \fi
1962   \noindent
1963 % 追加終わり
1964   \@ifstar
1965     {\@ssect{#3}{#4}{#5}{#6}}%
1966     {\@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。

```

1967 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
1968   \let\bxjs@if@ceph\@firstofone
1969 \else \let\bxjs@if@ceph\@gobble
1970 \fi

```

```

1971 \def\@sect#1#2#3#4#5#6[#7]#8{%
1972   \ifnum #2>\c@secnumdepth
1973     \let\@svsec\@empty
1974   \else
1975     \refstepcounter{#1}%
1976     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1977   \fi
1978 % 見出し後の空きを \@tempskipa にセット
1979   \@tempskipa #5\relax
1980 % 条件判断の順序を入れ替えました
1981   \ifdim \@tempskipa<\z@
1982     \def\@svsechd{%
1983       #6{\hskip #3\relax
1984         \@svsec #8}%
1985       \csname #1mark\endcsname{#7}%
1986       \addcontentsline{toc}{#1}{%
1987         \ifnum #2>\c@secnumdepth \else
1988           \protect\numberline{\bxjs@label@sect{#1}}%
1989         \fi
1990       #7}}% 目次にフルネームを載せるなら #8
1991   \else
1992     \begingroup
1993       \interlinepenalty \@M % 下から移動
1994       #6{%
1995         \@hangfrom{\hskip #3\relax\@svsec}%
1996       % \interlinepenalty \@M % 上に移動
1997       #8\@@par}%
1998     \endgroup
1999     \csname #1mark\endcsname{#7}%
2000     \addcontentsline{toc}{#1}{%
2001       \ifnum #2>\c@secnumdepth \else
2002         \protect\numberline{\bxjs@label@sect{#1}}%
2003       \fi
2004       #7}% 目次にフルネームを載せるならここは #8
2005   \fi
2006   \@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。

```

2007 \def\@xsect#1{%
2008 % 見出しの後ろの空きを \@tempskipa にセット
2009   \@tempskipa #1\relax
2010 % 条件判断の順序を変えました
2011   \ifdim \@tempskipa<\z@
2012     \@nbreakfalse
2013   \global\@noskipsecttrue

```

```

2014 \everypar{%
2015   \if@noskipsec
2016     \global\@noskipsecfalse
2017     {\setbox\z@\lastbox}%
2018     \clubpenalty\@M
2019     \begingroup \@svsechd \endgroup
2020     \unskip
2021     \@tempskipa #1\relax
2022     \hskip -\@tempskipa
2023   \else
2024     \clubpenalty \@clubpenalty
2025     \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
2026     \fi\bxjs@if@ceph\everyparhook}%
2027 \else
2028   \par \nobreak
2029   \vskip \@tempskipa
2030   \@afterheading
2031 \fi
2032 \if@slide
2033   {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
2034    \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
2035    \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
2036 \fi
2037 \par % 2000-12-18
2038 \ignorespaces}
2039 \def\@ssect#1#2#3#4#5{%
2040   \@tempskipa #3\relax
2041   \ifdim \@tempskipa<\z@
2042     \def\@svsechd{#4{\hskip #1\relax #5}}%
2043   \else
2044     \begingroup
2045       #4{%
2046         \@hangfrom{\hskip #1}%
2047         \interlinepenalty \@M #5\@par}%
2048     \endgroup
2049   \fi
2050   \@xsect{#3}}

```

■柱関係の命令

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

```

\subsectionmark 2051 \newcommand*\chaptermark[1]{}
2052 % \newcommand*\sectionmark[1]{}
\subsubsectionmark 2053 % \newcommand*\subsectionmark[1]{}
2054 % \newcommand*\subsubsectionmark[1]{}
\paragraphmark 2055 % \newcommand*\paragraphmark[1]{}
2056 % \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。

```
2057 %<!book&!report>\setcounter{secnumdepth}{3}
```

```
2058 %<book|report>\setcounter{secnumdepth}{2}
```

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

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

```
\c@subsection 2059 \newcounter{part}
```

```
2060 %<book|report>\newcounter{chapter}
```

```
\c@subsubsection 2061 %<book|report>\newcounter{section}[chapter]
```

```
\c@paragraph 2062 %<!book&!report>\newcounter{section}
```

```
\c@subparagraph 2063 \newcounter{subsection}[section]
```

```
2064 \newcounter{subsubsection}[subsection]
```

```
2065 \newcounter{paragraph}[subsubsection]
```

```
2066 \newcounter{subparagraph}[paragraph]
```

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

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

<code>\thesection</code>	<code>\arabic{COUNTER}</code>	1, 2, 3, ...
--------------------------	-------------------------------	--------------

<code>\thesubsection</code>	<code>\roman{COUNTER}</code>	i, ii, iii, ...
-----------------------------	------------------------------	-----------------

<code>\thesubsubsection</code>	<code>\Roman{COUNTER}</code>	I, II, III, ...
--------------------------------	------------------------------	-----------------

<code>\theparagraph</code>	<code>\alph{COUNTER}</code>	a, b, c, ...
----------------------------	-----------------------------	--------------

<code>\thesubparagraph</code>	<code>\Alph{COUNTER}</code>	A, B, C, ...
-------------------------------	-----------------------------	--------------

	<code>\kansuji{COUNTER}</code>	一, 二, 三, ...
--	--------------------------------	--------------

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```
2067 \renewcommand{\thepart}{\@Roman\c@part}
```

```
2068 %<*&!book&!report>
```

```
2069 \ifnum\bxjs@label@section=\bxjs@label@section@@compat
```

```
2070 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
```

```
2071 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
```

```
2072 \else
```

```
2073 \renewcommand{\thesection}{\@arabic\c@section}
```

```
2074 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
```

```
2075 \fi
```

```
2076 %</!book&!report>
```

```
2077 %<*&book|report>
```

```
2078 \renewcommand{\thechapter}{\@arabic\c@chapter}
```

```
2079 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
```

```
2080 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
```

```
2081 %</book|report>
```

```
2082 \renewcommand{\thesubsubsection}{%
```

```
2083 \thesubsection.\@arabic\c@subsubsection}
```

```
2084 \renewcommand{\theparagraph}{%
```

```
2085 \thesubsubsection.\@arabic\c@paragraph}
```

```

2086 \renewcommand{\thesubparagraph}{%
2087   \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。

```

2088 %<book|report>\newcommand{\@chapapp}{\prechaptername}
2089 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

[2017-03-05] `\frontmatter` と `\mainmatter` の2つの命令は, 改丁または改ページした後で `\pagenumbering{...}` でノンブルを1にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが, これではノンブルをリセットする際に偶奇逆転が起こる場合があります。 `openany` がどうにかに依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)。

```

2090 %<*book|report>
2091 \newcommand\frontmatter{%
2092   \pltx@cleartooddpage
2093   \@mainmatterfalse
2094   \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。

```

2095 \newcommand\mainmatter{%
2096   \pltx@cleartooddpage
2097   \@mainmattertrue
2098   \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。

```

2099 \newcommand\backmatter{%
2100   \if@openleft
2101     \cleardoublepage
2102   \else\if@openright
2103     \cleardoublepage
2104   \else
2105     \clearpage
2106   \fi\fi
2107   \@mainmatterfalse}
2108 %</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` は次のように使います。

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

まず `book` と `report` のクラス以外です。

```
2109 %<!*book&!report>
2110 \newcommand\part{%
2111   \if@noskipsec \leavevmode \fi
2112   \par
2113   \addvspace{4ex}%
2114   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2115   \secdef\@part\@spart}
2116 %</!*book&!report>
```

`book` および `report` クラスの場合は、少し複雑です。

```
2117 %<*book|report>
2118 \newcommand\part{%
2119   \if@openleft
2120     \cleardoublepage
2121   \else\if@openright
2122     \cleardoublepage
2123   \else
2124     \clearpage
2125   \fi\fi
2126   \thispagestyle{empty}% 欧文用標準スタイルでは plain
2127   \if@twocolumn
2128     \onecolumn
2129     \@restonecoltrue
2130   \else
2131     \@restonecolfalse
2132   \fi
2133   \null\vfil
2134   \secdef\@part\@spart}
2135 %</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。

```
2136 %<!*book&!report>
2137 \def\@part[#1]#2{%
2138   \ifnum \c@secnumdepth >\m@ne
2139     \refstepcounter{part}%
```

```

2140 \addcontentsline{toc}{part}{%
2141 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2142 \else
2143 \addcontentsline{toc}{part}{#1}%
2144 \fi
2145 \markboth{}{}%
2146 {\parindent\z@
2147 \raggedright
2148 \interlinepenalty \@M
2149 \normalfont
2150 \ifnum \c@secnumdepth >\m@ne
2151 \Large\headfont\prepartname\thepart\postpartname
2152 \par\nobreak
2153 \fi
2154 \huge \headfont #2%
2155 \markboth{}{}\par}%
2156 \nobreak
2157 \vskip 3ex
2158 \@afterheading}
2159 %</!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。

```

2160 %<*book|report>
2161 \def\@part[#1]#2{%
2162 \ifnum \c@secnumdepth >-2\relax
2163 \refstepcounter{part}%
2164 \addcontentsline{toc}{part}{%
2165 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2166 \else
2167 \addcontentsline{toc}{part}{#1}%
2168 \fi
2169 \markboth{}{}%
2170 {\centering
2171 \interlinepenalty \@M
2172 \normalfont
2173 \ifnum \c@secnumdepth >-2\relax
2174 \huge\headfont \prepartname\thepart\postpartname
2175 \par\vskip20\p@?
2176 \fi
2177 \Huge \headfont #2\par}%
2178 \@endpart}
2179 %</book|report>

```

\@spart 番号を付けない部です。

```

2180 %<*!book&!report>
2181 \def\@spart#1{%
2182 \parindent \z@ \raggedright
2183 \interlinepenalty \@M
2184 \normalfont

```

```

2185     \huge \headfont #1\par}%
2186     \nobreak
2187     \vskip 3ex
2188     \@afterheading}
2189 %</!book&!report>
2190 %<*book|report>
2191 \def\@spart#1{%
2192     \centering
2193     \interlinepenalty \@M
2194     \normalfont
2195     \Huge \headfont #1\par}%
2196     \@endpart}
2197 %</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。

```

2198 %<*book|report>
2199 \def\@endpart{\vfil\newpage
2200     \if@twoside
2201     \if@openleft %% added (2017/02/24)
2202     \null\thispagestyle{empty}\newpage
2203     \else\if@openright %% added (2016/12/13)
2204     \null\thispagestyle{empty}\newpage
2205     \fi\fi %% added (2016/12/13, 2017/02/24)
2206     \fi
2207     \if@restonecol
2208     \twocolumn
2209     \fi}
2210 %</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。

```

2211 %<*book|report>
2212 \newcommand{\chapter}{%
2213     \if@openleft\cleardoublepage\else
2214     \if@openright\cleardoublepage\else\clearpage\fi\fi
2215     \plainifnotempty % 元: \thispagestyle{plain}
2216     \global\@topnum\z@
2217     \if@english \@afterindentfalse \else \@afterindenttrue \fi
2218     \secdef
2219     {\@omit@numberfalse\@chapter}%
2220     {\@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。

```

2221 \def\@chapter[#1]#2{%
2222   \ifnum \c@secnumdepth >\m@ne
2223     \if@mainmatter
2224       \refstepcounter{chapter}%
2225       \typeout{\@chapapp\thechapter\@chappos}%
2226       \addcontentsline{toc}{chapter}%
2227         {\protect\numberline
2228 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2229         {\@chapapp\thechapter\@chappos}%
2230         #1}%
2231     \else\addcontentsline{toc}{chapter}{#1}\fi
2232   \else
2233     \addcontentsline{toc}{chapter}{#1}%
2234   \fi
2235   \chaptermark{#1}%
2236   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}%
2237   \addtocontents{lot}{\protect\addvspace{10\jsc@empt}}%
2238   \if@twocolumn
2239     \@topnewpage[\@makechapterhead{#2}]%
2240   \else
2241     \@makechapterhead{#2}%
2242     \@afterheading
2243   \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。

```

2244 \def\@makechapterhead#1{%
2245   \vspace*{2\Cvs}% 欧文は 50pt
2246   {\parindent \z@ \raggedright \normalfont
2247     \ifnum \c@secnumdepth >\m@ne
2248       \if@mainmatter
2249         \huge\headfont \@chapapp\thechapter\@chappos
2250         \par\nobreak
2251         \vskip \Cvs % 欧文は 20pt
2252       \fi
2253     \fi
2254     \interlinepenalty\@M
2255     \Huge \headfont #1\par\nobreak
2256     \vskip 3\Cvs}} % 欧文は 40pt

```

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

```

2257 \def\@schapter#1{%
2258   \chaptermark{#1}%
2259   \if@twocolumn
2260     \@topnewpage[\@makeschapterhead{#1}]%
2261   \else
2262     \@makeschapterhead{#1}\@afterheading
2263   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```
2264 \def\@makeschapterhead#1{%
2265   \vspace*{2\Cvs}% 欧文は 50pt
2266   {\parindent \z@ \raggedright
2267    \normalfont
2268    \interlinepenalty\@M
2269    \Huge \headfont #1\par\nobreak
2270    \vskip 3\Cvs}} % 欧文は 40pt
2271 %</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。

```
2272 \if@twocolumn
2273   \newcommand{\section}{%
2274     <jspf>\ifx\maketitle\relax\else\maketitle\fi
2275     \@startsection{section}{1}{\z@}%
2276     <!kiyou> {0.6\Cvs}{0.4\Cvs}%
2277     <kiyou> {\Cvs}{0.5\Cvs}%
2278     % {\normalfont\large\headfont\@secapp}}
2279     {\normalfont\large\headfont\raggedright}}
2280 \else
2281   \newcommand{\section}{%
2282     \if@slide\clearpage\fi
2283     \@startsection{section}{1}{\z@}%
2284     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2285     {.5\Cvs \@plus.3\Cdp}% 後アキ
2286     % {\normalfont\Large\headfont\@secapp}}
2287     {\normalfont\Large\headfont\raggedright}}
2288 \fi
```

`\subsection` 同上です。

```
2289 \if@twocolumn
2290   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2291     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2292     {\normalfont\normalsize\headfont}}
2293 \else
2294   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
2295     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2296     {.5\Cvs \@plus.3\Cdp}% 後アキ
2297     {\normalfont\large\headfont}}
2298 \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)。

```
2299 \if@twocolumn
```

```

2300 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2301     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2302     {\normalfont\normalsize\headfont}}
2303 \else
2304 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
2305     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2306     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2307     {\normalfont\normalsize\headfont}}
2308 \fi

```

`\paragraph` 見出しの後ろで改行されません。

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

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

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

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

段落のマーク（■）が必ず和文フォントで出力されるようにする。

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

```

2309 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2310 \let\bxjs@org@paragraph@mark\jsParagraphMark
2311 \ifx\bxjs@paragraph@mark\empty
2312   \let\jsParagraphMark\empty
2313 \else\ifx\bxjs@paragraph@mark\undefined\else
2314   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2315 \fi\fi
2316 \let\jsJaChar\empty
2317 \if@twocolumn
2318   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2319     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2320     <jspf> {\normalfont\normalsize\headfont}}
2321     <!\jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2322 \else
2323   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2324     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2325     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2326     <jspf> {\normalfont\normalsize\headfont}}
2327     <!\jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
2328 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

2329 \if@twocolumn
2330   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2331     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2332     {\normalfont\normalsize\headfont}}
2333 \else
2334   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2335     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2336     {\normalfont\normalsize\headfont}}
2337 \fi

```

8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

2338 \if@slide
2339   \setlength\leftmargini{1\jsZw}
2340 \else
2341   \if@twocolumn
2342     \setlength\leftmargini{2\jsZw}
2343   \else
2344     \setlength\leftmargini{3\jsZw}
2345   \fi
2346 \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 2347 \if@slide
\leftmarginiv 2348   \setlength\leftmarginii {1\jsZw}
\leftmarginiv 2349   \setlength\leftmarginiii{1\jsZw}
\leftmarginiv 2350   \setlength\leftmarginiv {1\jsZw}
2351   \setlength\leftmarginv {1\jsZw}
2352   \setlength\leftmarginvi {1\jsZw}
2353 \else
2354   \setlength\leftmarginii {2\jsZw}
2355   \setlength\leftmarginiii{2\jsZw}
2356   \setlength\leftmarginiv {2\jsZw}
2357   \setlength\leftmarginv {1\jsZw}
2358   \setlength\leftmarginvi {1\jsZw}
2359 \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。

```

2360 \setlength \labelsep {0.5\jsZw} % .5em
2361 \setlength \labelwidth{\leftmargini}
2362 \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。

```

2363 \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 2364 \@beginparpenalty -\@lowpenalty
\@itempenalty 2365 \@endparpenalty -\@lowpenalty
2366 \@itempenalty -\@lowpenalty

```

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

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

```

2367 \def\@listi{\leftmargin\leftmargini
2368 \parsep \z@
2369 \topsep 0.5\baselineskip
2370 \itemsep \z@ \relax}
2371 \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）。

```

2372 \@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 2373 \def\@listii{\leftmargin\leftmarginii
\@listiv 2374 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2375 \topsep \z@
\@listv 2376 \parsep \z@
\@listvi 2377 \itemsep\parsep}
2378 \def\@listiii{\leftmargin\leftmarginiii
2379 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2380 \topsep \z@
2381 \parsep \z@
2382 \itemsep\parsep}
2383 \def\@listiv {\leftmargin\leftmarginiv
2384 \labelwidth\leftmarginiv
2385 \advance\labelwidth-\labelsep}
2386 \def\@listv {\leftmargin\leftmarginv
2387 \labelwidth\leftmarginv
2388 \advance\labelwidth-\labelsep}

```

```

2389 \def\@listvi {\leftmargin\leftmarginvi
2390             \labelwidth\leftmarginvi
2391             \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。

```

2392 \renewcommand{\theenumi}{\@arabic\c@enumi}
2393 \renewcommand{\theenumii}{\@alph\c@enumii}
2394 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2395 \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。

```

2396 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2397 \newcommand*{\jsInJaParen}[1]{%
2398   \bxjs@dust\jsInhibitGlue (#1) \jsInhibitGlue\bxjs@dust}
2399 \newcommand{\labelenumi}{\theenumi.}
2400 \newcommand{\labelenumii}{\jsInJaParen{\theenumii}}
2401 \newcommand{\labelenumiii}{\theenumiii.}
2402 \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 2403 \renewcommand{\p@enumii}{\theenumi}
2404 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii )}
2405 \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` 2406 `\newcommand\labelitemi{\textbullet}`

`\labelitemiii` 2407 `\newcommand\labelitemii{\normalfont\bfseries \textendash}`

`\labelitemiv`

```

2408 \newcommand\labelitemiii{\textasteriskcentered}
2409 \newcommand\labelitemiv{\textperiodcentered}

```

■description 環境

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

```

2410 \newenvironment{description}{%
2411   \list{}{%
2412     \labelwidth=\leftmargin
2413     \labelsep=1\jsZw
2414     \advance \labelwidth by -\labelsep
2415     \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。

```

2416 \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。

```

2417 %<*book|report>
2418 \newenvironment{chapterabstract}{%
2419   \begin{list}{}{%
2420     \listparindent=1\jsZw
2421     \itemindent=\listparindent
2422     \rightmargin=0pt
2423     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2424 %</book|report>

```

“普通の” **abstract** 環境の定義。

```

2425 %<*article|report|slide>

```

```

2426 \newbox\@abstractbox
2427 \if@titlepage
2428   \newenvironment{abstract}{%
2429     \titlepage
2430     \null\vfil
2431     \@beginparpenalty\@lowpenalty
2432     \begin{center}%
2433       \headfont \abstractname
2434       \@endparpenalty\@M
2435     \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。

```

2436   \par}%
2437   {\par\vfil\null\endtitlepage}
2438 \else
2439   \newenvironment{abstract}{%
2440     \if@twocolumn
2441       \ifx\maketitle\relax
2442         \section*{\abstractname}%
2443       \else
2444         \global\setbox\@abstractbox\hbox\bgroup
2445         \begin{minipage}[b]{\textwidth}
2446           \small\parindent1\jsZw
2447           \begin{center}%
2448             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2449           \end{center}%
2450           \list{}{%
2451             \listparindent\parindent
2452             \itemindent \listparindent
2453             \rightmargin \leftmargin}%
2454           \item\relax
2455         \fi
2456       \else
2457         \small
2458         \begin{center}%
2459           {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2460         \end{center}%
2461         \list{}{%
2462           \listparindent\parindent
2463           \itemindent \listparindent
2464           \rightmargin \leftmargin}%
2465         \item\relax
2466       \fi}{\if@twocolumn
2467         \ifx\maketitle\relax
2468         \else
2469           \endlist\end{minipage}\egroup
2470         \fi
2471       \else
2472         \endlist

```



```

2473     \fi}
2474 \fi
2475 %</article|report|slide>
2476 %<*jspf>
2477 \newbox\@abstractbox
2478 \newenvironment{abstract}{%
2479     \global\setbox\@abstractbox\hbox\bgroup
2480     \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2481     \small
2482     \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2483     {\end{minipage}\egroup}
2484 %</jspf>

```

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

```

2485 %<*book|report>
2486 \ifbxjs@force@chapterabstract
2487     \let\abstract\chapterabstract
2488     \let\endabstract\endchapterabstract
2489 \fi
2490 %</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。

```

2491 %<*jspf>
2492 %\newbox\@keywordsbox
2493 %\newenvironment{keywords}{%
2494 %     \global\setbox\@keywordsbox\hbox\bgroup
2495 %     \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2496 %         \small\parindent0\jsZw}%
2497 %     {\end{minipage}\egroup}
2498 %</jspf>

```

■verse 環境

verse 詩のための verse 環境です。

```

2499 \newenvironment{verse}{%
2500     \let \=\@centercr
2501     \list{}{%
2502         \itemsep \z@
2503         \itemindent -2\jsZw % 元: -1.5em
2504         \listparindent\itemindent
2505         \rightmargin \z@
2506         \advance\leftmargin 2\jsZw}% 元: 1.5em
2507     \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。

```
2508 \newenvironment{quotation}{%
2509   \list{}{%
2510     \listparindent\parindent
2511     \itemindent\listparindent
2512     \rightmargin \z@}%
2513   \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。

```
2514 \newenvironment{quote}%
2515   {\list{}{\rightmargin\z@}\item\relax}{\endlist}
```

■定理など `ltthm.dtx` 参照。たとえば次のように定義します。

```
\newtheorem{definition}{定義}
\newtheorem{axiom}{公理}
\newtheorem{theorem}{定理}
```

[2001-04-26] 定理の中はイタリック体になりましたが、これでは和文がゴシック体になってしまうので、`\itshape` を削除しました。

[2009-08-23] `\bfseries` を `\headfont` に直し、`\labelsep` を 1zw にし、括弧を全角にしました。

```
2516 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2517   \item[\hskip \labelsep{\headfont #1\ #2}]}
2518 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2519   \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 p_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 p_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。

```
2520 \newenvironment{titlepage}{%
2521   %<book> \pltx@cleartooddpage %% 2017-02-24
2522   \if@twocolumn
2523     \@restonecoltrue\onecolumn
2524   \else
2525     \@restonecolfalse\newpage
2526   \fi}
```

```

2527 \thispagestyle{empty}%
2528 \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
24
2529 }%
2530 {\if@restonecol\twocolumn \else \newpage \fi
2531 \if@twoside\else
2532 \setcounter{page}\@ne
2533 \fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```

2534 %<*!book&!report>
2535 \newcommand{\appendix}{\par
2536 \setcounter{section}{0}%
2537 \setcounter{subsection}{0}%
2538 \ifnum\bxjs@label@section=\bxjs@label@section@@compat
2539 \gdef\presectionname{\appendixname}%
2540 \gdef\postsectionname{}}%
2541 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2542 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2543 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2544 \else
2545 \gdef\@secapp{\appendixname}%
2546 \gdef\@secpos{}}%
2547 \gdef\thesection{\@Alph\c@section}%
2548 \fi}
2549 %</!book&!report>
2550 %<*book|report>
2551 \newcommand{\appendix}{\par
2552 \setcounter{chapter}{0}%
2553 \setcounter{section}{0}%
2554 \gdef\@chapapp{\appendixname}%
2555 \gdef\@chappos{}}%
2556 \gdef\thechapter{\@Alph\c@chapter}}
2557 %</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。

```
2558 \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。

```
2559 \setlength\tabcolsep{6\p@?}
```

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

2560 \setlength\arrayrulewidth{.4\p@}

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

2561 \setlength\doublerulesep{2\p@}

■tabbing 環境

\tabbingsep \' コマンドで入るアキです。

2562 \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。

2563 \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。

2564 \setlength\fboxsep{3\p@?}

2565 \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。

2566 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}

2567 %<*book|report>

2568 \@addtoreset{equation}{chapter}

2569 \renewcommand\theequation

2570 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}

2571 %</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。

2572 % \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。

2573 % \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。

2574 % \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。

```
2575 %<*!book&!report>
2576 \newcounter{figure}
2577 \renewcommand \thefigure {\@arabic\c@figure}
2578 %</!book&!report>
2579 %<*book|report>
2580 \newcounter{figure}[chapter]
2581 \renewcommand \thefigure
2582     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2583 %</book|report>
```

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

```
\ext@figure 2584 \def\fps@figure{tbp}
2585 \def\ftype@figure{1}
\fnum@figure 2586 \def\ext@figure{lof}
2587 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2588 \newenvironment{figure}%
2589     {\@float{figure}}%
2590     {\end@float}
2591 \newenvironment{figure*}%
2592     {\@dblfloat{figure}}%
2593     {\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。

```

2594 %<!*book&!report>
2595 \newcounter{table}
2596 \renewcommand\thetable{\@arabic\c@table}
2597 %</!*book&!report>
2598 %<*book|report>
2599 \newcounter{table}[chapter]
2600 \renewcommand \thetable
2601     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2602 %</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 2603 \def\fps@table{tbp}
2604 \def\ftype@table{2}
\fnun@table 2605 \def\ext@table{lot}
2606 \def\fnun@table{\tablename\nobreak\thetable}

```

table * は段抜きのフロートです。

```

table* 2607 \newenvironment{table}%
2608     {\@float{table}}%
2609     {\end@float}
2610 \newenvironment{table*}%
2611     {\@dblfloat{table}}%
2612     {\end@dblfloat}

```

8.6 キャプション

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

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

```

2613 \newlength\abovecaptionskip
2614 \newlength\belowcaptionskip
2615 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2616 \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。

```

2617 %<!*jspf>
2618 % \long\def\@makecaption#1#2{\small
2619 %     \advance\leftskip10\jsc@mmm
2620 %     \advance\rightskip10\jsc@mmm
2621 %     \vskip\abovecaptionskip
2622 %     \sbox\@tempboxa{#1\hskip1\jszw\relax #2}%

```

```

2623 % \ifdim \wd\@tempboxa >\hsize
2624 % #1\hskip1\jsZw\relax #2\par
2625 % \else
2626 % \global \@minipagefalse
2627 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2628 % \fi
2629 % \vskip\belowcaptionskip}}
2630 \long\def\@makecaption#1#2{\small
2631 \advance\leftskip .0628\linewidth
2632 \advance\rightskip .0628\linewidth
2633 \vskip\abovecaptionskip
2634 \sbox\@tempboxa{#1\zwspace#2}%
2635 \ifdim \wd\@tempboxa <\hsize \centering \fi
2636 #1\zwspace#2\par
2637 \vskip\belowcaptionskip}}
2638 %</!jspf>
2639 %<*jspf>
2640 \long\def\@makecaption#1#2{%
2641 \vskip\abovecaptionskip
2642 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2643 \ifdim \wd\@tempboxa >\hsize
2644 {\small\sffamily
2645 \list{#1}{%
2646 \renewcommand{\makelabel}[1]{##1\hfil}
2647 \itemsep \z@
2648 \itemindent \z@
2649 \labelsep \z@
2650 \labelwidth 11\jsc@mmm
2651 \listparindent\z@
2652 \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2653 \else
2654 \global \@minipagefalse
2655 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2656 \fi
2657 \vskip\belowcaptionskip}
2658 %</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 `\scrDeclareOldFontCommand` に倣い、これらの命令を使うときには警告を発することにしました。

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

```

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

```

```

2659 \newif\if@jsc@warnoldfontcmd
2660 \@jsc@warnoldfontcmdtrue
2661 \newif\if@jsc@warnoldfontcmdexception
2662 \@jsc@warnoldfontcmdexceptionfalse

```

```

\jsc@DeclareOldFontCommand

```

```

2663 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2664   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2665   \DeclareOldFontCommand{#1}{%
2666     \bxjs@oldfontcmd{#1}#2%
2667   }{%
2668     \bxjs@oldfontcmd{#1}#3%
2669   }%
2670 }
2671 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2672   \ClassInfo\bxjs@clsname
2673   {Old font command '\string#1' is used!!\MessageBreak
2674     The first occurrence is}%
2675 }

```

```

\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。

```

```

2676 \newcommand*{\allowoldfontcommands}{%
2677   \@jsc@warnoldfontcmdexceptiontrue}
2678 \newcommand*{\disallowoldfontcommands}{%
2679   \@jsc@warnoldfontcmdexceptionfalse}

2680 \let\bxjs@oldfontcmd@list\@empty
2681 \def\bxjs@oldfontcmd#1{%
2682   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2683 \def\bxjs@oldfontcmd@a#1#2{%
2684   \if@jsc@warnoldfontcmdexception\else
2685     \global\@jsc@warnoldfontcmdfalse
2686     \ifx#1\relax
2687       \global\let#1=t%
2688       \jsc@warnoldfontcmd{#2}%
2689     \fi
2690   \fi}
2691 \def\bxjs@warnoldfontcmd@final{%
2692 % \par
2693   \global\let\bxjs@warnoldfontcmd@final\@empty

```



```

2694 \let\@tempa\@empty
2695 \def\do##1{%
2696   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2697     \edef\@tempa{\@tempa \space\string##1}\fi}
2698 \bxjs@oldfontcmd@list
2699 \ifx\@tempa\@empty\else
2700   \ClassWarningNoLine\bxjs@clsname
2701     {Some old font commands were used in text:\MessageBreak
2702       \space\@tempa\MessageBreak
2703       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2704       new font selection scheme called NFSS2 with several\MessageBreak
2705       new, combinable font commands. The
2706       class provides\MessageBreak
2707       the old font commands only for compatibility}
2708 \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。

```

2709 \def\bxjs@warnoldfontcmd@kick@final{%
2710   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2711 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2712 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2713 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2714 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2715 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2716 \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。

```

2717 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2718 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2719 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2720 \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 2721 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2722 \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。

```
2723 \newcommand\@pnumwidth{1.55em}
2724 \newcommand\@tocrmarg{2.55em}
2725 \newcommand\@dotsep{4.5}
2726 %<!book&!report>\setcounter{tocdepth}{2}
2727 %<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)

```
2728 \newdimen\jsc@tocl@width
2729 \newcommand{\tableofcontents}{%
2730 %<*book|report>
2731   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2732   \settowidth\@tempdima{\headfont\appendixname}%
2733   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2734   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2735   \if@twocolumn
2736     \@restonecoltrue\onecolumn
2737   \else
2738     \@restonecolfalse
2739   \fi
2740   \chapter*{\contentsname}%
2741   \@mkboth{\contentsname}{}%
2742 %</book|report>
2743 %<*!book&!report>
2744   \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2745   \settowidth\@tempdima{\headfont\appendixname}%
2746   \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2747   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2748   \section*{\contentsname}%
2749   \@mkboth{\contentsname}{\contentsname}%
2750 %</!book&!report>
2751   \@starttoc{toc}%
2752 %<book|report> \if@restonecol\twocolumn\fi
2753 }
```

`\l@part` 部の目次です。

```
2754 \newcommand*{\l@part}[2]{%
2755   \ifnum \c@tocdepth >-2\relax
2756   %<!book&!report> \addpenalty\@secpenalty
2757   %<book|report> \addpenalty{-\@highpenalty}%
2758   \addvspace{2.25em \@plus\p@}%
2759   \begingroup
2760     \parindent \z@
2761     % \@pnumwidth should be \@tocrmarg
2762     % \rightskip \@pnumwidth
2763     \rightskip \@tocrmarg
2764     \parfillskip -\rightskip
2765     {\leavevmode
2766       \large \headfont
2767       \setlength\@lnumwidth{4\jsZw}%
2768       #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2769     \nobreak
2770 %<book|report> \global\@nobreaktrue
2771 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
```

```

2772 \endgroup
2773 \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)

```

2774 %<*book|report>
2775 \newcommand*{\l@chapter}[2]{%
2776 \ifnum \c@tocdepth >\m@ne
2777 \addpenalty{-\@highpenalty}%
2778 \addvspace{1.0em \@plus\p@?}
2779 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2780 \begingroup
2781 \parindent\z@
2782 % \rightskip\@pnumwidth
2783 \rightskip\@tocrmarg
2784 \parfillskip-\rightskip
2785 \leavevmode\headfont
2786 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2787 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2788 \advance\leftskip\@lnumwidth \hskip-\leftskip
2789 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2790 \penalty\@highpenalty
2791 \endgroup
2792 \fi}
2793 %</book|report>

```

`\l@section` 節の目次です。

```

2794 %<!*book&!report>
2795 \newcommand*{\l@section}[2]{%
2796 \ifnum \c@tocdepth >\z@
2797 \addpenalty{\@secpenalty}%
2798 \addvspace{1.0em \@plus\p@?}%
2799 \begingroup
2800 \parindent\z@
2801 % \rightskip\@pnumwidth
2802 \rightskip\@tocrmarg
2803 \parfillskip-\rightskip
2804 \leavevmode\headfont
2805 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2806 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2807 \advance\leftskip\@lnumwidth \hskip-\leftskip
2808 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2809 \endgroup
2810 \fi}
2811 %</!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。

```

2812 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

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

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

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

```
\l@subparagraph 2813 %<!*book&!report>
2814 % \newcommand*{\l@section} {\@dottedtocline{2}{1.5em}{2.3em}}
2815 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2816 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2817 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2818 %
2819 % \newcommand*{\l@section} {\@dottedtocline{2}{1zw}{3zw}}
2820 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2821 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2822 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2823 %
2824 \newcommand*{\l@section}{%
2825     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2826     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2827 \newcommand*{\l@subsubsection}{%
2828     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2829     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2830 \newcommand*{\l@paragraph}{%
2831     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2832     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2833 \newcommand*{\l@subparagraph}{%
2834     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2835     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2836 %</!*book&!report>
2837 %<!*book|report>
2838 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2839 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2840 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2841 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2842 \newcommand*{\l@section}{%
2843     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2844     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2845 \newcommand*{\l@section}{%
2846     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2847     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2848 \newcommand*{\l@subsubsection}{%
2849     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2850     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2851 \newcommand*{\l@paragraph}{%
2852     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2853     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2854 \newcommand*{\l@subparagraph}{%
2855     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
```

```

2856 \dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2857 %</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。

```

2858 \newdimen\@lnumwidth
2859 \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 \cdot\hss}\hfill}
```

とします。

```

2860 \def\jsTocLine{\leaders\hbox{%
2861   $\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep mu$}\hfill}
2862 \def\dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2863   \vskip \z@ \@plus.2\p@?
2864   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2865     \parindent #2\relax\@afterindenttrue
2866     \interlinepenalty\@M
2867     \leavevmode
2868     \@lnumwidth #3\relax
2869     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2870     {#4}\nobreak
2871     \jsTocLine \nobreak\hb@xt@\@pnumwidth{%
2872       \hfil\normalfont \normalcolor #5\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2873 \newcommand{\listoffigures}{%
2874 %<*book|report>
2875 \if@twocolumn\@restonecoltrue\onecolumn
2876 \else\@restonecolfalse\fi
2877 \chapter*{\listfigurename}%
2878 \mkboth{\listfigurename}{}%
2879 %</book|report>
2880 %<!*book&!report>
2881 \section*{\listfigurename}%
2882 \mkboth{\listfigurename}{\listfigurename}%
2883 %</*!book&!report>
2884 \@starttoc{lof}%

```

```
2885 %<book|report> \if@restonecol\twocolumn\fi
2886 }
```

`\l@figure` 図目次の項目を出力します。

```
2887 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```
2888 \newcommand{\listoftables}{%
2889 %<*book|report>
2890 \if@twocolumn\@restonecoltrue\onecolumn
2891 \else\@restonecolfalse\fi
2892 \chapter*{\listtablename}%
2893 \@mkboth{\listtablename}{}%
2894 %</book|report>
2895 %<!*book&!report>
2896 \section*{\listtablename}%
2897 \@mkboth{\listtablename}{\listtablename}%
2898 %<!/book&!report>
2899 \@starttoc{lot}%
2900 %<book|report> \if@restonecol\twocolumn\fi
2901 }
```

`\l@table` 表目次は図目次と同じです。

```
2902 \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。

```
2903 \newdimen\bibindent
2904 \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。

```
2905 \newenvironment{thebibliography}[1]{%
2906 \@jsc@warnoldfontcmdexceptiontrue
2907 \global\let\presectionname\relax
2908 \global\let\postsectionname\relax
2909 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2910 %<*kiyou>
2911 \vspace{1.5\baselineskip}
2912 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2913 \vspace{0.5\baselineskip}
2914 %</kiyou>
2915 %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2916 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%

```

```

2917 \list{\@biblabel{\@arabic\c@enumiv}}%
2918 {\settowidth\labelwidth{\@biblabel{#1}}%
2919 \leftmargin\labelwidth
2920 \advance\leftmargin\labelsep
2921 \@openbib@code
2922 \usecounter{enumiv}%
2923 \let\p@enumiv\@empty
2924 \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2925 %<kiyou> \small
2926 \sloppy
2927 \clubpenalty4000
2928 \@clubpenalty\clubpenalty
2929 \widowpenalty4000%
2930 \sfcode`\.\@m}
2931 {\def\@noitemerr
2932 {\@latex@warning{Empty `thebibliography' environment}}}%
2933 \endlist}

```

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

```

2934 \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。

```

2935 \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。

```

2936 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

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

```

2937 % \def\@citex[#1]#2{\leavevmode
2938 % \let\@citea\@empty
2939 % \@cite{\@for\@citeb:=#2\do
2940 % {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2941 % \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
2942 % \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2943 % \ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2944 % \G@refundefinedtrue
2945 % \@latex@warning
2946 % {Citation `{\@citeb' on page \thepage \space undefined}}%
2947 % {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2948 % \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。

```
2949 % \DeclareRobustCommand\cite{\unskip
2950 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2951 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempwa
2952 %   , \jsInhibitGlue\ #2\fi}) }}$}
```

10.3 索引

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

```
2953 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2954   \if@twocolumn
2955     \onecolumn\@restonecolfalse
2956   \else
2957     \clearpage\@restonecoltrue
2958   \fi
2959   \columnseprule.4pt \columnsep 2\jsZw
2960   \ifx\multicols\@undefined
2961 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2962 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2963 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2964 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2965   \else
2966     \ifdim\textwidth<\fullwidth
2967       \setlength{\evensidemargin}{\oddsidemargin}
2968       \setlength{\textwidth}{\fullwidth}
2969       \setlength{\linewidth}{\fullwidth}
2970 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
2971 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2972 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2973 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2974   \else
2975 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}%
2976 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2977 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2978 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2979   \fi
2980   \fi
2981 %<book|report>      \@mkboth{\indexname}{}%
2982 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2983   \plainifnotempty % \thispagestyle{plain}
2984   \parindent\z@
2985   \parskip\z@ \@plus .3\p@?\relax
2986   \let\item\@idxitem
2987   \raggedright
2988   \footnotesize\narrowbaselines
2989 }{
```

```

2990 \ifx\multicols\@undefined
2991 \if@restonecol\onecolumn\fi
2992 \else
2993 \end{multicols}
2994 \fi
2995 \clearpage
2996 }

```

`\@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 2997 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2998 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2999 \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。

```

3000 \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。

```

3001 \newcommand\seename{\if@english see\else →\fi}
3002 \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。p_{La}T_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。

```

3003 \ifx\pltx@foot@penalty\@undefined
3004 \let\footnotes@ve=\footnote
3005 \def\footnote{\jsInhibitGlue\footnotes@ve}
3006 \let\footnotemarks@ve=\footnotemark
3007 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
3008 \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 p_TE_X では脚注番号のまわりにスペースが入りすぎることを防ぐため、北川さんのパッチ [qa:57090] を取り込みました。

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

[2016-07-11] コミュニティ版 p_AT_EX の変更に従いました (Thanks: 角藤さん)。p_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。

p_TE_X 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
3009 \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。

```
3010 % \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。

```
3011 \renewcommand{\footnoterule}{%
3012   \kern-2.6\p@? \kern-.4\p@
3013   \hrule width .4\columnwidth
3014   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
3015 %<book|report>\@addtoreset{footnote}{chapter}
```

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

[2016-08-25] コミュニティ版 p_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 p_AT_EX のバグ修正に従いました。

[2016-11-29] 古い p_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。

```
3016 \long\def\@footnotetext{%
3017   \insert\footins\bgroup
3018     \normalfont\footnotesize
3019     \interlinepenalty\interfootnotelinepenalty
3020     \splittopskip\footnotesep
3021     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
3022     \hsize\columnwidth \@parboxrestore
3023     \protected@edef\@currentlabel{%
3024       \csname p@footnote\endcsname\@thefnmark
```

```

3025 }%
3026 \color@begingroup
3027 \makefnintext{%
3028 \rule{z@\footnotesep\ignorespaces}%
3029 \futurelet\jsc@next\jsc@fo@t}
3030 \def\jsc@fo@t{\ifcat\bgroup\noexpand\jsc@next \let\jsc@next\jsc@f@t
3031 \else \let\jsc@next\jsc@f@t\fi \jsc@next}
3032 \def\jsc@f@t{\bgroup\aftergroup\jsc@@foot\let\jsc@next}
3033 \def\jsc@f@t#1{#1\jsc@@foot}
3034 \def\jsc@@foot{\@finalstrut\strutbox\color@endgroup\egroup
3035 \ifx\pltx@foot@penalty\undefined\else
3036 \ifhmode\null\fi
3037 \ifnum\pltx@foot@penalty=z@\else
3038 \penalty\pltx@foot@penalty
3039 \pltx@foot@penaltyz@
3040 \fi
3041 \fi}

```

`\@makefnintext` 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。

```

3042 \newcommand\@makefnintext[1]{%
3043 \advance\leftskip 3\jsZw
3044 \parindent 1\jsZw
3045 \noindent
3046 \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。

```

3047 % \def\@xfootnotenext[#1]{%
3048 % \begingroup
3049 % \ifnum#1>z@
3050 % \csname c@\@mpfn\endcsname #1\relax
3051 % \unrestored@protected@xdef\@thefnmark{\thempfn}%
3052 % \else
3053 % \unrestored@protected@xdef\@thefnmark{}%
3054 % \fi
3055 % \endgroup
3056 % \@footnotetext}

```

ここまでのコードは JS クラスを踏襲する。

11 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

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

```
3057 \let\jsInhibitGlueAtParTop\@empty
```

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

```
3058 \def\everyparhook{\jsInhibitGlueAtParTop}
3059 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
3060 \g@addto@macro\bxjs@begin@document@hook{\everypar{\everyparhook}}
3061 \fi
```

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

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

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

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

```
3062 \ifx j\jsEngine
3063 \def\@inhibitglue{%
3064   \futurelet\@let@token\@@inhibitglue}
3065 \begingroup
3066 \let\GDEF=\gdef
3067 \let\CATCODE=\catcode
3068 \let\ENDGROUP=\endgroup
3069 \CATCODE`k=12
3070 \CATCODE`a=12
3071 \CATCODE`n=12
3072 \CATCODE`j=12
3073 \CATCODE`i=12
3074 \CATCODE`c=12
3075 \CATCODE`h=12
3076 \CATCODE`r=12
3077 \CATCODE`t=12
```

```

3078 \CATCODE`e=12
3079 \GDEF\KANJI@CHARACTER{kanji character }
3080 \ENDGROUP
3081 \def\@@inhibitglue{%
3082   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
3083 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
3084   \def\jsc@ig@temp{#1}%
3085   \ifx\jsc@ig@temp\@empty
3086     \ifnum\the\inhibitxspcode`#2=2\relax
3087       \inhibitglue
3088     \fi
3089   \fi}
3090 \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。

```

3091 \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。

```

3092 \def\@doendpe{%
3093   \@endpetrue
3094   \def\par{%
3095     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
3096   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

3097 \def\@setminipage{%
3098   \@minipagetrue
3099   \everypar{\@minipagefalse\everypar{\everyparhook}}}%
3100 }

```

`\item` 命令の直後です。

```

3101 \def\@item[#1]{%
3102   \if@noparitem
3103     \@donoparitem
3104   \else
3105     \if@inlabel
3106       \indent \par
3107     \fi
3108     \ifhmode
3109       \unskip\unskip \par
3110     \fi
3111     \if@newlist

```

```

3112     \if@nobreak
3113         \@nbitem
3114     \else
3115         \addpenalty\@beginparpenalty
3116         \addvspace\@topsep
3117         \addvspace{-\parskip}%
3118     \fi
3119 \else
3120     \addpenalty\@itempenalty
3121     \addvspace\itemsep
3122 \fi
3123 \global\@inlabeltrue
3124 \fi
3125 \everypar{%
3126     \@minipagefalse
3127     \global\@newlistfalse
3128     \if@inlabel
3129         \global\@inlabelfalse
3130         {\setbox\z@\lastbox
3131         \ifvoid\z@
3132             \kern-\itemindent
3133         \fi}%
3134     \box\@labels
3135     \penalty\z@
3136 \fi
3137 \if@nobreak
3138     \@nobreakfalse
3139     \clubpenalty \@M
3140 \else
3141     \clubpenalty \@clubpenalty
3142     \everypar{\everyparhook}%
3143 \fi
3144 \everyparhook}%
3145 \if@noitemarg
3146     \@noitemargfalse
3147     \if@nmbrlist
3148         \refstepcounter\@listctr
3149     \fi
3150 \fi
3151 \sbox\@tempboxa{\makelabel{#1}}%
3152 \global\setbox\@labels\hbox{%
3153     \unhbox\@labels
3154     \hskip \itemindent
3155     \hskip -\labelwidth
3156     \hskip -\labelsep
3157     \ifdim \wd\@tempboxa >\labelwidth
3158         \box\@tempboxa
3159     \else
3160         \hbox to\labelwidth {\unhbox\@tempboxa}%

```

```

3161 \fi
3162 \hskip \labelsep}%
3163 \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。

```

3164 \def\@afterheading{%
3165 \@nbreaktrue
3166 \everypar{%
3167 \if@nbreak
3168 \@nbreakfalse
3169 \clubpenalty \@M
3170 \if@afterindent \else
3171 {\setbox\z@\lastbox}%
3172 \fi
3173 \else
3174 \clubpenalty \@clubpenalty
3175 \everypar{\everyparhook}%
3176 \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

```

3177 \fi

```

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

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

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

```

3178 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none\else
3179 \AtEndOfPackage{%
3180 \def\@gnewline #1{%
3181 \ifvmode
3182 \@nolnerr
3183 \else
3184 \unskip \reserved@e {\reserved@f#1}\nbreak \hfil \break \null
3185 \jsInhibitGlue \ignorespaces
3186 \fi}
3187 }
3188 \fi

```


12 いろいろなロゴ

L^AT_EX 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。

```
3189 \if@jslogo
3190   \IfFileExists{jslogo.sty}{%
3191     \RequirePackage{jslogo}%
3192   }{%
3193     \ClassWarningNoLine{bxjs@clsname}
3194       {The package 'jslogo' is not installed.\MessageBreak
3195        It is included in the recent release of\MessageBreak
3196        the 'jsclasses' bundle}
3197   }
3198 \fi
```

13 amsmath との衝突の回避

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

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

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

```
3199 %\let\ltx@ifnextchar\@ifnextchar
3200 %\def\ProvidesFile#1{%
3201 %   \begingroup
3202 %     \catcode`\ 10 %
3203 %     \ifnum \endlinechar<256 %
3204 %       \ifnum \endlinechar>\m@ne
3205 %         \catcode\endlinechar 10 %
3206 %       \fi
3207 %     \fi
3208 %     \@makeother\/%
```

```

3209 % \makeother\&%
3210 % \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 3211 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3212 \newcommand{\postpartname}{\if@english\else 部\fi}
3213 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3214 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3215 \newcommand{\presectionname}{}% 第
\postsectionname 3216 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3217 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3218 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3219 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname

\bibname 3220 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3221 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
3222 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 3223 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3224 %<jspf>\newcommand{\figurename}{Fig.~}
3225 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3226 %<jspf>\newcommand{\tablename}{Table~}

\appendixname

\abstractname 3227 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3228 \newcommand{\appendixname}{\if@english \else 付録\fi}
3229 %<book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

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

※ 2.0 版より、完全に `bxwareki` に任せる。

```

3230 \@onlypreamble\bxjs@decl@Seireki@cmds

```

```

3231 \@tempswafalse
3232 \if p\jsEngine \@tempswatrue \fi
3233 \if n\jsEngine \@tempswatrue \fi
3234 \bxjs@cond\if@tempswa\fi{%
3235 % 欧文 8bitTeX の場合
3236 \newif\ifjsSeireki \jsSeirekitrue
3237 \def\bxjs@decl@Seireki@cmds{%
3238   \def\西暦{\jsSeirekitrue}%
3239   \def\和暦{\jsSeirekifalse\bxjs@wareki@used}}
3240 \def\Seireki{\jsSeirekitrue}
3241 \def\Wareki{\jsSeirekifalse\bxjs@wareki@used}
3242 \def\bxjs@if@use@seireki{\bxjs@cond\ifjsSeireki\fi}
3243 \def\bxjs@iai{\noexpand~}
3244 }{%
3245 \newif\if 西暦 \西暦 true
3246 \def\bxjs@decl@Seireki@cmds{%
3247   \def\西暦{\西暦 true}%
3248   \def\和暦{\西暦 false\bxjs@wareki@used}}
3249 \def\Seireki{\西暦 true}
3250 \def\Wareki{\西暦 false\bxjs@wareki@used}
3251 \def\bxjs@if@use@seireki{\bxjs@cond\if 西暦\fi}
3252 \let\bxjs@iai@empty
3253 }
3254 \bxjs@decl@Seireki@cmds
3255 \let\bxjs@unxp@firstofone \let\bxjs@onxp@firstofone
3256 \bxjs@test@engine\unexpanded{%
3257   \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。

```

3258 \newif\ifbxjs@bxwareki@avail
3259 \IfFileExists{bxwareki.sty}{%
3260   \RequirePackage{bxwareki}[]%
3261   \bxjs@bxwareki@availtrue}{%

```

`\bxjs@wareki@used` 和暦が非対応の場合に警告を出す。

```

3262 \ifbxjs@bxwareki@avail \let\bxjs@wareki@used\empty
3263 \else
3264   \bxjs@robust@def\bxjs@wareki@used{%
3265     \global\let\bxjs@wareki@used\empty
3266     \ClassWarning\bxjs@clsname
3267       {Wareki mode is not supported, since\MessageBreak
3268         'bxwareki' is unavailable, found}}
3269   \g@addto@macro\bxjs@begin@document@hook{%
3270     \let\bxjs@wareki@used\empty}
3271 \fi

```

`\jayear` 和暦における年の表記（元号 + 年数）。

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

```

3272 \ifbxjs@bxwareki@avail
3273   \let\jyear\warekiyear
3274   \def\bxjs@tmpa{H}\ifx\bxjs@tmpa\warekigengoinitial
3275     \newcount\heisei \heisei=\value{warekiyear}
3276   \fi

    bxwareki が使えない場合は西暦表示にする。

3277 \else
3278   \edef\jyear{\the\year \bxjs@iai}
3279 \fi

```

\today 英語、西暦、和暦で場合分けをする。

```

3280 \let\bxjs@next\relax
3281 \ifbxjs@bxwareki@avail \ifx\warekigengo\@empty\else
3282   \def\bxjs@next{\bxjs@onxp{\warekitoday}}
3283 \fi\fi
3284 \edef\bxjs@today{%
3285   \if@english
3286     \ifcase\month\or
3287       January\or February\or March\or April\or May\or June\or
3288       July\or August\or September\or October\or November\or December\fi
3289     \space\number\day, \number\year
3290   \else
3291     \ifx\bxjs@next\relax \expandafter\@firstoftwo
3292     \else \noexpand\bxjs@if@use@seireki
3293   \fi {%
3294     \number\year\bxjs@iai\bxjs@unxp{年}%
3295     \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3296     \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3297   }\bxjs@next}%
3298 \fi}
3299 \let\today\bxjs@today

```

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

```

3300 \g@addto@macro\bxjs@begin@document@hook{%
3301   \ifx\bbl@jpn@maybekansuji\@undefined\else
3302     \bxjs@decl@Seireki@cmds
3303   \g@addto@macro\datejapanese{%
3304     \let\today\bxjs@today}%
3305   \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)


```

3306 \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。

```
3307 %<slide>\pagestyle{empty}%
3308 %<article|report>\pagestyle{plain}%
3309 %<book>\pagestyle{headings}%
3310 \pagenumbering{arabic}
3311 \if@twocolumn
3312   \twocolumn
3313   \sloppy
3314   \flushbottom
3315 \else
3316   \onecolumn
3317   \raggedbottom
3318 \fi
3319 %<*slide>
3320 \renewcommand\familydefault{\sfdefault}
3321 \raggedright
3322 %</slide>
```

■BXJS 独自の追加処理 

フックを実行する。

```
3323 \bxjs@pre@jadriver@hook
```

和文ドライバのファイルを読み込む。

```
3324 \input{bxjsja-\bxjs@jadriver.def}
```

おしまい。

```
3325 %</class>
```

以上です。

付録 A 和文ドライバの仕様

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

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

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

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

以下で標準で用意されている和文ドライバの実装を示す。

```
3326 %<*drv>
```

付録 B 和文ドライバ : minimal

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

このドライバでは、各エンジンについての必要最低限の処理だけを行っている。日本語処理のためのパッケージ (`xeCJK` や `LuaTeX-jā` 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。

B.1 補助マクロ

```
3327 %<*minimal>
```

```
3328 %% 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.
```

```
3329 \def\DeclareJaTextFontCommand#1#2{%
```

```
3330   \DeclareRobustCommand#1[1]{%
```

```
3331     \relax
```

```
3332     \ifmmode \expandafter\nfss@text \fi
```

```
3333     {#2##1}}%
```

```
3334 }
```

```
\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。
```

```
3335 \def\DeclareJaMathFontCommand#1#2{%
```

```
3336   \DeclareRobustCommand#1[1]{%
```

```
3337     \relax
```

```
3338     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3339     \nfss@text{\fontfamily\familydefault
```

```
3340       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3341       #2##1}}%
```

```

3342 }%
3343 }

```

`\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。

```

3344 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3345 \@onlypreamble\bxjs@if@sf@default
3346 \def\bxjs@if@sf@default#1{%
3347   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3348   \g@addto@macro\bxjs@begin@document@hook{%
3349     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3350 }

```

`\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。

```

3351 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3352 \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。

```

3353 \def\jsLetHeadChar#1#2{%
3354   \begingroup
3355     \escapechar=`\ %
3356     \let\bxjs@tmpa={% brace-match-hack
3357       \bxjs@let@hchar@exp#2}%
3358   \endgroup
3359   \let#1\bxjs@g@tmpa}
3360 \def\bxjs@let@hchar@exp{%
3361   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3362 \def\bxjs@let@hchar@exp@a{%
3363   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3364   \bxjs@let@hchar@out\let\relax
3365   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3366   \bxjs@let@hchar@out\let\space%
3367   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3368   \bxjs@let@hchar@out\let\@backslashchar
3369   }{\bxjs@let@hchar@exp@b}}}%
3370 \def\bxjs@let@hchar@exp@b#1{%
3371   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3372 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3373   %\message{<#1#2>}%
3374   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```



```

3375 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3376 \bxjs@let@hchar@out\let\relax
3377 }{%else
3378 \expandafter\bxjs@let@hchar@exp
3379 }%
3380 }{%else
3381 \bxjs@let@hchar@chr#1%
3382 }}
3383 \def\bxjs@let@hchar@chr#1{%
3384 \bxjs@let@hchar@out\def{{#1}}}}
3385 \def\bxjs@let@hchar@out#1#2{%
3386 \global#1\bxjs@g@tmpa#2\relax
3387 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3388 \chardef\bxjs@let@hchar@csta=128
3389 \chardef\bxjs@let@hchar@cstb=192
3390 \chardef\bxjs@let@hchar@cstc=224
3391 \chardef\bxjs@let@hchar@cstd=240
3392 \chardef\bxjs@let@hchar@cste=248
3393 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3394 \def\bxjs@let@hchar@chr@ue#1{%
3395 \@tempcnta=`#1\relax
3396 %\message{\the\@tempcnta}%
3397 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3398 \bxjs@let@hchar@chr@ue@a#1%
3399 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3400 \bxjs@let@hchar@out\let\relax
3401 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3402 \bxjs@let@hchar@chr@ue@b
3403 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3404 \bxjs@let@hchar@chr@ue@c
3405 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3406 \bxjs@let@hchar@chr@ue@d
3407 }{%else
3408 \bxjs@let@hchar@out\let\relax
3409 }}}}}
3410 \def\bxjs@let@hchar@chr@ue@a#1{%
3411 \bxjs@let@hchar@out\def{{#1}}}}
3412 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3413 \bxjs@let@hchar@out\def{{#1#2}}}}
3414 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3415 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3416 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3417 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
3418 \ifx j\jsEngine
```

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

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

```
3419 \def\bxjs@let@hchar@chr@pp#1#2{%
3420   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#2\relax#1#2}
3421 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3#4{%
3422   \%message{(\meaning#3:\meaning#4)}%
3423   \bxjs@cond\if#1k\fi{%
3424     \bxjs@let@hchar@out\def{#{#4}}%
3425   }{%else
3426     \bxjs@let@hchar@chr@ue#3#4%
3427   }}
3428 \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。

```
3429 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3430 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3431 \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。

```
3432 \@onlypreamble\bxjs@declarefontshape
3433 \ifjsWithupTeX
3434 \def\bxjs@declarefontshape{%
3435   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3436   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3437   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3438   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3439 }
3440 \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。

```
3441 \else
3442 \def\bxjs@declarefontshape{%
3443   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3444   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3445   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3446 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3447 }
3448 \def\bxjs@sizereference{jis}
3449 \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。

```

3450 \def\bxjs@next#1/#2/#3/#4/#5\relax{%
3451   \def\bxjs@tmpb{#5}}
3452 \ifjsWithpTeXng \def\bxjs@tmpb{10}%
3453 \else
3454 \expandafter\expandafter\expandafter\bxjs@next
3455 \expandafter\string\the\jfont\relax
3456 \fi
3457 \@for\bxjs@tmpa:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3458                  \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3459   {\expandafter\let\csname\bxjs@tmpa/10\endcsname=\@undefined
3460    \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。

```

3461 \begingroup
3462 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3463 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3464 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3465 % 幅が丁度 10pt なら補正は不要
3466 \ifdim\wd\z@=10pt
3467   \global\let\bxjs@scale\jsScale
3468 \else
3469 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3470   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3471   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3472   \bxjs@invscale\@tempdima\bxjs@tmpa
3473   \xdef\bxjs@scale{\strip@pt\@tempdima}
3474 \fi
3475 \endgroup
3476 %\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。

```

3477 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3478 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3479 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3480 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3481 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3482 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3483 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3484 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3485 \DeclareFontShape{\jsc@JYn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3486 \DeclareFontShape{\jsc@JYn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3487 \DeclareFontShape{\jsc@JYn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3488 \DeclareFontShape{\jsc@JYn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3489 \DeclareFontShape{\jsc@JYn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3490 \DeclareFontShape{\jsc@JYn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3491 \DeclareFontShape{\jsc@JYn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3492 \DeclareFontShape{\jsc@JYn}{gt}{b}{sl}{<->ssub*gt/bx/n}{-}
3493 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3494 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
3495 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
3496 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3497 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3498 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3499 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3500 \DeclareFontShape{\jsc@JTn}{gt}{bx}{it}{<->ssub*gt/m/n}{-}
3501 \DeclareFontShape{\jsc@JTn}{gt}{bx}{sl}{<->ssub*gt/m/n}{-}
3502 \DeclareFontShape{\jsc@JTn}{mc}{b}{n}{<->ssub*mc/bx/n}{-}
3503 \DeclareFontShape{\jsc@JTn}{mc}{b}{it}{<->ssub*mc/bx/n}{-}
3504 \DeclareFontShape{\jsc@JTn}{mc}{b}{sl}{<->ssub*mc/bx/n}{-}
3505 \DeclareFontShape{\jsc@JTn}{gt}{b}{n}{<->ssub*gt/bx/n}{-}
3506 \DeclareFontShape{\jsc@JTn}{gt}{b}{it}{<->ssub*gt/bx/n}{-}
3507 \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。

```

3508 \@ifl@t@r\fmtversion{2020/10/01}
3509     {\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
3510 \ifjsc@needspace@tch          % --- for 2020-02-02 or older BEGIN
3511 \ifx\@rmfamilyhook\@undefined % old
3512 \DeclareRobustCommand\rmfamily
3513     {\not@math@alphabet\rmfamily\mathrm
3514       \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3515 \DeclareRobustCommand\sffamily
3516     {\not@math@alphabet\sffamily\mathsf
3517       \romanfamily\sffamily\kanjifamily\gtdefault\selectfont}
3518 \DeclareRobustCommand\ttfamily
3519     {\not@math@alphabet\ttfamily\mathtt
3520       \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3521 \g@addto@macro\bxjs@begin@document@hook{%
3522     \ifx\mweights@init\@undefined\else % mweights.sty is loaded
3523     % my definitions above should have been overwritten, recover it!
3524     % \selectfont is executed twice but I don't care about speed...
3525     \expandafter\g@addto@macro\csname rmfamily \endcsname

```

```

3526      {\kanjifamily\mcdefault\selectfont}%
3527      \expandafter\g@addto@macro\csname sffamily \endcsname
3528      {\kanjifamily\gtdefault\selectfont}%
3529      \expandafter\g@addto@macro\csname ttfamily \endcsname
3530      {\kanjifamily\gtdefault\selectfont}%
3531      \fi}
3532 \else                                     % 2020-02-02
3533 \g@addto@macro\@rmfamilyhook
3534      {\prepare@family@series@update@kanji{mc}\mcdefault}
3535 \g@addto@macro\@sffamilyhook
3536      {\prepare@family@series@update@kanji{gt}\gtdefault}
3537 \g@addto@macro\@ttfamilyhook
3538      {\prepare@family@series@update@kanji{gt}\gtdefault}
3539 \fi
3540 \else % --- for 2020-02-02 or older END & for 2020-10-01 BEGIN
3541 \AddToHook{rmfamily}%
3542      {\prepare@family@series@update@kanji{mc}\mcdefault}
3543 \AddToHook{sffamily}%
3544      {\prepare@family@series@update@kanji{gt}\gtdefault}
3545 \AddToHook{ttfamily}%
3546      {\prepare@family@series@update@kanji{gt}\gtdefault}
3547 \fi % --- for 2020-10-01 END
3548 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
3549 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3550 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3551 \fi
3552 \bxjs@if@sf@default{%
3553      \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3554 \selectfont

```

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

```

3555 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3556 \let\bxjs@parse@qh@a\undefined
3557 \let\bxjs@parse@qh@b\undefined

```

■パラメタの設定

```

3558 \prebreakpenalty\jis"2147=10000
3559 \postbreakpenalty\jis"2148=10000
3560 \prebreakpenalty\jis"2149=10000
3561 \inhibitxspcode`!=1
3562 \inhibitxspcode`〒=2
3563 \xspcode`+=3
3564 \xspcode`\%=3

```

"80～"FF の範囲の `\spcode` を 3 に変更。

```

3565 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%

```

```
3566 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}
```

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

```
3567 \let\jsInhibitGlueAtParTop\@inhibitglue
```

`\jsResetDimen` は空のままでよい。

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

```
3568 \begingroup
```

```
3569 \catcode`\!=0
```

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

```
3570 \gdef\bxjs@ptex@dir{%
```

```
3571 !iftdir t%
```

```
3572 !else!ifydir y%
```

```
3573 !else ?%
```

```
3574 !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。

```
3575 % 古い \@makefnmark の定義
```

```
3576 \long\def\bxjs@tmpa{\hbox{%
```

```
3577 !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
```

```
3578 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
```

```
3579 \ifx\@makefnmark\bxjs@tmpa
```

```
3580 \long\gdef\@makefnmark{%
```

```
3581 !ifydir \hbox{}\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}%
```

```
3582 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
```

```
3583 \fi
```

エスケープ文字の変更はここまで。

```
3584 \endgroup
```

■minijs パッケージのブロック やっておく。

```
3585 \@namedef{ver@minijs.sty}{}
```

B.3 pdfTeX 用の処理

```
3586 \else\ifx p\jsEngine
```

```
3587 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
```

```
3588 \@onlypreamble\bxjs@cjk@loaded
```

```
3589 \def\bxjs@cjk@loaded{%
```

```
3590 \def\@footnotemark{%
```

```
3591 \leavevmode
```

```
3592 \ifhmode
```

```

3593     \edef\x@sf{\the\spacefactor}%
3594     \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3595         \unkern\unkern
3596         \ifdim\lastskip>\z@ \unskip \fi
3597     \fi\fi
3598     \nobreak
3599 \fi
3600 \@makefnmark
3601 \ifhmode \spacefactor\x@sf \fi
3602 \relax}%
3603 \let\bxjs@cjk@loaded\relax
3604 }
3605 \g@addto@macro\bxjs@begin@document@hook{%
3606     \ifpackageloaded{CJK}{%
3607         \bxjs@cjk@loaded
3608     }{}%
3609 }

```

B.4 XeTeX 用の処理

```

3610 \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。

```

3611 \def\bxjs@let@hchar@chr#1{%
3612     \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3613     \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3614         \bxjs@let@hchar@chr@xe
3615     }\bxjs@let@hchar@out\def{{#1}}}}
3616 \def\bxjs@let@hchar@chr@xe#1{%
3617     \lccode`0=`#1\relax
3618     \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3619 \@onlypreamble\bxjs@do@precisetext
3620 \ifx\XeTeXgenerateactualtext\@undefined\else
3621     \def\bxjs@do@precisetext{%
3622         \XeTeXgenerateactualtext=\@ne}
3623 \fi

```

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

```

3624 \@onlypreamble\bxjs@do@simplejasetup
3625 \def\bxjs@do@simplejasetup{%
3626     \ifnum\XeTeXinterchartokenstate>\z@
3627     \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3628         \jsSimpleJaSetup
3629         \ClassInfo\bxjs@clsname
3630         {'\string\jsSimpleJaSetup' is applied\@gobble}%
3631     \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```
3632 \newcommand*{\jsSimpleJaSetup}{%
3633   \XeTeXlinebreaklocale "ja"\relax
3634   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3635   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

```
3636 \fi\fi\fi
```

`simplejasetup` オプションの処理。

```
3637 \ifx\bxjs@do@simplejasetup\@undefined\else
3638   \g@addto@macro\bxjs@begin@document@hook{%
3639     \ifbxjs@simplejasetup
3640       \bxjs@do@simplejasetup
3641     \fi}
3642 \fi

precisetext オプションの処理。
3643 \ifbxjs@precisetext
3644   \ifx\bxjs@do@precisetext\@undefined
3645     \ClassWarning\bxjs@clsname
3646       {The current engine does not support the\MessageBreak
3647         'precise-text' option\@gobble}
3648   \else
3649     \bxjs@do@precisetext
3650   \fi
3651 \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。

```
3652 \g@addto@macro\bxjs@begin@document@hook{%
3653   \ifx\jsInhibitGlueAtParTop\@empty
3654     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3655     \ifx\everyparhook\bxjs@tmpa
3656       \let\everyparhook\@empty
3657     \fi
3658   \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。

```
3659 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@modern

まず \everypar を“乗っ取る” 処理を行う。
3660 \let\bxjs@everypar\everypar
3661 \newtoks\everypar
3662 \everypar\bxjs@everypar
```

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


```

3663 \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3664 \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。

```

3665 \ifbxjs@fancyhdr

```

\bxjs@adjust@fancyhdr fancyhdr の初期設定に関する改変の処理。fancyhdr 読込完了と \pagestyle{fancy} 実行の間で実行されるべき。

```

3666 \@onlypreamble\bxjs@adjust@fancyhdr
3667 \def\bxjs@adjust@fancyhdr{%

```

ヘッダ・フッタの要素の書式について、それが既定のままであれば、“二文字フォント命令”を除去したものに置き換える。

※和文なので \sl は無い方がよいはず。

```

3668 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3669 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3670 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3671 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3672 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3673 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3674 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3675 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3676 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3677 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3678 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3679 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3680 \def\bxjs@tmpa{\rm\thepage\strut}%
3681 \def\bxjs@tmpb{\thepage\strut}%
3682 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3683 \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。

```

3684 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth
3685 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3686 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3687 }\bxjs@tmpa
3688 \fi\fi
3689 \PackageInfo\bxjs@clsname
3690 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3691 \def\bxjs@pagestyle@hook{%
3692 \@ifpackageloaded{fancyhdr}{%

```

```

3693 \bxjs@adjust@fancyhdr
3694 \global\let\bxjs@adjust@fancyhdr\relax
3695 }{}

```

\pagestyle にフックを入れ込む。

```

3696 \let\bxjs@org@pagestyle\pagestyle
3697 \def\pagestyle{%
3698 \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。

```

3699 \g@addto@macro\bxjs@begin@document@hook{%
3700 \bxjs@pagestyle@hook
3701 \global\let\bxjs@pagestyle@hook\relax}

```

■和文空白命令

```

3702 \ifbxjs@jaspace@cmd

```

\jaenspace 半角幅の水平空き。

```

3703 \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。

```

3704 \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。

```

3705 \def\_ {\zwspace}

```

\jaspace jlreq クラスと互換の命令。

```

3706 \DeclareRobustCommand*{\jaspace}[1]{%
3707 \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3708 \ClassError\bxjs@clsname
3709 {Unknown jaspac: #1}{\@eha}%
3710 \else
3711 \csname bxjs@jaspace@@#1\endcsname
3712 \fi}
3713 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3714 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3715 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}

```

```

3716 \fi

```

終わり。

```

3717 \fi

```

以上で終わり。

```

3718 %</minimal>

```

付録 C 和文ドライバ : standard

標準のドライバ。

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

■和文フォント指定の扱い

C.1 準備

まず `minimal` ドライバを読み込む。

```
3719 %<*standard>
3720 %% このファイルは日本語文字を含みます
3721 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3722 \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。

```
3723 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3724 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3725 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3726 \define@key{bxjsStd}{jis2004}[true]{%
3727   \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。

```
3728 \newif\ifbxjs@jp@units

units オプションの処理。
```

```

3729 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3730 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3731 \define@key{bxjsStd}{units}[true]{%
3732   \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。

```

3733 \let\bxjs@jp@font\@empty

font オプションの処理。
3734 \define@key{bxjsStd}{font}{%
3735   \edef\bxjs@jp@font{#1}}

```

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

```

3736 \newif\ifbxjs@jp@strong@cmd \bxjs@jp@strong@cmdtrue

strong-cmd オプションの処理。
3737 \let\bxjs@kv@strongcmd@true\bxjs@jp@strong@cmdtrue
3738 \let\bxjs@kv@strongcmd@false\bxjs@jp@strong@cmdfalse
3739 \define@key{bxjs}{strong-cmd}[true]{\bxjs@set@keyval{strongcmd}{#1}{}}

```

実際の `japaram` の値を適用する。

```

3740 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3741 \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。

```

3742 \@onlypreamble\bxjs@apply@mmiv
3743 \def\bxjs@apply@mmiv{%
3744   \g@addto@macro\@classoptionslist{,jis2004}
3745 % \@ifpackagewith 判定への対策
3746   \PassOptionsToPackage{jis2004}{otf}
3747   \global\let\bxjs@apply@mmiv\relax}
3748 \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。

```

3749 \if jsEngine
3750   \bxjs@jp@unitsfalse
3751 \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。

```

3752 \ifbxjs@jp@units
3753   \IfFileExists{bxcalc.sty}{%
3754     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3755     \ifx\usepTeXunits\@undefined

```

```

3756 \PackageWarningNoLine\bxjs@clsname
3757 {Cannot support pTeX units (zw etc.), since\MessageBreak
3758 the package 'bxcalc' is too old}%
3759 \bxjs@jp@unitsfalse
3760 \else \usepTeXunits
3761 \fi
3762 }{%else
3763 \PackageWarningNoLine\bxjs@clsname
3764 {Cannot support pTeX units (zw etc.), since\MessageBreak
3765 the package 'bxcalc' is unavailable}%
3766 \bxjs@jp@unitsfalse
3767 }
3768 \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。

```

3769 \ifbxjs@jp@units
3770 \def\bxjs@parse@qh#1{\let\bxjs@tmpb\relax}
3771 \let\bxjs@parse@qh@a\@undefined
3772 \let\bxjs@parse@qh@b\@undefined
3773 \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。

```

3774 \ifbxjs@jp@units
3775 \def\bxjs@let@lenexpr#1#2{%
3776 \edef#1{#2}%
3777 \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3778 \else
3779 \def\bxjs@let@lenexpr{\edef}
3780 \fi

```

■\strong 命令の補填

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

※

```

3781 \ifbxjs@jp@strong@cmd\jsAtEndOfClass{%
3782 \ifx\strong\@undefined\ifx\strongenv\@undefined
3783 \DeclareRobustCommand{\strongenv}{\bxjs@strong@font}%
3784 \DeclareTextFontCommand{\strong}{\strongenv}%

```

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

※\strongfontdeclare は試験的機能とする。

```

3785 \@ifdefinable{\strongfontdeclare}{\bxjs@protected\def\strongfontdeclare#1{%
3786 \bxjs@strong@font@declare@a#1,\@nil}}%

```

```

3787 \def\bxjs@strong@font@declare@a#1,#2\@nil{\def\bxjs@strong@font{#1}}%
3788 \def\bxjs@strong@font{\bfseries}%
3789 \fi\fi
3790 }\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。

```

3791 \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 3792 \@tempdima=0.25mm
3793 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3794 \let\jH\jQ

```

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

```

\trueH 3795 \ifjsc@mag
3796 \@tempdimb=\jsBaseFontSize\relax
3797 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3798 \@tempdima=2.5mm
3799 \bxjs@invscale\@tempdima\bxjs@tmpa
3800 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3801 \@tempdima=10pt
3802 \bxjs@invscale\@tempdima\bxjs@tmpa
3803 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3804 \else \let\trueQ\jQ \let\bxjs@truept\p@
3805 \fi
3806 \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。

```

3807 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3808 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3809 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3810 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3811 \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。

```

3812 \def\jafontsize#1#2{%
3813 \begingroup
3814 \bxjs@jafontsize@a{#1}%
3815 \@tempdimb\jsInverseScale\@tempdima
3816 \bxjs@jafontsize@a{#2}%

```

```

3817 \xdef\bxjs@g@tmpa{%
3818 \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3819 \endgroup\bxjs@g@tmpa}
3820 \def\bxjs@jafontsize@a#1{%
3821 \bxjs@parse@qh{#1}%
3822 \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3823 \@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。

```
3824 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```

3825 \newcommand*\setkanjiskip[1]{%
3826 \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3827 \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3828 \newcommand*\getkanjiskip{%
3829 \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)autospaceing` での制御を用いるのでこの変数は常に真とする。

```
3830 \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 3831 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3832 \bxjs@kanjiskip@enabledtrue
3833 \bxjs@reset@kanjiskip}
3834 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3835 \bxjs@kanjiskip@enabledfalse
3836 \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。

```

3837 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3838 \ifbxjs@kanjiskip@enabled
3839 \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3840 \else \@tempskipa\z@
3841 \fi
3842 \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 3843 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3844 \newcommand*\setxkanjiskip[1]{%
```

```

3845 \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3846 \bxjs@reset@xkanjiskip}

```

```
\bxjs@enable@xkanjiskip 3847 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3848 \bxjs@xkanjiskip}
```

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

```

3849 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
3850 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3851   \bxjs@xkanjiskip@enabledtrue
3852   \bxjs@reset@xkanjiskip}
3853 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3854   \bxjs@xkanjiskip@enabledfalse
3855   \bxjs@reset@xkanjiskip}
3856 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3857   \ifbxjs@xkanjiskip@enabled
3858     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3859   \else \@tempskipa\z@
3860   \fi
3861   \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。

```

3862 \g@addto@macro\jsResetDimen{%
3863   \bxjs@reset@kanjiskip
3864   \bxjs@reset@xkanjiskip}
3865 \let\bxjs@apply@kanjiskip\relax
3866 \let\bxjs@apply@xkanjiskip\relax

```

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

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

```

3867 \@onlypreamble\bxjs@adjust@jafont
3868 \def\bxjs@adjust@jafont#1{%
3869   \ifx\jsJaFont\bxjs@auto
3870     \bxjs@get@kanjiEmbed
3871     \ifx\bxjs@jaEmbed\relax
3872       \let\bxjs@tmpa\@empty
3873     \else
3874       \let\bxjs@tmpa\bxjs@jaEmbed
3875       \ifx\bxjs@jaVariant\bxjs@hziv
3876         \bxjs@apply@mmiv
3877       \fi
3878     \fi
3879   \else
3880     \let\bxjs@tmpa\jsJaFont
3881   \fi
3882   \if f#1\ifx\bxjs@tmpa\bxjs@noEmbed
3883     \ClassWarningNoLine\bxjs@clsname
3884     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak

```



```

3885      not available on the current situation}%
3886      \let\bxjs@tmpa\@empty
3887      \fi\fi
3888  }
3889  \def\bxjs@@auto{auto}
3890  \def\bxjs@@noEmbed{noEmbed}
3891  \def\bxjs@@hziv{-04}

```

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

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

```

3892 \let\bxjs@jaEmbed\relax
3893 \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。

```

3894 \@onlypreamble\bxjs@get@kanjiEmbed
3895 \def\bxjs@get@kanjiEmbed{%
3896   \begingroup\setbox\z@=\hbox{%
3897     \global\let\bxjs@tmpdo\@empty
3898     \def\bxjs@next##1##2##3{%
3899       \def##1###1##3 ###2\@nil###3\@nnil{%
3900         \ifx$###1$\gdef##2{###2}\fi}%
3901       \g@addto@macro\bxjs@tmpdo{%
3902         \expandafter##1\bxjs@tmpa\@nil##3 \@nil\@nnil}}%
3903       \bxjs@next\bxjs@tmpdo@a\bxjs@g@tmpa{kanjiEmbed}%
3904       \bxjs@next\bxjs@tmpdo@b\bxjs@g@tmpa{jaEmbed}%
3905       \bxjs@next\bxjs@tmpdo@c\bxjs@g@tmpb{kanjiVariant}%
3906       \bxjs@next\bxjs@tmpdo@d\bxjs@g@tmpb{jaVariant}%
3907   }%
3908   \global\let\bxjs@g@tmpa\relax
3909   \global\let\bxjs@g@tmpb\relax
3910   \endlinechar\m@ne
3911   \let\do\@makeother\dospecials
3912   \catcode32=10 \catcode12=10 %form-feed
3913   \let\bxjs@tmpa\@empty
3914   \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3915   \ifeof\@inputcheck\else
3916     \read\@inputcheck to\bxjs@tmpa
3917     \closein\@inputcheck
3918   \fi
3919   \ifx\bxjs@tmpa\@empty\else
3920     \openin\@inputcheck="\bxjs@tmpa"\relax
3921     \@tempswatrue
3922     \loop\if@tempswa
3923       \read\@inputcheck to\bxjs@tmpa
3924       \bxjs@tmpdo
3925       \ifeof\@inputcheck \@tempswafalse \fi
3926     \repeat
3927   \fi

```

```

3928 } \endgroup
3929 \let\bxjs@jaEmbed\bxjs@g@tmpa
3930 \let\bxjs@jaVariant\bxjs@g@tmpb
3931 }

```

`\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。

```

3932 \@onlypreamble\bxjs@resolve@jafont@paren
3933 \def\bxjs@resolve@jafont@paren#1{%
3934   \def\bxjs@tmpb{\let#1}%
3935   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\@nnil#1}
3936 \@onlypreamble\bxjs@resolve@jafont@paren@a
3937 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\@nnil#5{%
3938   \ifx\relax#4\relax \bxjs@tmpb#5%
3939   \else
3940     \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3941     \bxjs@tmpb\bxjs@tmpa
3942   \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。

```

3943 \newcommand*\jachar[1]{%
3944   \begingroup
3945     \jsLetHeadChar\bxjs@tmpa{#1}%
3946     \ifx\bxjs@tmpa\relax
3947       \ClassWarningNoLine\bxjs@clsname
3948       {Illegal argument given to \string\jachar}%
3949     \else
3950       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3951     \fi
3952   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3953 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3954 \let\bxjs@jachar\@firstofone

```

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

```

3955 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3956 \@onlypreamble\bxjs@fix@hyperref@unicode
3957 \def\bxjs@fix@hyperref@unicode#1{%

```

```

3958 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3959 \@namedef{KV@Hyp@bxjs/hook}##1{%
3960   \KV@Hyp@unicode{##1}%
3961   \def\KV@Hyp@unicode####1{%
3962     \expandafter\ifx\csname if##1\expandafter\endcsname
3963       \csname if####1\endcsname\else
3964       \ClassWarningNoLine{bxjs@clsname
3965         {Blcoked hyperref option 'unicode=####1'}}%
3966     \fi
3967   }%
3968 }%
3969 }

```

`\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。

```

3970 \@onlypreamble\jsCheckHyperrefUnicode
3971 \let\jsCheckHyperrefUnicode\@empty
3972 \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。

```

3973 \@onlypreamble\bxjs@check@hyperref@unicode
3974 \def\bxjs@check@hyperref@unicode#1{%
3975   \g@addto@macro\jsCheckHyperrefUnicode{%
3976     \@tempwafalse
3977     \begingroup
3978       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3979       \aftergroup\@tempwattrue \fi
3980       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3981         \csname if#1\endcsname
3982       \aftergroup\@tempwattrue \fi
3983     \endgroup
3984     \if@tempwa\else
3985       \ClassError{bxjs@clsname
3986         {The value of hyperref 'unicode' key is not suitable\MessageBreak
3987           for the present engine (must be #1)}}%
3988       {\@ehc}%
3989     \fi}}

```

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

```

3990 \@onlypreamble\bxjs@urgent@special
3991 \@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。

```

3992 \ifbxjs@old@hook@system
3993 \def\bxjs@urgent@special#1{%
3994   \AtBeginDvi{\special{#1}}%
3995   \g@addto@macro\bxjs@begin@document@hook{%
3996     \ifpackageloaded{atbegshi}{%
3997       \begingroup
3998       \toks\z@{\special{#1}}%

```

```

3999      \toks\tw@\expandafter{\AtBegShi@HookFirst}%
4000      \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw}%
4001      \endgroup
4002    }{}%
4003  }%
4004 }

```

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。

```

4005 \else
4006   \def\bxjs@urgent@special#1{%
4007     \bxjs@urgent@special@a
4008     \AddToHook{shipout/firstpage}[pxjahyper/enc]{\special{#1}}
4009   \def\bxjs@urgent@special@a{%
4010     \DeclareHookRule{shipout/firstpage}{pxjahyper/enc}{<}{hyperref}%
4011     \global\let\bxjs@urgent@special@a\relax}
4012 \fi

```

C.4 pTeX 用設定

```

4013 \if j\jsEngine

```

■ 共通命令の実装

```

4014 \def\bxjs@apply@kanjiskip{%
4015   \kanjiskip\@tempskipa}
4016 \def\bxjs@apply@xkanjiskip{%
4017   \xkanjiskip\@tempskipa}

\jaJaChar のサブマクロ。
4018 \def\bxjs@jachar#1{%
4019   \bxjs@jachar@a#1...\@nil}
4020 \def\bxjs@jachar@a#1#2#3#4#5\@nil{%

```

引数が単一トークンなら和文文字トークンが得られたと見なし、それをそのまま出力する。

```

4021   \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。

```

4022   \else\ifx.#3%
4023     \@tempcnta`#1 \multiply\@tempcnta64
4024     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
4025     \bxjs@jachar@b
4026   \else\ifx.#4%
4027     \@tempcnta`#1 \multiply\@tempcnta64
4028     \advance\@tempcnta`#2 \multiply\@tempcnta64
4029     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
4030     \bxjs@jachar@b
4031   \else

```

```

4032 \tempcnta`#1 \multiply\tempcnta64
4033 \advance\tempcnta`#2 \multiply\tempcnta64
4034 \advance\tempcnta`#3 \multiply\tempcnta64
4035 \advance\tempcnta`#4 \advance\tempcnta-"3C82080
4036 \bxjs@jachar@b
4037 \fi\fi\fi}

```

符号値が \tempcnta の和文文字を出力する処理。

```

4038 \ifjsWithupTeX
4039 \def\bxjs@jachar@b{\kchar\tempcnta}
4040 \else
4041 \def\bxjs@jachar@b{%
4042 \ifx\bxUInt\@undefined\else
4043 \bxUInt{\tempcnta}%
4044 \fi}
4045 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

4046 \ifbxjs@jaspace@cmd
4047 \def\jathinspace{\hskip\xkanjiskip}
4048 \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。

```

4049 \ifbxjs@jp@jismmiv
4050 \PassOptionsToPackage{prefer2004jis}{pxchfon}
4051 \PassOptionsToPackage{prefer2004jis}{pxbabel}
4052 \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。

```

4053 \let\bxjs@jafont@paren\@firstofone
4054 \let\bxjs@tmpa\jsJaFont
4055 \ifx\bxjs@tmpa\bxjs@@auto
4056 \let\bxjs@tmpa\@empty
4057 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
4058 \def\bxjs@tmpa{noembed}
4059 \fi\fi
4060 \bxjs@resolve@jafont@paren\bxjs@tmpa
4061 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4062 \ifx\bxjs@tmpa\@empty\else
4063 \edef\bxjs@next{%
4064 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
4065 }\bxjs@next
4066 \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……。)

```

4067 \begingroup
4068   \global\let\bxjs@g@tmpa\relax
4069   \catcode`\|=0 \catcode`\|=12
4070   |def|bxjs@tmpdo@a#1|@nil{%
4071     |bxjs@tmpdo@a#1|@nil\RequirePackage|@nnil}%
4072   |def|bxjs@tmpdo@a#1\RequirePackage#2|@nnil{%
4073     |ifx$#1$|bxjs@tmpdo@b#2|@nil keyval|@nnil |fi}%
4074   |catcode`\|=0 \catcode`\|=12
4075   \def\bxjs@tmpdo@b#1keyval#2\@nnil{%
4076     \ifx$#2$\else
4077       \xdef\bxjs@g@tmpa{%
4078         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
4079     \fi}
4080 \@firstofone{%
4081   \catcode10=12 \endlinechar\m@ne
4082   \let\do\@makeother \dospecials \catcode32=10
4083   \openin\@inputcheck=otf.sty\relax
4084   \@tempswatrue
4085   \loop\if@tempswa
4086     \ifeof\@inputcheck \@tempswafalse \fi
4087     \if@tempswa
4088       \read\@inputcheck to\bxjs@next
4089       \expandafter\bxjs@tmpdo\bxjs@next\@nil
4090     \fi
4091   \repeat
4092   \closein\@inputcheck
4093 \endgroup}
4094 \bxjs@g@tmpa

```

■hyperref 対策 unicode にしてはいけない。

```

4095 \ifbxjs@hyperref@enc
4096   \bxjs@check@hyperref@unicode{false}
4097 \fi

```

tounicode special 命令を出力する。

```

4098 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
4099   \else\ifjsWithpTeXng T\else F\fi\fi T%
4100   \IfFileExists{pxjahyper-enc.sty}{\@tempswatrue}{\@tempswafalse}
4101   \if@tempswa
4102     \RequirePackage{pxjahyper-enc}[2020/10/05]%v0.6
4103     \ifbxjs@bigcode\else \suppressbigcode \fi
4104   \else

```

```

4105 \ifnum\jis"2121="A1A1 %euc
4106 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
4107 \else\ifnum\jis"2121="8140 %sjis
4108 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
4109 \else\ifnum\jis"2121="3000 %uptex
4110 \ifbxjs@bigcode
4111 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
4112 \PassOptionsToPackage{bigcode}{pxjahyper}
4113 \else
4114 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
4115 \PassOptionsToPackage{nobigcode}{pxjahyper}
4116 \fi
4117 \fi\fi\fi
4118 \let\bxToUnicodeSpecialDone=t
4119 \fi
4120 \fi

```

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

```

4121 \ifx f\bxjs@enablejfam\else
4122 \@enablejfamtrue
4123 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

4124 \if@enablejfam
4125 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
4126 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
4127 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
4128 \jfam\symmincho
4129 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
4130 \g@addto@macro\bxjs@begin@document@hook{%
4131 \ifx\reDeclareMathAlphabet\undefined\else
4132 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
4133 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
4134 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
4135 \fi}
4136 \fi

```

C.5 pdf_TE_X 用設定 : CJK + bxcjkatype

```

4137 \else\if p\jsEngine

```

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

※ Pandoc モードでは `autotilde` を指定しない。

```

4138 \bxjs@adjust@jafont{f}

```

```

4139 \let\bxjs@jafont@paren\@firstofone
4140 \bxjs@resolve@jafont@paren\bxjs@tmpa
4141 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4142 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
4143 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
4144   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
4145 \fi
4146 \edef\bxjs@next{%
4147   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
4148 }\bxjs@next
4149 \bxjs@cjk@loaded

```

■hyperref 対策 bxcjkatype 使用時は unicode にするべき。

※取りあえず固定はしない。

```

4150 \ifbxjs@hyperref@enc
4151   \PassOptionsToPackage{unicode}{hyperref}
4152 \fi

```

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

※ bxcjkatype を whole 付きで使っていることが前提。

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

```

4153 \ifx\bxjcjkatypeHyperrefPatchDone\@undefined
4154 \begingroup
4155   \CJK@input{UTF8.bdg}
4156 \endgroup
4157 \g@addto@macro\pdfstringdefPreHook{%
4158   \@nameuse{CJK@UTF8Binding}%
4159 }
4160 \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。

```

4161 \ifx\bxjcjkatypeHyperrefPatchDone\@undefined
4162 \g@addto@macro\pdfstringdefPreHook{%
4163   \ifx~\bxjs@@CJKtilde
4164     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
4165     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
4166     \let~\@empty
4167   \fi
4168 }
4169 \def\bxjs@@CJKtilde{\CJKe glue\ignorespaces}
4170 \def\bxjs@@tildecmd{~}
4171 \def\bxjs@LetUnexpandableSpace#1{%
4172   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
4173     \bxjs@org@LetUnexpandableSpace#1%
4174   \fi}
4175 \fi

```


■共通命令の実装

```
4176 \newskip\jsKanjiSkip
4177 \newskip\jsXKanjiSkip
4178 \ifx\CJKecglue\@undefined
4179   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4180 \fi
4181 \let\autospacing\bxjs@enable@kanjiskip
4182 \let\noautospacing\bxjs@disable@kanjiskip
4183 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4184 \def\bxjs@apply@kanjiskip{%
4185   \jsKanjiSkip\@tempskipa
4186   \let\CJKglue\bxjs@CJKglue}
4187 \let\autoxspacing\bxjs@enable@xkanjiskip
4188 \let\noautoxspacing\bxjs@disable@xkanjiskip
4189 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4190 \def\bxjs@apply@xkanjiskip{%
4191   \jsXKanjiSkip\@tempskipa
4192   \let\CJKecglue\bxjs@CJKecglue}
```

\jachar のサブマクロの実装。

```
4193 \def\bxjs@jachar#1{%
4194   \CJKforced{#1}}
```

和欧文間空白の命令 \jathinspace の実装。

```
4195 \ifbxjs@jaspace@cmd
4196   \protected\def\jathinspace{\CJKecglue}
4197 \fi
```

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

```
4198 \ifx t\bxjs@enablejfam
4199   \ClassWarningNoLine\bxjs@clsname
4200     {You cannot use 'enablejfam=true', since the\MessageBreak
4201     CJK package does not support Japanese math}
4202 \fi
```

C.6 Xe_{La}TeX 用設定：xeCJK + zxjatype

```
4203 \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。

```
4204 \RequirePackage{zxjatype}
4205 \PassOptionsToPackage{no-math}{fontspec}%!
4206 \PassOptionsToPackage{xetex}{graphicx}%!
4207 \PassOptionsToPackage{xetex}{graphics}%!
4208 \ifx\zxJaFamilyName\@undefined
4209   \ClassError\bxjs@clsname
4210     {xeCJK or zxjatype is too old}\@ehc
```

4211 \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。

```
4212 \bxjs@adjust@jafont{f}
4213 \let\bxjs@jafont@paren\gobble
4214 \bxjs@resolve@jafont@paren\bxjs@tmpa
4215 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4216 \ifx\bxjs@tmpa\@empty
4217   \setCJKmainfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiMincho-
     Regular.otf}
4218   \setCJKsansfont[BoldFont=HaranoAjiGothic-Medium.otf]{HaranoAjiGothic-
     Medium.otf}
4219 \else
4220   \edef\bxjs@next{%
4221     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4222   }\bxjs@next
4223 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X_ƎTeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、X_ƎTeX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「X_ƎTeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```
4224 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4225   \ifbxjs@hyperref@enc
4226     \PassOptionsToPackage{unicode}{hyperref}
4227   \fi
4228 \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。

```
4229 \AtEndOfPackage{%
4230 \def\@gnewline #1{%
```

```

4231 \ifvmode \@nolnerr
4232 \else
4233 \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
4234 \nobreak \hskip-1sp\hskip1sp\relax
4235 \ignorespaces
4236 \fi}
4237 }

```

■ 共通命令の実装

```

4238 \newskip\jsKanjiSkip
4239 \newskip\jsXKanjiSkip
4240 \ifx\CJKecglue\@undefined
4241 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4242 \fi
4243 \let\autospacing\bxjs@enable@kanjiskip
4244 \let\noautospacing\bxjs@disable@kanjiskip
4245 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4246 \def\bxjs@apply@kanjiskip{%
4247 \jsKanjiSkip\@tempskipa
4248 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4249 \let\autoxspacing\bxjs@enable@xkanjiskip
4250 \let\noautoxspacing\bxjs@disable@xkanjiskip
4251 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4252 \def\bxjs@apply@xkanjiskip{%
4253 \jsXKanjiSkip\@tempskipa
4254 \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。

```

4255 \ifx\mcfamily\@undefined
4256 \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4257 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4258 \fi

```

\jachar のサブマクロの実装。

```

4259 \def\bxjs@jachar#1{%
4260 \xeCJKDeclareCharClass{CJK}{`#1}\relax
4261 #1}

```

\jathinspace の実装。

```

4262 \ifbxjs@jaspace@cmd
4263 \protected\def\jathinspace{\CJKecglue}
4264 \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。

```

4265 \ifx t\bxjs@enablejfam
4266 \@enablejfamtrue
4267 \fi

```

実際に和文用の数式ファミリの設定を行う。

※ FIXME: 要検討。

```
4268 \if@enablejfam
4269   \xeCJKsetup{CJKmath=true}
4270 \fi
```

C.7 LuaTeX 用設定 : LuaTeX-ja

```
4271 \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。

```
4272 \let\zw\@undefined
4273 \RequirePackage{luatexja}
4274 \edef\bxjs@next{%
4275   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4276 }\bxjs@next
```

フォント代替の明示的定義。

```
4277 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4278 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4279 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4280 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4281 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4282 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4283 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4284 \DeclareFontShape{JY3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
4285 \DeclareFontShape{JY3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4286 \DeclareFontShape{JY3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4287 \DeclareFontShape{JY3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4288 \DeclareFontShape{JY3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4289 \DeclareFontShape{JY3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4290 \DeclareFontShape{JY3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4291 \DeclareFontShape{JY3}{gt}{b}{sl}{<->ssub*gt/bx/n}{ }
4292 \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{ }
4293 \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{ }
4294 \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
4295 \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{ }
4296 \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
4297 \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
4298 \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }
4299 \DeclareFontShape{JT3}{gt}{bx}{it}{<->ssub*gt/m/n}{ }
```

```

4300 \DeclareFontShape{JT3}{gt}{bx}{sl}{<->ssub*gt/m/n}{ }
4301 \DeclareFontShape{JT3}{mc}{b}{n}{<->ssub*mc/bx/n}{ }
4302 \DeclareFontShape{JT3}{mc}{b}{it}{<->ssub*mc/bx/n}{ }
4303 \DeclareFontShape{JT3}{mc}{b}{sl}{<->ssub*mc/bx/n}{ }
4304 \DeclareFontShape{JT3}{gt}{b}{n}{<->ssub*gt/bx/n}{ }
4305 \DeclareFontShape{JT3}{gt}{b}{it}{<->ssub*gt/bx/n}{ }
4306 \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。

```

4307 \bxjs@adjust@jafont{t}
4308 \ifx\bxjs@tmpa\bxjs@noEmbed
4309   \def\bxjs@tmpa{noembed}
4310 \fi
4311 \let\bxjs@jafont@paren\gobble
4312 \bxjs@resolve@jafont@paren\bxjs@tmpa
4313 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4314 \ifx\bxjs@tmpa\empty
4315   \defaultjfontfeatures{ Kerning=Off }
4316   \setmainfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiMincho-
     Regular.otf}
4317   \setsansfont[BoldFont=HaranoAjiGothic-Medium.otf,JFM=ujis]{HaranoAjiGothic-
     Medium.otf}
4318 \else
4319   \edef\bxjs@next{%
4320     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4321   }\bxjs@next
4322 \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。

```

4323 \ifpackagelater{luatexja}{2016/03/31}{ }{%else
4324 \DeclareRobustCommand\rmfamily
4325   {\not@math@alphabet\rmfamily\mathrm
4326     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
4327 \DeclareRobustCommand\sffamily
4328   {\not@math@alphabet\sffamily\mathsf
4329     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
4330 \DeclareRobustCommand\ttfamily
4331   {\not@math@alphabet\ttfamily\mathtt
4332     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4333 }
4334 \long\def\jttdefault{\gtdefault}
4335 \unless\ifx\@ltj@match@familytrue\undefined
4336   \@ltj@match@familytrue
4337 \fi

```

```

4338 \g@addto@macro\bxjs@begin@document@hook{%
4339   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}%
4340   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4341   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
4342 \bxjs@if@sf@default{%
4343   \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

4344 % 次の3つは既定値の通り
4345 %\ltjsetparameter{prebreakpenalty={`',10000}}
4346 %\ltjsetparameter{postbreakpenalty={`",10000}}
4347 %\ltjsetparameter{prebreakpenalty={`",10000}}
4348 \ltjsetparameter{jaxspmode={`!',1}}
4349 \ltjsetparameter{jaxspmode={`〒,2}}
4350 \ltjsetparameter{alxspmode={`+,3}}
4351 \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-j` 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。

```

4352 \ifx\ltjfakeparbegin\undefined
4353   \protected\def\ltjfakeparbegin{%
4354     \ifhmode
4355       \relax\directlua{%
4356         luatexja.jfmglue.create_beginpar_node()}}
4357   \fi}
4358 \fi

ltjs* クラスの定義と同等になるようにパッチを当てる。

4359 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@@none
4360 \begingroup
4361   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4362   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4363   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@@compat
4364     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4365   \directlua{
4366     local function patchcmd(cs, code, from, to)
4367       tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4368         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4369     end
4370     patchcmd(\@@xsect, [[\meaning\xsect]],
4371       \@{\hskip-\@tempskipa}, \@{\ltjfakeparbegin})
4372     patchcmd(\@@item, [[\meaning\@item]],
4373       \bxjs@tmpa, \@{\ltjfakeparbegin})}
4374 \endgroup
4375 \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。

```
4376 \ifbxjs@hyperref@enc
4377   \PassOptionsToPackage{unicode}{hyperref}
4378   \bxjs@check@hyperref@unicode{true}
4379 \fi
```

■共通命令の実装

```
4380 \protected\def\autospacing{%
4381   \ltjsetparameter{autospacing=true}}
4382 \protected\def\noautospacing{%
4383   \ltjsetparameter{autospacing=false}}
4384 \protected\def\autoxspacing{%
4385   \ltjsetparameter{autoxspacing=true}}
4386 \protected\def\noautoxspacing{%
4387   \ltjsetparameter{autoxspacing=false}}
4388 \def\bxjs@apply@kanjiskip{%
4389   \ltjsetparameter{kanjiskip={\@tempskipa}}}
4390 \def\bxjs@apply@xkanjiskip{%
4391   \ltjsetparameter{xkanjiskip={\@tempskipa}}}
```

\jachar のサブマクロの実装。

```
4392 \def\bxjs@jachar#1{%
4393   \ltjjachar`#1\relax}
```

\jathinspace の実装。

```
4394 \ifbxjs@jaspace@cmd
4395   \protected\def\jathinspace{%
4396     \hskip\ltjgetparameter{xkanjiskip}\relax}
4397 \fi
```

■和文数式ファミリ LuaTeX-j_a では和文数式ファミリは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablej_{fam} は常に真になる。

```
4398 \ifx f\bxjs@enablejfam
4399   \ClassWarningNoLine\bxjs@clsname
4400     {You cannot use 'enablejfam=false', since the\MessageBreak
4401       LuaTeX-ja always provides Japanese math families}
4402 \fi
```

C.8 共通処理 (2)

```
4403 \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 4404 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
4405   \DeclareJaTextFontCommand{\textmc}{\mcfamily}
```

```
4406 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4407 \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。

```
4408 \ifx\mathmc\@undefined
4409   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4410   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4411 \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)。

```
4412 \ifbxjs@jaspace@cmd
4413   \bxjs@protected\def\bxjs@choice@jathinspace{%
4414     \relax\ifmmode \mskip\medmuskip
4415     \else \jathinspace\ignorespaces
4416     \fi}
4417   \jsAtEndOfClass{%
4418     \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4419     \else \def\>{\protect\bxjs@choice@jathinspace}%
4420     \fi}
4421 \fi
```

■和文・和欧文間空白の初期値

```
4422 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4423 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4424 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4425 \fi
```

以上で終わり。

```
4426 %</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。

```
4427 %<*modern>
4428 \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。


```

4429 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4430 \def\encodingdefault{T1}%
4431 \input{t1enc.def}%
4432 \fontencoding\encodingdefault\selectfont
4433 \fi

```

基本フォントを Latin Modern フォントファミリに変更する。

※以下は `\usepackage[noamth]{lmodern}` と同じ。ユーザは後で `lmodern` を好きなオプションを付けて読み込むことができる。

```

4434 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
4435 \renewcommand{\rmdefault}{lrm}
4436 \renewcommand{\sfdefault}{lss}
4437 \renewcommand{\ttdefault}{lmtt}
4438 \fi

```

大型演算子用の数式フォントの設定。

※`amsmath` パッケージと同等にする。

```

4439 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4440   <-7.5>cmex7<7.5-8.5>cmex8%
4441   <8.5-9.5>cmex9<9.5->cmex10}{}%
4442 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

```

`amsmath` 読み込時に上書きされるのを防ぐ。

```

4443 \def\cmex@opt{10}

```

D.2 fixltx2e 読み込

※`fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```

4444 \ifx\@IncludeInRelease\@undefined
4445 \RequirePackage{fixltx2e}
4446 \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。

```

4447 \RequirePackage{bxjscjkat}

```

D.4 完了

おしまい。

```

4448 %</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。

```
4449 %<*pandoc>
```

```
4450 \input{bxjsja-standard.def}
```

bxjspandoc パッケージを読み込む。

```
4451 \RequirePackage{bxjspandoc}
```

\bxjs@endpreamble@hook etoolbox の \AtEndPreamble で実行される BXJS クラス用のフック。

※ ε -TeX 以外では無効になる。(将来 pandoc の外に出す可能性あり。)

```
4452 \@onlypreamble\bxjs@endpreamble@hook
```

```
4453 \let\bxjs@endpreamble@hook\@empty
```

```
4454 \ifjsWiththeTeX
```

```
4455 \RequirePackage{etoolbox}[2010/08/21]% v2.0
```

```
4456 \AtEndPreamble{\bxjs@endpreamble@hook}
```

```
4457 \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。

```
4458 \newif\ifbxjs@jp@fix@strong \bxjs@jp@fix@strongtrue
```

fix-strong オプションの処理。

```
4459 \let\bxjs@kv@fixstrong@true\bxjs@jp@fix@strongtrue
```

```
4460 \let\bxjs@kv@fixstrong@false\bxjs@jp@fix@strongfalse
```

```
4461 \define@key{bxjsPan}{fix-strong}[true]{%
```

```
4462 \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。

```
4463 \newif\ifbxjs@jp@fix@code \bxjs@jp@fix@codetrue
```

fix-code オプションの処理。

```
4464 \let\bxjs@kv@fixcode@true\bxjs@jp@fix@codetrue
```

```
4465 \let\bxjs@kv@fixcode@false\bxjs@jp@fix@codefalse
```

```
4466 \define@key{bxjsPan}{fix-code}[true]{%
```

```
4467 \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。

```
4468 \chardef\bxjs@jp@strong=0
```

strong オプションの処理。

```
4469 \def\bxjs@kv@strong@bold{\chardef\bxjs@jp@strong=0 }
```

```
4470 \def\bxjs@kv@strong@sans{\chardef\bxjs@jp@strong=1 }
```

```

4471 \def\bxjs@kv@strong@boldsans{\chardef\bxjs@jp@strong=2 }
4472 \define@key{bxjsPan}{strong}{%
4473   \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 4474 \newif\ifbxjs@jp@or@indent \bxjs@jp@or@indenttrue
\ifbxjs@jp@or@block@heading 4475 \newif\ifbxjs@jp@or@secnumdepth \bxjs@jp@or@secnumdepthtrue
4476 \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。

```

4477 \define@key{bxjsPan}{_plus}[]{%
4478   \bxjs@jp@or@indentfalse
4479   \bxjs@jp@or@secnumdepthfalse
4480   \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。

```

4481 \let\bxjs@kv@orindent@true\bxjs@jp@or@indenttrue
4482 \let\bxjs@kv@orindent@false\bxjs@jp@or@indentfalse
4483 \define@key{bxjsPan}{or-indent}[true]{%
4484   \bxjs@set@keyval{orindent}{#1}{}}
4485 \let\bxjs@kv@orsecnumdepth@true\bxjs@jp@or@secnumdepthtrue
4486 \let\bxjs@kv@orsecnumdepth@false\bxjs@jp@or@secnumdepthfalse
4487 \define@key{bxjsPan}{or-secnumdepth}[true]{%
4488   \bxjs@set@keyval{orsecnumdepth}{#1}{}}
4489 \let\bxjs@kv@orblockheading@true\bxjs@jp@or@block@headingtrue
4490 \let\bxjs@kv@orblockheading@false\bxjs@jp@or@block@headingfalse
4491 \define@key{bxjsPan}{or-block-heading}[true]{%
4492   \bxjs@set@keyval{blockheading}{#1}{}}

```

実際の `japaram` の値を適用する。

```

4493 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsPan}{#1}}
4494 \expandafter\bxjs@next\expandafter{\jsJaParam}

```

E.3 dupload システム

パッケージが重複して読み込まれたときに “option clash” の検査をスキップする。この時に何らかのコードを実行させることができる。

\bxjs@set@dupload@proc \bxjs@set@dupload@proc{〈ファイル名〉}{〈定義本体〉} 特定のファイルの読込が \@filewithoptions で指示されて、しかもそのファイルが読込済である場合に、オプション重複検査をスキップして、代わりに 〈定義本体〉 のコードを実行する。このコード中で #1 は渡されたオプション列のテキストに置換される。

```

4495 \@onlypreamble\bxjs@set@dupload@proc
4496 \def\bxjs@set@dupload@proc#1{%

```

```

4497 \expandafter\bxjs@set@dupload@proc@a\csize bxjs@dlp/#1\endcsize}
4498 \@onlypreamble\bxjs@set@dupload@proc@a
4499 \def\bxjs@set@dupload@proc@a#1{%
4500 \@onlypreamble#1\def#1##1}
4501 \def\bxjs@unset@dupload@proc#1{%
4502 \expandafter\let\csize bxjs@dlp/#1\endcsize\@undefined}

```

`\@if@options` `\@if@options` の再定義。

```

4503 \@onlypreamble\bxjs@org@if@options
4504 \let\bxjs@org@if@options\@if@options
4505 \@onlypreamble\bxjs@org@reset@options
4506 \let\bxjs@org@reset@options\relax
4507 \def\@if@options#1#2#3{%
4508 \let\bxjs@next\@secondoftwo
4509 \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4510 \ifx\bxjs@tmpa\bxjs@tmpb
4511 \expandafter\ifx\csize bxjs@dlp/#2.#1\endcsize\relax\else
4512 \let\bxjs@next\@firstoftwo \fi
4513 \fi
4514 \bxjs@next\bxjs@do@dupload@proc\bxjs@org@if@options{#1}{#2}{#3}}
4515 \g@addto@macro\bxjs@begin@document@hook{%
4516 \let\@if@options\bxjs@org@if@options}
4517 \@onlypreamble\bxjs@do@dupload@proc
4518 \def\bxjs@do@dupload@proc#1#2#3{%
4519 \ifx\bxjs@org@reset@options\relax
4520 \let\bxjs@org@reset@options\@reset@options
4521 \fi
4522 \bxjs@csize\bxjs@next\bxjs@dlp/#2.#1}%
4523 \def\@reset@options{%
4524 \let\@reset@options\bxjs@org@reset@options
4525 \@reset@options
4526 \bxjs@next{#3}}%
4527 \@firstoftwo}

```

E.4 lang 変数

`lang=ja` という言語指定が行われると、Pandoc はこれに対応していないため不完全な Babel や Polyglossia の設定を出力してしまう。これを防ぐための対策を行う。

`\bxjs@polyglossia@options` Polyglossia のオプション列のテキスト。“実際には読み込まれていない” 場合は `\relax` になる。

```

4528 \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。

```

4529 \let\bxjs@babel@options\relax

```

■Polyglossia について つまり $X_{\text{E}}\text{TeX}$ ・ LuaTeX の場合。

※この場合 etoolbox が使用可能になっている。

```
4530 \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。

```
4531 \pandocSkipLoadPackage{polyglossia}
4532 \bxjs@set@dupload@proc{polyglossia.sty}{%
4533   \bxjs@unset@dupload@proc{polyglossia.sty}%
4534   \ClassWarning\bxjs@clsname
4535     {Package polyglossia is required}%
4536   \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。

```
4537   \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。

```
4538     \ifblank{##2}{%
4539       \ClassWarning\bxjs@clsname
4540         {Main language is invalid, thus the fallback\MessageBreak
4541           definitions will be employed}%
4542       \bxjs@pandoc@polyglossia@ja
```

言語名が japanese だった場合も同様。

```
4543     }{\ifstrequal{##2}{japanese}{%
4544       \ClassWarning\bxjs@clsname
4545         {Main language is japanese, thus the fallback\MessageBreak
4546           definitions will be employed}%
4547       \bxjs@pandoc@polyglossia@ja
```

言語名が非空ならば、本来の polyglossia の処理を実行する。

```
4548     }{%else
4549       \ClassWarning\bxjs@clsname
4550         {Main language is valid (##2),\MessageBreak
4551           thus polyglossia will be loaded}%
4552       \csundef{ver@polyglossia.sty}%
4553       \edef\bxjs@next{%
4554         \noexpand\RequirePackage[\bxjs@polyglossia@options]{polyglossia}[]%
4555       }\bxjs@next
4556       \setmainlanguage[##1]{##2}%
4557     }}}
```

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。

```
4558 \g@addto@macro\bxjs@endpreamble@hook{%
4559   \ifx\bxjs@polyglossia@options\relax
4560     \csundef{ver@polyglossia.sty}%
4561   \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.ldf`) 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。

```
4562 \@onlypreamble\bxjs@pandoc@polyglossia@ja
4563 \def\bxjs@pandoc@polyglossia@ja{%
4564   \renewcommand*\setmainlanguage[2] [] {}%
4565   \newcommand*\setotherlanguage[2] [] {}%
4566   \ifblank{##2}{\else
4567     \cslet{##2}\@empty \cslet{end##2}\@empty
4568     \cslet{text##2}\@firstofone}}%
4569   \newcommand*\setotherlanguages[2] [] {}%
4570   \@for\bxjs@tmpa:={##2}\do{%
4571     \setotherlangauge{\bxjs@tmpa}}}%
```

Polyglossia の読込済マークは外れるようにしておく。

```
4572 \let\bxjs@polyglossia@options\relax}%
4573 \fi
```

■Babel について Xe_{La}TeX 以外の場合。

※ Pandoc 2.6 版において、Lua_{TeX} で用いられる多言語パッケージが Polyglossia から Babel に変更された。

```
4574 \if x\jsEngine\else
    パッケージの読込を検知するため読込済のマークを付けて dupload の処理を仕込む。
4575 \pandocSkipLoadPackage{babel}
4576 \bxjs@set@dupload@proc{babel.sty}{%
4577   \bxjs@unset@dupload@proc{babel.sty}%
4578   \ClassWarning\bxjs@clsname
4579     {Package babel is required}}%
```

パッケージオプションに言語名が空の `main=` があるかを調べる。ある場合は `lang=ja` 対策を実行する。

※`\bxjs@babel@options` には `main=` を除いたオプション列を格納する。

```
4580 \@tempswafalse \let\bxjs@babel@options\@empty
4581 \def\bxjs@tmpb{main=}
4582 \@for\bxjs@tmpa:=#1\do{%
4583   \ifx\bxjs@tmpa\bxjs@tmpb \@tempswatrue
4584   \else \edef\bxjs@babel@options{\bxjs@babel@options,\bxjs@tmpa}%
4585   \fi}%
4586 \if@tempswa
4587   \ClassWarning\bxjs@clsname
4588     {Main language is invalid, so fallback\MessageBreak
4589       definitions will be employed}%
4590   \bxjs@pandoc@polyglossia@ja
```

ない場合は、本来の `babel` の処理を実行する。

```

4591 \else
4592   \ClassWarning\bxjs@clsname
4593     {Main language is valid,\MessageBreak
4594       thus babel will be loaded}%
4595   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4596   \RequirePackage[#1]{babel}[]%
4597 \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。

```

4598 \g@addto@macro\bxjs@endpreamble@hook{%
4599   \ifx\bxjs@babel@options\relax
4600     \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4601   \fi}

```

`\bxjs@pandoc@babel@ja` Pandoc 側で `lang=ja` が指定されていた場合の処理。

```

4602 \@onlypreamble\bxjs@pandoc@babel@ja
4603 \def\bxjs@pandoc@babel@ja{%
4604   \expandafter\let\csname ver@babel.sty\endcsname\@undefined
4605   \edef\bxjs@next{%
4606     \noexpand\RequirePackage[\bxjs@babel@options,english]{babel}[]%
4607   }\bxjs@next
4608   \if j\jsEngine
4609     \RequirePackage[main=japanese]{pxbabel}[]%
4610   \else
4611     \RequirePackage{bxorigcapt}[]%
4612   \fi}

```

`lang` 対策はこれで終わり。

```

4613 \fi

```

E.5 geometry 変数

`geometry` を “再度読み込んだ” 場合に、そのパラメタで `\setpagelayout*` が呼ばれるようにする。

```

4614 \bxjs@set@dupload@proc{geometry.sty}{%
4615   \setpagelayout*{#1}}

```

E.6 CJKmainfont 変数

LuaTeX (+ LuaTeX-jan) の場合に `CJKmainfont` 変数が指定された場合は `\setmainfont` の指定にまわす。

```

4616 \if l\jsEngine
4617   \pandocSkipLoadPackage{xeCJK}
4618   \providecommand*\setCJKmainfont{\setmainfont}
4619 \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。

```
4620 \if x\jsEngine
4621   \expandafter\g@addto@macro\csname opt@xeCJK.sty\endcsname{%
4622     ,space}
4623 \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。

```
4624 \ifjsWithTeX
4625 \@onlypreamble\bxjs@info@or@ban
4626 \def\bxjs@info@or@ban#1{%
4627   \PackageInfo\bxjs@clsname
4628   {Freeze layout on '#1',\MessageBreak detected}}
```

■indent について indent 変数を指定しない場合に「段落表現形式をインデント方式に変更する」動作を抑止する。

```
4629 \unless\ifbxjs@jp@or@indent
4630   \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。

```
4631 \IfFileExists{parskip.sty}{%
4632   \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。

```
4633 }{%else
4634   \eappto\bxjs@endpreamble@hook{%
4635     \parindent=\the\parindent\relax
4636     \parskip=\the\parskip\relax}}
4637 \fi
4638 %
4639 % \paragraph{secnumdepth について}
4640 % |secnumdepth| の値を決めるのは numbersections 変数
4641 % (|-N|/|--number-sections| オプションに連動する)
4642 % や secnumdepth 変数であるが、何れにしても |secnumdepth|
4643 % の値は書き換えられる。
4644 % そのため、|secnumdepth| を復帰させる。
4645 \ifbxjs@jp@or@secnumdepth\else
4646   \bxjs@info@or@ban{secnumdepth}
4647   \eappto\bxjs@endpreamble@hook{%
4648     \c@secnumdepth=\the\c@secnumdepth\relax}
4649 \fi
```



```

4650 %
4651 % \paragraph{block-heading について}
4652 \ifbxjs@jp@or@block@heading\else
4653   \let\bxjs@frozen@paragraph\paragraph
4654   \let\bxjs@frozen@subparagraph\subparagraph
4655   \bxjs@info@or@ban{block-heading}
4656   \appto\bxjs@endpreamble@hook{%
4657     \let\oldparagraph\@undefined
4658     \let\paragraph\bxjs@frozen@paragraph
4659     \let\subparagraph\bxjs@frozen@subparagraph}
4660 \fi
4661 \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。

```

4662 \g@addto@macro\bxjs@begin@document@hook{%
4663   \@tempswafalse

```

まず、マーク変更が必要かを調べる。`\oldparagraph` という制御綴が定義済の場合、Pandoc が `\paragraph` の様式を変更したということなので、マーク変更が必要である。

```

4664   \ifx\oldparagraph\@undefined\else
4665     \@tempswatruue
4666   \fi

```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```

4667   \ifnum\c@secnumdepth>3
4668     \@tempswatruue
4669   \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。

```

4670   \if@tempswa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4671     \let\jsParagraphMark\@empty
4672   \fi\fi}

```

E.10 全角空白文字

```

4673 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4674   \catcode"3000=\active

```

```

4675 \begingroup \catcode`\!=7
4676 \protected\gdef!!!3000{\zwspace}
4677 \endgroup
4678 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4679 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4680 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4681 \fi\fi

```

E.11 hyperref 対策

hyperref の unicode オプションの固定を行う。

```

4682 \if j\jsEngine
4683 \bxjs@fix@hyperref@unicode{false}
4684 \else
4685 \bxjs@fix@hyperref@unicode{true}
4686 \fi

```

E.12 Pandoc 要素に対する和文用の補正

■重要要素 重要 (Strong) 要素に対する L^AT_EX 出力は `\textbf` となるが、代わりに `\strong` を使いたいため、`\textbf` を書き換えてしまう (うわぁ)。

```

4687 \ifbxjs@jp@fix@strong\ifbxjs@jp@strong@cmd
4688 \let\orgtextbf\textbf
4689 \DeclareRobustCommand\bxjstextbf[1]{%
4690 \begingroup
4691 \let\textbf\orgtextbf
4692 \strong{#1}%
4693 \endgroup}%
4694 \g@addto@macro\bxjs@begin@document@hook{%
4695 \let\textbf\bxjstextbf}
4696 \fi\fi

```

`\strong` の書体を設定する。

```

4697 \jsAtEndOfClass{%
4698 \ifx\strongfontdeclare\@undefined\else
4699 \ifcase\bxjs@jp@strong
4700 \or \strongfontdeclare{\sffamily}%
4701 \or \strongfontdeclare{\sffamily\bfseries}%
4702 \fi
4703 \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。

```

4704 \ifbxjs@jp@fix@code
4705 \chardef\bxjs@eghost@c=23
4706 \ifx j\jsEngine \xspcode\bxjs@eghost@c=3
4707 \else\ifx l\jsEngine \ltjsetParameter{alxspmode={\bxjs@eghost@c,3}}
4708 \else\ifx x\jsEngine %no-op

```

```

4709 \else \let\bxjs@eghost@c\@undefined
4710 \fi\fi\fi
4711 \ifx\bxjs@eghost@c\@undefined\else
4712 \font\bxjs@eghost@f=ec-lmr10 at 1.23456pt
4713 \def\bxjs@pan@eghost{\bgroup\bxjs@eghost@f\bxjs@eghost@c\egroup}
4714 \let\orgtexttt\texttt
4715 \DeclareRobustCommand\bxjstexttt[1]{%
4716 \ifmmode \nfss@text{\ttfamily #1}%
4717 \else
4718 \ifvmode \leavevmode \fi
4719 \bxjs@pan@eghost\null
4720 \begingroup \ttfamily #1\endgroup
4721 \null\bxjs@pan@eghost
4722 \fi}
4723 \g@addto@macro\bxjs@begin@document@hook{%
4724 \let\texttt\bxjstexttt}

```

さらに `\verb` の外側にも欧文ゴーストが入るようにする。

※`bxjaghost` の実装を参考にした。

```

4725 \expandafter\def\expandafter\verb\expandafter{%
4726 \expandafter\bxjs@pan@eghost\verb}
4727 \g@addto@macro\verb@egroup{\bxjs@pan@eghost}
4728 \fi
4729 \fi

```

E.13 完了

おしまい。

```
4730 %</pandoc>
```

和文ドライバ実装はここまで。

```
4731 %</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。

```
4732 %<*anc>
```

付録 G 補助パッケージ : `bxjscompat`

ムニャムニャムニャ……。

G.1 準備

```
4733 %<*compat>
4734 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4735 \let\bxac@engine=n
4736 \def\bxac@do#1#2{%
4737   \edef\bxac@tmpa{\string#1}%
4738   \edef\bxac@tmpb{\meaning#1}%
4739   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
4740 \bxac@do\XeTeXversion{\let\bxac@engine=x}
4741 \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。

4742 \ifx\jsAtEndOfClass\@undefined
4743   \let\bxac@delayed@if@bxjs\@firstofone
4744 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
4745 \fi

\ImposeOldLuaTeXBehavior ムニャムニャ。

\RevokeOldLuaTeXBehavior 4746 \newif\ifbxac@in@old@behavior
4747 \let\ImposeOldLuaTeXBehavior\relax
4748 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 XeTeX 部分

```
4749 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。

4750 \@onlypreamble\bxac@adjust@charclass
4751 \bxac@delayed@if@bxjs{%
4752   \@ifpackageloaded{xeCJK}{%}{%else
4753     \ifx\xe@alloc@intercharclass\@undefined\else
4754       \ifnum\xe@alloc@intercharclass=\z@
4755         \PackageInfo\bxac@pkgname
4756           {Setting up interchar class for CJK...\@gobble}%
4757         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4758           \xe@alloc@intercharclass=3
4759         }{%else
4760           \PackageWarning\bxac@pkgname
4761             {Cannot find file 'load-unicode-xetex-classes.tex'%
4762             \@gobble}%
4763         }%
4764       \fi\fi
4765     \ifnum\XeTeXcharclass"3041=\z@
4766       \PackageInfo\bxac@pkgname
```

```

4767      {Adjusting interchar class for CJK...\@gobble}%
4768      \@for\bxac@x:={%
4769          3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4770          3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4771          30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4772          31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4773          31FF%
4774      }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4775  \fi
4776  }%
4777 }

```

以上。

```
4778 \fi
```

G.3 LuaTeX 部分

```

4779 \ifx l\bxac@engine
      ムニヤムニヤ。
4780 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
4781   \chardef\pdftexversion=200
4782   \def\pdftexrevision{0}
4783   \let\pdftexbanner\luatexbanner
4784 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior
4785 \begingroup\expandafter\expandafter\expandafter\endgroup
4786 \expandafter\ifx\csname outputmode\endcsname\relax\else
4787 \def\bxac@ob@list{%
4788   \do{\let}\pdfoutput{\outputmode}%
4789   \do{\let}\pdfpagewidth{\pagewidth}%
4790   \do{\let}\pdfpageheight{\pageheight}%
4791   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}%
4792   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}
4793 \def\bxac@ob@do#1#2{\begingroup
4794   \expandafter\bxac@ob@do@a\csname bxac@string#2\endcsname{#1}#2}
4795 \def\bxac@ob@do@a#1#2#3#4{\endgroup
4796   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
4797   \else \let#3#1\relax \let#1\@undefined
4798   \fi}
4799 \protected\def\ImposeOldLuaTeXBehavior{%
4800   \unless\ifbxac@in@old@behavior
4801     \bxac@in@old@behaviortrue
4802     \let\do\bxac@ob@do \bxac@ob@list
4803   \fi}
4804 \protected\def\RevokeOldLuaTeXBehavior{%
4805   \ifbxac@in@old@behavior
4806     \bxac@in@old@behaviorfalse
4807     \let\do\bxac@ob@do \bxac@ob@list
4808   \fi}

```

4809 \fi

漢字および完成形ハングルのカテゴリコードのムニャムニャ。

```
4810 \ifnum\luatexversion>64 \directlua{
4811   local function range(cs, ce, cc, ff)
4812     if ff or not tex.getcatcode(cs) == cc then
4813       local setcc = tex.setcatcode
4814       for c = cs, ce do setcc(c, cc) end
4815     end
4816   end
4817   range(0x3400, 0x4DB5, 11, false)
4818   range(0x4DB5, 0x4DBF, 11, true)
4819   range(0x4E00, 0x9FCC, 11, false)
4820   range(0x9FCD, 0x9FFF, 11, true)
4821   range(0xAC00, 0xD7A3, 11, false)
4822   range(0x20000, 0x2A6D6, 11, false)
4823   range(0x2A6D7, 0x2A6FF, 11, true)
4824   range(0x2A700, 0x2B734, 11, false)
4825   range(0x2B735, 0x2B73F, 11, true)
4826   range(0x2B740, 0x2B81D, 11, false)
4827   range(0x2B81E, 0x2B81F, 11, true)
4828   range(0x2B820, 0x2CEA1, 11, false)
4829   range(0x2CEA2, 0x2FFFD, 11, true)
4830 } \fi
```

以上。

4831 \fi

G.4 完了

おしまい。

4832 %</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

```
4833 %<*cjkat>
4834 \def\bxjx@pkgname{bxjscjkat}
4835 \newcount\bxjx@canta
4836 \@onlypreamble\bxjx@tmpdo
4837 \@onlypreamble\bxjx@tmpdo@a
4838 \@onlypreamble\bxjx@tmpdo@b
```

\bxjx@engine エンジンの種別。

```

4839 \let\bxjx@engine=n
4840 \def\bxjx@tmpdo#1#2{%
4841   \edef\bxjx@tmpa{\string#1}%
4842   \edef\bxjx@tmpb{\meaning#1}%
4843   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4844 \bxjx@tmpdo\kanjiskip{\let\bxjx@engine=j}
4845 \bxjx@tmpdo\enablecjktoken{\let\bxjx@engine=u}
4846 \bxjx@tmpdo\XeTeXversion{\let\bxjx@engine=x}
4847 \bxjx@tmpdo\pdfTeXversion{\let\bxjx@engine=p}
4848 \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。

```

4849 \def\bxjx@tmpdo#1#2{%
4850   \if#1\bxjx@engine
4851     \@ifpackageloaded{#2}{\}%else
4852     \PackageError\bxjx@pkgname
4853       {Package '#2' must be loaded}%
4854       {Package loading is aborted.\MessageBreak\@ehc}%
4855     \endinput}
4856   \fi}
4857 \bxjx@tmpdo{p}{bxcjkatype}
4858 \bxjx@tmpdo{x}{xeCJK}
4859 \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。

```

4860 \ifx\TextOrMath\@undefined
4861   \RequirePackage{fixltx2e}
4862 \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-j_a と（ほぼ）等価なものに変更
 する。

※ LuaT_EX-j_a との相違点：A830、A960、1B000。

```

4863 \if u\bxjx@engine
4864   \@for\bxjx@tmpa:={%
4865     0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4866     0700,0750,0780,07C0,0800,0840,0860,08A0,0900,0980,%
4867     0A00,0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,%
4868     0F00,1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,%
4869     1720,1740,1760,1780,1800,18B0,1900,1950,1980,19E0,%
4870     1A00,1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1C80,1CC0,%
4871     1CD0,1D00,1D80,1DC0,1E00,1F00,2440,27C0,27F0,2800,%
4872     2A00,2C00,2C60,2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,%
4873     A4D0,A500,A640,A6A0,A700,A720,A800,A830,A840,A880,%
4874     A8E0,A900,A930,A980,A9E0,AA00,AA60,AA80,AAE0,AB00,%
4875     AB30,AB70,ABC0,D800,DB80,DC00,E000,FB00,FB50,FE00,%

```

```

4876 FE70,FFF0,%
4877 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4878 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4879 10480,104B0,10500,10530,10600,10800,10840,10860,%
4880 10880,108E0,10900,10920,10980,109A0,10A00,10A60,%
4881 10A80,10AC0,10B00,10B40,10B60,10B80,10C00,10C80,%
4882 10E60,11000,11080,110D0,11100,11150,11180,111E0,%
4883 11200,11280,112B0,11300,11400,11480,11580,11600,%
4884 11660,11680,11700,118A0,11A00,11A50,11AC0,11C00,%
4885 11C70,11D00,12000,12400,12480,13000,14400,16800,%
4886 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4887 1D200,1D300,1D360,1D400,1D800,1E000,1E800,1E900,%
4888 1EE00,1F000,1F030,1F0A0,1F300,1F600,1F650,1F680,%
4889 1F700,1F780,1F800,1F900,E0000,E0100,F0000,100000,%
4890 00C0%
4891 }\do{%
4892 \@tempcnta="\bxjx@tmpa\relax
4893 \@tempcntb\@tempcnta \advance\@tempcntb\m@ne
4894 \chardef\bxjx@tmpb\kcatcode\@tempcntb
4895 \kcatcode\@tempcnta=15 \kcatcode\@tempcntb\bxjx@tmpb}
4896 \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-jā 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。

```

4897 \@onlypreamble\bxjx@grkcyr@list
4898 \def\bxjx@grkcyr@list{%
4899 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4900 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4901 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4902 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4903 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4904 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4905 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
4906 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4907 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4908 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4909 \do{039B}{LGR}{\textLambda}{\Lambda}%    % GR. C. L. LAMDA

```


4910	\do{039C}{LGR}{\textMu}{M}%	% GR. C. L. MU
4911	\do{039D}{LGR}{\textNu}{N}%	% GR. C. L. NU
4912	\do{039E}{LGR}{\textXi}{\Xi}%	% GR. C. L. XI
4913	\do{039F}{LGR}{\textOmicron}{O}%	% GR. C. L. OMICRON
4914	\do{03A0}{LGR}{\textPi}{\Pi}%	% GR. C. L. PI
4915	\do{03A1}{LGR}{\textRho}{P}%	% GR. C. L. RHO
4916	\do{03A3}{LGR}{\textSigma}{\Sigma}%	% GR. C. L. SIGMA
4917	\do{03A4}{LGR}{\textTau}{T}%	% GR. C. L. TAU
4918	\do{03A5}{LGR}{\textUpsilon}{\Upsilon}%	% GR. C. L. UPSILON
4919	\do{03A6}{LGR}{\textPhi}{\Phi}%	% GR. C. L. PHI
4920	\do{03A7}{LGR}{\textChi}{X}%	% GR. C. L. CHI
4921	\do{03A8}{LGR}{\textPsi}{\Psi}%	% GR. C. L. PSI
4922	\do{03A9}{LGR}{\textOmega}{\Omega}%	% GR. C. L. OMEGA
4923	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
4924	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
4925	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
4926	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4927	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4928	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4929	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4930	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4931	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4932	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4933	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4934	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4935	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4936	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4937	\do{03BF}{LGR}{\textomicron}{o}%	% GR. S. L. OMICRON
4938	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
4939	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
4940	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
4941	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
4942	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
4943	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
4944	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
4945	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
4946	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
4947	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
4948	\do{0401}{T2A}{\CYRYO}{}	% CY. C. L. IO
4949	\do{0410}{T2A}{\CYRA}{}	% CY. C. L. A
4950	\do{0411}{T2A}{\CYRB}{}	% CY. C. L. BE
4951	\do{0412}{T2A}{\CYRV}{}	% CY. C. L. VE
4952	\do{0413}{T2A}{\CYRG}{}	% CY. C. L. GHE
4953	\do{0414}{T2A}{\CYRD}{}	% CY. C. L. DE
4954	\do{0415}{T2A}{\CYRE}{}	% CY. C. L. IE
4955	\do{0416}{T2A}{\CYRZH}{}	% CY. C. L. ZHE
4956	\do{0417}{T2A}{\CYRZ}{}	% CY. C. L. ZE
4957	\do{0418}{T2A}{\CYRI}{}	% CY. C. L. I
4958	\do{0419}{T2A}{\CYRISHRT}{}	% CY. C. L. SHORT I

4959 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
4960 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
4961 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
4962 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
4963 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
4964 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
4965 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
4966 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
4967 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
4968 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
4969 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
4970 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
4971 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
4972 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
4973 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
4974 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
4975 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
4976 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
4977 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
4978 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
4979 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
4980 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
4981 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
4982 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
4983 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
4984 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
4985 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
4986 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
4987 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
4988 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
4989 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
4990 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
4991 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
4992 \do{043B}{T2A}{\cyr1}{}%	% CY. S. L. EL
4993 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
4994 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
4995 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
4996 \do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
4997 \do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
4998 \do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES
4999 \do{0442}{T2A}{\cyrt}{}%	% CY. S. L. TE
5000 \do{0443}{T2A}{\cyru}{}%	% CY. S. L. U
5001 \do{0444}{T2A}{\cyrf}{}%	% CY. S. L. EF
5002 \do{0445}{T2A}{\cyrh}{}%	% CY. S. L. HA
5003 \do{0446}{T2A}{\cyrc}{}%	% CY. S. L. TSE
5004 \do{0447}{T2A}{\cyrch}{}%	% CY. S. L. CHE
5005 \do{0448}{T2A}{\cyrsh}{}%	% CY. S. L. SHA
5006 \do{0449}{T2A}{\cyrshch}{}%	% CY. S. L. SHCHA
5007 \do{044A}{T2A}{\cyrhrdsn}{}%	% CY. S. L. HARD SIGN

```

5008 \do{044B}{T2A}{\cyrery}{}%           % CY. S. L. YERU
5009 \do{044C}{T2A}{\cyrsoftsn}{}%        % CY. S. L. SOFT SIGN
5010 \do{044D}{T2A}{\cyrerev}{}%          % CY. S. L. E
5011 \do{044E}{T2A}{\cyreru}{}%           % CY. S. L. YU
5012 \do{044F}{T2A}{\cyrer}{}%            % CY. S. L. YA
5013 \do{0451}{T2A}{\cyrer}{}%            % CY. S. L. IO
5014 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
5015 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
5016 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
5017 \do{00B1}{TS1}{\textpm}{\pm}%         % PLUS-MINUS SIGN
5018 \do{00B4}{TS1}{\textasciicircum}{}%   % ACUTE ACCENT
5019 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
5020 \do{00D7}{TS1}{\texttimes}{\times}%    % MULTIPLICATION SIGN
5021 \do{00F7}{TS1}{\textdiv}{\div}%        % DIVISION SIGN
5022 }

```

`\mathdegree` 面倒なので補っておく。

```

5023 \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。

```

5024 \newif\ifbxjx@gcc@cjk

```

`\greekasCJK` [公開命令]「特定 CJK 曖昧文字」を和文扱いにする。

```

5025 \newcommand*\greekasCJK{%
5026   \bxjx@gcc@cjktrue}

```

`\nogreekasCJK` [公開命令]「特定 CJK 曖昧文字」を欧文扱いにする。

```

5027 \newcommand*\nogreekasCJK{%
5028   \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。

```

5029 \def\bxjx@tmpdo#1\relax{%
5030   \def\bxjx@fake@grk##1##2{%
5031     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
5032   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
5033     \ifx\##1\%
5034       \bxjx@canta##4\divide\bxjx@canta\@cclvi
5035       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
5036       \mathchar\bxjx@canta
5037     \else ##3\fi}
5038 } \expandafter\bxjx@tmpdo\string\mathchar\relax

```

■pdf \TeX ・up \TeX の場合

```
5039 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0
```

- $\backslash\text{bxjx@KC}/\langle\text{符号値}\rangle$: 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 \LaTeX においても、必ず「`inputenc` が明示的に読み込まれた」状態になる。

```
5040 \@ifpackageloaded{inputenc}{\fi}{%else
5041   \RequirePackage[utf8]{inputenc}}
5042 \def\bxjx@tmpa{utf8}
5043 \ifx\bxjx@tmpa\inputencdoingname
5044   \PackageWarningNoLine\bxjx@pkgname
5045     {Input encoding changed to utf8}%
5046   \inputencoding{utf8}%
5047 \fi
```

up \TeX 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。

```
5048 \if u\bxjx@engine
5049   \kcatcode"0370=15
5050   \kcatcode"0400=15
5051   \kcatcode"0500=15
5052 \fi
```

各文字について `\DeclareUnicodeCharacter` を実行する。

```
5053 \def\bxjx@tmpdo#1{%
5054   \@tempcnta="#1\relax
5055   \expandafter\bxjx@tmpdo@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
5056 \def\bxjx@tmpdo@a#1#2#3#4#5{%
```

引数 = $\backslash\text{bxjx@KC}/\langle\text{符号値}\rangle\{\langle\text{符号値}\rangle\}\{\langle\text{fontenc}\rangle\}\{\langle\text{LICR}\rangle\}\{\langle\text{数式 LICR}\rangle\}$

“数式中の動作”を決定する。 $\langle\text{数式 LICR}\rangle$ が空（数式非対応）なら警告を出す。

```
5057   \ifx\@#5\%
5058     \def\bxjx@tmpa{\@inmathwarn#4}%
```

$\langle\text{数式 LICR}\rangle$ が英字である場合は `\bxjx@fake@grk` で出力する。大文字なら `\Pi`、小文字なら `\pi` を基準文字にする。

```
5059   \else\ifcat A\noexpand#5%
5060     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5061       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
```

それ以外は $\langle\text{数式 LICR}\rangle$ をそのまま実行する。

```
5062   \else \def\bxjx@tmpa{#5}%
5063   \fi\fi
5064   \def\bxjx@tmpb{\bxjx@tmpdo@b{#1}{#2}{#3}{#4}}%
5065   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
```

以降はエンジン種別で分岐する。up $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の場合。

```
5066 \if u\bxjx@engine
5067 \def\bxjx@tmpdo@b#1#2#3#4#5{%
```

引数 = $\backslash\mathrm{bxjx}@KC/\langle\text{符号値}\rangle\{\langle\text{符号値}\rangle\}\{\langle\text{fontenc}\rangle\}\{\langle\mathrm{LICR}\rangle\}\{\langle\text{数式中の動作}\rangle\}$

当該の Unicode 文字の動作は「テキストでは $\langle\mathrm{LICR}\rangle$ 、数式では $\langle\text{数式中の動作}\rangle$ 」となる。 LICR は現在エンコーディングで有効な定義がある場合はそれが実行されるはずである。(つまり、現在が LGR である場合はギリシャ文字は常に欧文扱いになる。) それ以外の場合は LICR を $\backslash\mathrm{bxjx}@ja@or@not$ に帰着させる。この際に、和文用の定義として当該の $\mathrm{kchardef}$ を使用し、その制御綴として $\backslash\mathrm{bxjx}@KC/\dots$ を流用している。

```
5068 \kchardef#1=\@tempcnta
5069 \DeclareTextCommandDefault{#4}\bxjx@ja@or@not{#1}{#3}{#4}}%
5070 \DeclareUnicodeCharacter{#2}\TextOrMath{#4}{#5}}}
```

$\mathrm{pdfT}_{\mathrm{E}}\mathrm{X}$ の場合も処理はほとんど同じ。ただし、和文用の定義として $\backslash\mathrm{UTF}\{\langle\text{符号値}\rangle\}$ を使う ($\backslash\mathrm{UTF}$ は $\mathrm{bxcjkjatype}$ の命令)。 $\backslash\mathrm{bxjx}@KC/\dots$ は使わないが定義済にする必要がある。

```
5071 \else\if p\bxjx@engine
5072 \def\bxjx@tmpdo@b#1#2#3#4#5{%
5073 \mathchardef#1=\@tempcnta
5074 \DeclareTextCommandDefault{#4}\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
5075 \DeclareUnicodeCharacter{#2}\TextOrMath{#4}{#5}}}
```

```
5076 \fi\fi
```

以上の処理を「特定 CJK 曖昧文字」の各々に適用する。

```
5077 \let\do\bxjx@tmpdo \bxjx@grkcyr@list
```

$\backslash\mathrm{bxjx}@DeclareUnicodeCharacter$ $\backslash\mathrm{bxjx}@DeclareUnicodeCharacter$ を改変して、「特定 CJK 曖昧文字」の場合に再定義を抑制したもの。

```
5078 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
5079 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
5080 \@onlypreamble\bxjx@DeclareUnicodeCharacter
5081 \def\bxjx@DeclareUnicodeCharacter#1#2{%
5082 \count@=#1\relax
5083 \expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
5084 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
5085 \else
5086 \wlog{ \space\space skipped defining Unicode char U+#1}%
5087 \fi}
```

$\backslash\mathrm{bxjx}@ja@or@not$ $\backslash\mathrm{bxjx}@ja@or@not\{\langle\text{和文用定義}\rangle\}\{\langle\text{対象 fontenc}\rangle\}\{\langle\mathrm{LICR}\rangle\}$: $\backslash\mathrm{[no]greekasCJK}$ の状態に応じて和文または欧文で文字を出力する。

```
5088 \def\bxjx@ja@or@not#1#2#3{%
```

$\backslash\mathrm{greekasCJK}$ の場合は、無条件に $\langle\text{和文用定義}\rangle$ を実行する。

```
5089 \ifbxjx@gcc@cjk #1%
```

$\backslash\mathrm{nogreekasCJK}$ 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。

```

5090 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
5091 \else \UseTextSymbol{#2}{#3}%
5092 \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。

```

5093 \begingroup
5094 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
5095 \xdef\next{\def\noexpand\DeclareFontEncoding@#1#2#3{%
5096   \noexpand\bxjx@swap@DUC@cmd
5097   \the\toks@
5098   \noexpand\bxjx@swap@DUC@cmd}}
5099 \endgroup\next
5100 \def\bxjx@swap@DUC@cmd{%
5101   \let\bxjx@tmpa\DeclareUnicodeCharacter
5102   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
5103   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa
5104   \let\bxjx@tmpa\relax}

```

以上。

■X_YTeX・LuaTeX の場合

```

5105 \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。

```

5106 \def\bxjx@tmpdo#1{%
5107   \bxjx@cmta="#1\relax
5108   \begingroup
5109     \lccode`~=\bxjx@cmta
5110     \lowercase{\endgroup
5111       \bxjx@tmpdo@a{~}}{#1}}
5112 \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、`upLaTeX` 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。

```

5113 \ifx\\#5\\\let\bxjx@tmpa\relax
5114 \else\ifcat A\noexpand#5%
5115   \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
5116     {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
5117 \else \def\bxjx@tmpa{#5}%
5118 \fi\fi
5119 \ifx\bxjx@tmpa\relax\else
5120   \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
5121 \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。

```

5122 \mathchardef\bxjx@tmpa="119
5123 \ifx\bxjx@tmpa\pi \let\do\bxjx@tmpdo \bxjx@grkcyr@list \fi

```

次に、テキストにおいて「特定 CJK 曖昧文字」の扱いが `\[no]greekasCJK` で切り替わるようにする。

LuaTeX の場合は、LuaTeX-ja の `jacharrange` の設定を変更する。

※ “範囲 2” がギリシャ・キリル文字、“範囲 8” が Latin-1 の記号。

```
5124 \if l\bxjx@engine
5125   \protected\def\greekasCJK{%
5126     \bxjx@gcc@cjctrue
5127     \ltjsetparameter{jacharrange={+2, +8}}
5128   \protected\def\nogreekasCJK{%
5129     \bxjx@gcc@cjcfalse
5130     \ltjsetparameter{jacharrange={-2, -8}}
5131 \fi
```

X_ƎTeX の場合、xeCJK は X_ƎTeX 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。

```
5132 \if x\bxjx@engine
5133   \let\bxjx@gcc@cjkl@list\@empty
5134   \def\do#1#2#3#4{%
5135     \edef\bxjx@gcc@cjkl@list{\bxjx@gcc@cjkl@list
5136       \noexpand\XeTeXcharclass"#1\bxjx@canta}}
5137   \bxjx@grkcy@list
5138   \protected\def\greekasCJK{%
5139     \bxjx@gcc@cjctrue
5140     \bxjx@canta=\@ne \bxjx@gcc@cjkl@list}
5141   \protected\def\nogreekasCJK{%
5142     \bxjx@gcc@cjcfalse
5143     \bxjx@canta=\z@ \bxjx@gcc@cjkl@list}
5144 \fi
```

以上。

```
5145 \fi\fi
```

H.4 初期設定

「特定 CJK 曖昧文字」を欧文扱いにする。

```
5146 \nogreekasCJK
```

H.5 完了

おしまい。

```
5147 %</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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ コードより前に読み込む必要があるため、専ら文書クラス内での読込に限られる。

I.1 準備

```
5148 %<*ancpandoc>
5149 %% このファイルは日本語文字を含みます.
5150 \def\bxjsp@pkgname{bxjscjkat}

\bxjsp@engine エンジンの種別。

5151 \let\bxjsp@engine=n
5152 \@onlypreamble\bxjsp@do
5153 \def\bxjsp@do#1#2{%
5154   \edef\bxjsp@tmpa{\string#1}%
5155   \edef\bxjsp@tmpb{\meaning#1}%
5156   \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
5157 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
5158 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
5159 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
5160 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}
```

I.2 パッケージオプション

`english` オプションが指定されている場合、`\ldots` の調整を抑止する。

※つまり、「グローバルの `english` オプション」が指定されている場合も抑止の対象になる。BXJS クラスの英語モードを想定しているが、それ以外の場合でも、一般的な $\mathrm{IAT}_{\mathrm{E}}\mathrm{X}$ の習慣として、グローバルの `english` は「その文書の基底言語が英語である」ことを示す。

```
5161 \newif\ifbxjsp@english
5162 \DeclareOption{english}{\bxjsp@englishttrue}

 オプション定義はおしまい。

5163 \ProcessOptions*
```

I.3 パッケージ読込の阻止

`\pandocSkipLoadFile` `\pandocSkipLoadFile{〈ファイル名〉}` : 特定のファイルを (`\@filewithoptions` の処理に関して) 読込済であるとマークする。

```
5164 \@onlypreamble\pandocSkipLoadFile
5165 \newcommand*\pandocSkipLoadFile[1]{%
5166   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
5167 \def\bxjsp@skip@load@file@a#1#2{%
5168   \ifx#1\relax
5169     \def#1{2001/01/01}%
5170     \PackageInfo\bxjsp@pkgname
5171       {File '#2' marked as loaded\@gobble}%
```



```
5172 \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。

```
5173 \@onlypreamble\pandocSkipLoadPackage
5174 \newcommand*\pandocSkipLoadPackage[1]{%
5175 \pandocSkipLoadFile{#1.sty}}
```

1.4 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近（2015 年版以降）の `LATEX` ではこれで警告が出る。これを抑止する。

`LATEX` カーネルが新しい場合は `fixltx2e` を読込済にする。

```
5176 \ifx\@IncludeInRelease\@undefined\else
5177 \pandocSkipLoadPackage{fixltx2e}
5178 \fi
```

1.5 cmap パッケージ

エンジンが (u)p`LATEX` のときに `cmap` パッケージが読み込まれるのを阻止する。（実際は警告が出るだけで無害であるが。）

```
5179 \if j\bxjisp@engine
5180 \pandocSkipLoadPackage{cmap}
5181 \fi
```

1.6 microtype パッケージ

警告が多すぎなので消す。

```
5182 \if j\bxjisp@engine \else
5183 \PassOptionsToPackage{verbose=silent}{microtype}
5184 \fi
```

エンジンが (u)p`LATEX` のときに `microtype` パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は `standard` ドライバでこの処理を行っていたが、元来は Pandoc 用の処理なので、1.5 版で `pandoc` に移動。

```
5185 \if j\bxjisp@engine
5186 \pandocSkipLoadPackage{microtype}
5187 \newcommand*\UseMicrotypeSet[2][]{\fi}
5188 \fi
```

1.7 Unicode 文字変換対策

Pandoc で `LATEX` 形式に書き出す場合は、元データ中の一部の Unicode 文字を「`LATEX` の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

…→\ldots{} ‘→` ’→’ “→` ”→’’

日本語 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。

```
5189 \DeclareRobustCommand{\pandocLdots}{%
5190   \relax\ifmmode \expandafter\bxjsp@org@ldots
5191   \else \expandafter\bxjsp@ja@ellipsis
5192   \fi}
5193 \def\bxjsp@ja@ellipsis{…}
5194 \let\bxjsp@org@ldots\ldots
```

`\ldots` の実装を置き換える。

※ Benglish オプション指定時は置き換えない。

```
5195 \AtBeginDocument{%
5196   \let\bxjsp@org@ldots\ldots
5197   \ifbxjsp@english\else \let\ldots\pandocLdots \fi}
```

`\ldots` の直後の文字が非英字の場合、Pandoc は「`\ldots。`」のように空白を入れずに並べて出力する。「Pandoc は非英字と見なすが X_YL^AT_EX・Lua_T_EX 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。

```
5198 \chardef\bxjsp@cc@other=12
5199 \@onlypreamble\bxjsp@makeother@range
5200 \def\bxjsp@makeother@range#1#2{%
5201   \@tempcnta"#1\relax \@tempcntb"#2\relax
5202   \loop\ifnum\@tempcnta<\@tempcntb
5203     \catcode\@tempcnta\bxjsp@cc@other
5204     \advance\@tempcnta\@ne
5205   \repeat}
5206 \ifnum0\if x\bxjsp@engine1\fi\if 1\bxjsp@engine1\fi>0
5207   \catcode"1F23B=\bxjsp@cc@other
5208   \bxjsp@makeother@range{9FCD}{A000}
5209   \bxjsp@makeother@range{1B002}{1B170}
5210   \bxjsp@makeother@range{2B820}{2EBF0}
5211 \fi
```

I.8 PandoLa モジュール

インストール済であれば読み込む。

```
5212 \IfFileExists{bxpandola.sty}{%  
5213   \RequirePackage{bxpandola}\relax  
5214   \PackageInfo{bxjsp@pkgname  
5215     {PandoLa module is loaded\@gobble}  
5216 }{}
```

I.9 完了

おしまい。

```
5217 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
5218 %</anc>
```