

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”)





v1.7c [2018/02/04]

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

目次

1	はじめに	3
2	オプション	7
3	和文フォントの変更	36
4	フォントサイズ	37
5	レイアウト	43
5.1	ページレイアウト	44
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)	56
7	ページスタイル	58
8	文書のマークアップ	61
8.1	表題	61
8.2	章・節	66
8.3	リスト環境	78
8.4	パラメータの設定	86
8.5	フロート	87
8.6	キャプション	89
9	フォントコマンド	90

10	相互参照	92
10.1	目次の類	92
10.2	参考文献	98
10.3	索引	99
10.4	脚注	101
11	段落の頭へのグルー挿入禁止	103
12	いろいろなロゴ	107
13	amsmath との衝突の回避	108
14	初期設定	108
付録 A	和文ドライバの仕様 罨	113
付録 B	和文ドライバ : minimal 罨	114
B.1	補助マクロ	114
B.2	(u)pTeX 用の設定	116
B.3	pdfTeX 用の処理	120
B.4	X _Y TeX 用の処理	121
B.5	後処理 (エンジン共通)	121
付録 C	和文ドライバ : standard 罨	124
C.1	共通処理 (1)	125
C.2	pTeX 用設定	133
C.3	pdfTeX 用設定 : CJK + bxcjkatype	136
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	138
C.5	LuaTeX 用設定 : LuaTeX-ja	141
C.6	共通処理 (2)	143
付録 D	和文ドライバ : modern 罨	144
D.1	フォント設定	145
D.2	fixltx2e 読込	145
D.3	和文カテゴリコード	145
D.4	完了	145
付録 E	和文ドライバ : pandoc 罨	146
E.1	dupload システム	146
E.2	lang 変数	147
E.3	geometry 変数	147
E.4	CJKmainfont 変数	148
E.5	paragraph のマーク	148

E.6	全角空白文字	148
E.7	完了	149
付録 F	補助パッケージ一覧 	149
付録 G	補助パッケージ : bxjscompat 	149
G.1	準備	149
G.2	X _Y TeX 部分	150
G.3	LuaTeX 部分	151
G.4	完了	152
付録 H	補助パッケージ : bxjscjkat 	152
H.1	準備	152
H.2	和文カテゴリコードの設定	153
H.3	ギリシャ・キリル文字の扱い	154
H.4	初期設定	160
H.5	完了	160
付録 I	補助パッケージ : bxjspandoc 	160
I.1	準備	160
I.2	パッケージ読込の阻止	161
I.3	fixltx2e パッケージ	161
I.4	cmap パッケージ	161
I.5	microtype パッケージ	161
I.6	Unicode 文字変換対策	162
I.7	PandoLa モジュール	163
I.8	完了	163

1 はじめに

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

```

<article>  bxjsarticle.cls  短いレポート（章なし）
<report>   bxjsreport.cls  長いレポート（章あり）
<book>     bxjsbook.cls    書籍用
<slide>    bxjsslide.cls   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「p_{La}TeX 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_EX3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいて奥村が改変したものです。権利については両者のものに従います。奥村は何の権利も主張しません。

[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。

以下では実際のコードに即して説明します。

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

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

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

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

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

■ BXJS クラス特有の設定

長さ値の指定で式を利用可能にするため `calc` を読み込む。

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

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

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

クラスの本体ではこの他に `geometry` パッケージが読み込まれる。

互換性のための補助パッケージを読み込む。

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

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

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

```

21 %<report>\let\jsDocClass\jsReport
22 %<slide>\let\jsDocClass\jsSlide

\jsEngine [暗黙文字トークン] エンジン (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。

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

\ifjsWithupTeX [スイッチ] エンジンが (内部漢字コードが Unicode の) upTEX であるか。
34 \newif\ifjsWithupTeX
35 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
36   \jsWithupTeXtrue
37 \fi\fi
38 \let\if@jsc@uplatex\ifjsWithupTeX

\ifjsWithpTeXng [スイッチ] エンジンが pTEX-ng であるか。
39 \newif\ifjsWithpTeXng
40 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}

\ifjsWitheTeX [スイッチ] エンジンが  $\epsilon$ -TEX 拡張をもつか。
41 \newif\ifjsWitheTeX
42 \bxjs@test@engine\epsilonTeXversion{\jsWitheTeXtrue}

非サポートのエンジンの場合は強制終了させる。
※ NTT jTEX と Omega 系。

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

\bxjs@protected  $\epsilon$ -TEX 拡張が有効な場合にのみ \protected の効果をもつ。
53 \ifjsWitheTeX \let\bxjs@protected\protected

```

```

54 \else \let\bxjs@protected\@empty
55 \fi

\bxjs@robust@def 無引数の頑強な命令を定義する。
56 \ifjsWitheTeX
57   \def\bxjs@robust@def{\protected\def}
58 \else
59   \def\bxjs@robust@def{\DeclareRobustCommand*}
60 \fi

\ifjsInPdfMode [スイッチ] pdfTEX / LuaTEX が PDF モードで動作しているか。
※ LuaTEX 0.8x 版でのプリミティブ名変更に対応。
61 \newif\ifjsInPdfMode
62 \@nameuse{ImposeOldLuaTeXBehavior}
63 \let\bxjs@tmpa\PackageWarningNoLine
64 \let\PackageWarningNoLine\PackageInfo % suppress warning
65 \RequirePackage{ifpdf}
66 \let\PackageWarningNoLine\bxjs@tmpa
67 \@nameuse{RevokeOldLuaTeXBehavior}
68 \let\ifjsInPdfMode\ifpdf

\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。
69 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
70   #1\expandafter\@firstoftwo
71   \else\expandafter\@secondoftwo
72   \fi}

\bxjs@cslet \bxjs@cslet{⟨名前 1⟩}\制御綴 :
73 \def\bxjs@cslet#1{%
74   \expandafter\let\csname#1\endcsname}

\bxjs@csletcs \bxjs@csletcs{⟨名前 1⟩}{⟨名前 2⟩} :
75 \def\bxjs@csletcs#1#2{%
76   \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。
77 \def\bxjs@catopt#1#2{%
78   #1\if\relax#1\relax\else\if\relax#2\relax\else,\fi\fi#2}

\jsAtEndOfClass このクラスの読込終了時に対するフック。(補助パッケージ中で用いられる。)
79 \def\jsAtEndOfClass{%
80   \expandafter\g@addto@macro\csname\bxjs@clsname.cls-h@k\endcsname}

```

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

※現在の Lua^AT_EX 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。

```

81 \@onlypreamble\bxjs@restore@jltrcc
82 \let\bxjs@restore@jltrcc\@empty
83 \if 1\jsEngine
84 \def\bxjs@change@jltrcc#1{%
85   \xdef\bxjs@restore@jltrcc{%
86     \bxjs@restore@jltrcc
87     \catcode`#1=\the\catcode`#1\relax}%
88   \catcode`#1=11\relax}
89 \@tfor\bxjs@x:=西曆\do
90   {\expandafter\bxjs@change@jltrcc\bxjs@x}
91 \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。

```

92 \bxjs@robust@def\jsInhibitGlue{%
93   \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。

```

94 \if@compatibility
95   \ClassError\bxjs@clsname
96   {Something went chaotic!\MessageBreak
97     (How come '\string\documentstyle' is there?)\MessageBreak
98     I cannot go a single step further...}
99   {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
100    then there'll still be hope....}
101   \expandafter\@firstofone
102 \else \expandafter\@gobble
103 \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。

```
104 \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。

```
105 \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。

```
106 %<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。
```

```
107 %<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。

```
108 %<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。

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

デフォルトは `a4paper` です。

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

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

```
110 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
111 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
112 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
113 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
114 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
115 \DeclareOption{b4paper}{\bxjs@setpaper{{257true mm}{364true mm}}}
116 \DeclareOption{b5paper}{\bxjs@setpaper{{182true mm}{257true mm}}}
117 \DeclareOption{b6paper}{\bxjs@setpaper{{128true mm}{182true mm}}}
118 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
119 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
120 \DeclareOption{b4j}{\bxjs@setpaper{{257true mm}{364true mm}}}
```



```

121 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
122 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
123 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}

```

※...var を Pandoc で使えるように後ろに paper をつけた形を用意する。

```

124 \DeclareOption{a4varpaper}{\bxjs@setpaper{{210truemm}{283truemm}}}
125 \DeclareOption{b5varpaper}{\bxjs@setpaper{{182truemm}{230truemm}}}
126 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
127 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
128 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

129 \newif\if@landscape
130 \@landscapefalse
131 \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。

```

132 \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。

```

133 %<!slide>\@slidefalse
134 %<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。

```
135 \newcommand{\@ptsize}{0}
136 \def\bxjs@param@basefontsize{10pt}
137 \def\jsUnusualPtSize{-20}
```

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

```
138 \def\bxjs@setbasefontsize#1{%
139   \bxjs@setbasefontlength\@tempdima{#1}%
140   \edef\bxjs@param@basefontsize{\the\@tempdima}%
141   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
142   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
143   \else\ifdim\@tempdima=12pt  \long\def\@ptsize{2}%
144   \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi}
```

`\bxjs@setbasefontlength` `base`、`jbase` で指定される長さ（式）のための特別な `\setlength`。与えられた式が“(実数)Q”の形の場合、Q 単位の長さを代入する（この場合“式”は使えない）。

※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。

```
145 \def\bxjs@setbasefontlength#1#2{%
```

ここで `true` の長さが使われるのは不合理なので、式が“true”を含む場合には警告を出す。

```
146   \bxjs@setbasefontlength@a#2true\@nnil
```

式の末尾が“Q”である時は特別に扱い、それ以外は `\setlength` に移譲する。

```
147   \ifx j\jsEngine \setlength#1{#2}%
148   \else
149     \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
150     \ifx\bxjs@tmpa\relax \setlength#1{#2}%
151     \else \@tempdimc0.25mm #1=\bxjs@tmpa\@tempdimc
152     \fi
153   \fi}
154 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
155   \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
156   \else \def\bxjs@tmpa{#1}%
157   \fi}
158 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
159   \ifx\@nnil#2\@nnil\else
160     \ClassWarningNoLine\bxjs@clsname
161     {You should not use 'true' lengths here}%
162   \fi}
```

`\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。

```
163 \newif\ifjsc@mag
164 \newif\ifjsc@mag@xreal
165 %\let\jsc@magscale\@undefined
166 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
```

```

167 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
168 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
169 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
170 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
171 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
172 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
173 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
174 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
175 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
176 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
177 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
178 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
179 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
180 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
181 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
182 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
183 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
184 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

185 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
186 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
187 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}

```

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

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

```

188 \if j\jsEngine
189 \hour\time \divide\hour by 60\relax
190 \@tempcnta\hour \multiply\@tempcnta 60\relax
191 \minute\time \advance\minute-\@tempcnta
192 \DeclareOption{tombow}{%
193   \tombowtrue \tombowdatetrue
194   \setlength{\@tombowwidth}{.1\p}%
195   \@bannertoken{%
196     \jobname\space(\number\year-\two@digits\month-\two@digits\day
197     \space\two@digits\hour:\two@digits\minute)}}%
198   \maketombowbox}
199 \DeclareOption{tombo}{%
200   \tombowtrue \tombowdatefalse
201   \setlength{\@tombowwidth}{.1\p}%

```

```
202 \maketombowbox}
203 \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。

```
204 \if j\jsEngine
205 \DeclareOption{mentuke}{%
206 \tombowtrue \tombowdatefalse
207 \setlength{\@tombowwidth}{\z@}%
208 \maketombowbox}
209 \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。

```
210 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
211 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
212 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
213 \DeclareOption{onecolumn}{\@twocolumnfalse}
214 \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。

```
215 \DeclareOption{titlepage}{\@titlepagetrue}
216 \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` と表すことにしています。 `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。

```
217 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
218 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
219 %<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。

```
220 \def\eqnarray{%
221 \stepcounter{equation}%
222 \def\@currentlabel{\p@equation\theequation}%
```

```

223 \global\@eqnswtrue
224 \m@th
225 \global\@eqcnt\z@
226 \tabskip\@centering
227 \let\\\@eqnocr
228 $$\everycr{}\halign to\displaywidth\bgroup
229     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
230     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
231     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
232     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
233     \tabskip\z@skip
234 \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。森本さんにしただけで訂正しました。

```

235 \DeclareOption{leqno}{\input{leqno.clo}}
236 \DeclareOption{fleqn}{\input{fleqn.clo}}
237 % fleqn 用の eqnarray 環境の再定義
238 \def\eqnarray{%
239     \stepcounter{equation}%
240     \def\@currentlabel{\p@equation\theequation}%
241     \global\@eqnswtrue\m@th
242     \global\@eqcnt\z@
243     \tabskip\mathindent
244     \let\=\@eqnocr
245     \setlength\abovedisplayskip{\topsep}%
246     \ifvmode
247         \addtolength\abovedisplayskip{\partopsep}%
248     \fi
249     \addtolength\abovedisplayskip{\parskip}%
250     \setlength\belowdisplayskip{\abovedisplayskip}%
251     \setlength\belowdisplayshortskip{\abovedisplayskip}%
252     \setlength\abovedisplayshortskip{\abovedisplayskip}%
253     $$\everycr{}\halign to\linewidth% $$
254     \bgroup
255     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
256     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
257     &\global\@eqcnt\tw@
258     $\displaystyle{##}$\hfil \tabskip\@centering
259     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
260     \tabskip\z@skip\cr
261 }

```

■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。

```

262 % \DeclareOption{openbib}{%
263 %     \AtEndOfPackage{%
264 %         \renewcommand\@openbib@code{%

```

```

265 %      \advance\leftmargin\bibindent
266 %      \itemindent -\bibindent
267 %      \listparindent \itemindent
268 %      \parsep \z@}%
269 %      \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：

```

270 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。
271 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
272 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
273 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
274 \define@key{bxjs}{enablejfam}[true]{%
275   \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。

```

276 \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 JS クラスは \ifdraft という公開名のスイッチを用いているが、これは ifdraft パッケージと衝突するので、代わりに \ifjsDraft の名前を用い、本文開始時に \ifdraft が未定義の場合に限り、\ifjsDraft を \ifdraft にコピーする処理にする。

※ JS クラスの \ifdraft が廃止されたので、\ifdraft は 2.0 版で廃止を予定する。

```

277 \let\ifjsDraft\iffalse
278 \@onlypreamble\bxjs@draft
279 \def\bxjs@draft#1{%
280   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}

```

```

281 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
282 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
283 \AtBeginDocument{%
284   \expandafter\ifx\csname ifdraft\endcsname\relax
285     \expandafter\let\csname ifdraft\endcsname\expandafter\endcsname
286     \csname ifjsDraft\endcsname
287   \fi}

```

■和文フォントメトリックの選択 ここでは新しい jis フォントメトリックを標準で使いますが、古い 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 メトリックが使えます。

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。

```

288 \newif\ifbxjs@papersize
289 \bxjs@papersizetrue
290 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
291 \DeclareOption{papersize}{\bxjs@papersizetrue}

```

■英語化 オプション english を新設しました。

```

292 \newif\if@english
293 \@englishfalse
294 \DeclareOption{english}{\@englishtrue}

```

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

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

BXJS では当初から bxjsreport クラスが用意されている。

■jslogo パッケージの読み込み L^AT_EX 関連のロゴを再定義する jslogo パッケージを読み込まないオプション nojslogo を新設しました。jslogo オプションの指定で従来どおりの

動作となります。デフォルトは `jslogo` で、すなわちパッケージを読み込みます。

BXJS クラスでは、`nojslogo` を既定とする。

```
295 \newif\if@jslogo \@jslogofalse
296 \DeclareOption{jslogo}{\@jslogotrue}
297 \DeclareOption{nojslogo}{\@jslogofalse}
```

■BXJS 特有のオプションの一覧

- エンジンオプション：`xelatex` 等。
- ドライバオプション：`dvipdfmx` 等。
- 複合設定オプション：`pandoc` 等。
- `nopapersize`：`papersize`（既定で有効）の否定。
- `zw` / `noz`：`\jsZw` と等価な命令として `\zw` を定義する／しない。
- `js` / `nojs`：JS クラスを読込済として扱う／扱わない。
- `precisetext` / `noprecisetext`： \XeTeX の “`generateactualtext`” を有効／無効にする。
- `simplejasetup` / `nosimplejasetup`： \XeTeX の “`linebreaklocale`” を有効／無効にする。
- `bigcode` / `nobigcode`： \upTeX で CMap として UTF8-UCS2 の代わりに UTF8-UTF16 を使う／使わない。
- `oldfontcommands` / `nooldfontcommands`：古い “二文字フォント命令” に対する警告を抑止する／しない。
- `base=<dimen>`：基底フォントサイズを直接指定する。（`xxpt` オプションの代用なので、既定値は 10pt である。）
- `jbase=<dimen>`：基底フォントサイズを “和文規準で” 直接指定する。
- `scale=<real>`：和文フォントのスケールを表すマクロ `\jsScale` の値を設定する。もちろんこの値を何らかの方法で和文処理モジュールに渡さないと意味を成さない。既定値は 0.924715 (= 13Q/10pt)。
- `noscale`：`scale=1` と等価。
- `mag=<int>`：`\mag` 値の直接設定。既定は `base` から算出する。
- `paper={<dimen:width>}{<dimen:height>}`：用紙サイズ設定。用紙サイズオプションの代用で、既定値は `a4paper` 相当。
- `ja=<name>`：使用する和文ドライバの指定。
- `jafont=<name>`：和文フォントプリセットの指定。
- `japaram=<name>`：和文フォントパラメタの指定。
- `magstyle=<name>`：“版面拡大” の実現方法の選択。
- `dvi=<name>`：DVI モードの時のみに参照されるドライバ指定。
- `geometry={class | user}`：`geometry` パッケージの読込を自動的に行うかユーザに

任せるか。

- `fancyhdr=<bool>` : `fancyhdr` パッケージ用の調整を行うか。
- `layout=<name>` : レイアウト変種の指定。
- `textwidth-limit=<number>` : `bxjsbook` における、`\textwidth` の上限の全角単位での値。
- `paragraph-mark=<char>` : パラグラフのマーク。
- `whole-zw-lines=<bool>` : 行長を全角単位に丸めるか。
- `hyperref-enc=<bool>` : `hyperref` の文字コード指定補正を行うか。

`\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 128 pt 以上の場合でも動作するように修正した。

```
298 \mathchardef\bxjs@isc@ll=128
299 \mathchardef\bxjs@isc@sl=259
300 \def\bxjs@isc@sl@h{65539 }
301 \def\bxjs@invscale#1#2{%
302   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
303   \ifdim\@tempdima<\bxjs@isc@ll\p@
304     \@tempcnta\@tempdima \multiply\@tempcnta\@ccclvi
305     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@ccclvi
306   \else
307     \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
308     \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
309   \fi
310   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
311   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
312   \@tempdimb\@tempcnta\@ne
313   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
314   \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
315   \@whiledim\@tempdimb<\@tempdimc\do{%
316     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
317     \advance\@tempcntb\@ne \divide\@tempcntb\tw@
318     \ifdim #2\@tempcntb>\@tempdima
319       \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
320     \else \@tempdimb=\@tempcntb\@ne \fi}%
321   \xdef\bxjs@gtmpa{\the\@tempdimb}%
322   \endgroup #1=\bxjs@gtmpa\relax}
```

■複合設定オプション

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

れる場合に用意される。

`\bxjs@composite@proc` 複合設定オプションのための遅延処理マクロ。

```
323 \let\bxjs@composite@proc\relax
```

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

```
324 \DeclareOption{pandoc}{%
```

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

※実際の和文ドライバ・エンジン設定より優先される。

```
325 \def\bxjs@composite@proc{%
326   \bxjs@oldfontcommandstrue
327   \setkeys{bxjs}{ja=pandoc}%
328   \let\bxjs@engine@given=*\%
```

ドライバオプションを `dvi=dvipdfmx` 相当に変更する。

※これは実際のドライバ設定で上書きできる (オプション宣言順に注意)。

TODO: できない気がする…。

```
329 \def\bxjs@driver@opt{dvipdfmx}%
330 \bxjs@dvi@opttrue}
```

■エンジン・ドライバオプション

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

```
331 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
332 %\let\bxjs@engine@opt\@undefined
```

エンジン明示指定のオプションの処理。

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

```
333 \DeclareOption{autodetect-engine}{%
334   \let\bxjs@engine@given=*\%
335 \DeclareOption{latex}{%
336   \def\bxjs@engine@opt{latex}%
337   \let\bxjs@engine@given=n}%
338 \DeclareOption{platex}{%
339   \def\bxjs@engine@opt{platex}%
340   \let\bxjs@engine@given=j}%
341 \DeclareOption{uplatex}{%
342   \def\bxjs@engine@opt{uplatex}%
343   \let\bxjs@engine@given=u}%
344 \DeclareOption{xelatex}{%
345   \def\bxjs@engine@opt{xelatex}%
```

```

346 \let\bxjs@engine@given=x}
347 \DeclareOption{pdflatex}{%
348 \def\bxjs@engine@opt{pdflatex}%
349 \let\bxjs@engine@given=p}
350 \DeclareOption{lualatex}{%
351 \def\bxjs@engine@opt{lualatex}%
352 \let\bxjs@engine@given=l}
353 \DeclareOption{platex-ng}{%
354 \def\bxjs@engine@opt{platex-ng}%
355 \let\bxjs@engine@given=g}
356 \DeclareOption{platex-ng*}{%
357 \def\bxjs@engine@opt{platex-ng*}%
358 \let\bxjs@platexng@nodrv=t%
359 \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。

```

360 %\let\bxjs@driver@given\@undefined
361 \let\bxjs@driver@@dvimode=0
362 \let\bxjs@driver@@dvipdfmx=1
363 \let\bxjs@driver@@pdfmode=2
364 \let\bxjs@driver@@xetex=3
365 \let\bxjs@driver@@dvips=4
366 \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。

```

367 %\let\bxjs@driver@opt\@undefined

368 \DeclareOption{dvips}{%
369 \def\bxjs@driver@opt{dvips}%
370 \let\bxjs@driver@given\bxjs@driver@@dvips}
371 \DeclareOption{dviout}{%
372 \def\bxjs@driver@opt{dviout}%
373 \let\bxjs@driver@given\bxjs@driver@@dvimode}
374 \DeclareOption{xdvi}{%
375 \def\bxjs@driver@opt{xdvi}%
376 \let\bxjs@driver@given\bxjs@driver@@dvimode}
377 \DeclareOption{dvipdfmx}{%
378 \def\bxjs@driver@opt{dvipdfmx}%
379 \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
380 \DeclareOption{nodvidriver}{%
381 \def\bxjs@driver@opt{nodvidriver}%
382 \let\bxjs@driver@given\bxjs@driver@@none}
383 \DeclareOption{pdftex}{%
384 \def\bxjs@driver@opt{pdftex}%
385 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
386 \DeclareOption{luatex}{%
387 \def\bxjs@driver@opt{luatex}%
388 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
389 \DeclareOption{xetex}{%

```

```

390 \def\bxjs@driver@opt{xetex}%
391 \let\bxjs@driver@given\bxjs@driver@@xetex}

```

「もし DVI モードであればドライバを dvipdfmx にする」というオプション。
 ※ 1.2 版で dvi オプションが新設されたが、互換性のためこのオプションも残す。

```

392 \DeclareOption{dvipdfmx-if-dvi}{%
393 \setkeys{bxjs}{dvi=dvipdfmx}}

```

■その他の BXJS 独自オプション

`\ifbxjs@usezw` `\jsZw` の同義語として `\zw` を使えるようにするか。既定は真。

```

394 \newif\ifbxjs@usezw \bxjs@usezwtrue

```

`zw`、`noz` オプションの定義。

```

395 \DeclareOption{noz}{%
396 \bxjs@usezwfalse}
397 \DeclareOption{zw}{%
398 \bxjs@usezwtrue}

```

`\ifbxjs@disguise@js` JS クラスの派生クラスのふりをするか。既定は真。

```

399 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue

```

`nojs`、`js` オプションの定義。

```

400 \DeclareOption{nojs}{%
401 \bxjs@disguise@jsfalse}
402 \DeclareOption{js}{%
403 \bxjs@disguise@jstrue}

```

`\ifbxjs@precisetext` \TeX の “generateactualtext” を有効にするか。既定は偽。

```

404 \newif\ifbxjs@precisetext

```

`noprecisetext` / `precisetext` オプションの定義。

```

405 \DeclareOption{noprecisetext}{%
406 \bxjs@precisetextfalse}
407 \DeclareOption{precisetext}{%
408 \bxjs@precisetexttrue}

```

`\ifbxjs@simplejasetup` \TeX の “linebreaklocale” を有効にするか。既定は真（であるが多くの場合は後に無効化される）。

```

409 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

```

`nosimplejasetup` / `simplejasetup` オプションの定義。

```

410 \DeclareOption{nosimplejasetup}{%
411 \bxjs@simplejasetupfalse}
412 \DeclareOption{simplejasetup}{%
413 \bxjs@simplejasetuptrue}

```

`\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、オプションで指定することとする。

```
414 \newif\ifbxjs@bigcode \bxjs@bigcodefalse
```

その上で、「`TeX` 環境がある程度新しければ利用可能であろう」と判断し `bxjs@bigcode` の既定値を真とする。具体的な判断基準として、「`TeX` のバージョンが 3.14159265 (2014 年 1 月) 以上であるか」を採用する。

```
415 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
416 \def\bxjs@tmpb#1 #2#3\@nil{%
417   \ifx1#2\bxjs@bigcodetrue \fi}
418 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil
```

`nobigcode` / `bigcode` オプションの定義。

```
419 \DeclareOption{nobigcode}{%
420   \bxjs@bigcodefalse}
421 \DeclareOption{bigcode}{%
422   \bxjs@bigcodetrue}
```

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

```
423 \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。

```
424 \DeclareOption{nooldfontcommands}{%
425   \bxjs@oldfontcommandsfalse}
426 \DeclareOption{oldfontcommands}{%
427   \bxjs@oldfontcommandstrue}
```

■keyval 型のオプション

```
428 \def\bxjs@setkey{%
429   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
430 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
431 \DeclareOption*{\bxjs@setkey}
```

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

```
432 \def\bxjs@safe@setkeys#1#2{%
433   \let\bxjs@KV@errx\KV@errx
434   \let\KV@errx\bxjs@safe@setkeys@a
435   \setkeys{#1}{#2}%
436   \let\KV@errx\bxjs@KV@errx}
437 \def\bxjs@safe@setkeys@a#1{}
```

```
\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。

```
438 \def\bxjs@set@keyval#1#2#3{%
439   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
440   \ifx\bxjs@next\relax
441     \bxjs@error@keyval{#1}-{#2}%
442     #3%
443   \else \bxjs@next
444   \fi}
445 \@onlypreamble\bxjs@error@keyval
446 \def\bxjs@error@keyval#1#2{%
447   \ClassError\bxjs@clsname
448   {Invalid value '#2' for option #1}\@ehc}
```

`\ifbxjs@scaleset` 和文スケール値が指定されたか。

```
449 \newif\ifbxjs@scaleset
```

`\jsScale` 〔実数値マクロ〕和文スケール値。

```
450 \def\jsScale{0.924715}
```

`base` オプションの処理。

```
451 \define@key{bxjs}{base}{\bxjs@setbasefontsize{#1}}
```

`jbase` オプションの処理。ここでは `\jsScale` の値を使用する。`scale` の処理との順序依存を消すため、`jbase` の処理の実行を遅延させている。

```
452 \@onlypreamble\bxjs@do@opt@jbase
453 \let\bxjs@do@opt@jbase\relax
454 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
455 \def\bxjs@setjbasefontsize#1{%
456   \def\bxjs@do@opt@jbase{%
457     \bxjs@setbasefontlength\@tempdima{#1}%
458     \bxjs@invscale\@tempdima\jsScale
459     \bxjs@setbasefontsize{\@tempdima}}}
```

`scale` オプションの処理。

```
460 \define@key{bxjs}{scale}{%
461   \bxjs@scalesettrue
462   \edef\jsScale{#1}}
```

`noscale` オプションの処理。

```
463 \DeclareOption{noscale}{%
464   \bxjs@scalesettrue
465   \def\jsScale{1}}
```

`\bxjs@param@mag` `mag` オプションの値。

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

`mag` オプションの処理。

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

`paper` オプションの処理。

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

`\bxjs@jadriver` 和文ドライバの名前。

```
469 \let\bxjs@jadriver\relax
470 %\let\bxjs@jadriver@given\@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)。

```
471 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
472 \define@key{bxjs}{ja}[\relax]{%
473   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}
```

`\jsJaFont` 和文フォント設定の名前。

```
474 \let\jsJaFont\@empty
```

`jafont` オプションの処理。

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

`\jsJaParam` 和文ドライバパラメタの文字列。

```
476 \let\jsJaParam\@empty
```

`japaram` オプションの処理。

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

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

```
478 \let\bxjs@magstyle@mag=m
479 \let\bxjs@magstyle@real=r
480 \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。

```
481 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
482 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
483 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

`\bxjs@magstyle@default` は既定の値を表す。

```
484 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
485 \ifx l\jsEngine \ifnum\luatexversion>86
486   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
487 \fi\fi
488 \ifjsWithpTeXng
489   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
490 \fi
491 \let\bxjs@magstyle\bxjs@magstyle@default
```

`magstyle` オプションの処理。

```
492 \define@key{bxjs}{magstyle}{%
493   \expandafter\let\expandafter\bxjs@magstyle\csname
```

```

494   bxjs@magstyle@#1\endcsname
495   \ifx\bxjs@magstyle\relax
496     \ClassError\bxjs@clsname
497     {Invalid value '#1' for option magstyle}\@ehc
498     \let\bxjs@magstyle\bxjs@magstyle@default
499   \fi}

```

\bxjs@geometry geometry オプションの値。

```

500 \let\bxjs@geometry@class=c
501 \let\bxjs@geometry@user=u
502 \let\bxjs@geometry\bxjs@geometry@class

```

geometry オプションの処理。

```

503 \define@key{bxjs}{geometry}{%
504   \expandafter\let\expandafter\bxjs@geometry\csname
505     bxjs@geometry@#1\endcsname
506   \ifx\bxjs@geometry\relax
507     \ClassError\bxjs@clsname
508     {Invalid value '#1' for option geometry}\@ehc
509     \let\bxjs@geometry\bxjs@geometry@class
510   \fi}

```

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

```

511 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

```

fancyhdr オプションの処理。

```

512 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
513 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
514 \define@key{bxjs}{fancyhdr}[true]{%
515   \bxjs@set@keyval{fancyhdr}{#1}{}}

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

516 \newif\ifbxjs@dvi@opt

```

DVI モードのドライバとドライバ種別との対応。

```

517 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
518 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
519 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
520 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
521 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

```

dvi オプションの処理。

```

522 \define@key{bxjs}{dvi}{%
523   \expandafter\let\expandafter\bxjs@tmpa\csname
524     bxjs@dvidriver@@#1\endcsname
525   \ifx\bxjs@tmpa\relax
526     \ClassError\bxjs@clsname
527     {Invalid value '#1' for option dvi}\@ehc
528   \else

```



```

\bxjs@driver@given を未定義にしていることに注意。
529 \def\bxjs@driver@opt{#1}%
530 \let\bxjs@driver@given\@undefined
531 \bxjs@dvi@opttrue
532 \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。
533 \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。
534 \newif\ifbxjs@force@chapterabstract
535 %<book>\bxjs@force@chapterabstracttrue

layout オプションの処理。
536 \@namedef{bxjs@kv@layout@v1}{%
537 %<book>\bxjs@layout@buggyhmargintrue
538 %<report>\bxjs@force@chapterabstracttrue
539 }
540 \@namedef{bxjs@kv@layout@v2}{%
541 %<book>\bxjs@layout@buggyhmarginfalse
542 %<report>\bxjs@force@chapterabstractfalse
543 }
544 \define@key{bxjs}{layout}{%
545 \bxjs@set@keyval{layout}{#1}{}}

\bxjs@textwidth@limit textwidth-limit の指定値。 \textwidth の上限。
546 %\let\bxjs@textwidth@limit\@undefined
547 \define@key{bxjs}{textwidth-limit}{%
548 \edef\bxjs@textwidth@limit{#1}}

\bxjs@paragraph@mark paragraph-mark の指定値。 パラグラフのマーク。
549 %\let\bxjs@paragraph@mark\@undefined
550 \define@key{bxjs}{paragraph-mark}{%
551 \edef\bxjs@paragraph@mark{#1}}

\ifbxjs@whole@zw@lines [スイッチ] whole-zw-lines の指定値。
552 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
553 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
554 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
555 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}

\ifbxjs@jaspace@cmd [スイッチ] jaspacemcmd の指定値。
556 \newif\ifbxjs@jaspace@cmd \bxjs@jaspace@cmdtrue
557 \let\bxjs@kv@jaspacemcmd@true\bxjs@jaspace@cmdtrue
558 \let\bxjs@kv@jaspacemcmd@false\bxjs@jaspace@cmdfalse
559 \define@key{bxjs}{jaspacemcmd}[true]{\bxjs@set@keyval{jaspacemcmd}{#1}{}}
560 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{jaspacemcmd}{#1}{}}

```

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

```
561 \newif\ifbxjs@fix@at@cmd \bxjs@fix@at@cmdtrue
562 \let\bxjs@kv@fixatcmd@true\bxjs@fix@at@cmdtrue
563 \let\bxjs@kv@fixatcmd@false\bxjs@fix@at@cmdfalse
564 \define@key{bxjs}{fix-at-cmd}[true]{\bxjs@set@keyval{fixatcmd}{#1}{}}
```

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

```
565 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
566 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
567 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
568 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#1}{}}
```

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

```
569 \chardef\bxjs@everyparhook@none=0
570 \chardef\bxjs@everyparhook@compat=1
571 \chardef\bxjs@everyparhook@modern=2
572 \if j\jsEngine
573   \let\bxjs@everyparhook\bxjs@everyparhook@compat
574 \else
575   \let\bxjs@everyparhook\bxjs@everyparhook@modern
576 \fi
577 \def\bxjs@kv@everyparhook@none{\let\bxjs@everyparhook\bxjs@everyparhook@none}
578 \def\bxjs@kv@everyparhook@compat{\let\bxjs@everyparhook\bxjs@everyparhook@compat}
579 \def\bxjs@kv@everyparhook@modern{\let\bxjs@everyparhook\bxjs@everyparhook@modern}
580 \define@key{bxjs}{everyparhook}{\bxjs@set@keyval{everyparhook}{#1}{}}
```

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

```
581 \chardef\bxjs@label@section@none=0
582 \chardef\bxjs@label@section@compat=1
583 \chardef\bxjs@label@section@modern=2
584 \let\bxjs@label@section\bxjs@label@section@compat
585 \def\bxjs@kv@labelsection@none{\let\bxjs@label@section\bxjs@label@section@none}
586 \def\bxjs@kv@labelsection@compat{\let\bxjs@label@section\bxjs@label@section@compat}
587 \def\bxjs@kv@labelsection@modern{\let\bxjs@label@section\bxjs@label@section@modern}
588 \define@key{bxjs}{label-section}{\bxjs@set@keyval{labelsection}{#1}{}}
```

■ 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。

```
589 \let\bxjs@ltx@removeelement\@removeelement
590 \def\@removeelement#1#2#3{%
591   \def\reserved@a{#2}%
```

```

592 \ifx\reserved@a\@empty \let#3\@empty
593 \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
594 \fi}

```

デフォルトのオプションを実行し、dvi ファイルの先頭に dvips の papersize special を書き込みます。この special は dvips や最近の dviout が対応しています。multicols や url を \RequirePackage するのはやめました。

```

595 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
596 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
597 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
598 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
599 \ProcessOptions\relax
600 \bxjs@composite@proc

```

後処理

```

601 \if@slide
602 \def\maybeblue{\@ifundefined{ver@color.sty}{}{\color{blue}}}
603 \fi
604 \if@landscape
605 \setlength\@tempdima {\paperheight}
606 \setlength\paperheight{\paperwidth}
607 \setlength\paperwidth {\@tempdima}
608 \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込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。

```

609 \@onlypreamble\bxjs@purge@brace@elts
610 \def\bxjs@purge@brace@elts{%
611 \def\bxjs@tmpa{\@gobble}%
612 \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
613 \let\@classoptionslist\bxjs@tmpa}
614 \@onlypreamble\bxjs@purge@be@a
615 \def\bxjs@purge@be@a#1,{%
616 \ifx\@nil#1\relax\else
617 \bxjs@purge@be@b#1{\@nil}
618 \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
619 \expandafter\bxjs@purge@be@a
620 \fi}
621 \@onlypreamble\bxjs@purge@be@b
622 \def\bxjs@purge@be@b#1#{\bxjs@purge@be@c}
623 \@onlypreamble\bxjs@purge@be@c
624 \def\bxjs@purge@be@c#1\@nil{%
625 \ifx\@nil#1\@nil \@tempswatrue \else \@tempswafalse \fi}
626 \bxjs@purge@brace@elts

```

`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。

```
627 \@expandtwoargs\@removeelement
628   {papersize}\@classoptionslist\@classoptionslist
629 \@expandtwoargs\@removeelement
630   {10pt}\@classoptionslist\@classoptionslist
631 \@expandtwoargs\@removeelement
632   {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。

[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。

現在の（正規化前の）和文ドライバの値を `\bxjs@jadriver@given` に保存する。

```
633 \ifx\bxjs@jadriver\relax\else
634   \let\bxjs@jadriver@given\bxjs@jadriver
635 \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。

```
636 \let\bxjs@tmpb\jsEngine
637 \ifx j\bxjs@tmpb\ifjsWithpTeXng
638   \let\bxjs@tmpb=g
639 \fi\fi
640 \ifx j\bxjs@tmpb\ifjsWithupTeX
641   \let\bxjs@tmpb=u
642 \fi\fi
643 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
644   \let\bxjs@tmpb=n
645 \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。）

```
646 \ifx *\bxjs@engine@given
647   \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。

```

648 \ifx j\bxjs@engine@given
649   \g@addto@macro\@classoptionslist{,latex}
650 \else\ifx u\bxjs@engine@given
651   \g@addto@macro\@classoptionslist{,uplatex}
652 \fi\fi
653 \fi
654 \ifx\bxjs@engine@given\@undefined\else
655   \ifx\bxjs@engine@given\bxjs@tmpb\else
656     \ClassError\bxjs@clsname
657       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
658   \fi
659 \fi


```

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

```

660 \ifjsWithpTeXng
661   \g@addto@macro\@classoptionslist{,uplatex}
662 \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。

```

663 \@tempswatrue
664 \ifx \bxjs@driver@given\@undefined\else
665   \ifjsInPdfMode
666     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
667       \@tempswafalse
668     \fi
669   \else\ifx x\jsEngine
670     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
671       \@tempswafalse
672     \fi
673   \else
674     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
675       \@tempswafalse
676     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
677       \@tempswafalse
678     \fi\fi
679   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
680     \@tempswafalse
681   \fi\fi
682 \fi\fi
683 \fi
684 \if@tempswa\else
685   \ClassError\bxjs@clsname
686     {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
687 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

688 \ifjsInPdfMode \@tempswafalse
689 \else\ifx x\jsEngine \@tempswafalse

```

```

690 \else\ifjsWithpTeXng \@tempswafalse
691 \else \@tempswatrue
692 \fi\fi\fi
693 \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)。

```

694 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
695 \ClassWarningNoLine\bxjs@clsname
696 {No driver option is given}
697 \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。)

```

698 \ifbxjs@dvi@opt
699 \edef\bxjs@nxt{%
700 \let\noexpand\bxjs@driver@given
701 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
702 \noexpand\g@addto@macro\noexpand\@classoptionslist
703 {,\bxjs@driver@opt}%
704 }\bxjs@nxt
705 \fi
706 \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。

```

707 \ifjsWithpTeXng
708 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
709 \let\bxjs@platexng@nodrv\@undefined
710 \else\ifx t\bxjs@platexng@nodrv\else
711 \g@addto@macro\@classoptionslist{,dvipdfmx}
712 \fi\fi
713 \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。

```

714 \ifx\bxjs@driver@given\bxjs@driver@@none
715 \bxjs@papersizefalse
716 \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。

```

717 \def\bxjs@@minimal{minimal}
718 \ifx\bxjs@jadriver\relax
719 \ifx j\jsEngine
720 \def\bxjs@jadriver{standard}
721 \else
722 \let\bxjs@jadriver\bxjs@@minimal
723 \fi

```

724 \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。

```
725 \ifx\bxjs@jadriver@given\@undefined\else
726   \ifx\bxjs@engine@given\@undefined
727     \ClassError\bxjs@clsname
728     {An engine option must be explicitly given}%
729     {When you use a Japanese-driver you must specify a correct\MessageBreak
730       engine option.\MessageBreak \@ehc}
731 \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。)

```
732 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
733   \ifx\bxjs@magstyle\bxjs@magstyle@mag
734     \let\bxjs@magstyle\bxjs@magstyle@default
735     \ClassError\bxjs@clsname
736     {The engine does not support 'magstyle=usemag'}%
737     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
738       The default value 'nomag*' is used instead.\MessageBreak \@ehc}
739   \fi
740 \fi
```

オプション処理時に遅延させていた jbase の処理をここで実行する。

```
741 \bxjs@do@opt@jbase
```

\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。

```
742 \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。)

```
743 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
744   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
745     \catcode\@tempcnta\active
746     \advance\@tempcnta\@ne
747   \repeat
748 \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。

```
749 \ifbxjs@disguise@js
750 %<book|report>\def\bxjs@js@clsname{jsbook}
751 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
```

```

752 \namedef{ver@bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
753 \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。

```

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

```

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

```

755 \ifbxjs@oldfontcommands
756 \AtEndOfClass{\allowoldfontcommands}
757 \fi

```

■papersize スペシャルの出力 dvips の papersize スペシャルを出力します。これで dvips や新しい dviout で出力領域が自動設定できます。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。

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。

```

758 \newdimen\stockwidth \newdimen\stockheight
759 \begingroup\expandafter\expandafter\expandafter\endgroup
760 \expandafter\ifx\csname iftombow\expandafter\endcsname\csname iftrue\endcsname
761 % \newdimen\stockwidth \newdimen\stockheight
762 \setlength{\stockwidth}{\paperwidth}
763 \setlength{\stockheight}{\paperheight}
764 \advance \stockwidth 2in
765 \advance \stockheight 2in
766 \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。

```

767 %<slide>\def\n@baseline{13}%
768 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%

```



```
769 %<!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。

```
770 \ifx\bxjs@magstyle\bxjs@magstyle@mag
771   \jsc@magtrue
772 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
773   \jsc@mag@xrealtrue
774 \fi\fi
```

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

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

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

設定すべき `\mag` 値を (基底サイズ)/(10 pt) \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。

```
775 \ifx\bxjs@param@mag\relax
776   \@tempdima=\bxjs@param@basefontsize
777   \advance\@tempdima.001pt \multiply\@tempdima25
778   \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
779   \edef\bxjs@param@mag{\the\@tempcmta}
780 \else
781 % mag 値が直接指定された場合
782   \let\c@bxjs@cmta\@tempcmta
783   \setcounter{bxjs@cmta}{\bxjs@param@mag}
784   \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
785 % 有効な mag 値の範囲は 1--32768
786   \edef\bxjs@param@mag{\the\@tempcmta}
787   \advance\@tempcmta100000
788   \def\bxjs@tpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
789   \expandafter\bxjs@tpa\the\@tempcmta\@nil
790   \edef\bxjs@param@basefontsize{\the\@tempdima}
791 \fi
792 \@tempcmta\bxjs@param@mag \advance\@tempcmta100000
793 \def\bxjs@tpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
794 \expandafter\bxjs@tpa\the\@tempcmta\@nil
795 \edef\jsc@magscale{\strip@pt\@tempdima}
796 \let\jsBaseFontSize\bxjs@param@basefontsize
797 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
```

```

798 %\typeout{\string\jsEngine: \meaning\jsEngine}
799 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
800 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
801 %\typeout{\string\jsc@magscale: \jsc@magscale}
802 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
803 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}

```

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

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

```

804 \newdimen\jsc@mpt
805 \newdimen\jsc@mmm
806 \ifjsc@mag
807   \jsc@mpt=1\p@
808   \jsc@mmm=1mm
809   \catcode\?=9 % \p@? read as \p@
810 \else
811   \jsc@mpt=\jsc@magscale\p@
812   \jsc@mmm=\jsc@magscale mm
813   \catcode\?=11 \let\p@?\jsc@mpt
814 \fi
815 \chardef\bxjs@qmcc=\catcode\?\relax

```

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

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

```

816 \newdimen\jsZw
817 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
818 \ifbxjs@usezw
819   \providecommand*\zw{\jsZw}
820 \fi

```

`\zwspace` 全角幅の水平空き。

```

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

```

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

```

822 \ifjsc@mag@xreal
823   \RequirePackage{type1cm}
824   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

825 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax

```

```

826 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
827 \let\jsc@get@external@font\get@external@font
828 \def\get@external@font{%
829   \jsc@preadjust@extract@font
830   \jsc@get@external@font}
831 \def\jsc@fstrunc#1{%
832   \edef\jsc@tmpa{\strip@pt#1}%
833   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
834 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
835   \if#5*\else
836     \edef\jsc@tmpa{#1%
837       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
838     \fi}
839 \def\jsc@preadjust@extract@font{%
840   \let\jsc@req@size\f@size
841   \dimen@\f@size\p@ \jsc@invscale\dimen@\jsc@magscale
842   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
843   \let\jsc@ref@size\jsc@tmpa
844   \let\f@size\jsc@ref@size}
845 \def\execute@size@function#1{%
846   \let\jsc@cref@size\f@size
847   \let\f@size\jsc@req@size
848   \csname s@fct@#1\endcsname}
849 \let\jsc@DeclareErrorFont\DeclareErrorFont
850 \def\DeclareErrorFont#1#2#3#4#5{%
851   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
852   \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
853   \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
854 \def\gen@sfcnt{%
855   \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
856   \empty@sfcnt}
857 \def\genb@sfcnt{%
858   \edef\mandatory@arg{%
859     \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@}%
860   \empty@sfcnt}
861 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
862 \fi

```

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

```

\jsc@smallskip
\jsc@medskip 863 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 864 %\def\jsc@medskip{\vspace\jsc@medskipamount}
865 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 866 \newskip\jsc@smallskipamount
\jsc@bigskipamount

```

```

867 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
868 %\newskip\jsc@medskipamount
869 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
870 %\newskip\jsc@bigskipamount
871 %\jsc@bigskipamoun =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

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

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

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

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

```

872 % \ifpapersize
873 %   \setlength{\@tempdima}{\paperwidth}
874 %   \setlength{\@tempdimb}{\paperheight}
875 %   \iftombow
876 %     \advance \@tempdima 2truein
877 %     \advance \@tempdimb 2truein
878 %   \fi
879 %   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima space height \the\@tempdimb}}
880 % \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 `\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。
-

```

881 \chardef\bxjs@periodchar=`\ .
882 \bxjs@protected\def\bxjs@SE{%

```

```

883 \ifnum\spacefactor<\@m \spacefactor\@m
884 \else \spacefactor\sffcode\bxjs@periodchar
885 \fi}
886 \ifbxjs@fix@at@cmd
887 \def\@{\bxjs@SE{}}
888 \fi

```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

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

```

889 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
890 \expandafter\bxjs@tmpa\expandafter{%

```

```

891 \set@fontsize{#1}{#2}{#3}%
892 % 末尾にコードを追加
893 \expandafter\def\expandafter\size@update\expandafter{%
894     \size@update
895     \jsFontSizeChanged}%
896 }

```

`\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。

```

897 \newcommand*\jsFontSizeChanged{%
898     \jsZw=\f@size\p@
899     \jsZw=\jsScale \jsZw
900     \ifdim\parindent>\z@
901         \if@english \parindent=1em
902         \else        \parindent=1\jsZw
903     \fi
904     \fi\relax
905     \jsResetDimen}

```

`\jsResetDimen` ユーザ定義用のフック。

```

906 \newcommand*\jsResetDimen{}

```

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

```

907 \ifjsc@mag
908     \let\jsc@setfontsize\@setfontsize
909 \else
910     \def\jsc@setfontsize#1#2#3{%
911         \@setfontsize#1{#2\jsc@mpt}{#3\jsc@mpt}}
912 % microtype 対策
913 \ifjsWithTeX\if j\jsEngine\else
914     \def\jsc@setfontsize#1#2#3{%
915         \edef\bxjs@sfs@next{%
916             \unexpanded{\@setfontsize#1}%
917             {\the\dimexpr#2\jsc@mpt\relax}{\the\dimexpr#3\jsc@mpt\relax}%
918         }\bxjs@sfs@next}
919     \fi\fi
920 \fi

```

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

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

```

921 \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。

```

922 \newif\ifnarrowbaselines
923 \if@english
924   \narrowbaselinestrue
925 \fi
926 \def\narrowbaselines{%
927   \narrowbaselinestrue
928   \skip0=\abovedisplayskip
929   \skip2=\abovedisplayshortskip
930   \skip4=\belowdisplayskip
931   \skip6=\belowdisplayshortskip
932 % 一時的に警告を無効化する
933   \let\bxjs@ltx@nomath\@nomath
934   \let\@nomath\@gobble
935   \@currsize\selectfont
936   \let\@nomath\bxjs@ltx@nomath
937   \abovedisplayskip=\skip0
938   \abovedisplayshortskip=\skip2
939   \belowdisplayskip=\skip4
940   \belowdisplayshortskip=\skip6\relax}
941 \def\widebaselines{\narrowbaselinesfalse\@currsize\selectfont}

```

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

```

942 \def\bxjs@ifnarrowbaselines{%
943   \ifnarrowbaselines\expandafter\@firstoftwo
944   \else \expandafter\@secondoftwo
945   \fi
946 }

```

`\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。

```
947 \renewcommand{\normalsize}{%
948   \bxjs@if@narrowbaselines{%
949     \jsc@setfontsize\normalsize\@xpt\@xipt
950   }{%else
951     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
952   }%
```

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。

```
953 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
954 \abovedisplayshortskip \z@ \@plus3\p@?
955 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
956 \belowdisplayshortskip \belowdisplayskip
```

最後に、リスト環境のトップレベルのパラメータ `\@listI` を、`\@listi` にコピーしておきます。`\@listI` の設定は後で出てきます。

```
957 \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。

```
958 \normalsize
```

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

`\Cdp` 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。

`\Cvs`

`\Chs`

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

```
959 \ifx\Cht\undefined \newdimen\Cht \fi
960 \ifx\Cdp\undefined \newdimen\Cdp \fi
961 \ifx\Cwd\undefined \newdimen\Cwd \fi
962 \ifx\Cvs\undefined \newdimen\Cvs \fi
963 \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。

```

964 \setlength\Cht{0.88\jsZw}
965 \setlength\Cdp{0.12\jsZw}
966 \setlength\Cwd{1\jsZw}
967 \setlength\Cvs{\baselineskip}
968 \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。

```

969 \newcommand{\small}{%
970   \bxjs@if@narrowbaselines{%
971     <!kiyou> \jsc@setfontsize\small\@ixpt{11}%
972     <kiyou> \jsc@setfontsize\small{8.8888}{11}%
973   }{%else
974     <!kiyou> \jsc@setfontsize\small\@ixpt{13}%
975     <kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
976   }%
977   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
978   \abovedisplayshortskip \z@ \@plus3\p@?
979   \belowdisplayskip \abovedisplayskip
980   \belowdisplayshortskip \belowdisplayskip
981   \def\@listi{\leftmargin\leftmargini
982             \topsep \z@
983             \parsep \z@
984             \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。

```

985 \newcommand{\footnotesize}{%
986   \bxjs@if@narrowbaselines{%
987     <!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
988     <kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
989   }{%else
990     <!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
991     <kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
992   }%
993   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
994   \abovedisplayshortskip \z@ \@plus2\p@?
995   \belowdisplayskip \abovedisplayskip
996   \belowdisplayshortskip \belowdisplayskip

```

```

997 \def\@listi{\leftmargin\leftmargini
998 \topsep \z@
999 \parsep \z@
1000 \itemsep \parsep}}

```

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

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 1001 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
1002 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 1003 \if@twocolumn
\HUGE 1004 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
1005 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
1006 \else
1007 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
1008 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
1009 \fi
1010 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
1011 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
1012 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
1013 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
1014 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
1015 \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。

```

1016 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}

```

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

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

```

1017 % \newcommand{\headfont}{\bfseries}
1018 \newcommand{\headfont}{\sffamily}
1019 % \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。

```
1020 %<!kiyou>\setlength\columnsep{2\Cwd}  
1021 %<kiyou>\setlength\columnsep{28truebp}  
1022 \setlength\columnseprule{0\jsc@empt}
```

■段落

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

```
\lineskiplimit 1023 \setlength\lineskip{1\jsc@empt}  
1024 \setlength\normallineskip{1\jsc@empt}  
\normallineskiplimit 1025 \setlength\lineskiplimit{1\jsc@empt}  
1026 \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。

```
1027 \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。

```
1028 \setlength\parskip{0\jsc@empt}  
1029 \if@slide  
1030 \setlength\parindent{0\p@}  
1031 \else  
1032 \setlength\parindent{1\Cwd}  
1033 \fi
```

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

```
\@highpenalty 1034 \@lowpenalty 51  
1035 \@medpenalty 151  
1036 \@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。

```
1037 % \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。

```
1038 % \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

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
1039 \ifjsc@mag
1040 \mag=\bxjs@param@mag
1041 \fi
1042 \setlength{\topskip}{10\p0?}
```

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

```
1043 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
1044 \def\bxjs@read@b{%
1045   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
1046   \else \expandafter\bxjs@read@d \fi}
1047 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
1048 \def\bxjs@read@d#1\@nil{}
1049 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
1050 \edef\bxjs@layout@paper{%
1051   \ifjsc@mag truedimen,\fi
1052   \if@landscape landscape,\fi
1053   \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。

```
1054 %<*article|report>
1055 \def\bxjs@layout{%
1056   headheight=\topskip,footskip=0.03367\paperheight,%
1057   headsep=\footskip-\topskip,includeheadfoot,%
1058   hscale=0.76,hmarginratio=1:1,%
1059   vscale=0.83,vmarginratio=1:1,%
1060 }
1061 %</article|report>
1062 %<*book>
1063 \ifbxjs@layout@buggyhmargin %---
1064 % アレ
1065 \def\bxjs@layout{%
```

```

1066 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1067 hmargin=36\jsc@mmm,hmarginratio=1:1,%
1068 vscale=0.83,vmarginratio=1:1,%
1069 }
1070 \else %---
1071 % 非アレ
1072 \def\bxjs@layout{%
1073 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
1074 hmargin=18\jsc@mmm,%
1075 vscale=0.83,vmarginratio=1:1,%
1076 }
1077 \fi %---
1078 %</book>
1079 %<*slide>
1080 \def\bxjs@layout{%
1081 noheadfoot,%
1082 hscale=0.9,hmarginratio=1:1,%
1083 vscale=0.944,vmarginratio=1:1,%
1084 }
1085 %</slide>

```

\fullwidth 〔寸法レジスタ〕ヘッダ・フッタ領域の横幅。

```

1086 \newdimen\fullwidth

```

\jsTextWidthLimit 〔実数値マクロ〕bxjsbook における、\textwidth の上限の全角単位での値。

```

1087 %<*book>
1088 \newcommand\jsTextWidthLimit{40}
1089 \ifx\bxjs@textwidth@limit\undefined\else
1090 \let\c@bxjs@canta\@tempcanta
1091 \setcounter{bxjs@canta}{\bxjs@textwidth@limit}
1092 \long\edef\jsTextWidthLimit{\the\@tempcanta}
1093 \fi
1094 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

1095 \def\bxjs@postproc@layout{%
1096 % ドライバ再設定
1097 \ifx\bxjs@geometry@driver\relax\else
1098 \let\Gm@driver\bxjs@geometry@driver
1099 \fi
1100 % textwidth 調整
1101 \ifbxjs@whole@zw@lines
1102 \@tempdimb=\textwidth
1103 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1104 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1105 \advance\@tempdimb-\textwidth
1106 \advance\oddsidemargin 0.5\@tempdimb
1107 \advance\evensidemargin 0.5\@tempdimb
1108 \fi

```

```

1109 \fullwidth=\textwidth
1110 %<*book>
1111 \ifdim\textwidth>\jsTextWidthLimit\Cwd
1112 \textwidth=\jsTextWidthLimit\Cwd
1113 \addtolength\evensidemargin{\fullwidth-\textwidth}
1114 \fi
1115 %</book>
1116 % textheight 調整
1117 \@tempdimb=\textheight
1118 \advance\textheight-\topskip
1119 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1120 \advance\textheight\topskip
1121 \advance\@tempdimb-\textheight
1122 \advance\topmargin0.5\@tempdimb
1123 % headheight 調整
1124 \@tempdima=\topskip
1125 \advance\headheight\@tempdima
1126 \advance\topmargin-\@tempdima
1127 % marginpar 設定
1128 \setlength\marginparsep{\columnsep}
1129 \setlength\marginparpush{\baselineskip}
1130 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1131 -\textwidth-10\jsc@mmm-\marginparsep}
1132 \ifbxjs@whole@zw@lines
1133 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1134 \fi
1135 % 連動する変数
1136 \maxdepth=.5\topskip
1137 \stockwidth=\paperwidth
1138 \stockheight=\paperheight
1139 }

```

\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。

```

1140 \edef\jsGeometryOptions{%
1141 \bxjs@layout@paper,\bxjs@layout}

```

■geometry パッケージ読込

geoemtry オプションの値に応じて分岐する。

まずは geometry=class の場合。

```

1142 \ifx\bxjs@geometry\bxjs@geometry@class

```

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。

```

1143 \@onlypreamble\bxjs@revert
1144 \let\bxjs@revert\@empty
1145 \edef\bxjs@tmpa{\the\ht\strutbox}
1146 \ht\strutbox=10\p@?
1147 \g@addto@macro\bxjs@revert{\ht\strutbox=\bxjs@tmpa\relax}

```

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。

```

1148 \ifbxjs@papersize
1149   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1150     \PassOptionsToPackage{dvipdfm}{geometry}
1151   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1152     \PassOptionsToPackage{dvipdfm}{geometry}
1153   \fi\fi
1154   \let\bxPapersizeSpecialDone=t
1155 \else
1156   \PassOptionsToPackage{driver=none}{geometry}
1157 \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。

```

1158 \edef\bxjs@nxt{%
1159   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1160 \AtBeginDocument{\bxjs@pre@geometry@hook}
1161 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1162 \bxjs@nxt \bxjs@revert
1163 \AtBeginDocument{\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。

```

1164 \let\bxjs@geometry@driver\Gm@driver
1165 \bxjs@postproc@layout

```

\bxjs@pre@geometry@hook 1.2 版より、geometry の 4.x 版の使用は非推奨とする。

※将来サポートを廃止する予定。

```

1166 \@onlypreamble\bxjs@pre@geometry@hook
1167 \def\bxjs@pre@geometry@hook{%
1168   \@ifpackageloaded{geometry}{%
1169     \@ifpackagelater{geometry}{2010/02/12}{-}{%else
1170       \PackageWarningNoLine\bxjs@clsname
1171       {The 'geometry' package installed\MessageBreak
1172        is too old (< v5.0)}}%
1173     \if x\jsEngine \ifnum\mag=\@m\else
1174       \def\bxjs@Gm@driver{pdftex}
1175       \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi

```

```

1176     \ifx\pdfvorigin\undefined \newdimen\pdfvorigin \fi
1177     \fi\fi
1178   }%
1179   \ifjsWithpTeXng
1180     \ifx\Gm@driver\empty
1181       \def\Gm@driver{pdftex}%
1182     \fi
1183   \fi
1184 }{}

```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

- `\setpagelayout{<text>}` : 現在の geometry の設定の一部を修正する。
- `\setpagelayout*{<text>}` : 用紙以外の設定をリセットして、改めて設定を行う。

どちらも設定の後で後処理 `\bxjs@postproc@layout` を実行する。

```

1185 \def\setpagelayout{\@ifstar
1186   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1187 \def\bxjs@modify@layout#1{%
1188   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1189   \bxjs@nxt\bxjs@postproc@layout}
1190 \def\bxjs@reset@layout#1{%
1191   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1192   \bxjs@nxt\bxjs@postproc@layout}

```

`geometry=class` の場合の処理はここで終わり。

次に、`geometry=user` の場合の処理。

```

1193 \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。

```

1194 \AtBeginDocument{\bxjs@check@page@layout}
1195 \@onlypreamble\bxjs@check@page@layout
1196 \def\bxjs@check@page@layout{%
1197   \ifdim\textwidth=.5\maxdimen
1198     \ClassError\bxjs@clsname
1199     {Page layout is not properly set}%
1200     {\@ehd}
1201   \fi}
1202 \def\jsUseMinimalPageLayout{%
1203   \setlength{\textwidth}{6.5in}%
1204   \setlength{\textheight}{8in}}

```

`\setpagelayout` はとりあえず無効にしておく。

```

1205 \let\bxjs@geometry@driver\relax
1206 \def\setpagelayout{\@ifstar
1207   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}

```



```

1208 \def\bxjs@pagelayout@a#1{%
1209   \ClassError\bxjs@clsname
1210   {Command '\string\setpagelayout' is not supported,\MessageBreak
1211     because 'geometry' value is not 'class'}\@eha}

```

geometry=user の場合の処理はここで終わり。

```
1212 \fi\fi
```

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。

```
<*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。

```

1213 \setlength\topskip{1.38zw}%% from 10\jsc@mpt (2016-08-17)
1214 \if@slide
1215   \setlength\headheight{0\jsc@mpt}
1216 \else
1217   \setlength\headheight{20\jsc@mpt}%% from 2\topskip (2016-08-17); from \topskip (2003-
1218   06-26)
1218 \fi

```

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

```

1219 %<*article|kiyou>
1220 \if@slide
1221   \setlength\footskip{0pt}
1222 \else
1223   \setlength\footskip{0.03367\paperheight}

```

```

1224 \ifdim\footskip<\baselineskip
1225 \setlength\footskip{\baselineskip}
1226 \fi
1227 \fi
1228 %</article|kiyou>
1229 %<jspf>\setlength\footskip{9\jsc@mmm}
1230 %<*book>
1231 \if@report
1232 \setlength\footskip{0.03367\paperheight}
1233 \ifdim\footskip<\baselineskip
1234 \setlength\footskip{\baselineskip}
1235 \fi
1236 \else
1237 \setlength\footskip{0pt}
1238 \fi
1239 %</book>
1240 %<*report>
1241 \setlength\footskip{0.03367\paperheight}
1242 \ifdim\footskip<\baselineskip
1243 \setlength\footskip{\baselineskip}
1244 \fi
1245 %</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)。

```

1246 %<*article>
1247 \if@slide
1248 \setlength\headsep{0\jsc@mpt}
1249 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1250 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1251 \else
1252 \setlength\headsep{\footskip}
1253 \addtolength\headsep{-\topskip}
1254 \fi
1255 %</article>
1256 %<*book>
1257 \if@report
1258 \setlength\headsep{\footskip}
1259 \addtolength\headsep{-\topskip}
1260 \else
1261 \setlength\headsep{6\jsc@mmm}
1262 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1263 \addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
1264 \fi
1265 %</book>

```

```

1266 %<*report>
1267 \setlength\headsep{\footskip}
1268 \addtolength\headsep{-\topskip}
1269 %</report>
1270 %<*jspf>
1271 \setlength\headsep{9\jsc@mmm}
1272 \addtolength\headsep{-\topskip}
1273 %</jspf>
1274 %<*kiyou>
1275 \setlength\headheight{0\jsc@empt}
1276 \setlength\headsep{0\jsc@empt}
1277 \addtolength\headsep{-\topskip}%% added (2016-10-08)
1278 \addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
1279 %</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。

```

1280 \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。

```

1281 \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。

```

1282 %<*article>
1283 \if@slide
1284   \setlength\fullwidth{0.9\paperwidth}
1285 \else
1286   \setlength\fullwidth{0.76\paperwidth}
1287 \fi
1288 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1289 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1290 \setlength\textwidth{\fullwidth}
1291 %</article>

```

```

1292 %<*book>
1293 \if@report
1294   \setlength\fullwidth{0.76\paperwidth}
1295 \else
1296   \setlength\fullwidth{\paperwidth}
1297   \addtolength\fullwidth{-36\jsc@mmm}
1298 \fi
1299 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1300 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1301 \setlength\textwidth{\fullwidth}
1302 \if@report \else
1303   \if@twocolumn \else
1304     \ifdim \fullwidth>40zw
1305       \setlength\textwidth{40zw}
1306     \fi
1307   \fi
1308 \fi
1309 %</book>
1310 %<*report>
1311 \setlength\fullwidth{0.76\paperwidth}
1312 \if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
1313 \divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
1314 \setlength\textwidth{\fullwidth}
1315 %</report>
1316 %<*jspf>
1317 \setlength\fullwidth{50zw}
1318 \addtolength\fullwidth{8\jsc@mmm}
1319 \setlength\textwidth{\fullwidth}
1320 %</jspf>
1321 %<*kiyou>
1322 \setlength\fullwidth{48zw}
1323 \addtolength\fullwidth{\columnsep}
1324 \setlength\textwidth{\fullwidth}
1325 %</kiyou>

```

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

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

某学会誌スタイルでは 44 行にします。

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

[2016-08-26] `\topskip` を 10pt から 1.38zw に増やしましたので、その分 `\textheight`

を増やします (2016-08-17 での修正漏れ)。

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

```
1326 %<*article|book|report>
1327 \if@slide
1328   \setlength{\textheight}{0.95\paperheight}
1329 \else
1330   \setlength{\textheight}{0.83\paperheight}
1331 \fi
1332 \addtolength{\textheight}{-10\jsc@mpt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1333 \addtolength{\textheight}{-\headsep}
1334 \addtolength{\textheight}{-\footskip}
1335 \addtolength{\textheight}{-\topskip}
1336 \divide\textheight\baselineskip
1337 \multiply\textheight\baselineskip
1338 %</article|book|report>
1339 %<jspf>\setlength{\textheight}{51\baselineskip}
1340 %<kiyou>\setlength{\textheight}{47\baselineskip}
1341 \addtolength{\textheight}{\topskip}
1342 \addtolength{\textheight}{0.1\jsc@mpt}
1343 %<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。

```
1344 \def\flushbottom{%
1345   \def\@textbottom{\vskip \z@ \@plus.1\jsc@mpt}%
1346   \let\@texttop\relax}
```

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

```
1347 \setlength\marginparsep{\columnsep}
1348 \setlength\marginparpush{\baselineskip}
```

`\oddsidemargin` それぞれ奇数ページ、偶数ページの左マージンから 1 インチ引いた値です。片面印刷では `\evensidemargin` `\oddsidemargin` が使われます。 \TeX は上・左マージンに `1truein` を挿入しますが、トンボ関係のオプションが指定されると $\text{p}\text{\LaTeX 2}_\epsilon$ (`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。

```
1349 \setlength{\oddsidemargin}{\paperwidth}
1350 \addtolength{\oddsidemargin}{-\fullwidth}
1351 \setlength{\oddsidemargin}{.5\oddsidemargin}
1352 \iftombow
```

```

1353 \addtolength{\oddsidemargin}{-1in}
1354 \else
1355 \addtolength{\oddsidemargin}{-\inv@mag in}
1356 \fi
1357 \setlength{\evensidemargin}{\oddsidemargin}
1358 \if@mparswitch
1359 \addtolength{\evensidemargin}{\fullwidth}
1360 \addtolength{\evensidemargin}{-\textwidth}
1361 \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。

```

1362 \setlength\marginparwidth{\paperwidth}
1363 \addtolength\marginparwidth{-\oddsidemargin}
1364 \addtolength\marginparwidth{-\inv@mag in}
1365 \addtolength\marginparwidth{-\textwidth}
1366 \addtolength\marginparwidth{-10\jsc@mmm}
1367 \addtolength\marginparwidth{-\marginparsep}
1368 \@tempdima=1zw
1369 \divide\marginparwidth\@tempdima
1370 \multiply\marginparwidth\@tempdima

```

`\topmargin` 上マージン (紙の上端とヘッダ上端の距離) から 1 インチ引いた値です。

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

[2016-08-17] `\topskip` を 10pt から 1.38zw に直しましたが、`\topmargin` は従来の値から変わらないように調節しました。…のつもりでしたが、`\textheight` を増やし忘れていたので変わってしまっていました (2016-08-26 修正済み)。

```

1371 \setlength\topmargin{\paperheight}
1372 \addtolength\topmargin{-\textheight}
1373 \if@slide
1374 \addtolength\topmargin{-\headheight}
1375 \else
1376 \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -
    \headheight (2003-06-26)
1377 \fi
1378 \addtolength\topmargin{-\headsep}
1379 \addtolength\topmargin{-\footskip}
1380 \setlength\topmargin{0.5\topmargin}
1381 %<kiyou>\setlength\topmargin{81truebp}
1382 \iftombow
1383 \addtolength\topmargin{-1in}
1384 \else
1385 \addtolength\topmargin{-\inv@mag in}
1386 \fi

```

</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。

```
1387 \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。

```
1388 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

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

[2003-08-23] ちょっと増やしました。

```
1389 \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。

```
1390 \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。

```
1391 \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。

```
1392 \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。

```
1393 \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。

```
1394 \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。

```
1395 \renewcommand{\floatpagefraction}{.8}
```

`\cdbltopnumber` 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。

```
1396 \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。

```
1397 \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。

```
1398 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1399 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1400 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1401 \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。

```
1402 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1403 \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。

```
1404 \setlength\@fptop{0\p@? \@plus 1fil}
1405 \setlength\@fpsep{8\p@? \@plus 2fil}
1406 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
1407 \setlength\@dblfpsep{0\p@? \@plus 1fil}
1408 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1409 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ（日本語 TeX 開発コミュニティ版のみ）

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

`\pltx@cleartoleftpage`

`\pltx@cleartooddpage`

`\pltx@cleartoevenpage`

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

となっています。

```
1410 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1411 % \ifodd\c@page
1412 % \iftdir
1413 % \hbox{}\thispagestyle{empty}\newpage
1414 % \if@twocolumn\hbox{}\newpage\fi
1415 % \fi
1416 % \else
1417 % \ifydir
1418 % \hbox{}\thispagestyle{empty}\newpage
1419 % \if@twocolumn\hbox{}\newpage\fi
1420 % \fi
1421 % \fi\fi}
1422 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1423 % \ifodd\c@page
1424 % \ifydir
1425 % \hbox{}\thispagestyle{empty}\newpage
1426 % \if@twocolumn\hbox{}\newpage\fi
1427 % \fi
1428 % \else
1429 % \iftdir
1430 % \hbox{}\thispagestyle{empty}\newpage
1431 % \if@twocolumn\hbox{}\newpage\fi
1432 % \fi
1433 % \fi\fi}
1434 \def\pltx@cleartooddpage{\clearpage\if@twoside
1435 \ifodd\c@page\else
1436 \hbox{}\thispagestyle{empty}\newpage
1437 \if@twocolumn\hbox{}\newpage\fi
1438 \fi\fi}
1439 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1440 \ifodd\c@page
1441 \hbox{}\thispagestyle{empty}\newpage
1442 \if@twocolumn\hbox{}\newpage\fi
1443 \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。

```
1444 \let\pltx@cleartorightpage\pltx@cleartooddpage
1445 \let\pltx@cleartoleftpage\pltx@cleartoevenpage
```

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

```
1446 %<*book|report>
1447 \if@openleft
1448 \let\cleardoublepage\pltx@cleartoleftpage
1449 \else\if@openright
```

```

1450 \let\cleardoublepage\pltx@cleartorightpage
1451 \fi\fi
1452 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

`\@evenhead` `\@oddhead`, `\@oddfoot`, `\@evenhead`, `\@evenfoot` は偶数・奇数ページの柱 (ヘッダ, フッタ) を出力する命令です。これらは `\fullwidth` 幅の `\hbox` の中で呼び出されます。
`\@evenfoot` `\ps@...` の中で定義しておきます。
`\@oddfoot` 柱の内容は、`\chapter` が呼び出す `\chaptermark{何々}`, `\section` が呼び出す `\sectionmark{何々}` で設定します。柱を扱う命令には次のものがあります。

```

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

```

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

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

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

```

1453 % \def\ps@empty{%
1454 %   \let\@mkboth\@gobbletwo
1455 %   \let\@oddhead\@empty
1456 %   \let\@oddfoot\@empty
1457 %   \let\@evenhead\@empty
1458 %   \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。

```

1459 \def\ps@plainfoot{%

```

```

1460 \let\@mkboth\@gobbletwo
1461 \let\@oddhead\@empty
1462 \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1463 \let\@evenhead\@empty
1464 \let\@evenfoot\@oddfoot}
1465 \def\ps@plainhead{%
1466 \let\@mkboth\@gobbletwo
1467 \let\@oddfoot\@empty
1468 \let\@evenfoot\@empty
1469 \def\@evenhead{%
1470 \if@mparswitch \hss \fi
1471 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1472 \if@mparswitch\else \hss \fi}%
1473 \def\@oddhead{%
1474 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1475 %<book>\let\ps@plain\ps@plainhead
1476 %<!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。

```

1477 %<*article|slide>
1478 \if@twoside
1479 \def\ps@headings{%
1480 \let\@oddfoot\@empty
1481 \let\@evenfoot\@empty
1482 \def\@evenhead{\if@mparswitch \hss \fi
1483 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1484 \if@mparswitch\else \hss \fi}%
1485 \def\@oddhead{%
1486 \underline{%
1487 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1488 \let\@mkboth\markboth
1489 \def\sectionmark##1{\markboth{%
1490 \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1491 ##1}{}}%
1492 \def\subsectionmark##1{\markright{%
1493 \ifnum \c@secnumdepth >\@ne \bxjs@label@sect{subsection}\hskip1\jsZw\fi
1494 ##1}}%
1495 }
1496 \else % if not twoside
1497 \def\ps@headings{%
1498 \let\@oddfoot\@empty
1499 \def\@oddhead{%
1500 \underline{%
1501 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1502 \let\@mkboth\markboth
1503 \def\sectionmark##1{\markright{%

```

```

1504         \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1505         ##1}}}}
1506 \fi
1507 %</article|slide>

```

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

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

```

1508 %<*book|report>
1509 \def\bxjs@maybe@autoxspacing{%
1510   \ifx\autoxspacing\@undefined\else \autoxspacing \fi}
1511 \newif\if@omit@number
1512 \def\ps@headings{%
1513   \let\@oddfoot\@empty
1514   \let\@evenfoot\@empty
1515   \def\@evenhead{%
1516     \if@mparswitch \hss \fi
1517     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1518       \textbf{\thepage}\hfil\leftmark}}}%
1519     \if@mparswitch\else \hss \fi}%
1520   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1521     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1522   \let\@mkboth\markboth
1523   \def\chaptermark##1{\markboth{%
1524     \ifnum \c@secnumdepth >\m@ne
1525       \if@mainmatter
1526         \if@omit@number\else
1527           \@chapapp\thechapter\@chappos\hskip1\jsZw
1528         \fi
1529       \fi
1530     \fi
1531     ##1-}}}%
1532   \def\sectionmark##1{\markright{%
1533     \ifnum \c@secnumdepth >\z@ \bxjs@label@sect{section}\hskip1\jsZw\fi
1534     ##1}}}%
1535 %</book|report>

```

最後は学会誌の場合です。

```

1536 %<*jspf>
1537 \def\ps@headings{%
1538   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1539   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1540   \def\@oddhead{\normalfont\hfil \@title \hfil}
1541   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1542 %</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。

```
1543 \def\ps@myheadings{%
1544   \let\@oddfoot\@empty\let\@evenfoot\@empty
1545   \def\@evenhead{%
1546     \if@mparswitch \hss \fi%
1547     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1548     \if@mparswitch\else \hss \fi}%
1549   \def\@oddhead{%
1550     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1551   \let\mkboth\@gobbletwo
1552 %<book|report> \let\chaptermark\@gobble
1553   \let\sectionmark\@gobble
1554 %<!book&!report> \let\subsectionmark\@gobble
1555 }
```

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 1556 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1557 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1558 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1559 % \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。

```
1560 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1561 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1562 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1563 \AtBeginDocument{\bxjs@decl@subtitle}
1564 \def\bxjs@decl@subtitle{%
1565   \global\let\bxjs@decl@subtitle\relax
1566   \ifx\subtitle\@undefined
1567     \global\let\subtitle\jsSubtitle
1568   \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。

```
1569 \def\bxjs@annihilate@subtitle{%
1570   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1571   \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 1572 %<*jspf>
1573 \newcommand*\etitle[1]{\gdef\etitle{#1}}
\keywords 1574 \newcommand*\eauthor[1]{\gdef\eauthor{#1}}
1575 \newcommand*\keywords[1]{\gdef\keywords{#1}}
1576 \newcommand*\email[1]{\gdef\authors@mail{#1}}
1577 \newcommand*\AuthorsEmail[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1578 %</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。

```
1579 \def\plainifnotempty{%
1580   \ifx \@oddhead \@empty
1581     \ifx \@oddfoot \@empty
1582       \else
1583         \thispagestyle{plainfoot}%
1584       \fi
1585     \else
1586       \thispagestyle{plainhead}%
1587     \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。

```
1588 %<*article|book|report|slide>
1589 \if@titlepage
1590   \newcommand*\maketitle{%
1591     \begin{titlepage}%
1592       \let\footnotesize\small
1593       \let\footnoterule\relax
1594       \let\footnote\thanks
1595       \null\vfil
1596       \if@slide
1597         {\footnotesize \@date}%
```

```

1598     \begin{center}
1599         \mbox{} \\\[1\jsZw]
1600         \large
1601         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1602         \jsc@smallskip
1603         \@title
1604         \ifx\bxjs@subtitle\@undefined\else
1605             \par\vskip\z@
1606             {\small \bxjs@subtitle\par}
1607         \fi
1608         \jsc@smallskip
1609         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1610         \vfill
1611         {\small \@author}%
1612     \end{center}
1613 \else
1614 \vskip 60\p@?
1615 \begin{center}%
1616     {\LARGE \@title \par}%
1617     \ifx\bxjs@subtitle\@undefined\else
1618         \vskip5\p@?
1619         {\normalsize \bxjs@subtitle\par}
1620     \fi
1621     \vskip 3em%
1622     {\large
1623         \lineskip .75em
1624         \begin{tabular}[t]{c}%
1625             \@author
1626         \end{tabular}\par}%
1627     \vskip 1.5em
1628     {\large \@date \par}%
1629 \end{center}%
1630 \fi
1631 \par
1632 \@thanks\vfil\null
1633 \end{titlepage}%
1634 \setcounter{footnote}{0}%
1635 \global\let\thanks\relax
1636 \global\let\maketitle\relax
1637 \global\let\@thanks\@empty
1638 \global\let\@author\@empty
1639 \global\let\@date\@empty
1640 \global\let\@title\@empty
1641 \global\let\title\relax
1642 \global\let\author\relax
1643 \global\let\date\relax
1644 \global\let\and\relax
1645 \bxjs@annihilate@subtitle
1646 }%

```

```

1647 \else
1648   \newcommand{\maketitle}{\par
1649     \begin{group}
1650       \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1651       \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1652       \long\def\@makefnmark##1{\advance\leftskip 3\jsZw
1653         \parindent 1\jsZw\noindent
1654         \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1655       \if@twocolumn
1656         \ifnum \col@number=\@one
1657           \@maketitle
1658         \else
1659           \twocolumn[\@maketitle]%
1660         \fi
1661       \else
1662         \newpage
1663         \global\@topnum\z@ % Prevents figures from going at top of page.
1664         \@maketitle
1665       \fi
1666       \plainifnotempty
1667       \@thanks
1668     \end{group}
1669     \setcounter{footnote}{0}%
1670     \global\let\thanks\relax
1671     \global\let\maketitle\relax
1672     \global\let\@thanks\@empty
1673     \global\let\@author\@empty
1674     \global\let\@date\@empty
1675     \global\let\@title\@empty
1676     \global\let\title\relax
1677     \global\let\author\relax
1678     \global\let\date\relax
1679     \global\let\and\relax
1680     \bxjs@annihilate@subtitle
1681   }

```

`\@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。

```

1682 \def\@maketitle{%
1683   \newpage\null
1684   \vskip 2em
1685   \begin{center}%
1686     \let\footnote\thanks
1687     {\LARGE \@title \par}%
1688     \ifx\bxjs@subtitle\@undefined\else
1689       \vskip3\p@?
1690       {\normalsize \bxjs@subtitle\par}
1691     \fi
1692     \vskip 1.5em
1693     {\large

```



```

1694         \lineskip .5em
1695         \begin{tabular}[t]{c}%
1696             \@author
1697         \end{tabular}\par}%
1698         \vskip 1em
1699         {\large \@date}%
1700     \end{center}%
1701     \par\vskip 1.5em
1702 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1703 }
1704 \fi
1705 %</article|book|report|slide>
1706 %<*jspf>
1707 \newcommand\maketitle{\par
1708     \begingroup
1709         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1710         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1711         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1712             \parindent 1\jsZw\noindent
1713             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1714         \twocolumn[\@maketitle]%
1715         \plainifnotempty
1716         \@thanks
1717     \endgroup
1718     \setcounter{footnote}{0}%
1719     \global\let\thanks\relax
1720     \global\let\maketitle\relax
1721     \global\let\@thanks\@empty
1722     \global\let\@author\@empty
1723     \global\let\@date\@empty
1724 % \global\let\@title\@empty % \@title は柱に使う
1725     \global\let\title\relax
1726     \global\let\author\relax
1727     \global\let\date\relax
1728     \global\let\and\relax
1729     \ifx\authors@mail\@undefined\else{%
1730         \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1731         \footnotetext[0]{\itshape\authors@mail}%
1732     }\fi
1733     \global\let\authors@mail\@undefined}
1734 \def\@maketitle{%
1735     \newpage\null
1736     \vskip 6em % used to be 2em
1737     \begin{center}
1738         \let\footnote\thanks
1739         \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1740         \lineskip .5em
1741         \ifx\@author\@undefined\else
1742             \vskip 1em

```

```

1743     \begin{tabular}[t]{c}%
1744         \@author
1745     \end{tabular}\par
1746 \fi
1747 \ifx\@etitle\@undefined\else
1748     \vskip 1em
1749     {\large \@etitle \par}%
1750 \fi
1751 \ifx\@eauthor\@undefined\else
1752     \vskip 1em
1753     \begin{tabular}[t]{c}%
1754         \@eauthor
1755     \end{tabular}\par
1756 \fi
1757 \vskip 1em
1758 \@date
1759 \end{center}
1760 \vskip 1.5em
1761 \centerline{\box\@abstractbox}
1762 \ifx\@keywords\@undefined\else
1763     \vskip 1.5em
1764     \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\small\@keywords}}
1765 \fi
1766 \vskip 1.5em}
1767 %</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。

```

1768 \def\bxjs@label@sect#1{%
1769     \expandafter\ifx\csname label#1\endcsname\relax
1770         \csname the#1\endcsname
1771     \else \csname label#1\endcsname
1772     \fi}
1773 \def\@seccntformat#1{\bxjs@label@sect{#1}\quad}

```

`\@secapp` 節番号の接頭辞。

`\@secpos` 節番号の接尾辞。

```

1774 \ifnum\bxjs@label@section=\bxjs@label@section@compat\else
1775 \def\@secapp{\presectionname}
1776 \def\@secpos{\postsectionname}
1777 \fi

```

`\labelsection` 節番号の出力書式。

```
1778 \ifnum\bxjs@label@section=\bxjs@label@section@modern
1779 \def\labelsection{\@secapp\thesection\@secpos}
1780 \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。

```
1781 \def\@startsection#1#2#3#4#5#6{%
1782   \if@noskipsec \leavevmode \fi
1783   \par
1784 % 見出し上の空きを \@tempskipa にセットする
1785   \@tempskipa #4\relax
1786 % \@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
1787   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1788 % 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
1789   \ifdim \@tempskipa <\z@
1790     \@tempskipa -\@tempskipa \@afterindentfalse
1791   \fi
1792   \if@nobreak
1793 %   \everypar{\everyparhook}% これは間違い
1794   \everypar{}\%
```

```

1795 \else
1796 \addpenalty\@secpenalty
1797 % 次の行は削除
1798 % \addvspace\@tempskipa
1799 % 次の \noindent まで追加
1800 \ifdim \@tempskipa >\z@
1801 \if@slide\else
1802 \null
1803 \vspace*{-\baselineskip}%
1804 \fi
1805 \vskip\@tempskipa
1806 \fi
1807 \fi
1808 \noindent
1809 % 追加終わり
1810 \@ifstar
1811 {\@ssect{#3}{#4}{#5}{#6}}%
1812 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

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

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

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

```

1813 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
1814 \let\bxjs@if@ceph\@firstofone
1815 \else \let\bxjs@if@ceph\@gobble
1816 \fi

```

```

1817 \def\@sect#1#2#3#4#5#6[#7]#8{%
1818 \ifnum #2>\c@secnumdepth
1819 \let\@svsec\@empty
1820 \else
1821 \refstepcounter{#1}%
1822 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1823 \fi
1824 % 見出し後の空きを \@tempskipa にセット
1825 \@tempskipa #5\relax
1826 % 条件判断の順序を入れ替えました
1827 \ifdim \@tempskipa<\z@
1828 \def\@svsechd{%
1829 #6{\hskip #3\relax
1830 \@svsec #8}%
1831 \csname #1mark\endcsname{#7}}%
1832 \addcontentsline{toc}{#1}{%
1833 \ifnum #2>\c@secnumdepth \else
1834 \protect\numberline{\bxjs@label@sect{#1}}%

```

```

1835     \fi
1836     #7}}% 目次にフルネームを載せるなら #8
1837 \else
1838     \begingroup
1839     \interlinepenalty \@M % 下から移動
1840     #6{%
1841         \@hangfrom{\hskip #3\relax\@svsec}%
1842 %     \interlinepenalty \@M % 上に移動
1843     #8\@@par}%
1844 \endgroup
1845 \csname #1mark\endcsname{#7}%
1846 \addcontentsline{toc}{#1}{%
1847     \ifnum #2>\c@secnumdepth \else
1848         \protect\numberline{\bxjs@label@sect{#1}}%
1849     \fi
1850     #7}}% 目次にフルネームを載せるならここは #8
1851 \fi
1852 \@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。

```

1853 \def\@xsect#1{%
1854 % 見出しの後ろの空きを \@tempskipa にセット
1855     \@tempskipa #1\relax
1856 % 条件判断の順序を変えました
1857     \ifdim \@tempskipa<\z@
1858         \@nobreakfalse
1859         \global\@noskipsectrue
1860         \everypar{%
1861             \if@noskipsec
1862                 \global\@noskipsecfalse
1863             {\setbox\z@\lastbox}%
1864             \clubpenalty\@M
1865             \begingroup \@svsechd \endgroup
1866             \unskip
1867             \@tempskipa #1\relax
1868             \hskip -\@tempskipa
1869         \else
1870             \clubpenalty \@clubpenalty
1871             \everypar\expandafter{\bxjs@if@ceph\everyparhook}%
1872             \fi\bxjs@if@ceph\everyparhook}%
1873     \else
1874         \par \nobreak
1875         \vskip \@tempskipa
1876         \@afterheading
1877     \fi

```

```

1878 \if@slide
1879 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1880 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1881 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1882 \fi
1883 \par % 2000-12-18
1884 \ignorespaces}
1885 \def\@ssect#1#2#3#4#5{%
1886 \@tempskipa #3\relax
1887 \ifdim \@tempskipa<\z@
1888 \def\@svsechd{#4{\hskip #1\relax #5}}%
1889 \else
1890 \begingroup
1891 #4{%
1892 \@hangfrom{\hskip #1}%
1893 \interlinepenalty \@M #5\@@par}%
1894 \endgroup
1895 \fi
1896 \@xsect{#3}}

```

■柱関係の命令

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

```

\subsectionmark 1897 \newcommand*\chaptermark[1]{}
1898 % \newcommand*\sectionmark[1]{}
\subsubsectionmark 1899 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1900 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1901 % \newcommand*\paragraphmark[1]{}
1902 % \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。

```

1903 %<!book&!report>\setcounter{secnumdepth}{3}
1904 %<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 1905 \newcounter{part}
1906 %<book|report>\newcounter{chapter}
\c@subsubsection 1907 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1908 %<!book&!report>\newcounter{section}
\c@subparagraph 1909 \newcounter{subsection}[section]
1910 \newcounter{subsubsection}[subsection]
1911 \newcounter{paragraph}[subsubsection]
1912 \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`

`\thesection`

`\thesubsection`

`\thesubsubsection`

`\theparagraph`

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```
1913 \renewcommand{\thepart}{\@Roman\c@part}
1914 %<*!book&!report>
1915 \ifnum\bxjs@label@section=\bxjs@label@section@compat
1916 \renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1917 \renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1918 \else
1919 \renewcommand{\thesection}{\@arabic\c@section}
1920 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1921 \fi
1922 %</!book&!report>
1923 %<*book|report>
1924 \renewcommand{\thechapter}{\@arabic\c@chapter}
1925 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1926 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1927 %</book|report>
1928 \renewcommand{\thesubsubsection}{%
1929   \thesubsection.\@arabic\c@subsubsection}
1930 \renewcommand{\theparagraph}{%
1931   \thesubsubsection.\@arabic\c@paragraph}
1932 \renewcommand{\thesubparagraph}{%
1933   \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。

```
1934 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1935 %<book|report>\newcommand{\@chappos}{\postchaptername}
```

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

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

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

修正することで、問題を解消しました。実は、 \LaTeX の標準クラスでは 1998 年に修正されていた問題です (コミュニティ版 $\text{p}\text{\LaTeX}$ の標準クラス 2017/03/05 も参照)。

```
1936 %<*book|report>
1937 \newcommand\frontmatter{%
1938   \pltx@cleartooddpage
1939   \@mainmatterfalse
1940   \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。

```
1941 \newcommand\mainmatter{%
1942   \pltx@cleartooddpage
1943   \@mainmattertrue
1944   \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。

```
1945 \newcommand\backmatter{%
1946   \if@openleft
1947     \cleardoublepage
1948   \else\if@openright
1949     \cleardoublepage
1950   \else
1951     \clearpage
1952   \fi\fi
1953   \@mainmatterfalse}
1954 %</book|report>
```

■部

`\part` 新しい部を始めます。

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

`\secdef{星なし}{星あり}`

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

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

まず `book` と `report` のクラス以外です。

```
1955 %<*&!book&!report>
1956 \newcommand\part{%
1957   \if@noskipsec \leavevmode \fi
1958   \par
1959   \addvspace{4ex}%
1960   \if@english \@afterindentfalse \else \@afterindenttrue \fi
```



```

1961 \secdef\@part\@spart}
1962 %</*!book&!report>

book および report クラスの場合は、少し複雑です。

1963 %<*book|report>
1964 \newcommand\part{%
1965 \if@openleft
1966 \cleardoublepage
1967 \else\if@openright
1968 \cleardoublepage
1969 \else
1970 \clearpage
1971 \fi\fi
1972 \thispagestyle{empty}% 欧文用標準スタイルでは plain
1973 \if@twocolumn
1974 \onecolumn
1975 \@restonecoltrue
1976 \else
1977 \@restonecolfalse
1978 \fi
1979 \null\vfil
1980 \secdef\@part\@spart}
1981 %</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。

```

1982 %</*!book&!report>
1983 \def\@part[#1]#2{%
1984 \ifnum \c@secnumdepth >\m@ne
1985 \refstepcounter{part}%
1986 \addcontentsline{toc}{part}{%
1987 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1988 \else
1989 \addcontentsline{toc}{part}{#1}%
1990 \fi
1991 \markboth{}{}%
1992 {\parindent\z@
1993 \raggedright
1994 \interlinepenalty \@M
1995 \normalfont
1996 \ifnum \c@secnumdepth >\m@ne
1997 \Large\headfont\prepartname\thepart\postpartname
1998 \par\nobreak
1999 \fi
2000 \huge \headfont #2%
2001 \markboth{}{}\par}%
2002 \nobreak
2003 \vskip 3ex

```

```

2004 \@afterheading}
2005 %</!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。

```

2006 %<*book|report>
2007 \def\@part[#1]#2{%
2008   \ifnum \c@secnumdepth >-2\relax
2009     \refstepcounter{part}%
2010     \addcontentsline{toc}{part}{%
2011       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
2012   \else
2013     \addcontentsline{toc}{part}{#1}%
2014   \fi
2015   \markboth{}{}%
2016   {\centering
2017     \interlinepenalty \@M
2018     \normalfont
2019     \ifnum \c@secnumdepth >-2\relax
2020       \huge\headfont \prepartname\thepart\postpartname
2021       \par\vskip20\p@?
2022     \fi
2023     \Huge \headfont #2\par}%
2024   \@endpart}
2025 %</book|report>

```

\@spart 番号を付けない部です。

```

2026 %<*!book&!report>
2027 \def\@spart#1{%
2028   \parindent \z@ \raggedright
2029   \interlinepenalty \@M
2030   \normalfont
2031   \huge \headfont #1\par}%
2032   \nobreak
2033   \vskip 3ex
2034   \@afterheading}
2035 %</!book&!report>
2036 %<*book|report>
2037 \def\@spart#1{%
2038   \centering
2039   \interlinepenalty \@M
2040   \normalfont
2041   \Huge \headfont #1\par}%
2042   \@endpart}
2043 %</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。

```
2044 %<*book|report>
2045 \def\@endpart{\vfil\newpage
2046   \if@twoside
2047     \if@openleft %% added (2017/02/24)
2048       \null\thispagestyle{empty}\newpage
2049     \else\if@openright %% added (2016/12/13)
2050       \null\thispagestyle{empty}\newpage
2051     \fi\fi %% added (2016/12/13, 2017/02/24)
2052   \fi
2053   \if@restonecol
2054     \twocolumn
2055   \fi}
2056 %</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。

```
2057 %<*book|report>
2058 \newcommand{\chapter}{%
2059   \if@openleft\cleardoublepage\else
2060   \if@openright\cleardoublepage\else\clearpage\fi\fi
2061   \plainifnotempty % 元: \thispagestyle{plain}
2062   \global\@topnum\z@
2063   \if@english \@afterindentfalse \else \@afterindenttrue \fi
2064   \secdef
2065     {\@omit@numberfalse\@chapter}%
2066     {\@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。

```
2067 \def\@chapter[#1]#2{%
2068   \ifnum \c@secnumdepth >\m@ne
2069     \if@mainmatter
2070       \refstepcounter{chapter}%
2071       \typeout{\@chapapp\thechapter\@chappos}%
2072       \addcontentsline{toc}{chapter}%
2073         {\protect\numberline
2074 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
2075         {\@chapapp\thechapter\@chappos}%
2076         #1}%
2077     \else\addcontentsline{toc}{chapter}{#1}\fi
2078   \else
2079     \addcontentsline{toc}{chapter}{#1}%
2080   \fi
2081   \chaptermark{#1}%
2082   \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}}
```

```

2083 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}}%
2084 \if@twocolumn
2085   \@topnewpage[\@makechapterhead{#2}]%
2086 \else
2087   \@makechapterhead{#2}%
2088   \@afterheading
2089 \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。

```

2090 \def\@makechapterhead#1{%
2091   \vspace*{2\Cvs}% 欧文は 50pt
2092   {\parindent \z@ \raggedright \normalfont
2093     \ifnum \c@secnumdepth >\m@ne
2094       \if@mainmatter
2095         \huge\headfont \@chapapp\thechapter\@chappos
2096         \par\nobreak
2097         \vskip \Cvs % 欧文は 20pt
2098       \fi
2099     \fi
2100     \interlinepenalty\@M
2101     \Huge \headfont #1\par\nobreak
2102     \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。

```

2103 \def\@schapter#1{%
2104   \chaptermark{#1}%
2105   \if@twocolumn
2106     \@topnewpage[\@makeschapterhead{#1}]%
2107   \else
2108     \@makeschapterhead{#1}\@afterheading
2109   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

2110 \def\@makeschapterhead#1{%
2111   \vspace*{2\Cvs}% 欧文は 50pt
2112   {\parindent \z@ \raggedright
2113     \normalfont
2114     \interlinepenalty\@M
2115     \Huge \headfont #1\par\nobreak
2116     \vskip 3\Cvs}} % 欧文は 40pt
2117 %</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。

```

2118 \if@twocolumn

```

```

2119 \newcommand{\section}{%
2120 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
2121 \startsection{section}{1}{\z@}%
2122 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
2123 %<kiyou> {\Cvs}{0.5\Cvs}%
2124 % {\normalfont\large\headfont\@secapp}%
2125 {\normalfont\large\headfont\raggedright}}
2126 \else
2127 \newcommand{\section}{%
2128 \if@slide\clearpage\fi
2129 \startsection{section}{1}{\z@}%
2130 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2131 {.5\Cvs \@plus.3\Cdp}% 後アキ
2132 % {\normalfont\Large\headfont\@secapp}%
2133 {\normalfont\Large\headfont\raggedright}}
2134 \fi

```

\subsection 同上です。

```

2135 \if@twocolumn
2136 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2137 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2138 {\normalfont\normalsize\headfont}}
2139 \else
2140 \newcommand{\subsection}{\startsection{subsection}{2}{\z@}%
2141 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
2142 {.5\Cvs \@plus.3\Cdp}% 後アキ
2143 {\normalfont\large\headfont}}
2144 \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)。

```

2145 \if@twocolumn
2146 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2147 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
2148 {\normalfont\normalsize\headfont}}
2149 \else
2150 \newcommand{\subsubsection}{\startsection{subsubsection}{3}{\z@}%
2151 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2152 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
2153 {\normalfont\normalsize\headfont}}
2154 \fi

```

\paragraph 見出しの後ろで改行されません。

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

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

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

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

段落のマーク (■) が必ず和文フォントで出力されるようにする。

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

```
2155 \newcommand\jsParagraphMark{\relax\jsJaChar{■}}
2156 \let\bxjs@org@paragraph@mark\jsParagraphMark
2157 \ifx\bxjs@paragraph@mark\@empty
2158   \let\jsParagraphMark\@empty
2159 \else\ifx\bxjs@paragraph@mark\@undefined\else
2160   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
2161 \fi\fi
2162 \let\jsJaChar\@empty
2163 \if@twocolumn
2164   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2165     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2166 %<jspf>    {\normalfont\normalsize\headfont}}
2167 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2168 \else
2169   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
2170     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
2171     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
2172 %<jspf>    {\normalfont\normalsize\headfont}}
2173 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
2174 \fi
```

`\subparagraph` 見出しの後ろで改行されません。

```
2175 \if@twocolumn
2176   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2177     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2178     {\normalfont\normalsize\headfont}}
2179 \else
2180   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
2181     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
2182     {\normalfont\normalsize\headfont}}
2183 \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 ですが、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。

```
2184 \if@slide
2185   \setlength\leftmargini{1\jsZw}
2186 \else
2187   \if@twocolumn
2188     \setlength\leftmargini{2\jsZw}
2189   \else
2190     \setlength\leftmargini{3\jsZw}
2191   \fi
2192 \fi
```

`\leftmarginii` ii, iii, iv は `\labelsep` とそれぞれ ‘(m)’, ‘vii.’, ‘M.’ の幅との和より大きくすること
`\leftmarginiii` になっています。ここでは全角幅の整数倍に丸めました。

```
\leftmarginiv 2193 \if@slide
\leftmarginv 2194   \setlength\leftmarginii {1\jsZw}
2195   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 2196   \setlength\leftmarginiv {1\jsZw}
2197   \setlength\leftmarginv {1\jsZw}
2198   \setlength\leftmarginvi {1\jsZw}
2199 \else
2200   \setlength\leftmarginii {2\jsZw}
2201   \setlength\leftmarginiii{2\jsZw}
2202   \setlength\leftmarginiv {2\jsZw}
2203   \setlength\leftmarginv {1\jsZw}
2204   \setlength\leftmarginvi {1\jsZw}
2205 \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。

```
2206 \setlength \labelsep {0.5\jsZw} % .5em
2207 \setlength \labelwidth{\leftmargini}
2208 \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。

```
2209 \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 2210 \@beginparpenalty -\@lowpenalty
\@itempenalty 2211 \@endparpenalty -\@lowpenalty
2212 \@itempenalty -\@lowpenalty
```

`\@listi` `\@listi` は `\leftmargin`, `\parsep`, `\topsep`, `\itemsep` などのトップレベルの定義を
`\@listI` します。この定義は、フォントサイズコマンドによって変更されます（たとえば `\small` の
中では小さい値に設定されます）。このため、`\normalsize` がすべてのパラメータを戻せる

ように、`\@listI` で `\@listi` のコピーを保存します。元の値はかなり複雑ですが、ここでは簡素化してしまいました。特に最初と最後に行送りの半分の空きが入るようにしてあります。アスキーの標準スタイルではトップレベルの `itemize`, `enumerate` 環境でだけ最初と最後に行送りの半分の空きが入るようになっていました。

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

```
2213 \def\@listi{\leftmargin\leftmarginI
2214   \parsep \z@
2215   \topsep 0.5\baselineskip
2216   \itemsep \z@ \relax}
2217 \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）。

```
2218 \@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 2219 \def\@listii{\leftmargin\leftmarginii
\@listiv 2220   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
2221   \topsep \z@
\@listv 2222   \parsep \z@
\@listvi 2223   \itemsep\parsep}
2224 \def\@listiii{\leftmargin\leftmarginiii
2225   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
2226   \topsep \z@
2227   \parsep \z@
2228   \itemsep\parsep}
2229 \def\@listiv {\leftmargin\leftmarginiv
2230   \labelwidth\leftmarginiv
2231   \advance\labelwidth-\labelsep}
2232 \def\@listv {\leftmargin\leftmarginv
2233   \labelwidth\leftmarginv
2234   \advance\labelwidth-\labelsep}
2235 \def\@listvi {\leftmargin\leftmarginvi
2236   \labelwidth\leftmarginvi
2237   \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。

```
2238 \renewcommand{\theenumi}{\@arabic\c@enumi}
2239 \renewcommand{\theenumii}{\@alph\c@enumii}
2240 \renewcommand{\theenumiii}{\@roman\c@enumiii}
2241 \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
`\labelenumii` 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。

```

2242 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
2243 \newcommand*\jsInJaParen[1]{%
2244   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
2245 \newcommand{\labelenumi}{\theenumi.}
2246 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
2247 \newcommand{\labelenumiii}{\theenumiii.}
2248 \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
`\p@enumiii` の書式です。これも第 2 レベルは和文用かっこにしました。

```

\p@enumiv 2249 \renewcommand{\p@enumii}{\theenumi}
2250 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
2251 \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 2252 \newcommand\labelitemi{\textbullet}
\labelitemiii 2253 \newcommand\labelitemii{\normalfont\bfseries \textendash}
2254 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 2255 \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。

```

2256 \newenvironment{description}{%
2257   \list{}{%
2258     \labelwidth=\leftmargin
2259     \labelsep=1\jsZw
2260     \advance \labelwidth by -\labelsep
2261     \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
(たとえば `\hspace{1\jsZw}`) を入れるのもいいと思います。

```

2262 \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 (+titlepage)** 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。

```
2263 %<*book|report>
2264 \newenvironment{chapterabstract}{%
2265   \begin{list}{}{%
2266     \listparindent=1\jsZw
2267     \itemindent=\listparindent
2268     \rightmargin=\z@
2269     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
2270 %</book|report>
```

“普通の” **abstract** 環境の定義。

```
2271 %<*article|report|slide>
2272 \newbox\@abstractbox
2273 \if@titlepage
2274   \newenvironment{abstract}{%
2275     \titlepage
2276     \null\vfil
2277     \@beginparpenalty\@lowpenalty
2278     \begin{center}%
2279       \headfont \abstractname
2280       \@endparpenalty\@M
2281     \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。

```
2282   \par}%
2283   {\par\vfil\null\endtitlepage}
2284 \else
2285   \newenvironment{abstract}{%
2286     \if@twocolumn
2287       \ifx\maketitle\relax
2288         \section*{\abstractname}%
```

```

2289 \else
2290 \global\setbox\@abstractbox\hbox\bgroup
2291 \begin{minipage}[b]{\textwidth}
2292 \small\parindent1\jsZw
2293 \begin{center}%
2294 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2295 \end{center}%
2296 \list{}{%
2297 \listparindent\parindent
2298 \itemindent \listparindent
2299 \rightmargin \leftmargin}%
2300 \item\relax
2301 \fi
2302 \else
2303 \small
2304 \begin{center}%
2305 {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2306 \end{center}%
2307 \list{}{%
2308 \listparindent\parindent
2309 \itemindent \listparindent
2310 \rightmargin \leftmargin}%
2311 \item\relax
2312 \fi}{\if@twocolumn
2313 \ifx\maketitle\relax
2314 \else
2315 \endlist\end{minipage}\egroup
2316 \fi
2317 \else
2318 \endlist
2319 \fi}
2320 \fi
2321 %</article|report|slide>
2322 %<*jspf>
2323 \newbox\@abstractbox
2324 \newenvironment{abstract}{%
2325 \global\setbox\@abstractbox\hbox\bgroup
2326 \begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
2327 \small
2328 \if@english \parindent6\jsc@mmm \else \parindent1\jsZw \fi}%
2329 {\end{minipage}\egroup}
2330 %</jspf>

```

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

```

2331 %<*book|report>
2332 \ifbxjs@force@chapterabstract
2333 \let\abstract\chapterabstract
2334 \let\endabstract\endchapterabstract

```

```
2335 \fi
2336 %</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。

```
2337 %<*jspf>
2338 %\newbox\@keywordsbox
2339 %\newenvironment{keywords}{%
2340 % \global\setbox\@keywordsbox\hbox\bgroup
2341 % \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
2342 % \small\parindent0\jsZw}%
2343 % {\end{minipage}\egroup}
2344 %</jspf>
```

■verse 環境

verse 詩のための `verse` 環境です。

```
2345 \newenvironment{verse}{%
2346 \let \=\@centercr
2347 \list{}{%
2348 \itemsep \z@
2349 \itemindent -2\jsZw % 元: -1.5em
2350 \listparindent\itemindent
2351 \rightmargin \z@
2352 \advance\leftmargin 2\jsZw}% 元: 1.5em
2353 \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。

```
2354 \newenvironment{quotation}{%
2355 \list{}{%
2356 \listparindent\parindent
2357 \itemindent\listparindent
2358 \rightmargin \z@}%
2359 \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。

```
2360 \newenvironment{quote}%
2361 {\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。

```
2362 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2363   \item[\hskip \labelsep{\headfont #1\ #2}]}
2364 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2365   \item[\hskip \labelsep{\headfont #1\ #2 (#3) }]}
```

`titlepage` タイトルを独立のページに出力するのに使われます。

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

```
2366 \newenvironment{titlepage}{%
2367   %<book> \pltx@cleartooddpage %% 2017-02-24
2368   \if@twocolumn
2369     \@restonecoltrue\onecolumn
2370   \else
2371     \@restonecolfalse\newpage
2372   \fi
2373   \thispagestyle{empty}%
2374   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
2375   24
2376   }%
2377   {\if@restonecol\twocolumn \else \newpage \fi
2378   \if@twoside\else
2379     \setcounter{page}\@ne
2380   \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2380 %<*&!book&!report>
2381 \newcommand{\appendix}{\par
2382   \setcounter{section}{0}%
2383   \setcounter{subsection}{0}%
2384   \ifnum\bxjs@label@section=\bxjs@label@section@compat
```

```

2385 \gdef\presectionname{\appendixname}%
2386 \gdef\postsectionname{}%
2387 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2388 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2389 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2390 \else
2391 \gdef\@secapp{\appendixname}%
2392 \gdef\@secpos{}%
2393 \gdef\thesection{\@Alph\c@section}%
2394 \fi}
2395 %</!book&!report>
2396 %<*book|report>
2397 \newcommand{\appendix}{\par
2398 \setcounter{chapter}{0}%
2399 \setcounter{section}{0}%
2400 \gdef\@chapapp{\appendixname}%
2401 \gdef\@chappos{}%
2402 \gdef\thechapter{\@Alph\c@chapter}}
2403 %</book|report>

```

8.4 パラメータの設定

■array と tabular 環境

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

```
2404 \setlength\arraycolsep{5\p@?}
```

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

```
2405 \setlength\tabcolsep{6\p@?}
```

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

```
2406 \setlength\arrayrulewidth{.4\p@}
```

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

```
2407 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2408 \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。

```
2409 \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。

```
2410 \setlength\fboxsep{3\p@?}
2411 \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。

```
2412 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2413 %<*book|report>
2414 \@addtoreset{equation}{chapter}
2415 \renewcommand\theequation
2416 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2417 %</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。

```
2418 % \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。

```
2419 % \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。

```
2420 % \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。

```
2421 %<!*book&!report>
2422 \newcounter{figure}
2423 \renewcommand \thefigure {\@arabic\c@figure}
2424 %</!book&!report>
2425 %<*book|report>
2426 \newcounter{figure}[chapter]
2427 \renewcommand \thefigure
2428     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2429 %</book|report>
```

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

```
\ext@figure 2430 \def\fps@figure{tbp}
2431 \def\ftype@figure{1}
\fnun@figure 2432 \def\ext@figure{lof}
2433 \def\fnun@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2434 \newenvironment{figure}%
2435         {\@float{figure}}%
2436         {\end@float}
2437 \newenvironment{figure*}%
2438         {\@dblfloat{figure}}%
2439         {\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。

```
2440 %<!*book&!report>
2441 \newcounter{table}
2442 \renewcommand\thetable{\@arabic\c@table}
2443 %</!book&!report>
2444 %<*book|report>
2445 \newcounter{table}[chapter]
2446 \renewcommand \thetable
2447     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2448 %</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 2449 \def\fps@table{tbp}
2450 \def\ftype@table{2}
\fnun@table 2451 \def\ext@table{lot}
2452 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` * は段抜きのフロートです。

`table*`


```

2453 \newenvironment{table}%
2454         {\@float{table}}%
2455         {\end@float}
2456 \newenvironment{table*}%
2457         {\@dblfloat{table}}%
2458         {\end@dblfloat}

```

8.6 キャプション

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

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

```

2459 \newlength\abovecaptionskip
2460 \newlength\belowcaptionskip
2461 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2462 \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。

```

2463 %<!*jspf>
2464 % \long\def\@makecaption#1#2{{\small
2465 %   \advance\leftskip 10\jsc@mmm
2466 %   \advance\rightskip 10\jsc@mmm
2467 %   \vskip\abovecaptionskip
2468 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2469 %   \ifdim \wd\@tempboxa >\hsize
2470 %     #1\hskip1\jsZw\relax #2\par
2471 %   \else
2472 %     \global \minipagefalse
2473 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2474 %   \fi
2475 %   \vskip\belowcaptionskip}}
2476 \long\def\@makecaption#1#2{{\small
2477   \advance\leftskip .0628\linewidth
2478   \advance\rightskip .0628\linewidth
2479   \vskip\abovecaptionskip
2480   \sbox\@tempboxa{#1\zwspace#2}%
2481   \ifdim \wd\@tempboxa <\hsize \centering \fi
2482   #1\zwspace#2\par
2483   \vskip\belowcaptionskip}}
2484 %</!*jspf>
2485 %<!*jspf>
2486 \long\def\@makecaption#1#2{%

```

```

2487 \vskip\abovecaptionskip
2488 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2489 \ifdim \wd\@tempboxa >\hsize
2490   {\small\sffamily
2491     \list{#1}{%
2492       \renewcommand{\makelabel}[1]{##1\hfil}
2493       \itemsep \z@
2494       \itemindent \z@
2495       \labelsep \z@
2496       \labelwidth 11\jsc@mmm
2497       \listparindent\z@
2498       \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
2499 \else
2500   \global \@minipagefalse
2501   \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2502 \fi
2503 \vskip\belowcaptionskip}
2504 %</jspf>

```

9 フォントコマンド

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

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

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

```

\if@jsc@warnoldfontcmd


---


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


---


2505 \newif\if@jsc@warnoldfontcmd
2506 \@jsc@warnoldfontcmdtrue
2507 \newif\if@jsc@warnoldfontcmdexception
2508 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand
2509 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2510   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2511   \DeclareOldFontCommand{#1}{%
2512     \bxjs@oldfontcmd{#1}#2%
2513   }{%

```

```

2514     \bxjs@oldfontcmd{#1}#3%
2515 }%
2516 }
2517 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2518   \ClassInfo\bxjs@clsname
2519   {Old font command '\string#1' is used!!\MessageBreak
2520     The first occurrence is}%
2521 }

```

`\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。

```

2522 \newcommand*{\allowoldfontcommands}{%
2523   \@jsc@warnoldfontcmdexceptiontrue}
2524 \newcommand*{\disallowoldfontcommands}{%
2525   \@jsc@warnoldfontcmdexceptionfalse}

```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2526 \let\bxjs@oldfontcmd@list\@empty
2527 \def\bxjs@oldfontcmd#1{%
2528   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2529 \def\bxjs@oldfontcmd@a#1#2{%
2530   \if@jsc@warnoldfontcmdexception\else
2531     \global\@jsc@warnoldfontcmdfalse
2532     \ifx#1\relax
2533       \global\let#1=t%
2534       \jsc@warnoldfontcmd{#2}%
2535     \fi
2536   \fi}
2537 \def\bxjs@warnoldfontcmd@final{%
2538 % \par
2539   \global\let\bxjs@warnoldfontcmd@final\@empty
2540   \let\@tempa\@empty
2541   \def\do##1{%
2542     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2543       \edef\@tempa{\@tempa \space\string##1}\fi}
2544   \bxjs@oldfontcmd@list
2545   \ifx\@tempa\@empty\else
2546     \ClassWarningNoLine\bxjs@clsname
2547     {Some old font commands were used in text\MessageBreak
2548       (see the log file for detail)}%
2549     \ClassInfo\bxjs@clsname
2550     {Some old font commands were used in text:\MessageBreak
2551       \space\@tempa\MessageBreak
2552       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2553       new font selection scheme called NFSS2 with several\MessageBreak
2554       new, combinable font commands. The

```

```

2555     class provides\MessageBreak
2556     the old font commands
2557     only for compatibility%
2558     \@gobble}%
2559 \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。

```

2560 \def\bxjs@warnoldfontcmd@kick@final{%
2561   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2562 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2563 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2564 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2565 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2566 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2567 \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。

```

2568 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2569 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2570 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2571 \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 2572 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2573 \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。

```
2574 \newcommand\@pnumwidth{1.55em}
2575 \newcommand\@tocrmarg{2.55em}
2576 \newcommand\@dotsep{4.5}
2577 %<!book&!report>\setcounter{tocdepth}{2}
2578 %<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)

```
2579 \newdimen\jsc@tocl@width
2580 \newcommand{\tableofcontents}{%
2581 %<*book|report>
2582 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2583 \settowidth\@tempdima{\headfont\appendixname}%
2584 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2585 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
```

```

2586 \if@twocolumn
2587 \@restonecoltrue\onecolumn
2588 \else
2589 \@restonecolfalse
2590 \fi
2591 \chapter*{\contentsname}%
2592 \mkboth{\contentsname}{}%
2593 %</book|report>
2594 %<*&book&!report>
2595 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2596 \settowidth\@tempdima{\headfont\appendixname}%
2597 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2598 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2599 \section*{\contentsname}%
2600 \mkboth{\contentsname}{\contentsname}%
2601 %</!book&!report>
2602 \@starttoc{toc}%
2603 %<book|report> \if@restonecol\twocolumn\fi
2604 }

```

\l@part 部の目次です。

```

2605 \newcommand*{\l@part}[2]{%
2606 \ifnum \c@tocdepth >-2\relax
2607 %<!book&!report> \addpenalty\@secpenalty
2608 %<book|report> \addpenalty{-\@highpenalty}%
2609 \addvspace{2.25em \@plus\p@}%
2610 \begingroup
2611 \parindent \z@
2612 % \@pnumwidth should be \@tocrmarg
2613 % \rightskip \@pnumwidth
2614 \rightskip \@tocrmarg
2615 \parfillskip -\rightskip
2616 {\leavevmode
2617 \large \headfont
2618 \setlength\@lnumwidth{4\jsZw}%
2619 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2620 \nobreak
2621 %<book|report> \global\@nobreaktrue
2622 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2623 \endgroup
2624 \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)

```

2625 %<*&book|report>
2626 \newcommand*{\l@chapter}[2]{%
2627 \ifnum \c@tocdepth >\m@ne
2628 \addpenalty{-\@highpenalty}%

```

```

2629 \addvspace{1.0em \@plus\p@?}
2630 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2631 \begingroup
2632 \parindent\z@
2633 % \rightskip\@pnumwidth
2634 \rightskip\@tocrmarg
2635 \parfillskip-\rightskip
2636 \leavevmode\headfont
2637 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2638 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2639 \advance\leftskip\@lnumwidth \hskip-\leftskip
2640 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2641 \penalty\@highpenalty
2642 \endgroup
2643 \fi}
2644 %</book|report>

```

\l@section 節の目次です。

```

2645 %<*&book&!report>
2646 \newcommand*{\l@section}[2]{%
2647 \ifnum \c@tocdepth >\z@
2648 \addpenalty{\@secpenalty}%
2649 \addvspace{1.0em \@plus\p@?}%
2650 \begingroup
2651 \parindent\z@
2652 % \rightskip\@pnumwidth
2653 \rightskip\@tocrmarg
2654 \parfillskip-\rightskip
2655 \leavevmode\headfont
2656 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2657 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2658 \advance\leftskip\@lnumwidth \hskip-\leftskip
2659 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2660 \endgroup
2661 \fi}
2662 %</!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。

```

2663 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

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

\l@subsubsection しれません。

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

```

\l@subparagraph 2664 %<*&book&!report>
2665 % \newcommand*{\l@subsection}{\@dottedtocline{2}{1.5em}{2.3em}}
2666 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2667 % \newcommand*{\l@paragraph}{\@dottedtocline{4}{7.0em}{4.1em}}

```

```

2668 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2669 %
2670 % \newcommand*{\l@section} {\@dottedtocline{2}{1zw}{3zw}}
2671 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2672 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2673 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2674 %
2675 \newcommand*{\l@subsection}{%
2676     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2677     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2678 \newcommand*{\l@subsubsection}{%
2679     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2680     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2681 \newcommand*{\l@paragraph}{%
2682     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2683     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2684 \newcommand*{\l@subparagraph}{%
2685     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2686     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2687 %<!/book&!report>
2688 %<*book|report>
2689 % \newcommand*{\l@subsection} {\@dottedtocline{2}{3.8em}{3.2em}}
2690 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2691 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2692 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2693 \newcommand*{\l@section}{%
2694     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2695     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2696 \newcommand*{\l@subsection}{%
2697     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2698     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2699 \newcommand*{\l@subsubsection}{%
2700     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2701     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2702 \newcommand*{\l@paragraph}{%
2703     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2704     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2705 \newcommand*{\l@subparagraph}{%
2706     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2707     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2708 %</book|report>

```

\numberline 欧文版 L^AT_EX では \numberline{...} は幅 \@tempdima の箱に左詰めで出力する命令で
 \@lnumwidth 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。

```

2709 \newdimen\@lnumwidth
2710 \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` に変えています。

```

2711 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2712   \vskip \z@ \@plus.2\p@?
2713   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2714    \parindent #2\relax\@afterindenttrue
2715    \interlinepenalty\@M
2716    \leavevmode
2717    \@lnumwidth #3\relax
2718    \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2719    {#4}\nobreak
2720    \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2721     mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2722     \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2723 \newcommand{\listoffigures}{%
2724 %<*book|report>
2725 \if@twocolumn\@restonecoltrue\onecolumn
2726 \else\@restonecolfalse\fi
2727 \chapter*{\listfigurename}%
2728 \@mkboth{\listfigurename}{}%
2729 %</book|report>
2730 %<*&!book&!report>
2731 \section*{\listfigurename}%
2732 \@mkboth{\listfigurename}{\listfigurename}%
2733 %</&!book&!report>
2734 \@starttoc{lof}%
2735 %<book|report> \if@restonecol\twocolumn\fi
2736 }

```

`\l@figure` 図目次の項目を出力します。

```

2737 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2738 \newcommand{\listoftables}{%
2739 %<*book|report>
2740 \if@twocolumn\@restonecoltrue\onecolumn
2741 \else\@restonecolfalse\fi
2742 \chapter*{\listtablename}%
2743 \@mkboth{\listtablename}{}%
2744 %</book|report>
2745 %<*&!book&!report>
2746 \section*{\listtablename}%
2747 \@mkboth{\listtablename}{\listtablename}%
2748 %</&!book&!report>

```

```

2749 \starttoc{lot}%
2750 %<book|report> \if@restonecol\twocolumn\fi
2751 }

```

`\l@table` 表目次は図目次と同じです。

```

2752 \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。

```

2753 \newdimen\bibindent
2754 \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。

```

2755 \newenvironment{thebibliography}[1]{%
2756   \@jsc@warnoldfontcmdexceptiontrue
2757   \global\let\presectionname\relax
2758   \global\let\postsectionname\relax
2759   %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2760   %<*kiyou>
2761   \vspace{1.5\baselineskip}
2762   \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2763   \vspace{0.5\baselineskip}
2764   %</kiyou>
2765   %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2766   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2767   \list{\@biblabel{\@arabic\c@enumiv}}%
2768     {\settowidth\labelwidth{\@biblabel{#1}}%
2769     \leftmargin\labelwidth
2770     \advance\leftmargin\labelsep
2771     \@openbib@code
2772     \usecounter{enumiv}%
2773     \let\p@enumiv\empty
2774     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2775   %<kiyou> \small
2776   \sloppy
2777   \clubpenalty4000
2778   \@clubpenalty\clubpenalty
2779   \widowpenalty4000%
2780   \sfcode`\.\@m}
2781   {\def\@noitemerr
2782     {\@latex@warning{Empty `thebibliography' environment}}}%
2783   \endlist}

```

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

```
2784 \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。

```
2785 \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。

```
2786 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

`\cite` 文献の番号を出力する部分は ltbibl.dtx で定義されていますが、コンマとカッコを和文
`\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必
`\@citex` 要に応じて生かしてください。かっこの前後に入るグルーを `\jsInhibitGlue` で取っていますので、オリジナル同様、Knuth-`\cite{knu}` のように半角空白で囲んでください。

```
2787 % \def\@citex[#1]#2{\leavevmode
2788 %   \let\@citea\@empty
2789 %   \@cite{\@for\@citeb:=#2\do
2790 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2791 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2792 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2793 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2794 %       \G@refundefinedtrue
2795 %       \@latex@warning
2796 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2797 %     {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2798 % \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。

```
2799 % \DeclareRobustCommand\cite{\unskip
2800 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2801 %   \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2802 %     , \jsInhibitGlue\ #2\fi}}}$}
```

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)。

```
2803 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2804   \if@twocolumn
2805     \onecolumn\@restonecolfalse
2806   \else
2807     \clearpage\@restonecoltrue
2808   \fi
```

```

2809 \columnseprule.4pt \columnsep 2\jsZw
2810 \ifx\multicols\@undefined
2811 %<book|report> \twocolumn[\@makeschapterhead{\indexname}%
2812 %<book|report> \addcontentsline{toc}{chapter}{\indexname}}%
2813 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2814 %<!book&!report> \twocolumn[\section*{\indexname}]%
2815 \else
2816 \ifdim\textwidth<\fullwidth
2817 \setlength{\evensidemargin}{\oddsidemargin}
2818 \setlength{\textwidth}{\fullwidth}
2819 \setlength{\linewidth}{\fullwidth}
2820 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}%
2821 %<book|report> \addcontentsline{toc}{chapter}{\indexname}}%
2822 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2823 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]%
2824 \else
2825 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}%
2826 %<book|report> \addcontentsline{toc}{chapter}{\indexname}}%
2827 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2828 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]%
2829 \fi
2830 \fi
2831 %<book|report> \mkboth{\indexname}{}%
2832 %<!book&!report> \mkboth{\indexname}{\indexname}%
2833 \plainifnotempty % \thispagestyle{plain}
2834 \parindent\z@
2835 \parskip\z@ \@plus .3\p@?\relax
2836 \let\item\@idxitem
2837 \raggedright
2838 \footnotesize\narrowbaselines
2839 }{
2840 \ifx\multicols\@undefined
2841 \if@restonecol\onecolumn\fi
2842 \else
2843 \end{multicols}
2844 \fi
2845 \clearpage
2846 }

```

\@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 2847 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2848 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2849 \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。

```
2850 \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。

```
2851 \newcommand\seename{\if@english see\else →\fi}  
2852 \newcommand\alsoname{\if@english see also\else →\fi}
```

10.4 脚注

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

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

```
2853 \ifx\pltx@foot@penalty\@undefined  
2854 \let\footnotes@ve=\footnote  
2855 \def\footnote{\jsInhibitGlue\footnotes@ve}  
2856 \let\footnotemarks@ve=\footnotemark  
2857 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}  
2858 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

pT_EX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
2859 \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。

```
2860 % \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。

```
2861 \renewcommand{\footnoterule}{%
2862   \kern-2.6\p@? \kern-.4\p@
2863   \hrule width .4\columnwidth
2864   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2865 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

```
2866 \long\def\@footnotetext{%
2867   \insert\footins\bgroup
2868     \normalfont\footnotesize
2869     \interlinepenalty\interfootnotelinepenalty
2870     \splittopskip\footnotesep
2871     \splitmaxdepth \dp\strutbox \floatingpenalty \MM
2872     \hsize\columnwidth \@parboxrestore
2873     \protected@edef\@currentlabel{%
2874       \csname p@footnote\endcsname\@thefnmark
2875     }%
2876     \color@begingroup
2877       \@makefnmark{%
2878         \rule{\z@\footnotesep}{\ignorespaces}%
2879         \futurelet\next\fo@t}
2880 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@@
2881                                     \else \let\next\fo@t\fi \next}
2882 \def\fo@@{\bgroup\aftergroup\@foot\let\next}
2883 \def\fo@t#1{#1\@foot}
2884 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2885   \ifx\pltx@foot@penalty\undefined\else
2886     \ifhmode\null\fi
2887     \ifnum\pltx@foot@penalty=\z@\else
2888       \penalty\pltx@foot@penalty
2889       \pltx@foot@penalty\z@
2890     \fi
2891   \fi}
```

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

```

2892 \newcommand\@makefntext[1]{%
2893   \advance\leftskip 3\jsZw
2894   \parindent 1\jsZw
2895   \noindent
2896   \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。

```

2897 % \def\@xfootnotenext[#1]{%
2898 %   \begingroup
2899 %     \ifnum#1>\z@
2900 %       \csname c@\@mpfn\endcsname #1\relax
2901 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2902 %     \else
2903 %       \unrestored@protected@xdef\@thefnmark{}%
2904 %     \fi
2905 %   \endgroup
2906 %   \@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。

```

2907 \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。

```

2908 \def\everyparhook{\jsInhibitGlueAtParTop}
2909 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
2910 \AtBeginDocument{\everypar{\everyparhook}}
2911 \fi

```

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

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

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

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

```
2912 \ifx j\jsEngine
2913 \def\@inhibitglue{%
2914   \futurelet\@let@token\@inhibitglue}
2915 \begingroup
2916 \let\GDEF=\gdef
2917 \let\CATCODE=\catcode
2918 \let\ENDGROUP=\endgroup
2919 \CATCODE`k=12
2920 \CATCODE`a=12
2921 \CATCODE`n=12
2922 \CATCODE`j=12
2923 \CATCODE`i=12
2924 \CATCODE`c=12
2925 \CATCODE`h=12
2926 \CATCODE`r=12
2927 \CATCODE`t=12
2928 \CATCODE`e=12
2929 \GDEF\KANJI@CHARACTER{kanji character }
2930 \ENDGROUP
2931 \def\@inhibitglue{%
2932   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
2933   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
2934   \def\jsc@ig@temp{#1}%
2935   \ifx\jsc@ig@temp\empty
2936     \ifnum\the\inhibitxspcode`#2=2\relax
2937       \inhibitglue
2938     \fi
2939   \fi}
2940 \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。

```
2941 \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。

```
2942 \def\@doendpe{%
2943   \@endpetrue
2944   \def\par{%
2945     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2946   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

[2017-08-31] minipage 環境にも対策します。

```
2947 \def\@setminipage{%
2948   \@minipagetrue
2949   \everypar{\@minipagefalse\everypar{\everyparhook}}%
2950 }
```

`\item` 命令の直後です。

```
2951 \def\@item[#1]{%
2952   \if@noparitem
2953     \@donoparitem
2954   \else
2955     \if@inlabel
2956       \indent \par
2957     \fi
2958     \ifhmode
2959       \unskip\unskip \par
2960     \fi
2961     \if@newlist
2962       \if@nobreak
2963         \@nbitem
2964       \else
2965         \addpenalty\@beginparpenalty
2966         \addvspace\@topsep
2967         \addvspace{-\parskip}%
2968       \fi
2969     \else
2970       \addpenalty\@itempenalty
2971       \addvspace\itemsep
2972     \fi
2973     \global\@inlabeltrue
2974   \fi
2975   \everypar{%
2976     \@minipagefalse
2977     \global\@newlistfalse
2978     \if@inlabel
2979       \global\@inlabelfalse
2980     {\setbox\z@\lastbox
2981       \ifvoid\z@
2982         \kern-\itemindent
2983       \fi}%
```

```

2984     \box\@labels
2985     \penalty\z@
2986     \fi
2987     \if@nobreak
2988         \@nobreakfalse
2989         \clubpenalty \@M
2990     \else
2991         \clubpenalty \@clubpenalty
2992         \everypar{\everyparhook}%
2993     \fi
2994     \everyparhook}%
2995 \if@noitemarg
2996     \@noitemargfalse
2997 \if@nmbrlist
2998     \refstepcounter\@listctr
2999 \fi
3000 \fi
3001 \sbox\@tempboxa{\makelabel{#1}}}%
3002 \global\setbox\@labels\hbox{%
3003     \unhbox\@labels
3004     \hskip \itemindent
3005     \hskip -\labelwidth
3006     \hskip -\labelsep
3007     \ifdim \wd\@tempboxa >\labelwidth
3008         \box\@tempboxa
3009     \else
3010         \hbox to\labelwidth {\unhbox\@tempboxa}%
3011     \fi
3012     \hskip \labelsep}%
3013 \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。

```

3014 \def\@afterheading{%
3015     \@nobreaktrue
3016     \everypar{%
3017         \if@nobreak
3018             \@nobreakfalse
3019             \clubpenalty \@M
3020         \if@afterindent \else
3021             {\setbox\z@\lastbox}%
3022         \fi
3023     \else
3024         \clubpenalty \@clubpenalty
3025         \everypar{\everyparhook}%
3026     \fi\everyparhook}}

```

「`\everyparhook` 用のパッチ処理」はここまで。

3027 \fi

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

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

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

```
3028 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@none\else
3029 \AtEndOfPackage{%
3030 \def\@gnewline #1{%
3031   \ifvmode
3032     \@nolnerr
3033   \else
3034     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
3035     \jsInhibitGlue \ignorespaces
3036   \fi}
3037 }
3038 \fi
```

12 いろいろなロゴ

\LaTeX 関連のロゴを作り直します。

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

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

※`\小`、`\上小` の制御綴は定義しない。

```
3039 \if@jslogo
3040   \IfFileExists{jslogo.sty}{%
3041     \RequirePackage{jslogo}%
3042   }{%
3043     \ClassWarningNoLine\bxjs@clsname
3044       {The package 'jslogo' is not installed.\MessageBreak
3045        It is included in the recent release of\MessageBreak
3046        the 'jsclasses' bundle}
3047   }
```

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。

```

3049 %\let\ltx@ifnextchar\@ifnextchar
3050 %\def\ProvidesFile#1{%
3051 %   \begingroup
3052 %     \catcode`\ 10 %
3053 %     \ifnum \endlinechar<256 %
3054 %       \ifnum \endlinechar>\m@ne
3055 %         \catcode\endlinechar 10 %
3056 %       \fi
3057 %     \fi
3058 %     \@makeother\/%
3059 %     \@makeother\&%
3060 %     \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 3061 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 3062 \newcommand{\postpartname}{\if@english\else 部\fi}
3063 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 3064 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 3065 \newcommand{\presectionname}{}% 第
\postsectionname 3066 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 3067 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 3068 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
3069 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 3070 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 3071 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}

```

```

3072 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 3073 %<!jspf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
3074 %<jspf>\newcommand{\figurename}{Fig.~}
3075 %<!jspf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
3076 %<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname 3077 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
3078 \newcommand{\appendixname}{\if@english \else 付録\fi}
3079 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

環境変数 SOURCE_DATE_EPOCH / FORCE_SOURCE_DATE が設定されている場合は“今日”が過去・未来の日付になる可能性がある。その場合、和暦表記は 1970 年から平成の終わりまでの期間でのみサポートする。
 ※新元号が公表されたら対応する。

```

3080 \@tempwafalse
3081 \if p\jsEngine \@tempwattrue \fi
3082 \if n\jsEngine \@tempwattrue \fi
3083 \if@tempswa \expandafter\@firstoftwo
3084 \else \expandafter\@secondoftwo
3085 \fi
3086 {%
3087 % 欧文 8bitTeX の場合
3088 \newif\ifjsSeireki \jsSeirekitrue
3089 \def\bxjs@decl@Seireki@cmds{%
3090 \def\西暦{\jsSeirekitrue}%
3091 \def\和暦{\jsSeirekifalse}}
3092 \def\Seireki{\jsSeirekitrue}
3093 \def\Wareki{\jsSeirekifalse}
3094 \def\bxjs@if@use@seireki{%
3095 \ifjsSeireki \expandafter\@firstoftwo
3096 \else \expandafter\@secondoftwo \fi}
3097 }{%
3098 \newif\if 西暦 \西暦 true
3099 \def\bxjs@decl@Seireki@cmds{%
3100 \def\西暦{\西暦 true}%
3101 \def\和暦{\西暦 false}}
3102 \def\Seireki{\西暦 true}
3103 \def\Wareki{\西暦 false}
3104 \def\bxjs@if@use@seireki{%
3105 \if 西暦 \expandafter\@firstoftwo

```

```

3106 \else \expandafter\@secondoftwo \fi}
3107 }
3108 \bxjs@decl@Seireki@cmds
3109 \let\bxjs@unxp\@firstofone \let\bxjs@onxp\@firstofone
3110 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded
3111 \def\bxjs@onxp{\unexpanded\expandafter}}
3112 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
3113 \def\bxjs@iai{\noexpand~}
3114 \else \def\bxjs@iai{}
3115 \fi

```

`\jyear` 和暦における年の表記（元号 + 年数）。

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

※一般の年数の整数レジスタは `\bxjs@jyear` で、`\heisei` はこれの別名になる。

```

3116 \newcount\bxjs@jyear
3117 \let\bxjs@gengo\@empty \bxjs@jyear\year
3118 \@tempcnta=\the\year\two@digits\month\two@digits\day\relax
3119 \ifnum\@tempcnta<19700101
3120 \else\ifnum\@tempcnta<19890108
3121 \def\bxjs@gengo{昭和}\advance\bxjs@jyear-1925\relax
3122 \let\heisei\bxjs@jyear
3123 \else\ifnum\@tempcnta<20190501
3124 \def\bxjs@gengo{平成}\advance\bxjs@jyear-1988\relax
3125 \let\heisei\bxjs@jyear
3126 \else
3127 % 新元号対応
3128 % \def\bxjs@gengo{新元号}\advance\bxjs@jyear-2018\relax
3129 \fi\fi\fi
3130 \edef\jyear{%
3131 \ifx\bxjs@gengo\@empty \the\bxjs@jyear\bxjs@iai
3132 \else \bxjs@onxp{\bxjs@gengo}%
3133 \ifnum\bxjs@jyear=\@ne \bxjs@unxp{元}%
3134 \else \bxjs@iai\the\bxjs@jyear\bxjs@iai
3135 \fi
3136 \fi}

```

`\today` 英語、西暦、和暦で場合分けをする。

```

3137 \edef\bxjs@today{%
3138 \if@english
3139 \ifcase\month\or
3140 January\or February\or March\or April\or May\or June\or
3141 July\or August\or September\or October\or November\or December\fi
3142 \space\number\day, \number\year
3143 \else
3144 \noexpand\bxjs@if@use@seireki{%
3145 \number\year\bxjs@iai\bxjs@unxp{年}%
3146 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3147 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%

```

```

3148 }{%
3149 \bxjs@onxp{\jayear}\bxjs@unxp{年}%
3150 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
3151 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
3152 }%
3153 \fi}
3154 \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。

```

3155 \AtBeginDocument{%
3156 \ifx\bbj@jpn@Seirekitrue\@undefined\else
3157 \bxjs@decl@Seireki@cmds
3158 \g@addto@macro\datejapanese{%
3159 \let\today\bxjs@today}%
3160 \fi}

```

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

```

3161 \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。

```

3162 %<slide>\pagestyle{empty}%
3163 %<article|report>\pagestyle{plain}%
3164 %<book>\pagestyle{headings}%
3165 \pagenumbering{arabic}
3166 \if@twocolumn
3167 \twocolumn
3168 \sloppy
3169 \flushbottom
3170 \else
3171 \onecolumn
3172 \raggedbottom
3173 \fi
3174 %<*slide>
3175 \renewcommand\familydefault{\sfdefault}
3176 \raggedright
3177 %</slide>

```

■BXJS 独自の追加処理 

和文ドライバのファイルを読み込む。

```

3178 \catcode`\?=12
3179 \ifx\bxjs@jadriver\relax\else

```

```
3180 \input{bxjsja-\bxjs@jadriver.def}
```

```
3181 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
3182 \bxjs@restore@jltrcc
```

```
3183 %</cls>
```

以上です。

付録 A 和文ドライバの仕様

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

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

```
3184 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

ただし、(u)p_T_EX 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

```
3185 %<*minimal>
```

```
3186 %% 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.
```

```
3187 \def\DeclareJaTextFontCommand#1#2{%
```

```
3188   \DeclareRobustCommand#1[1]{%
```

```
3189     \relax
```

```
3190     \ifmmode \expandafter\nfss@text \fi
```

```
3191     {#2##1}}%
```

```
3192 }
```

```
\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。
```

```
3193 \def\DeclareJaMathFontCommand#1#2{%
```

```
3194   \DeclareRobustCommand#1[1]{%
```

```
3195     \relax
```

```
3196     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
3197     \nfss@text{\fontfamily\familydefault
```

```
3198       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
3199       #2##1}}%
```

```

3200 }%
3201 }

```

`\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。

```

3202 \long\def\bxjs@@CSsfdefault{\sfdefault}%
3203 \@onlypreamble\bxjs@if@sf@default
3204 \def\bxjs@if@sf@default#1{%
3205   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
3206   \AtBeginDocument{%
3207     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
3208 }

```

`\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。

```

3209 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
3210 \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。

```

3211 \def\jsLetHeadChar#1#2{%
3212   \begingroup
3213   \escapechar=`\ %
3214   \let\bxjs@tmpa={% brace-match-hack
3215     \bxjs@let@hchar@exp#2}%
3216   \endgroup
3217   \let#1\bxjs@g@tmpa}
3218 \def\bxjs@let@hchar@exp{%
3219   \futurelet\@let@token\bxjs@let@hchar@exp@a}
3220 \def\bxjs@let@hchar@exp@a{%
3221   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
3222   \bxjs@let@hchar@out\let\relax
3223   }{\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi% 空白
3224   \bxjs@let@hchar@out\let\space%
3225   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
3226   \bxjs@let@hchar@out\let\@backslashchar
3227   }{\bxjs@let@hchar@exp@b}}}%
3228 \def\bxjs@let@hchar@exp@b#1{%
3229   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
3230 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
3231   %\message{<#1#2>}%
3232   \bxjs@cond@if#1\@backslashchar\fi% 制御綴

```

```

3233 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
3234 \bxjs@let@hchar@out\let\relax
3235 }{%else
3236 \expandafter\bxjs@let@hchar@exp
3237 }%
3238 }{%else
3239 \bxjs@let@hchar@chr#1%
3240 }}
3241 \def\bxjs@let@hchar@chr#1{%
3242 \bxjs@let@hchar@out\def{{#1}}}}
3243 \def\bxjs@let@hchar@out#1#2{%
3244 \global#1\bxjs@g@tmpa#2\relax
3245 \toks@{\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

3246 \chardef\bxjs@let@hchar@csta=128
3247 \chardef\bxjs@let@hchar@cstb=192
3248 \chardef\bxjs@let@hchar@cstc=224
3249 \chardef\bxjs@let@hchar@cstd=240
3250 \chardef\bxjs@let@hchar@cste=248
3251 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
3252 \def\bxjs@let@hchar@chr@ue#1{%
3253 \@tempcnta=#1\relax
3254 %\message{\the\@tempcnta}%
3255 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
3256 \bxjs@let@hchar@chr@ue@a#1%
3257 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
3258 \bxjs@let@hchar@out\let\relax
3259 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
3260 \bxjs@let@hchar@chr@ue@b
3261 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
3262 \bxjs@let@hchar@chr@ue@c
3263 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
3264 \bxjs@let@hchar@chr@ue@d
3265 }{%else
3266 \bxjs@let@hchar@out\let\relax
3267 }}}}
3268 \def\bxjs@let@hchar@chr@ue@a#1{%
3269 \bxjs@let@hchar@out\def{{#1}}}}
3270 \def\bxjs@let@hchar@chr@ue@b#1#2{%
3271 \bxjs@let@hchar@out\def{{#1#2}}}}
3272 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
3273 \bxjs@let@hchar@out\def{{#1#2#3}}}}
3274 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
3275 \bxjs@let@hchar@out\def{{#1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
3276 \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。

```
3277 \def\bxjs@let@hchar@chr@pp#1{%
3278   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
3279 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
3280   \%message{(#1)}%
3281   \bxjs@cond\if#1t\fi{%
3282     \bxjs@let@hchar@chr@ue#3%
3283   }{%else
3284     \bxjs@let@hchar@out\def{#3}}%
3285   }}
3286 \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。

```
3287 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
3288 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
3289 \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。

```
3290 \@onlypreamble\bxjs@declarefontshape
3291 \ifjsWithupTeX
3292 \def\bxjs@declarefontshape{%
3293   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
3294   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
3295   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
3296   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
3297 }
3298 \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。

```
3299 \else
3300 \def\bxjs@declarefontshape{%
3301   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
3302   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
3303   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%

```

```

3304 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
3305 }
3306 \def\bxjs@sizereference{jis}
3307 \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。

```

3308 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
3309   \def\bxjs@y{#5}}
3310 \ifjsWithpTeXng \def\bxjs@y{10}%
3311 \else
3312 \expandafter\expandafter\expandafter\bxjs@tmpa
3313 \expandafter\string\the\jfont\relax
3314 \fi
3315 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
3316               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
3317   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
3318    \expandafter\let\csname\bxjs@x/\bxjs@y\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。

```

3319 \begingroup
3320 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
3321 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
3322 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
3323 % 幅が丁度 10pt なら補正は不要
3324 \ifdim\wd\z@=10pt
3325   \global\let\bxjs@scale\jsScale
3326 \else
3327 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3328   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3329   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3330   \bxjs@invscale\@tempdima\bxjs@tmpa
3331   \xdef\bxjs@scale{\strip@pt\@tempdima}
3332 \fi
3333 \endgroup
3334 %\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。

```

3335 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3336 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{%
3337 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{%
3338 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{%

```

```

3339 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3340 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3341 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3342 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
3343 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
3344 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
3345 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
3346 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
3347 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
3348 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
3349 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}

```

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。

```

3350 \DeclareRobustCommand\rmfamily
3351   {\not@math@alphabet\rmfamily\mathrm
3352     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3353 \DeclareRobustCommand\sffamily
3354   {\not@math@alphabet\sffamily\mathsf
3355     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3356 \DeclareRobustCommand\ttfamily
3357   {\not@math@alphabet\ttfamily\mathtt
3358     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3359 \ifx\DeclareFixJFMCJKTextFontCommand\@undefined
3360 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3361 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3362 \fi
3363 \bxjs@if@sf@default{%
3364   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3365 \selectfont

```

■パラメタの設定

```

3366 \prebreakpenalty\jis"2147=10000
3367 \postbreakpenalty\jis"2148=10000
3368 \prebreakpenalty\jis"2149=10000
3369 \inhibitxspcode`!=1
3370 \inhibitxspcode`〒=2
3371 \xspcode`+=3
3372 \xspcode`%=3

```

"80～"FF の範囲の \xspcode を 3 に変更。

```

3373 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3374   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3375 \let\jsInhibitGlueAtParTop\@inhibitglue

```

\jsResetDimen は空のままでよい。

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

```
3376 \begingroup
3377 \catcode`\!=0
```

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

```
3378 \gdef\bxjs@ptex@dir{%
3379   !iftdir t%
3380   !else!ifydir y%
3381   !else ?%
3382   !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。

```
3383 % 古い \@makefnmark の定義
3384 \long\def\bxjs@tmpa{\hbox{%
3385   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3386   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3387 \ifx\@makefnmark\bxjs@tmpa
3388 \long\gdef\@makefnmark{%
3389   !ifydir \hbox{} \hbox{\@textsuperscript{\normalfont\@thefnmark}} \hbox{}%
3390   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3391 \fi
3392 \endgroup
```

B.3 pdfTeX 用の処理

```
3393 \else\ifx p\jsEngine
3394 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3395 \@onlypreamble\bxjs@cjk@loaded
3396 \def\bxjs@cjk@loaded{%
3397   \def\@footnotemark{%
3398     \leavevmode
3399     \ifhmode
3400       \edef\@x@sf{\the\spacefactor}%
3401       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3402         \unkern\unkern
3403         \ifdim\lastskip>\z@ \unskip \fi
3404       \fi\fi
3405       \nobreak
3406     \fi
3407     \@makefnmark
3408     \ifhmode \spacefactor\@x@sf \fi
3409     \relax}%
```



```

3410 \let\bxjs@CJK@loaded\relax
3411 }
3412 \AtBeginDocument{%
3413 \ifpackageloaded{CJK}{%
3414 \bxjs@CJK@loaded
3415 }{}%
3416 }

```

B.4 XeTeX 用の処理

```

3417 \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。

```

3418 \def\bxjs@let@hchar@chr#1{%
3419 \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3420 \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3421 \bxjs@let@hchar@chr@xe
3422 }{\bxjs@let@hchar@out\def{{#1}}}}
3423 \def\bxjs@let@hchar@chr@xe#1{%
3424 \lccode`0=`#1\relax
3425 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

3426 \ifx\XeTeXgenerateactualtext\@undefined\else
3427 \def\bxjs@do@precisetext{%
3428 \XeTeXgenerateactualtext=\@one}
3429 \fi

```

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

```

3430 \@onlypreamble\bxjs@do@simplejasetup
3431 \def\bxjs@do@simplejasetup{%
3432 \ifnum\XeTeXinterchartokenstate>\z@
3433 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3434 \jsSimpleJaSetup
3435 \ClassInfo\bxjs@clsname
3436 {'\string\jsSimpleJaSetup' is applied\@gobble}%
3437 \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

3438 \newcommand*{\jsSimpleJaSetup}{%
3439 \XeTeXlinebreaklocale "ja"\relax
3440 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3441 \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理（エンジン共通）

```

3442 \fi\fi\fi

```

simplejasetup オプションの処理。

```

3443 \ifx\bxjs@do@simplejasetup\@undefined\else
3444   \AtBeginDocument{%
3445     \ifbxjs@simplejasetup
3446       \bxjs@do@simplejasetup
3447     \fi}
3448 \fi

precisetext オプションの処理。
3449 \ifbxjs@precisetext
3450   \ifx\bxjs@do@precisetext\@undefined
3451     \ClassWarning\bxjs@clsname
3452       {The current engine does not support the\MessageBreak
3453         'precisetext' option\@gobble}
3454   \else
3455     \bxjs@do@precisetext
3456   \fi
3457 \fi

```

■段落頭でのグルー挿入禁止

`\bxjs@check@everyparhook` 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。

```

3458 \onlypreamble\bxjs@check@everyparhook
3459 \def\bxjs@check@everyparhook{%
3460   \ifx\jsInhibitGlueAtParTop\@empty
3461     \def\bxjs@tmpa{\jsInhibitGlueAtParTop}%
3462     \ifx\everyparhook\bxjs@tmpa
3463       \let\everyparhook\@empty
3464     \fi
3465   \fi}
3466 \AtBeginDocument{\bxjs@check@everyparhook}

```

`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。

```

3467 \ifnum\bxjs@everyparhook=\bxjs@everyparhook@modern

```

まず `\everypar` を“乗っ取る” 処理を行う。

```

3468   \let\bxjs@everypar\everypar
3469   \newtoks\everypar
3470   \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。

```

3471   \bxjs@everypar{\the\expandafter\everypar\everyparhook}%
3472 \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。

3473 \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。

3474 \@onlypreamble\bxjs@adjust@fancyhdr

3475 \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。

3476 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%

3477 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%

3478 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3479 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3480 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3481 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3482 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%

3483 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%

3484 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi

3485 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi

3486 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi

3487 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi

3488 \def\bxjs@tmpa{\rm\thepage\strut}%

3489 \def\bxjs@tmpb{\thepage\strut}%

3490 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi

3491 \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。

3492 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth

3493 \setlength{\@tempdima}{\fullwidth-\textwidth}%

3494 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%

3495 }\bxjs@tmpa

3496 \fi\fi

3497 \PackageInfo\bxjs@clsname

3498 {Patch to fancyhdr is applied\@gobble}}

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

3499 \def\bxjs@pagestyle@hook{%

3500 \@ifpackageloaded{fancyhdr}{%

3501 \bxjs@adjust@fancyhdr

3502 \global\let\bxjs@adjust@fancyhdr\relax

3503 }{}}

\pagestyle にフックを入れ込む。

3504 \let\bxjs@org@pagestyle\pagestyle

3505 \def\pagestyle{%

3506 \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。

```
3507 \AtBeginDocument{%
3508   \bxjs@pagestyle@hook
3509   \global\let\bxjs@pagestyle@hook\relax}
```

■和文空白命令

```
3510 \ifbxjs@jaspace@cmd
```

`\jaenspace` 半角幅の水平空き。

```
3511 \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。

```
3512 \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。

```
3513 \def\_ {\zwspace}
```

`\jaspace` `jlreq` クラスと互換の命令。

```
3514 \DeclareRobustCommand*{\jaspace}[1]{%
3515   \expandafter\ifx\csname bxjs@jaspace@@#1\endcsname\relax
3516   \ClassError\bxjs@clsname
3517   {Unknown jaspac: #1}{\@eha}%
3518   \else
3519   \csname bxjs@jaspace@@#1\endcsname
3520   \fi}
3521 \def\bxjs@jaspace@@zenkaku{\hskip 1\jsZw\relax}
3522 \def\bxjs@jaspace@@nibu{\hskip .5\jsZw\relax}
3523 \def\bxjs@jaspace@@shibu{\hskip .25\jsZw\relax}
3524 \fi
```

終わり。

```
3525 \fi
```

以上で終わり。

```
3526 %</minimal>
```

付録 C 和文ドライバ : standard

標準のドライバ。

- `\rmfamily/\sffamily/\ttfamily` での和文ファミリ連動
- `\mcfamily/\gtfamily`
- `\textmc/\textgt`
- `\zw`

- `\jQ/\jH`
- `\trueQ/\trueH/\ascQ`
- `\setkanjiskip/\getkanjiskip`
- `\setxkanjiskip/\getxkanjiskip`
- `\autospacing/\noautospacing`
- `\autoxspacing/\noautoxspacing`

■和文フォント指定の扱い standard 和文ドライバでは `\jsJaFont` の値を和文フォントの“プリセット”の指定として用いる。プリセットの値は、 \TeX Live の `kanji-config-updmap` コマンドで使う“ファミリ”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。

C.1 共通処理 (1)

まず minimal ドライバを読み込む。

```
3527 %<*standard>
3528 %% このファイルは日本語文字を含みます
3529 \input{bxjsja-minimal.def}
```

`simplejasetup` は standard では無効になる。

```
3530 \bxjs@simplejasetupfalse
```

■japaram オプションの処理 japaram の値を key-value リストとして解釈する。keyval のファミリは `bxjsStd` とする。

`\ifbxjs@jp@jismmiv` 2004JIS 字形を優先させるか。

```
3531 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3532 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3533 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3534 \define@key{bxjsStd}{jis2004}[true]{%
3535   \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。

```
3536 \newif\ifbxjs@jp@units

units オプションの処理。
3537 \let\bxjs@kv@units@true\bxjs@jp@unitstrue
3538 \let\bxjs@kv@units@false\bxjs@jp@unitsfalse
3539 \define@key{bxjsStd}{units}[true]{%
3540   \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。

```
3541 \let\bxjs@jp@font\@empty
```

font オプションの処理。

```
3542 \define@key{bxjsStd}{font}{%
3543   \edef\bxjs@jp@font{#1}}
```

実際の japaram の値を適用する。

```
3544 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3545 \expandafter\bxjs@next\expandafter{\jsJaParam}
```

■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。

```
3546 \ifbxjs@jp@jismmiv
3547   \g@addto@macro\@classoptionslist{jis2004}
3548 % \ifpackagewith 判定への対策
3549   \PassOptionsToPackage{jis2004}{otf}
3550 \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。

```
3551 \if j\jsEngine
3552   \bxjs@jp@unitsfalse
3553 \fi
```

units パラメタが有効の場合は、bxcalc パッケージを読みこんで、\usepTeXunits 命令を実行する。

```
3554 \ifbxjs@jp@units
3555   \IfFileExists{bxcalc.sty}{%
3556     \RequirePackage{bxcalc}[2018/01/28]%v1.0a
3557     \ifx\usepTeXunits\undefined
3558       \PackageWarningNoLine\bxjs@clsname
3559         {Cannot support pTeX units (zw etc.), since\MessageBreak
3560           the package 'bxcalc' is too old}%
3561     \else \usepTeXunits
3562     \fi
3563   }{%else
3564     \PackageWarningNoLine\bxjs@clsname
3565       {Cannot support pTeX units (zw etc.), since\MessageBreak
3566         the package 'bxcalc' is unavailable}%
3567   }
3568 \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。

```
3569 \ifbxjs@jp@units
3570   \def\bxjs@let@lenexpr#1#2{%
3571     \edef#1{#2}%
3572     \expandafter\CUXParseExpr\expandafter#1\expandafter{#1}}
3573 \else
3574   \def\bxjs@let@lenexpr{\edef}
```

3575 \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。

3576 \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 3577 \@tempdima=0.25mm
      3578 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
      3579 \let\jH\jQ
```

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

```
\trueH 3580 \ifjscmag
      3581 \@tempdimb=\jsBaseFontSize\relax
      3582 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
      3583 \@tempdima=2.5mm
      3584 \bxjs@invscale\@tempdima\bxjs@tmpa
      3585 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
      3586 \@tempdima=10pt
      3587 \bxjs@invscale\@tempdima\bxjs@tmpa
      3588 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
      3589 \else \let\trueQ\jQ \let\bxjs@truept\p@
      3590 \fi
      3591 \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。

```
3592 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3593 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3594 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3595 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3596 \fi
```

\jsSetQHLLength \jsSetQHLLength\CS{〈長さ式〉} : \setlength の変種で、通常の calc の長さ式の代わ
りに、〈実数〉Q・〈実数〉H・〈実数〉trueQ・〈実数〉trueH が記述できる。

```
3597 \def\jsSetQHLLength#1#2{%
3598 \begingroup
3599 \bxjs@parse@qh{#2}%
3600 \ifx\bxjs@tmpb\relax \setlength\@tempdima{#2}%
3601 \else \@tempdima\bxjs@tmpb\relax
3602 \fi
3603 \xdef\bxjs@g@tmpa{\the\@tempdima}%
3604 \endgroup
3605 #1=\bxjs@g@tmpa\relax}
```

`\bxjs@parse@qh` `jsSetQHLlength` の下請け。#1 が Q/H/trueQ/trueH で終わる場合、`\bxjs@tmpb` にそれに等しい寸法の表現を返す。それ以外では、`\bxjs@tmpb` は `\relax` になる。

```
3606 \if j\jsEngine
```

(u)pL^AT_EX ではこの処理は不要なので無効化する。

```
3607 \def\bxjs@parse@qh#1{%
3608   \let\bxjs@tmpb\relax}
3609 \else\ifbxjs@jp@units
3610 \def\bxjs@parse@qh#1{%
3611   \let\bxjs@tmpb\relax}
3612 \else
3613 \def\bxjs@parse@qh#1{%
3614   \def\bxjs@tmpa{#1}\let\bxjs@tmpb\relax
3615   \bxjs@parse@qh@a{trueQ}\trueQ
3616   \bxjs@parse@qh@a{trueH}\trueH
3617   \bxjs@parse@qh@a{Q}\jQ
3618   \bxjs@parse@qh@a{H}\jH}
3619 \def\bxjs@parse@qh@a#1#2{%
3620   \ifx\bxjs@tmpb\relax
3621     \def\bxjs@next##1#1\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}}%
3622     \expandafter\bxjs@next\bxjs@tmpa\@nil#1\@nil\@nnil{#2}%
3623   \fi}
3624 \def\bxjs@parse@qh@b#1#2#3{%
3625   \ifx\@nnil#2\@nnil\else \def\bxjs@tmpb{#1#3}\fi}
3626 \fi\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。

```
3627 \def\jafontsize#1#2{%
3628   \begingroup
3629     \bxjs@jafontsize@a{#1}%
3630     \@tempdimb\jsInverseScale\@tempdima
3631     \bxjs@jafontsize@a{#2}%
3632     \xdef\bxjs@g@tmpa{%
3633       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3634   \endgroup\bxjs@g@tmpa}
3635 \def\bxjs@jafontsize@a#1{%
3636   \bxjs@parse@qh{#1}%
3637   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3638   \@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。

```
3639 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。


```

3640 \newcommand*\setkanjiskip[1]{%
3641   \bxjs@let@lenexpr\bxjs@kanjiskip{#1}%
3642   \bxjs@reset@kanjiskip}

```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```

3643 \newcommand*\getkanjiskip{%
3644   \bxjs@kanjiskip}

```

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

```

3645 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue

```

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

```

\bxjs@disable@kanjiskip 3646 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3647   \bxjs@kanjiskip@enabledtrue
3648   \bxjs@reset@kanjiskip}
3649 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3650   \bxjs@kanjiskip@enabledfalse
3651   \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。

```

3652 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3653   \ifbxjs@kanjiskip@enabled
3654     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3655   \else \@tempskipa\z@
3656   \fi
3657   \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 3658 \def\bxjs@xkanjiskip{0pt}

```

```

\getxkanjiskip 3659 \newcommand*\setxkanjiskip[1]{%

```

```

\ifbxjs@xkanjiskip@enabled 3660   \bxjs@let@lenexpr\bxjs@xkanjiskip{#1}%
3661   \bxjs@reset@xkanjiskip}

```

```

\bxjs@enable@xkanjiskip 3662 \newcommand*\getxkanjiskip{%

```

```

\bxjs@disable@xkanjiskip 3663   \bxjs@xkanjiskip}

```

```

3664 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue

```

```

\bxjs@reset@xkanjiskip 3665 \bxjs@robust@def\bxjs@enable@xkanjiskip{%

```

```

3666   \bxjs@xkanjiskip@enabledtrue

```

```

3667   \bxjs@reset@xkanjiskip}

```

```

3668 \bxjs@robust@def\bxjs@disable@xkanjiskip{%

```

```

3669   \bxjs@xkanjiskip@enabledfalse

```

```

3670   \bxjs@reset@xkanjiskip}

```

```

3671 \bxjs@robust@def\bxjs@reset@xkanjiskip{%

```

```

3672   \ifbxjs@xkanjiskip@enabled

```

```

3673     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%

```

```

3674   \else \@tempskipa\z@

```

```

3675   \fi

```

```

3676   \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。

```
3677 \g@addto@macro\jsResetDimen{%
3678   \bxjs@reset@kanjiskip
3679   \bxjs@reset@xkanjiskip}
3680 \let\bxjs@apply@kanjiskip\relax
3681 \let\bxjs@apply@xkanjiskip\relax
```

■和文フォント指定の扱い

\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。

```
3682 \@onlypreamble\bxjs@adjust@jafont
3683 \def\bxjs@adjust@jafont#1{%
3684   \ifx\jsJaFont\bxjs@@auto
3685     \bxjs@get@kanjiEmbed
3686     \ifx\bxjs@kanjiEmbed\relax
3687       \let\bxjs@tmpa\@empty
3688     \else
3689       \let\bxjs@tmpa\bxjs@kanjiEmbed
3690     \fi
3691   \else
3692     \let\bxjs@tmpa\jsJaFont
3693   \fi
3694   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3695     \ClassWarningNoLine\bxjs@clsname
3696     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3697     not available on the current situation}%
3698     \let\bxjs@tmpa\@empty
3699   \fi\fi
3700 }
3701 \def\bxjs@@auto{auto}
3702 \def\bxjs@@noEmbed{noEmbed}
```

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

```
3703 \let\bxjs@kanjiEmbed\relax
```

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

```
3704 \@onlypreamble\bxjs@get@kanjiEmbed
3705 \def\bxjs@get@kanjiEmbed{%
3706   \begingroup\setbox\z@=\hbox{%
3707     \global\let\bxjs@g@tmpa\relax
3708     \endlinechar\m@ne
3709     \let\do\@makeother\dospecials
3710     \catcode32=10 \catcode12=10 %form-feed
3711     \let\bxjs@tmpa\@empty
```

```

3712 \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3713 \ifeof\@inputcheck\else
3714 \read\@inputcheck to\bxjs@tmpa
3715 \closein\@inputcheck
3716 \fi
3717 \ifx\bxjs@tmpa\@empty\else
3718 \openin\@inputcheck="\bxjs@tmpa"\relax
3719 \@tempswatrue
3720 \loop\if@tempswa
3721 \read\@inputcheck to\bxjs@tmpa
3722 \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3723 \ifx\bxjs@tmpb\relax\else
3724 \global\let\bxjs@g@tmpa\bxjs@tmpb
3725 \@tempswafalse
3726 \fi
3727 \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3728 \ifx\bxjs@tmpb\relax\else
3729 \global\let\bxjs@g@tmpa\bxjs@tmpb
3730 \@tempswafalse
3731 \fi
3732 \ifeof\@inputcheck \@tempswafalse \fi
3733 \repeat
3734 \fi
3735 }\endgroup
3736 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3737 }
3738 \@onlypreamble\bxjs@get@ke@a
3739 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3740 \ifx$#1$\def\bxjs@tmpb{#2}%
3741 \else \let\bxjs@tmpb\relax
3742 \fi}
3743 \@onlypreamble\bxjs@get@ke@b
3744 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3745 \ifx$#1$\def\bxjs@tmpb{#2}%
3746 \else \let\bxjs@tmpb\relax
3747 \fi}

```

`\jachar \jachar{< 文字 >}` : 和文文字として出力する。

```

3748 \newcommand*\jachar[1]{%
3749 \begingroup

```

\jsLetHeadChar で先頭の“文字”を拾ってそれを \bxjs@jachar に渡す。

```

3750 \jsLetHeadChar\bxjs@tmpa{#1}%
3751 \ifx\bxjs@tmpa\relax
3752 \ClassWarningNoLine\bxjs@clsname
3753 {Illegal argument given to \string\jachar}%
3754 \else
3755 \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3756 \fi

```

```
3757 \endgroup}
```

\jsJaChar を \jachar と等価にする。

```
3758 \let\jsJaChar\jachar
```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```
3759 \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。

```
3760 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3761 \@onlypreamble\bxjs@fix@hyperref@unicode
3762 \def\bxjs@fix@hyperref@unicode#1{%
3763   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3764   \@namedef{KV@Hyp@bxjs/hook}##1{%
3765     \KV@Hyp@unicode{##1}%
3766     \def\KV@Hyp@unicode####1{%
3767       \expandafter\ifx\csname if##1\expandafter\endcsname
3768         \csname if####1\endcsname\else
3769         \ClassWarningNoLine\bxjs@clsname
3770         {Blcoked hyperref option 'unicode=####1'}%
3771       \fi
3772     }%
3773   }%
3774 }
```

\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。

```
3775 \@onlypreamble\jsCheckHyperrefUnicode
3776 \let\jsCheckHyperrefUnicode\@empty
3777 \AtBeginDocument{\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。

```
3778 \@onlypreamble\bxjs@check@hyperref@unicode
3779 \def\bxjs@check@hyperref@unicode#1{%
3780   \g@addto@macro\jsCheckHyperrefUnicode{%
3781     \@tempswafalse
3782     \begingroup
3783       \expandafter\ifx\csname ifHy@unicode\endcsname\relax
3784         \aftergroup\@tempswattrue \fi
3785       \expandafter\ifx\csname ifHy@unicode\expandafter\endcsname
3786         \csname if#1\endcsname
3787       \aftergroup\@tempswattrue \fi
3788     \endgroup
3789     \if@tempswa\else
3790       \ClassError\bxjs@clsname
3791       {The value of hyperref 'unicode' key is not suitable\MessageBreak
3792       for the present engine (must be #1)}}%
```

```

3793      {\@ehc}%
3794      \fi}}

```

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

```

3795 \@onlypreamble\bxjs@urgent@special
3796 \def\bxjs@urgent@special#1{%
3797   \AtBeginDvi{\special{#1}}%
3798   \AtBeginDocument{%
3799     \ifpackageloaded{atbegshi}{%
3800       \begingroup
3801         \toks\z@{\special{#1}}%
3802         \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3803         \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3804       \endgroup
3805     }{}%
3806   }%
3807 }

```

`\bxjs@resolve@jafont@paren` jafont パラメタ値内の () を解決する。

```

3808 \def\bxjs@resolve@jafont@paren#1{%
3809   \def\bxjs@tmpb{\let#1}%
3810   \expandafter\bxjs@resolve@jafont@paren@a#1\@nil()\@nil\relax}
3811 \def\bxjs@resolve@jafont@paren@a#1(#2)#3\@nil#4\relax{%
3812   \ifx\relax#4\relax \bxjs@tmpb\jsJaFont
3813   \else
3814     \edef\bxjs@tmpa{#1\bxjs@jafont@paren{#2}#3}%
3815     \bxjs@tmpb\bxjs@tmpa
3816   \fi}

```

C.2 pTeX 用設定

```

3817 \if j\jsEngine

```

■ 共通命令の実装

```

3818 \def\bxjs@apply@kanjiskip{%
3819   \kanjiskip\@tempskipa}
3820 \def\bxjs@apply@xkanjiskip{%
3821   \xkanjiskip\@tempskipa}

```

`\jaJaChar` のサブマクロ。

```

3822 \def\bxjs@jachar#1{%
3823   \bxjs@jachar@a#1...\@nil}
3824 \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。

```

3825   \ifx.#2#1%

```

引数が複数トークンの場合は、UTF-8 のバイト列であるに見なし、そのスカラー値を `\@tempcnta` に代入する。

```

3826 \else\ifx.#3%
3827 \@tempcnta`#1 \multiply\@tempcnta64
3828 \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3829 \bxjs@jachar@b
3830 \else\ifx.#4%
3831 \@tempcnta`#1 \multiply\@tempcnta64
3832 \advance\@tempcnta`#2 \multiply\@tempcnta64
3833 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3834 \bxjs@jachar@b
3835 \else
3836 \@tempcnta`#1 \multiply\@tempcnta64
3837 \advance\@tempcnta`#2 \multiply\@tempcnta64
3838 \advance\@tempcnta`#3 \multiply\@tempcnta64
3839 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3840 \bxjs@jachar@b
3841 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3842 \ifjsWithupTeX
3843 \def\bxjs@jachar@b{\kchar\@tempcnta}
3844 \else
3845 \def\bxjs@jachar@b{%
3846 \ifx\bxUInt\@undefined\else
3847 \bxUInt{\@tempcnta}%
3848 \fi}
3849 \fi

```

和欧文間空白の命令 \jathinspace の実装。

```

3850 \ifbxjs@jaspace@cmd
3851 \def\jathinspace{\hskip\xkanjiskip}
3852 \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。

```

3853 \ifbxjs@jp@jismmiv
3854 \PassOptionsToPackage{prefer2004jis}{pxchfon}
3855 \PassOptionsToPackage{prefer2004jis}{pxbabel}
3856 \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 が指定されいる場合は、その値を pxchfon のオプションに追加する。

```

3857 \let\bxjs@jafont@paren\@firstofone
3858 \let\bxjs@tmpa\jsJaFont
3859 \ifx\bxjs@tmpa\bxjs@@auto
3860 \let\bxjs@tmpa\@empty
3861 \else\ifx\bxjs@tmpa\bxjs@@noEmbed

```

```

3862 \def\bxjs@tmpa{noembed}
3863 \fi\fi
3864 \bxjs@resolve@jafont@paren\bxjs@tmpa
3865 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3866 \ifx\bxjs@tmpa\@empty\else
3867 \edef\bxjs@next{%
3868 \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3869 }\bxjs@next
3870 \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。

※ 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……。)

```

3871 \begingroup
3872 \global\let\@gtempa\relax
3873 \catcode`\|=0 \catcode`\|=12
3874 |def|bxjs@check@a#1|@nil{%
3875 |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3876 |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3877 |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3878 |catcode`\|=0 \catcode`\|=12
3879 \def\bxjs@check@b#1keyval#2\@nnil{%
3880 \ifx$#2$\else
3881 \xdef\@gtempa{%
3882 \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3883 \fi}
3884 \@firstofone{%
3885 \catcode10=12 \endlinechar\m@ne
3886 \let\do\@makeother \dospecials \catcode32=10
3887 \openin\@inputcheck=otf.sty\relax
3888 \@tempswatrue
3889 \loop\if@tempswa
3890 \ifeof\@inputcheck \@tempswafalse \fi
3891 \if@tempswa
3892 \read\@inputcheck to\bxjs@line
3893 \expandafter\bxjs@check\bxjs@line\@nil
3894 \fi
3895 \repeat
3896 \closein\@inputcheck
3897 \endgroup}
3898 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3899 \ifbxjs@hyperref@enc
3900 \bxjs@check@hyperref@unicode{false}
3901 \fi

```

tounicode special 命令を出力する。

```
3902 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3903   \else\ifjsWithpTeXng T\else F\fi\fi T%
3904   \ifnum\jis"2121="A1A1 %euc
3905   \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3906   \else\ifnum\jis"2121="8140 %sjis
3907   \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3908   \else\ifnum\jis"2121="3000 %uptex
3909   \ifbxjs@bigcode
3910   \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3911   \PassOptionsToPackage{bigcode}{pxjahyper}
3912   \else
3913   \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3914   \fi
3915   \fi\fi\fi
3916   \let\bxToUnicodeSpecialDone=t
3917 \fi
```

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

```
3918 \ifx f\bxjs@enablejfam\else
3919   \@enablejfamtrue
3920 \fi
```

実際に和文用の数式ファミリの設定を行う。

```
3921 \if@enablejfam
3922   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3923   \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3924   \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3925   \jfam\symmincho
3926   \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3927   \AtBeginDocument{%
3928     \ifx\reDeclareMathAlphabet\undefined\else
3929       \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3930       \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3931       \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3932     \fi}
3933 \fi
```

C.3 pdfTeX 用設定 : CJK + bxcjkatype

```
3934 \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。


```

3935 \bxjs@adjust@jafont{f}
3936 \let\bxjs@jafont@paren\@firstofone
3937 \bxjs@resolve@jafont@paren\bxjs@tmpa
3938 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3939 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole}}
3940 \def\bxjs@tmpb{pandoc}\ifx\bxjs@tmpb\bxjs@jadriver\else
3941   \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{autotilde}}
3942 \fi
3943 \edef\bxjs@next{%
3944   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkjatype}[2013/10/15]% v0.2c
3945 }\bxjs@next
3946 \bxjs@cjk@loaded

```

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

※取りあえず固定はしない。

```

3947 \ifbxjs@hyperref@enc
3948   \PassOptionsToPackage{unicode}{hyperref}
3949 \fi

```

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

※ bxcjkjatype を whole 付きで使っていることが前提。

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

```

3950 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3951 \begingroup
3952   \CJK@input{UTF8.bdg}
3953 \endgroup
3954 \g@addto@macro\pdfstringdefPreHook{%
3955   \@nameuse{CJK@UTF8Binding}}%
3956 }
3957 \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。

```

3958 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3959 \g@addto@macro\pdfstringdefPreHook{%
3960   \ifx~\bxjs@@CJKtilde
3961     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3962     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3963     \let~\@empty
3964   \fi
3965 }
3966 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3967 \def\bxjs@@tildecmd{~}
3968 \def\bxjs@LetUnexpandableSpace#1{%
3969   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
3970     \bxjs@org@LetUnexpandableSpace#1%
3971   \fi}

```

```
3972 \fi
```

■ 共通命令の実装

```
3973 \newskip\jsKanjiSkip
3974 \newskip\jsXKanjiSkip
3975 \ifx\CJKecglue\@undefined
3976   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3977 \fi
3978 \let\autospacing\bxjs@enable@kanjiskip
3979 \let\noautospacing\bxjs@disable@kanjiskip
3980 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3981 \def\bxjs@apply@kanjiskip{%
3982   \jsKanjiSkip\@tempskipa
3983   \let\CJKglue\bxjs@CJKglue}
3984 \let\autoxspacing\bxjs@enable@xkanjiskip
3985 \let\noautoxspacing\bxjs@disable@xkanjiskip
3986 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3987 \def\bxjs@apply@xkanjiskip{%
3988   \jsXKanjiSkip\@tempskipa
3989   \let\CJKecglue\bxjs@CJKecglue}
```

\jachar のサブマクロの実装。

```
3990 \def\bxjs@jachar#1{%
3991   \CJKforced{#1}}
```

和欧文間空白の命令 \jathinspace の実装。

```
3992 \ifbxjs@jaspace@cmd
3993   \protected\def\jathinspace{\CJKecglue}
3994 \fi
```

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

```
3995 \ifx t\bxjs@enablejfam
3996   \ClassWarningNoLine\bxjs@clsname
3997   {You cannot use 'enablejfam=true', since the\MessageBreak
3998     CJK package does not support Japanese math}
3999 \fi
```

C.4 Xe_{La}TeX 用設定 : xeCJK + zxjatype

```
4000 \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。

```
4001 \RequirePackage{zxjatype}
4002 \PassOptionsToPackage{no-math}{fontspec}%!
4003 \PassOptionsToPackage{xetex}{graphicx}%!
4004 \PassOptionsToPackage{xetex}{graphics}%!
4005 \ifx\zxJaFamilyName\@undefined
4006   \ClassError\bxjs@clsname
```

```

4007 {xeCJK or zxjatype is too old}\@ehc
4008 \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 IPAex フォントを使用する。

```

4009 \bxjs@adjust@jafont{f}
4010 \let\bxjs@jafont@paren\@gobble
4011 \bxjs@resolve@jafont@paren\bxjs@tmpa
4012 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4013 \ifx\bxjs@tmpa\@empty
4014   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
4015   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
4016 \else
4017   \edef\bxjs@next{%
4018     \noexpand\RequirePackage[\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
4019   }\bxjs@next
4020 \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。

```

4021 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
4022   \ifbxjs@hyperref@enc
4023     \PassOptionsToPackage{unicode}{hyperref}
4024   \fi
4025 \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。

```

4026 \AtEndOfPackage{%
4027 \def\@gnewline #1{%
4028   \ifvmode \@nolnerr
4029   \else
4030     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null

```

```

4031 \nobreak \hskip-1sp\hskip1sp\relax
4032 \ignorespaces
4033 \fi}
4034 }

```

■ 共通命令の実装

```

4035 \newskip\jsKanjiSkip
4036 \newskip\jsXKanjiSkip
4037 \ifx\CJKecglue\@undefined
4038 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
4039 \fi
4040 \let\autospacing\bxjs@enable@kanjiskip
4041 \let\noautospacing\bxjs@disable@kanjiskip
4042 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
4043 \def\bxjs@apply@kanjiskip{%
4044 \jsKanjiSkip\@tempskipa
4045 \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
4046 \let\autoxspacing\bxjs@enable@xkanjiskip
4047 \let\noautoxspacing\bxjs@disable@xkanjiskip
4048 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
4049 \def\bxjs@apply@xkanjiskip{%
4050 \jsXKanjiSkip\@tempskipa
4051 \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。

```

4052 \ifx\mcfamily\@undefined
4053 \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
4054 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
4055 \fi

```

\jachar のサブマクロの実装。

```

4056 \def\bxjs@jachar#1{%
4057 \xeCJKDeclareCharClass{CJK}{`#1}\relax
4058 #1}

```

\jathinspace の実装。

```

4059 \ifbxjs@jaspace@cmd
4060 \protected\def\jathinspace{\CJKecglue}
4061 \fi

```

■ 和文数式ファミリー 和文数式ファミリーは既定で無効とする。すなわち enablejfam=true の場合にのみ @enablejfam を真にする。

```

4062 \ifx t\bxjs@enablejfam
4063 \@enablejfamtrue
4064 \fi

```

実際に和文用の数式ファミリーの設定を行う。

※ FIXME: 要検討。

```

4065 \if@enablejfam
4066   \xeCJKsetup{CJKmath=true}
4067 \fi

```

C.5 LuaTeX 用設定 : LuaTeX-ja

```

4068 \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。

```

4069 \let\zw\undefined
4070 \RequirePackage{luatexja}
4071 \edef\bxjs@next{%
4072   \noexpand\RequirePackage[scale=\jsScale]{luatexja-fontspec}[2015/08/26]%
4073 }\bxjs@next

```

■ 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、`luatexja-preset` パッケージの `ipaex` オプション (IPAex フォント使用) と等価な設定を用いる (`luatexja-preset` は読み込まない)。

```

4074 \bxjs@adjust@jafont{t}
4075 \ifx\bxjs@tmpa\bxjs@@noEmbed
4076   \def\bxjs@tmpa{noembed}
4077 \fi
4078 \let\bxjs@jafont@paren\@gobble
4079 \bxjs@resolve@jafont@paren\bxjs@tmpa
4080 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
4081 \ifx\bxjs@tmpa\@empty
4082   \defaultfontfeatures{ Kerning=Off }
4083   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
4084   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
4085 \else
4086   \edef\bxjs@next{%
4087     \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
4088   }\bxjs@next
4089 \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。

```

4090 \DeclareRobustCommand\rmfamily
4091   {\not@math@alphabet\rmfamily\mathrm
4092     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}

```

```

4093 \DeclareRobustCommand\sffamily
4094   {\not@math@alphabet\sffamily\mathsf
4095     \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
4096 \DeclareRobustCommand\ttfamily
4097   {\not@math@alphabet\ttfamily\mathtt
4098     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
4099 \AtBeginDocument{%
4100   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
4101   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
4102   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}%
4103 \bxjs@if@sfddefault{%
4104   \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

4105 % 次の3つは既定値の通り
4106 \ltjsetparameter{prebreakpenalty={`',10000}}
4107 \ltjsetparameter{postbreakpenalty={`",10000}}
4108 \ltjsetparameter{prebreakpenalty={`",10000}}
4109 \ltjsetparameter{jaxspmode={`!,1}}
4110 \ltjsetparameter{jaxspmode={`〒,2}}
4111 \ltjsetparameter{alxspmode={`+,3}}
4112 \ltjsetparameter{alxspmode={`%,3}}

```

■段落頭でのグルー挿入禁止 基本的に現状の `ltjs*` クラスの処理に合わせる。

※`\jsInhibitGlueAtParTop` は使わない。

`\ltjfakeparbegin` 現在の `LuaTEX-ja` で定義されているマクロで、段落中で段落冒頭用の処理を発動する。未定義である場合にに備えて同等のものを用意する。

```

4113 \ifx\ltjfakeparbegin\@undefined
4114   \protected\def\ltjfakeparbegin{%
4115     \ifhmode
4116       \relax\directlua{%
4117         luatexja.jfmglue.create_beginpar_node()}}
4118   \fi}
4119 \fi

```

`ltjs*` クラスの定義と同等になるようにパッチを当てる。

```

4120 \unless\ifnum\bxjs@everyparhook=\bxjs@everyparhook@none
4121 \begingroup
4122   \let%\@percentchar \def\@#1{[[\detokenize{#1}]]}
4123   \@gobble\if\def\bxjs@tmpa{\@{\everypar{}}\fi}}
4124   \ifnum\bxjs@everyparhook=\bxjs@everyparhook@compat
4125     \@gobble\if\def\bxjs@tmpa{\@{\everypar{\everyparhook}}\fi}}\fi
4126   \directlua{
4127     local function patchcmd(cs, code, from, to)
4128       tex.sprint(code:gsub(from:gsub("%W", "%\\%\\%0"), "%0"..to)
4129         :gsub("macro:", \@gdef..cs, 1):gsub("->", "{", 1).."}")
4130     end

```

```

4131 patchcmd(\@xsect, [[\meaning\xsect]],
4132 \@{\hskip-\@tempskipa}, \@ltjfakeparbegin)
4133 patchcmd(\@item, [[\meaning@item]],
4134 \bxjs@tmpa, \@ltjfakeparbegin)}
4135 \endgroup
4136 \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。

```

4137 \ifbxjs@hyperref@enc
4138 \PassOptionsToPackage{unicode}{hyperref}
4139 \bxjs@check@hyperref@unicode{true}
4140 \fi

```

■共通命令の実装

```

4141 \protected\def\autospacing{%
4142 \ltjsetparameter{autospacing=true}}
4143 \protected\def\noautospacing{%
4144 \ltjsetparameter{autospacing=false}}
4145 \protected\def\autoxspacing{%
4146 \ltjsetparameter{autoxspacing=true}}
4147 \protected\def\noautoxspacing{%
4148 \ltjsetparameter{autoxspacing=false}}
4149 \def\bxjs@apply@kanjiskip{%
4150 \ltjsetparameter{kanjiskip={\@tempskipa}}}
4151 \def\bxjs@apply@xkanjiskip{%
4152 \ltjsetparameter{xkanjiskip={\@tempskipa}}}

```

\jachar のサブマクロの実装。

```

4153 \def\bxjs@jachar#1{%
4154 \ltjjachar`#1\relax}

```

\jathinspace の実装。

```

4155 \ifbxjs@jaspace@cmd
4156 \protected\def\jathinspace{%
4157 \hskip\ltjgetparameter{xkanjiskip}\relax}
4158 \fi

```

■和文数式ファミリ LuaTeX-ja 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 @enablejfam は常に真になる。

```

4159 \ifx f\bxjs@enablejfam
4160 \ClassWarningNoLine\bxjs@clsname
4161 {You cannot use 'enablejfam=false', since the\MessageBreak
4162 LuaTeX-ja always provides Japanese math families}
4163 \fi

```

C.6 共通処理 (2)

```
4164 \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 4165 \ifx\DeclareFixJFMCJKTextFontCommand\undefined
4166 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
4167 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
4168 \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。

```
4169 \ifx\mathmc\undefined
4170 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
4171 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
4172 \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)。

```
4173 \ifbxjs@jaspace@cmd
4174 \bxjs@protected\def\bxjs@choice@jathinspace{%
4175 \relax\ifmmode \mskip\medmuskip
4176 \else \jathinspace\ignorespaces
4177 \fi}
4178 \jsAtEndOfClass{%
4179 \ifjsWithTeX \let\>\bxjs@choice@jathinspace
4180 \else \def\>{\protect\bxjs@choice@jathinspace}%
4181 \fi}
4182 \fi
```

■和文・和欧文間空白の初期値

```
4183 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
4184 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
4185 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
4186 \fi
```

以上で終わり。

```
4187 %</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。

```
4188 %<*modern>
4189 \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。

```
4190 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4191 \def\encodingdefault{T1}%
4192 \input{t1enc.def}%
4193 \fontencoding\encodingdefault\selectfont
4194 \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。

```
4195 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
4196 \renewcommand{\rmdefault}{lmr}
4197 \renewcommand{\sfdefault}{lms}
4198 \renewcommand{\ttdefault}{lmtt}
4199 \fi
```

大型演算子用の数式フォントの設定。

※ `amsfonts` パッケージと同等にする。

```
4200 \DeclareFontShape{OMX}{cmex}{m}{n}{%
4201   <-7.5>cmex7<7.5-8.5>cmex8%
4202   <8.5-9.5>cmex9<9.5->cmex10}{}%
4203 \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。
4204 \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。

```
4205 \ifx\@IncludeInRelease\@undefined
4206 \RequirePackage{fixltx2e}
4207 \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。

```
4208 \RequirePackage{bxjsjckcat}
```

D.4 完了

おしまい。

```
4209 %</modern>
```

付録 E 和文ドライバ : pandoc

Pandoc 用の何か。

standard ドライバの設定を引き継ぐ。

```
4210 %<*pandoc>
4211 \input{bxjsja-standard.def}
4212 \RequirePackage{bxjspandoc}
```

E.1 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。

```
4213 \@onlypreamble\bxjs@set@dupload@proc
4214 \def\bxjs@set@dupload@proc#1{%
4215   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
4216 \@onlypreamble\bxjs@set@dupload@proc@a
4217 \def\bxjs@set@dupload@proc@a#1{%
4218   \@onlypreamble#1\def#1##1}
```

`\@if@ptions` `\@if@ptions` の再定義。

```
4219 \@onlypreamble\bxjs@org@if@ptions
4220 \let\bxjs@org@if@ptions\@if@ptions
4221 \newif\ifbxjs@dlp
4222 \def\@if@ptions#1#2#3{%
4223   \bxjs@dlpfalse
4224   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
4225   \ifx\bxjs@tmpa\bxjs@tmpb
4226     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
4227       \bxjs@dlptrue \fi
4228   \fi
4229   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
4230   \else \expandafter\bxjs@org@if@ptions
4231   \fi {#1}{#2}{#3}}
4232 \AtBeginDocument{%
4233   \let\@if@ptions\bxjs@org@if@ptions}
4234 \@onlypreamble\bxjs@do@dupload@proc
4235 \def\bxjs@do@dupload@proc#1#2#3{%
4236   \csname bxjs@dlp/#2.#1\endcsname{#3}%
4237   \@firstoftwo}
```

E.2 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、とりあえず両パッケージを無効化しておく。

```
4238 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0
```

Polyglossia について。

```
4239 \pandocSkipLoadPackage{polyglossia}
4240 \bxjs@set@dupload@proc{polyglossia.sty}{%
4241   \ClassWarning\bxjs@clsname
4242     {Loading of polyglossia is blocked}}
4243 \ifx\setmainlanguage\@undefined
4244 \newcommand*\setmainlanguage[2][]{%
4245 \newcommand*\setotherlanguage[2][]{%
4246   \ifcat_#2_\else
4247     \expandafter\let\csname #2\endcsname\@empty
4248     \expandafter\let\csname end#2\endcsname\@empty
4249     \expandafter\let\csname text#2\endcsname\@firstofone
4250   \fi}
4251 \newcommand*\setotherlanguages[2][]{%
4252   \@for\bxjs@tmpa:={#2}\do{%
4253     \setotherlanguage{\bxjs@tmpa}}}
4254 \fi
4255 \else
```

Babel について。

```
4256 \pandocSkipLoadPackage{babel}
4257 \bxjs@set@dupload@proc{babel.sty}{%
4258   \ClassWarning\bxjs@clsname
4259     {Loading of babel is blocked}}
4260 \let\foreignlanguage\@secondoftwo
4261 \let\otherlanguage\@gobble
4262 \let\endotherlanguage\@empty
4263 \fi
```

E.3 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。

```
4264 \bxjs@set@dupload@proc{geometry.sty}{%
4265   \setpagelayout*{#1}}
```

E.4 CJKmainfont 変数

LuaTeX (+ LuaTeX-ja) の場合に CJKmainfont 変数が指定された場合は `\setmainjfont` の指定にまわす。

```
4266 \if 1\jsEngine
4267   \pandocSkipLoadPackage{xCJK}
4268   \providecommand*\setCJKmainfont{\setmainjfont}
4269 \fi
```

E.5 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。

```
4270 \AtBeginDocument{%
4271   \@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。

```
4272   \ifx\oldparagraph\undefined\else
4273     \@tempwattrue
4274   \fi
```

`\paragraph` が番号付きの場合は、マーク変更が必要である。

```
4275   \ifnum\c@secnumdepth>3
4276     \@tempwattrue
4277   \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。

```
4278   \if@tempwa\ifx\jsParagraphMark\bxjs@org@paragraph@mark
4279     \let\jsParagraphMark\empty
4280   \fi\fi}
```

E.6 全角空白文字

```
4281 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>\z@
4282   \catcode"3000=\active
4283   \begingroup \catcode`\!=7
4284   \protected\gdef!!!3000{\zwspace}
```

```

4285 \endgroup
4286 \else\ifx\DeclareUnicodeCharacter\@undefined\else
4287 \DeclareUnicodeCharacter{3000}{\bxjs@zsp@char}
4288 \bxjs@protected\def\bxjs@zsp@char{\zwspace}
4289 \fi\fi

```

■hyperref 対策 hyperref の unicode オプションの固定を行う。

```

4290 \if j\jsEngine
4291 \bxjs@fix@hyperref@unicode{false}
4292 \else
4293 \bxjs@fix@hyperref@unicode{true}
4294 \fi

```

E.7 完了

おしまい。

```
4295 %</pandoc>
```

和文ドライバ実装はここまで。

```
4296 %</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。

```
4297 %<*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

```

4298 %<*compat>
4299 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

4300 \let\bxac@engine=n
4301 \def\bxac@do#1#2{%
4302 \edef\bxac@tmpa{\string#1}%
4303 \edef\bxac@tmpb{\meaning#1}%
4304 \ifx\bxac@tmpa\bxac@tmpb #2\fi}

```

```

4305 \brac@do\XeTeXversion{\let\bxac@engine=x}
4306 \brac@do\luatexversion{\let\bxac@engine=l}

```

`\brac@delayed@if@bxjs` もし BXJS クラスの読込中でこのパッケージが読み込まれているならば、BXJS のクラスの
 終わりまで実行を遅延する。

```

4307 \ifx\jsAtEndOfClass\@undefined
4308   \let\brac@delayed@if@bxjs\@firstofone
4309 \else \let\brac@delayed@if@bxjs\jsAtEndOfClass
4310 \fi

```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```

\RevokeOldLuaTeXBehavior 4311 \newif\ifbxac@in@old@behavior
4312 \let\ImposeOldLuaTeXBehavior\relax
4313 \let\RevokeOldLuaTeXBehavior\relax

```

G.2 XeTeX 部分

```

4314 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。
4315 \@onlypreamble\brac@adjust@charclass
4316 \brac@delayed@if@bxjs{%
4317   \@ifpackageloaded{xeCJK}{%}{%else
4318     \ifx\XeTeX@alloc@intercharclass\@undefined\else
4319       \ifnum\XeTeX@alloc@intercharclass=\z@
4320         \PackageInfo\bxac@pkgname
4321         {Setting up interchar class for CJK...\@gobble}%
4322         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
4323           \XeTeX@alloc@intercharclass=3
4324         }{%else
4325           \PackageWarning\bxac@pkgname
4326             {Cannot find file 'load-unicode-xetex-classes.tex'%
4327             \@gobble}%
4328         }%
4329       \fi\fi
4330     \ifnum\XeTeXcharclass"3041=\z@
4331       \PackageInfo\bxac@pkgname
4332       {Adjusting interchar class for CJK...\@gobble}%
4333       \@for\bxac@x={%
4334         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
4335         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
4336         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
4337         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
4338         31FF%
4339       }\do{\XeTeXcharclass"\bxac@x=\@ne}%
4340     \fi
4341   }%
4342 }

```

以上。

4343 \fi

G.3 LuaTeX 部分

4344 \ifx l\bxac@engine

ムニヤムニヤ。

4345 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85

4346 \chardef\pdftexversion=200

4347 \def\pdftexrevision{0}

4348 \let\pdftexbanner\luatexbanner

4349 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

\RevokeOldLuaTeXBehavior 4350 \begingroup\expandafter\expandafter\expandafter\endgroup

4351 \expandafter\ifx\cename outputmode\endcsname\relax\else

4352 \def\bxac@ob@list{%

4353 \do{\let}\pdfoutput{\outputmode}%

4354 \do{\let}\pdfpagewidth{\pagewidth}%

4355 \do{\let}\pdfpageheight{\pageheight}%

4356 \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%

4357 \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}

4358 \def\bxac@ob@do#1#2{\begingroup

4359 \expandafter\bxac@ob@do@a\cename bxac@\string#2\endcsname{#1}#2}

4360 \def\bxac@ob@do@a#1#2#3#4{\endgroup

4361 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax

4362 \else \let#3#1\relax \let#1\@undefined

4363 \fi}

4364 \protected\def\ImposeOldLuaTeXBehavior{%

4365 \unless\ifbxac@in@old@behavior

4366 \bxac@in@old@behaviortrue

4367 \let\do\bxac@ob@do \bxac@ob@list

4368 \fi}

4369 \protected\def\RevokeOldLuaTeXBehavior{%

4370 \ifbxac@in@old@behavior

4371 \bxac@in@old@behaviorfalse

4372 \let\do\bxac@ob@do \bxac@ob@list

4373 \fi}

4374 \fi

漢字および完成形ハングルのカテゴリコードのムニヤムニヤ。

4375 \ifnum\luatexversion>64 \directlua{

4376 local function range(cs, ce, cc, ff)

4377 if ff or not tex.getcatcode(cs) == cc then

4378 local setcc = tex.setcatcode

4379 for c = cs, ce do setcc(c, cc) end

4380 end

4381 end

4382 range(0x3400, 0x4DB5, 11, false)

4383 range(0x4DB5, 0x4DBF, 11, true)

```

4384     range(0x4E00, 0x9FCC, 11, false)
4385     range(0x9FCD, 0x9FFF, 11, true)
4386     range(0xAC00, 0xD7A3, 11, false)
4387     range(0x20000, 0x2A6D6, 11, false)
4388     range(0x2A6D7, 0x2A6FF, 11, true)
4389     range(0x2A700, 0x2B734, 11, false)
4390     range(0x2B735, 0x2B73F, 11, true)
4391     range(0x2B740, 0x2B81D, 11, false)
4392     range(0x2B81E, 0x2B81F, 11, true)
4393     range(0x2B820, 0x2CEA1, 11, false)
4394     range(0x2CEA2, 0x2FFFD, 11, true)
4395 } \fi

```

以上。

```
4396 \fi
```

G.4 完了

おしまい。

```
4397 %</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

```

4398 %<*cjkcat>
4399 \def\bxjx@pkgname{bxjscjkat}
4400 \newcount\bxjx@canta

\bxjx@engine エンジンの種別。

4401 \let\bxjx@engine=n
4402 \def\bxjx@do#1#2{%
4403   \edef\bxjx@tmpa{\string#1}%
4404   \edef\bxjx@tmpb{\meaning#1}%
4405   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
4406 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
4407 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
4408 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
4409 \bxjx@do\pdftexversion{\let\bxjx@engine=p}
4410 \bxjx@do\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。

```

4411 \def\bxjx@do#1#2{%
4412   \if#1\bxjx@engine

```



```

4413 \ifpackageloaded{#2}{}{%else
4414 \PackageError{bxjx@pkgname
4415 {Package '#2' must be loaded}%
4416 {Package loading is aborted.\MessageBreak\@ehc}%
4417 \endinput}
4418 \fi}
4419 \bxjx@do{p}{bxcjkjatype}
4420 \bxjx@do{x}{xeCJK}
4421 \bxjx@do{l}{luatexja}

```

古い \LaTeX の場合、 \TextOrMath は `fixltx2e` パッケージで提供される。

```

4422 \ifx\TextOrMath\undefined
4423 \RequirePackage{fixltx2e}
4424 \fi

```

H.2 和文カテゴリコードの設定

up\LaTeX の場合、和文カテゴリコードの設定を Lua\TeX-j a と（ほぼ）等価なものに変更する。

※ Lua\TeX-j a との相違点：A830、A960、1B000。

```

4425 \if u\bxjx@engine
4426 \@for\bxjx@x:={%
4427 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
4428 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
4429 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
4430 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
4431 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
4432 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
4433 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
4434 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
4435 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
4436 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
4437 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
4438 10000,10080,10100,10140,10190,101D0,10280,102A0,%
4439 102E0,10300,10330,10350,10380,103A0,10400,10450,%
4440 10480,10500,10530,10600,10800,10840,10860,10880,%
4441 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
4442 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
4443 11000,11080,110D0,11100,11150,11180,111E0,11200,%
4444 11280,112B0,11300,11480,11580,11600,11680,11700,%
4445 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
4446 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
4447 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
4448 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
4449 1F700,1F780,1F800,1F900,E0000,F0000,100000%
4450 }\do{\kcatcode"\bxjx@x=15 }
4451 \fi

```

H.3 ギリシャ・キリル文字の扱い

※ここで「ギリシャ・キリル文字」は Unicode と JIS X 0213 に共通して含まれるもののみを指すことにする。

`\bxjx@grkcyr@list` 対象のギリシャ・キリル文字に関するデータ。

```
4452 \def\bxjx@grkcyr@list{%
4453 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
4454 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
4455 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
4456 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
4457 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
4458 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
4459 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
4460 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
4461 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
4462 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
4463 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
4464 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
4465 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
4466 \do{039E}{LGR}{\textXi}{\Xi}%          % GR. C. L. XI
4467 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
4468 \do{03A0}{LGR}{\textPi}{\Pi}%          % GR. C. L. PI
4469 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
4470 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
4471 \do{03A4}{LGR}{\textTau}{T}%           % GR. C. L. TAU
4472 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
4473 \do{03A6}{LGR}{\textPhi}{\Phi}%        % GR. C. L. PHI
4474 \do{03A7}{LGR}{\textChi}{X}%           % GR. C. L. CHI
4475 \do{03A8}{LGR}{\textPsi}{\Psi}%        % GR. C. L. PSI
4476 \do{03A9}{LGR}{\textOmega}{\Omega}%    % GR. C. L. OMEGA
4477 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
4478 \do{03B2}{LGR}{\textbeta}{\beta}%      % GR. S. L. BETA
4479 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA
4480 \do{03B4}{LGR}{\textdelta}{\delta}%     % GR. S. L. DELTA
4481 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
4482 \do{03B6}{LGR}{\textzeta}{\zeta}%       % GR. S. L. ZETA
4483 \do{03B7}{LGR}{\texteta}{\eta}%          % GR. S. L. ETA
4484 \do{03B8}{LGR}{\texttheta}{\theta}%     % GR. S. L. THETA
4485 \do{03B9}{LGR}{\textiota}{\iota}%       % GR. S. L. IOTA
4486 \do{03BA}{LGR}{\textkappa}{\kappa}%     % GR. S. L. KAPPA
4487 \do{03BB}{LGR}{\textlambda}{\lambda}%   % GR. S. L. LAMDA
4488 \do{03BC}{LGR}{\textmu}{\mu}%          % GR. S. L. MU
4489 \do{03BD}{LGR}{\textnu}{\nu}%          % GR. S. L. NU
4490 \do{03BE}{LGR}{\textxi}{\xi}%          % GR. S. L. XI
4491 \do{03BF}{LGR}{\textomicron}{o}%         % GR. S. L. OMICRON
4492 \do{03C0}{LGR}{\textpi}{\pi}%          % GR. S. L. PI
```

4493 \do{03C1}{LGR}{\textrho}{\rho}% % GR. S. L. RHO
 4494 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
 4495 \do{03C3}{LGR}{\textsigma}{\sigma}% % GR. S. L. SIGMA
 4496 \do{03C4}{LGR}{\texttau}{\tau}% % GR. S. L. TAU
 4497 \do{03C5}{LGR}{\textupsilon}{\upsilon}% % GR. S. L. UPSILON
 4498 \do{03C6}{LGR}{\textphi}{\phi}% % GR. S. L. PHI
 4499 \do{03C7}{LGR}{\textchi}{\chi}% % GR. S. L. CHI
 4500 \do{03C8}{LGR}{\textpsi}{\psi}% % GR. S. L. PSI
 4501 \do{03C9}{LGR}{\textomega}{\omega}% % GR. S. L. OMEGA
 4502 \do{0401}{T2A}{\CYRYO}{}% % CY. C. L. IO
 4503 \do{0410}{T2A}{\CYRA}{}% % CY. C. L. A
 4504 \do{0411}{T2A}{\CYRB}{}% % CY. C. L. BE
 4505 \do{0412}{T2A}{\CYRV}{}% % CY. C. L. VE
 4506 \do{0413}{T2A}{\CYRG}{}% % CY. C. L. GHE
 4507 \do{0414}{T2A}{\CYRD}{}% % CY. C. L. DE
 4508 \do{0415}{T2A}{\CYRE}{}% % CY. C. L. IE
 4509 \do{0416}{T2A}{\CYRZH}{}% % CY. C. L. ZHE
 4510 \do{0417}{T2A}{\CYRZ}{}% % CY. C. L. ZE
 4511 \do{0418}{T2A}{\CYRI}{}% % CY. C. L. I
 4512 \do{0419}{T2A}{\CYRISHRT}{}% % CY. C. L. SHORT I
 4513 \do{041A}{T2A}{\CYRK}{}% % CY. C. L. KA
 4514 \do{041B}{T2A}{\CYRL}{}% % CY. C. L. EL
 4515 \do{041C}{T2A}{\CYRM}{}% % CY. C. L. EM
 4516 \do{041D}{T2A}{\CYRN}{}% % CY. C. L. EN
 4517 \do{041E}{T2A}{\CYRO}{}% % CY. C. L. O
 4518 \do{041F}{T2A}{\CYRP}{}% % CY. C. L. PE
 4519 \do{0420}{T2A}{\CYRR}{}% % CY. C. L. ER
 4520 \do{0421}{T2A}{\CYRS}{}% % CY. C. L. ES
 4521 \do{0422}{T2A}{\CYRT}{}% % CY. C. L. TE
 4522 \do{0423}{T2A}{\CYRU}{}% % CY. C. L. U
 4523 \do{0424}{T2A}{\CYRF}{}% % CY. C. L. EF
 4524 \do{0425}{T2A}{\CYRH}{}% % CY. C. L. HA
 4525 \do{0426}{T2A}{\CYRC}{}% % CY. C. L. TSE
 4526 \do{0427}{T2A}{\CYRCH}{}% % CY. C. L. CHE
 4527 \do{0428}{T2A}{\CYRSH}{}% % CY. C. L. SHA
 4528 \do{0429}{T2A}{\CYRSHCH}{}% % CY. C. L. SHCHA
 4529 \do{042A}{T2A}{\CYRHRDSN}{}% % CY. C. L. HARD SIGN
 4530 \do{042B}{T2A}{\CYRERY}{}% % CY. C. L. YERU
 4531 \do{042C}{T2A}{\CYRSFTSN}{}% % CY. C. L. SOFT SIGN
 4532 \do{042D}{T2A}{\CYREREV}{}% % CY. C. L. E
 4533 \do{042E}{T2A}{\CYRYU}{}% % CY. C. L. YU
 4534 \do{042F}{T2A}{\CYRYA}{}% % CY. C. L. YA
 4535 \do{0430}{T2A}{\cyra}{}% % CY. S. L. A
 4536 \do{0431}{T2A}{\cyrb}{}% % CY. S. L. BE
 4537 \do{0432}{T2A}{\cyrv}{}% % CY. S. L. VE
 4538 \do{0433}{T2A}{\cyrg}{}% % CY. S. L. GHE
 4539 \do{0434}{T2A}{\cyrd}{}% % CY. S. L. DE
 4540 \do{0435}{T2A}{\cyre}{}% % CY. S. L. IE
 4541 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE

```

4542 \do{0437}{T2A}{\cyrz}{}% % CY. S. L. ZE
4543 \do{0438}{T2A}{\cyri}{}% % CY. S. L. I
4544 \do{0439}{T2A}{\cyrishrt}{}% % CY. S. L. SHORT I
4545 \do{043A}{T2A}{\cyrk}{}% % CY. S. L. KA
4546 \do{043B}{T2A}{\cyrl}{}% % CY. S. L. EL
4547 \do{043C}{T2A}{\cyrm}{}% % CY. S. L. EM
4548 \do{043D}{T2A}{\cyrn}{}% % CY. S. L. EN
4549 \do{043E}{T2A}{\cyro}{}% % CY. S. L. O
4550 \do{043F}{T2A}{\cyrp}{}% % CY. S. L. PE
4551 \do{0440}{T2A}{\cyrr}{}% % CY. S. L. ER
4552 \do{0441}{T2A}{\cyrs}{}% % CY. S. L. ES
4553 \do{0442}{T2A}{\cyrt}{}% % CY. S. L. TE
4554 \do{0443}{T2A}{\cyru}{}% % CY. S. L. U
4555 \do{0444}{T2A}{\cyr f}{}% % CY. S. L. EF
4556 \do{0445}{T2A}{\cyrh}{}% % CY. S. L. HA
4557 \do{0446}{T2A}{\cyr c}{}% % CY. S. L. TSE
4558 \do{0447}{T2A}{\cyrch}{}% % CY. S. L. CHE
4559 \do{0448}{T2A}{\cyrsh}{}% % CY. S. L. SHA
4560 \do{0449}{T2A}{\cyrshch}{}% % CY. S. L. SHCHA
4561 \do{044A}{T2A}{\cyrhrdsn}{}% % CY. S. L. HARD SIGN
4562 \do{044B}{T2A}{\cyrery}{}% % CY. S. L. YERU
4563 \do{044C}{T2A}{\cyr sftsn}{}% % CY. S. L. SOFT SIGN
4564 \do{044D}{T2A}{\cyrerev}{}% % CY. S. L. E
4565 \do{044E}{T2A}{\cyryu}{}% % CY. S. L. YU
4566 \do{044F}{T2A}{\cyrya}{}% % CY. S. L. YA
4567 \do{0451}{T2A}{\cyryo}{}% % CY. S. L. IO
4568 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4569 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4570 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4571 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN
4572 \do{00B4}{TS1}{\textasciicute}{}% % ACUTE ACCENT
4573 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4574 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4575 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN
4576 }
4577 \providecommand*{\mathdegree}{\circ}

```

\ifbxjx@gcc@cjk [スイッチ] ギリシャ・キリル文字を和文扱いにするか。

```
4578 \newif\ifbxjx@gcc@cjk
```

\greekasCJK ギリシャ・キリル文字を和文扱いにする。

\nogreekasCJK ギリシャ・キリル文字を欧文扱いにする。

```

4579 \newcommand*\greekasCJK{%
4580   \bxjx@gcc@cjktrue}
4581 \newcommand*\nogreekasCJK{%
4582   \bxjx@gcc@cjkfalse}

```

\bx@fake@grk \bx@fake@grk{(出力文字)}{(基準文字)} :

```

4583 \def\bxjx@do#1\relax{%
4584   \def\bxjx@fake@grk##1##2{%
4585     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4586   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4587     \ifx\\##1\\%
4588       \bxjx@cmta##4\divide\bxjx@cmta\@cclvi
4589       \multiply\bxjx@cmta\@cclvi \advance\bxjx@cmta`##3\relax
4590       \mathchar\bxjx@cmta
4591     \else ##3\fi}
4592 }\expandafter\bxjx@do\string\mathchar\relax

```

■pdfLaTeX・upLaTeX の場合

```

4593 \ifnum0\if p\bxjx@engine1\fi\if u\bxjx@engine1\fi>0

```

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。

```

4594 \@ifpackageloaded{inputenc}{\else
4595   \RequirePackage[utf8]{inputenc}}
4596 \def\bxjx@tmpa{utf8}
4597 \ifx\bxjx@tmpa\inputencdoingname
4598   \PackageWarningNoLine\bxjx@pkgname
4599     {Input encoding changed to utf8}%
4600   \inputencoding{utf8}%
4601 \fi

```

upLaTeX の場合は当該の文字を含むブロックをの和文カテゴリコードを変更する。

```

4602 \if u\bxjx@engine
4603   \kcatcode"0370=15
4604   \kcatcode"0400=15
4605   \kcatcode"0500=15
4606 \fi

```

各文字について \DeclareUnicodeCharacter を実行する。

```

4607 \def\do#1{%
4608   \@tempcnta="#1\relax
4609   \@tempcntb=\@tempcnta \divide\@tempcntb256
4610   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4611   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4612 \def\bxjx@do@a#1#2#3#4#5{%
4613   \ifx\\#5\\%
4614     \def\bxjx@tmpa{\@inmathwarn#4}%
4615   \else\ifcat A\noexpand#5%
4616     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4617       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4618   \else \def\bxjx@tmpa{#5}%
4619   \fi\fi
4620   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4621   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}
4622 \if u\bxjx@engine
4623 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}

```

```

4624 \def\bxjx@do@b#1#2#3#4#5{%
4625   \kchardef#1=\@tempcnta
4626   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4627   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4628 \else\if p\bxjx@engine
4629 \def\bxjx@do@b#1#2#3#4#5{%
4630   \mathchardef#1=\@tempcnta
4631   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{UTF{#2}}{#3}{#4}}%
4632   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4633 \fi\fi
4634 \bxjx@grkcyr@list
4635 \let\bxjx@do@a\undefined
4636 \let\bxjx@do@b\undefined

```

`\bxjx@DeclareUnicodeCharacter` `\bxjx@DeclareUnicodeCharacter` を改変して、ギリシャ・キリル文字の場合に再定義を抑制したもの。

```

4637 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4638 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4639 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4640 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4641   \count@=#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
4642   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
4643     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4644   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4645     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4646   \else
4647     \wlog{ \space\space skipped defining Unicode char U+#1}%
4648   \fi\fi}

```

`\bxjx@ja@or@not` `\bxjx@ja@or@not`

```

4649 \def\bxjx@ja@or@not#1#2#3{%
  \greekasCJK の場合は、無条件に和文用コードを実行する。

```

```

4650   \ifbxjx@gcc@CJK #1%

```

`\nogreekasCJK` の場合は、エンコーディングを固定して欧文用のコードを実行するが、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）和文用コードを使う。

```

4651   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4652   \else \UseTextSymbol{#2}{#3}%
4653   \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。

```

4654 \begingroup
4655 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4656 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4657   \noexpand\bxjx@swap@DUC@cmd
4658   \the\toks@
4659   \noexpand\bxjx@swap@DUC@cmd}}

```

```

4660 \endgroup\next
4661 \def\bxjx@swap@DUC@cmd{%
4662   \let\bxjx@tmpa\DeclareUnicodeCharacter
4663   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4664   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■ Xe_{La}TeX・Lua_{La}TeX の場合

```

4665 \else\ifnum0\if x\bxjx@engine1\fi\if 1\bxjx@engine1\fi>0

```

各文字について、math active を設定する。

```

4666 \def\do#1{%
4667   \bxjx@cmta="#1\relax
4668   \begingroup
4669     \lccode`~=\bxjx@cmta
4670     \lowercase{\endgroup
4671       \bxjx@do@a{~}}{#1}}
4672 \def\bxjx@do@a#1#2#3#4#5{%
4673   \ifx\#5\\\let\bxjx@tmpa\relax
4674   \else\ifcat A\noexpand#5%
4675     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4676       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4677   \else \def\bxjx@tmpa{#5}%
4678   \fi\fi
4679   \ifx\bxjx@tmpa\relax\else
4680     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4681   \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、こちらの設定を有効にする。

```

4682 \mathchardef\bxjx@tmpa="119
4683 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4684 \let\bxjx@do@a\undefined

```

Lua_{La}TeX における \(\text{no}\)greekasCJK の定義。jacharrange の設定を変更する。

```

4685 \if 1\bxjx@engine
4686   \protected\def\greekasCJK{%
4687     \bxjx@gcc@cjktrue
4688     \ltjsetparameter{jacharrange={+2, +8}}
4689   \protected\def\nogreekasCJK{%
4690     \bxjx@gcc@cjkfalse
4691     \ltjsetparameter{jacharrange={-2, -8}}
4692   \fi

```

Xe_{La}TeX における \(\text{no}\)greekasCJK の定義。

```

4693 \if x\bxjx@engine
4694   \protected\def\greekasCJK{%
4695     \bxjx@gcc@cjktrue
4696     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%

```

```

4697 \bxjx@grkcyr@list}
4698 \protected\def\nogreekasCJK{%
4699 \bxjx@gcc@cjkgfalse
4700 \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4701 \bxjx@grkcyr@list}
4702 \fi

```

以上。

```

4703 \fi\fi

```

H.4 初期設定

ギリシャ・キリル文字を欧文扱いにする。

```

4704 \nogreekasCJK

```

H.5 完了

おしまい。

```

4705 %</cjkgcat>

```

付録 I 補助パッケージ : bxjspan doc

Pandoc の L^AT_EX 用標準テンプレートをより幸せに使うための設定。BXJS クラスの pandoc ドライバのコードの中の、“汎用的”に使える部分を切り出したもの。つまり現在の pandoc ドライバはこのパッケージを読みこむ。

※テンプレートの T_EX コードより前に読み込む必要があるため、専ら文書クラス内での読込に限られる。

I.1 準備

```

4706 %<*ancpandoc>
4707 %% このファイルは日本語文字を含みます。
4708 \def\bxjsp@pkgname{bxjscjkgcat}

\bxjsp@engine エンジンの種別。

4709 \let\bxjsp@engine=n
4710 \@onlypreamble\bxjsp@do
4711 \def\bxjsp@do#1#2{%
4712 \edef\bxjsp@tmpa{\string#1}%
4713 \edef\bxjsp@tmpb{\meaning#1}%
4714 \ifx\bxjsp@tmpa\bxjsp@tmpb #2\fi}
4715 \bxjsp@do\kanjiskip{\let\bxjsp@engine=j}
4716 \bxjsp@do\XeTeXversion{\let\bxjsp@engine=x}
4717 \bxjsp@do\pdftexversion{\let\bxjsp@engine=p}
4718 \bxjsp@do\luatexversion{\let\bxjsp@engine=l}

```


1.2 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。

```
4719 \newcommand*\pandocSkipLoadFile[1]{%
4720   \expandafter\bxjsp@skip@load@file@a\csname ver@#1\endcsname{#1}}
4721 \def\bxjsp@skip@load@file@a#1#2{%
4722   \ifx#1\relax
4723     \def#1{2001/01/01}%
4724     \PackageInfo\bxjsp@pkgname
4725       {File '#2' marked as loaded\@gobble}%
4726   \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。

```
4727 \newcommand*\pandocSkipLoadPackage[1]{%
4728   \pandocSkipLoadFile{#1.sty}}
```

1.3 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。

```
4729 \ifx\@IncludeInRelease\@undefined\else
4730   \pandocSkipLoadPackage{fixltx2e}
4731 \fi
```

1.4 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。)

```
4732 \if j\bxjsp@engine
4733   \pandocSkipLoadPackage{cmap}
4734 \fi
```

1.5 microtype パッケージ

警告が多すぎなので消す。

```
4735 \if j\bxjsp@engine \else
4736   \PassOptionsToPackage{verbose=silent}{microtype}
4737 \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。

```
4738 \if j\bxjsp@engine
4739   \pandocSkipLoadPackage{microtype}
4740   \newcommand*\UseMicrotypeSet[2][]{ }
4741 \fi
```

I.6 Unicode 文字変換対策

Pandoc で L^AT_EX 形式に書き出す場合は、元データ中の一部の Unicode 文字を「L^AT_EX の表記」に置き換える。その中には日本語文書で問題になるものが含まれる。

…→\ldots{ } ‘→` ’→' “→` ”→' ’

日本語 L^AT_EX では「L^AT_EX の表記」は欧文扱い、Unicode 文字は和文扱いとして使い分ける習慣があるので、このような置換が行われるのは好ましくない。

これらの置換のうち、後の 4 つは Pandoc の `--no-tex-ligatures` オプションを指定すれば抑止できるが、「…」の置換を抑止する機能はないようである。そこで、「\ldots{ } を『…』に戻す」という処置を行う。

`\pandocLdots` Pandoc 用の `\ldots` の実装。非数式でありかつ後続が `{ }` の場合は代わりに `…` を実行する。

```
4742 \DeclareRobustCommand{\pandocLdots}{%
4743   \relax\ifmmode \expandafter\bxjsp@org@ldots
4744   \else \expandafter\bxjsp@ldots@a
4745   \fi}
4746 \def\bxjsp@ja@ellipsis{…}
4747 \let\bxjsp@org@ldots\ldots
4748 \def\bxjsp@ldots@a{%
4749   \futurelet\bxjsp@tok\bxjsp@ldots@b}
4750 \def\bxjsp@ldots@b{%
4751   \ifx\bxjsp@tok\bgroup \expandafter\bxjsp@ldots@c
4752   \else \expandafter\bxjsp@org@ldots
4753   \fi}
4754 \def\bxjsp@ldots@c{%
4755   \afterassignment\bxjsp@ldots@d \let\bxjsp@tok=}
4756 \def\bxjsp@ldots@d{%
4757   \futurelet\bxjsp@tok\bxjsp@ldots@e}
4758 \def\bxjsp@ldots@e{%
4759   \ifx\bxjsp@tok\egroup \expandafter\bxjsp@ldots@f
4760   \else \expandafter\bxjsp@ldots@g
4761   \fi}
4762 \def\bxjsp@ldots@f{%
4763   \bxjsp@ja@ellipsis \let\bxjsp@tok=}
4764 \def\bxjsp@ldots@g{%
4765   \expandafter\bxjsp@org@ldots\expandafter{\romannumeral-`} }
```

`\ldots` の実装を置き換える。

```
4766 \AtBeginDocument{%  
4767   \let\bxjsp@org\ldots\ldots  
4768   \let\ldots\pandocLdots}
```

1.7 PandoLa モジュール

インストール済であれば読み込む。

```
4769 \IfFileExists{bxpandola.sty}{%  
4770   \RequirePackage{bxpandola}\relax  
4771   \PackageInfo\bxjsp@pkgname  
4772   {PandoLa module is loaded\@gobble}  
4773 }{}
```

1.8 完了

おしまい。

```
4774 %</ancpandoc>
```

補助パッケージ実装はここまで。

```
4775 %</anc>
```