

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”)

v1.5d [2017/07/07]

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

目次

1	はじめに	3
2	オプション	7
3	和文フォントの変更	33
4	フォントサイズ	34
5	レイアウト	39
5.1	ページレイアウト	40
6	改ページ (日本語 TeX 開発コミュニティ版のみ)	47
7	ページスタイル	49
8	文書のマークアップ	52
8.1	表題	52
8.2	章・節	57
8.3	リスト環境	69
8.4	パラメータの設定	76
8.5	フロート	77
8.6	キャプション	79
9	フォントコマンド	80

10	相互参照	83
10.1	目次の類	83
10.2	参考文献	88
10.3	索引	90
10.4	脚注	91
11	段落の頭へのグルー挿入禁止	93
12	いろいろなロゴ	97
13	amsmath との衝突の回避	97
14	初期設定	97
付録 A	和文ドライバの仕様 	101
付録 B	和文ドライバ : minimal 	102
B.1	補助マクロ	102
B.2	(u)pTeX 用の設定	104
B.3	pdfTeX 用の処理	108
B.4	X _Y TeX 用の処理	109
B.5	後処理 (エンジン共通)	109
付録 C	和文ドライバ : standard 	111
C.1	共通処理 (1)	112
C.2	pTeX 用設定	118
C.3	pdfTeX 用設定 : CJK + bxcjkatype	122
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	124
C.5	LuaTeX 用設定 : LuaTeX-ja	126
C.6	共通処理 (2)	128
付録 D	和文ドライバ : modern 	129
D.1	フォント設定	129
D.2	fixltx2e 読込	129
D.3	和文カテゴリコード	130
D.4	完了	130
付録 E	和文ドライバ : pandoc 	130
E.1	dupload システム	130
E.2	lang 変数	131
E.3	geometry 変数	132
E.4	CJKmainfont 変数	132
E.5	fixltx2e パッケージ	132

E.6	cmap パッケージ	132
E.7	microtype パッケージ	133
E.8	完了	133
付録 F	補助パッケージ一覧	133
付録 G	補助パッケージ: bxjscompat	133
G.1	準備	133
G.2	X _g TeX 部分	134
G.3	LuaTeX 部分	135
G.4	完了	136
付録 H	補助パッケージ: bxjscjkat	136
H.1	準備	136
H.2	和文カテゴリコードの設定	137
H.3	ギリシャ・キリル文字の扱い	138
H.4	初期設定	144
H.5	完了	144

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_ATeX 2_ε 新ドキュメントクラス」に改変を加えたものである。本ドキュメントクラスに関する説明は全てこの形式の枠の中に記す。枠の外にあるものは原版著者による原版に対する解説である。

これは L^AT_EX 3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいて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 up_ATeX 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。

■ BXJS クラス特有の設定

```

1 %<*cls>
2 %% このファイルは日本語文字を含みます。
長さ値の指定で式を利用可能にするため calc を読み込む。
3 \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。
4 \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。
5 \IfFileExists{bxjscompat.sty}{%
6   \let\jsAtEndOfClass@gobble
7   \RequirePackage{bxjscompat}%
8 }{}

```

`\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`。

```

9 \let\jsArticle=a
10 \let\jsBook=b
11 \let\jsReport=r
12 \let\jsSlide=s
13 %<article>\let\jsDocClass\jsArticle
14 %<article>\def\bxjs@clsname{bxjsarticle}
15 %<book>\let\jsDocClass\jsBook
16 %<book>\def\bxjs@clsname{bxjsbook}
17 %<report>\let\jsDocClass\jsReport
18 %<report>\def\bxjs@clsname{bxjsreport}
19 %<slide>\let\jsDocClass\jsSlide
20 %<slide>\def\bxjs@clsname{bxjsslide}

```

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

```

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

```

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

```

32 \newif\ifjsWithupTeX
33 \ifx\ucs@undefined\else \ifnum\ucs"3000="3000

```

```

34 \jsWithupTeXtrue
35 \fi\fi
36 \let\if@jsc@uplatex\ifjsWithupTeX

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

\ifjsWitheTeX [スイッチ] エンジンが  $\epsilon$ -TeX 拡張をもつか。
39 \newif\ifjsWitheTeX
40 \bxjs@test@engine\TeXversion{\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。
41 \let\bxjs@tmpa\relax
42 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
43 \ifx O\jsEngine \def\bxjs@tmpa{Omega}\fi
44 \ifx\bxjs@tmpa\relax \expandafter\@gobble
45 \else
46 \ClassError\bxjs@clsname
47 {The engine in use (\bxjs@tmpa) is not supported}
48 {It's a fatal error. I'll quit right now.}
49 \expandafter\@firstofone
50 \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。
51 \ifjsWitheTeX \let\bxjs@protected\protected
52 \else \let\bxjs@protected\@empty
53 \fi

\bxjs@robust@def 無引数の頑強な命令を定義する。
54 \ifjsWitheTeX
55 \def\bxjs@robust@def{\protected\def}
56 \else
57 \def\bxjs@robust@def{\DeclareRobustCommand*}
58 \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。
59 \newif\ifjsInPdfMode
60 \@nameuse{ImposeOldLuaTeXBehavior}
61 \let\bxjs@tmpa\PackageWarningNoLine
62 \let\PackageWarningNoLine\PackageInfo % suppress warning
63 \RequirePackage{ifpdf}
64 \let\PackageWarningNoLine\bxjs@tmpa
65 \@nameuse{RevokeOldLuaTeXBehavior}
66 \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。

```
67 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
68   #1\expandafter\@firstoftwo
69   \else\expandafter\@secondoftwo
70   \fi}
```

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

```
71 \def\bxjs@cslet#1{%
72   \expandafter\let\csname#1\endcsname}
```

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

```
73 \def\bxjs@csletcs#1#2{%
74   \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。

```
75 \def\bxjs@catopt#1#2{%
76   #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。)

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

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

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

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

```
90 \bxjs@robust\def\jsInhibitGlue{%
91   \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。

```
92 \if@compatibility
```

```

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

```
102 \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。

```
103 \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。

```
104 %<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。

```
105 %<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。

```
106 %<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。

```
107 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

B 判は JIS と ISO で定義が異なります。JIS では B0 判の面積が 1.5m^2 ですが，ISO では B1 判の辺の長さが A0 判と A1 判の辺の長さの幾何平均です。したがって ISO の B0 判

は 1000 mm × 1414 mm です。このため、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ の `b5paper` は 250 mm × 176 mm ですが、 $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ の `b5paper` は 257 mm × 182 mm になっています。ここでは $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ にならって JIS に従いました。

デフォルトは `a4paper` です。

`b5var` (B5 変形, 182mm×230mm), `a4var` (A4 変形, 210mm×283mm) 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。

```
108 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
109 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
110 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
111 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
112 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
113 \DeclareOption{b4paper}{\bxjs@setpaper{{257trueemm}{364trueemm}}}
114 \DeclareOption{b5paper}{\bxjs@setpaper{{182trueemm}{257trueemm}}}
115 \DeclareOption{b6paper}{\bxjs@setpaper{{128trueemm}{182trueemm}}}
116 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
117 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
118 \DeclareOption{b4j}{\bxjs@setpaper{{257trueemm}{364trueemm}}}
119 \DeclareOption{b5j}{\bxjs@setpaper{{182trueemm}{257trueemm}}}
120 \DeclareOption{a4var}{\bxjs@setpaper{{210trueemm}{283trueemm}}}
121 \DeclareOption{b5var}{\bxjs@setpaper{{182trueemm}{230trueemm}}}
122 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
123 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
124 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}
```

■横置き 用紙の縦と横の長さを入れ換えます。

```
125 \newif\if@landscape
126 \@landscapefalse
127 \DeclareOption{landscape}{\@landscapetrue}
```

■slide オプション `slide` を新設しました。

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

```
128 \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。

```
129 %<!slide>\@slidefalse
130 %<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。

```
131 \newcommand{\@ptsize}{0}
132 \def\bxjs@param@basefontsize{10pt}
133 \def\jsUnusualPtSize{-20}
```

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

```
134 \def\bxjs@setbasefontsize#1{%
135   \bxjs@setbasefontlength\@tempdima{#1}%
136   \edef\bxjs@param@basefontsize{\the\@tempdima}%
137   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
138   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
139   \else\ifdim\@tempdima=12pt  \long\def\@ptsize{2}%
140   \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。

```
141 \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。

```
142   \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。

```
143   \ifx j\jsEngine \setlength#1{#2}%
144   \else
145     \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
146     \ifx\bxjs@tmpa\relax \setlength#1{#2}%
147     \else \@tempdimc0.25mm #1=\bxjs@tmpa\@tempdim
148     \fi
```

```

149 \fi}
150 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
151 \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
152 \else \def\bxjs@tmpa{#1}%
153 \fi}
154 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
155 \ifx\@nnil#2\@nnil\else
156 \ClassWarningNoLine\bxjs@clsname
157 {You should not use 'true' lengths here}%
158 \fi}

```

`\ifjpsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

```

159 \newif\ifjpsc@mag
160 \newif\ifjpsc@mag@xreal
161 %\let\jpsc@magscale\@undefined
162 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
163 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
164 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
165 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
166 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
167 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
168 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
169 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
170 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
171 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
172 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
173 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
174 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
175 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
176 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
177 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
178 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
179 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
180 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の `magstyle` 設定オプション。

```

181 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
182 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
183 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}

```

■ トンボオプション

欧文 L^AT_EX のカーネルではサポートされないため削除。

ただし JS クラスとの互換のため以下の変数を用意する。

※ JS クラスでは、2017-01-11 の改修で、「トンボオプション指定時のみ

`\stockwidth/height` を定義する」という仕様に変更された。BXJS での対応は要検討。

```
184 \newdimen\stockwidth \newdimen\stockheight
```

■面付け

欧文 L^AT_EX のカーネルではサポートされないため削除。

■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。

```
185 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
186 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
187 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
188 \DeclareOption{onecolumn}{\@twocolumnfalse}
189 \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。

```
190 \DeclareOption{titlepage}{\@titlepagetrue}
191 \DeclareOption{notitlepage}{\@titlepagefalse}
```

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

[2017-02-24] `openright` は横組では奇数ページ起こし, 縦組では偶数ページ起こしを表します。ややこしいですが, これは L^AT_EX 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。

```
192 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
193 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
194 %<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` L^AT_EX 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。

```
195 \def\eqnarray{%
196   \stepcounter{equation}%
197   \def\@currentlabel{\p@equation\theequation}%
198   \global\@eqnswtrue
```

```

199 \m@th
200 \global\@eqcnt\z@
201 \tabskip\@centering
202 \let\@eqnocr
203 $$\everycr{}\halign to\displaywidth\bgroup
204     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
205     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
206     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
207     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
208     \tabskip\z@skip
209 \cr}

```

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

```

210 \DeclareOption{leqno}{\input{leqno.clo}}
211 \DeclareOption{fleqn}{\input{fleqn.clo}}
212 % fleqn 用の eqnarray 環境の再定義
213 \def\eqnarray{%
214     \stepcounter{equation}%
215     \def\@currentlabel{\p@equation\theequation}%
216     \global\@eqnswtrue\m@th
217     \global\@eqcnt\z@
218     \tabskip\mathindent
219     \let\@=\@eqnocr
220     \setlength\abovedisplayskip{\topsep}%
221     \ifvmode
222         \addtolength\abovedisplayskip{\partopsep}%
223     \fi
224     \addtolength\abovedisplayskip{\parskip}%
225     \setlength\belowdisplayskip{\abovedisplayskip}%
226     \setlength\belowdisplayshortskip{\abovedisplayskip}%
227     \setlength\abovedisplayshortskip{\abovedisplayskip}%
228     $$\everycr{}\halign to\linewidth% $$
229     \bgroup
230     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
231     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
232     &\global\@eqcnt\tw@
233         $\displaystyle{##}$\hfil \tabskip\@centering
234     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
235     \tabskip\z@skip\cr
236 }}

```

■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。

```

237 % \DeclareOption{openbib}{%
238 %     \AtEndOfPackage{%
239 %         \renewcommand\@openbib@code{%
240 %             \advance\leftmargin\bibindent

```

```

241 %      \itemindent -\bibindent
242 %      \listparindent \itemindent
243 %      \parsep \z}%
244 %      \renewcommand\newblock{\par}}

```

■数式フォントとして和文フォントを登録しないオプション 数式中では 16 通りのフォントしか使えません。AMSFonTS や mathpmtmx 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：

```

245 %\let\bxjs@enablejfam\undefined

enablejfam オプションの処理。
246 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
247 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
248 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
249 \define@key{bxjs}{enablejfam}[true]{%
250 \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。

```

251 \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。

\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 は 2016/07/13 版で廃止された。よって \ifdraft は 2.0 版で廃止を予定する。

```

252 \let\ifjsDraft\iffalse
253 \@onlypreamble\bxjs@draft
254 \def\bxjs@draft#1{%
255 \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
256 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}

```

```

257 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
258 \AtBeginDocument{%
259   \expandafter\ifx\csname ifdraft\endcsname\relax
260     \expandafter\let\csname ifdraft\endcsname
261       \csname ifjsDraft\endcsname
262   \fi}

```

■和文フォントメトリックの選択

ここは和文ドライバの管轄。

■papersize スペシャルの利用

`geometry` パッケージが行う。

`\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。

```

263 \newif\ifbxjs@papersize
264 \bxjs@papersizetrue
265 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
266 \DeclareOption{papersize}{\bxjs@papersizetrue}

```

■英語化 オプション `english` を新設しました。

```

267 \newif\if@english
268 \@englishfalse
269 \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 ‘`report`’ 相当のものは別に `bxjsreport` クラスとして用意する。

※ FIXME : `jsbook + report` と `jsreport` のどちらに合わせるか。

■`jslogo` パッケージの読み込み \LaTeX 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。

```
270 \newif\if@jslogo \@jslogofalse
271 \DeclareOption{jslogo}{\@jslogotrue}
272 \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 / nozw：\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：Xe_ΛTeX の “generateactualtext” を有効／無効にする。
- simplejasetup / nosimplejasetup：Xe_ΛTeX 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ーに任せるか。
- 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 \TeX 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。

```

273 \mathchardef\bxjs@isc@ll=128
274 \mathchardef\bxjs@isc@sl=259
275 \def\bxjs@isc@sl@h{65539 }
276 \def\bxjs@invscale#1#2{%
277   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
278     \ifdim\@tempdima<\bxjs@isc@ll\p@
279       \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
280       \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
281     \else
282       \@tempcnta\@tempdima \divide\@tempcnta\@tempdimb
283       \multiply\@tempcnta\p@ \let\bxjs@isc@sl\bxjs@isc@sl@h
284     \fi
285     \@tempcntb\p@ \divide\@tempcntb\@tempdimb
286     \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
287     \@tempdimb\@tempcnta\@ne
288     \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
289     \advance\@tempcnta\bxjs@isc@sl \@tempdimc\@tempcnta\@ne
290     \@whiledim\@tempdimb<\@tempdimc\do{%
291       \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
292       \advance\@tempcntb\@ne \divide\@tempcntb\tw@
293       \ifdim #2\@tempcntb>\@tempdima
294         \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
295       \else \@tempdimb=\@tempcntb\@ne \fi}%
296     \xdef\bxjs@gtmpa{\the\@tempdimb}%
297   \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。

```
298 \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。

```
299 \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。

```
300 \def\bxjs@composite@proc{%
```

```
301 \bxjs@oldfontcommandstrue
```

```
302 \setkeys{bxjs}{ja=pandoc}%
```

```
303 \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…。

```
304 \def\bxjs@driver@opt{dvipdfmx}%
```

```
305 \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。

```
306 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
307 %\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。

```
308 \DeclareOption{autodetect-engine}{%
```

```
309 \let\bxjs@engine@given=*}
```

```
310 \DeclareOption{latex}{%
```

```
311 \def\bxjs@engine@opt{latex}%
```

```
312 \let\bxjs@engine@given=n}
```

```
313 \DeclareOption{platex}{%
```

```
314 \def\bxjs@engine@opt{platex}%
```

```
315 \let\bxjs@engine@given=j}
```

```
316 \DeclareOption{uplatex}{%
```

```
317 \def\bxjs@engine@opt{uplatex}%
```

```
318 \let\bxjs@engine@given=u}
```

```
319 \DeclareOption{xelatex}{%
```

```
320 \def\bxjs@engine@opt{xelatex}%
```

```
321 \let\bxjs@engine@given=x}
```

```
322 \DeclareOption{pdflatex}{%
```

```
323 \def\bxjs@engine@opt{pdflatex}%
```

```

324 \let\bxjs@engine@given=p}
325 \DeclareOption{lualatex}{%
326 \def\bxjs@engine@opt{lualatex}%
327 \let\bxjs@engine@given=l}
328 \DeclareOption{platex-ng}{%
329 \def\bxjs@engine@opt{platex-ng}%
330 \let\bxjs@engine@given=g}
331 \DeclareOption{platex-ng*}{%
332 \def\bxjs@engine@opt{platex-ng*}%
333 \let\bxjs@platexng@nodrv=t%
334 \let\bxjs@engine@given=g}

```

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

```

335 %\let\bxjs@driver@given\undefined
336 \let\bxjs@driver@@dvimode=0
337 \let\bxjs@driver@@dvipdfmx=1
338 \let\bxjs@driver@@pdfmode=2
339 \let\bxjs@driver@@xetex=3
340 \let\bxjs@driver@@dvips=4
341 \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。

```

342 %\let\bxjs@driver@opt\undefined
343 \DeclareOption{dvips}{%
344 \def\bxjs@driver@opt{dvips}%
345 \let\bxjs@driver@given\bxjs@driver@@dvips}
346 \DeclareOption{dviout}{%
347 \def\bxjs@driver@opt{dviout}%
348 \let\bxjs@driver@given\bxjs@driver@@dvimode}
349 \DeclareOption{xdvi}{%
350 \def\bxjs@driver@opt{xdvi}%
351 \let\bxjs@driver@given\bxjs@driver@@dvimode}
352 \DeclareOption{dvi pdfmx}{%
353 \def\bxjs@driver@opt{dvi pdfmx}%
354 \let\bxjs@driver@given\bxjs@driver@@dvi pdfmx}
355 \DeclareOption{nodvidriver}{%
356 \def\bxjs@driver@opt{nodvidriver}%
357 \let\bxjs@driver@given\bxjs@driver@@none}
358 \DeclareOption{pdftex}{%
359 \def\bxjs@driver@opt{pdftex}%
360 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
361 \DeclareOption{lualatex}{%
362 \def\bxjs@driver@opt{lualatex}%
363 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
364 \DeclareOption{xetex}{%
365 \def\bxjs@driver@opt{xetex}%
366 \let\bxjs@driver@given\bxjs@driver@@xetex}

```

「もし DVI モードであればドライバを `dvi pdfmx` 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。

```
367 \DeclareOption{dvipdfmx-if-dvi}{%
368   \setkeys{bxjs}{dvi=dvipdfmx}}
```

■ その他の BXJS 独自オプション 密

`\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。

```
369 \newif\ifbxjs@usezw \bxjs@usezwtrue
```

`zw`、`noz` オプションの定義。

```
370 \DeclareOption{noz}{%
371   \bxjs@usezwfalse}
372 \DeclareOption{zw}{%
373   \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。

```
374 \newif\ifbxjs@disguise@js \bxjs@disguise@jstrue
```

`nojs`、`js` オプションの定義。

```
375 \DeclareOption{nojs}{%
376   \bxjs@disguise@jsfalse}
377 \DeclareOption{js}{%
378   \bxjs@disguise@jstrue}
```

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

```
379 \newif\ifbxjs@precisetext
```

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

```
380 \DeclareOption{noprecisetext}{%
381   \bxjs@precisetextfalse}
382 \DeclareOption{precisetext}{%
383   \bxjs@precisetexttrue}
```

`\ifbxjs@simplejasetup` X_YTeX 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）。

```
384 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue
```

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

```
385 \DeclareOption{nosimplejasetup}{%
386   \bxjs@simplejasetupfalse}
387 \DeclareOption{simplejasetup}{%
388   \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。

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

その上で、「 $\text{T}_{\text{E}}\text{X}$ 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、「 $\text{T}_{\text{E}}\text{X}$ のバージョンが 3.14159265 (2014 年 1 月) 以上であるか」を採用する。

```
390 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
391 \def\bxjs@tmpb#1 #2#3\@nil{%
392   \ifx1#2\bxjs@bigcodetrue \fi}
393 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa 0\@nil
```

`nobigcode` / `bigcode` オプションの定義。

```
394 \DeclareOption{nobigcode}{%
395   \bxjs@bigcodefalse}
396 \DeclareOption{bigcode}{%
397   \bxjs@bigcodetrue}
```

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

```
398 \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。

```
399 \DeclareOption{nooldfontcommands}{%
400   \bxjs@oldfontcommandsfalse}
401 \DeclareOption{oldfontcommands}{%
402   \bxjs@oldfontcommandstrue}
```

■ `keyval` 型のオプション

```
403 \def\bxjs@setkey{%
404   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
405 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
406 \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`。

```
407 \def\bxjs@safe@setkeys#1#2{%
408   \let\bxjs@KV@errx\KV@errx
409   \let\KV@errx\bxjs@safe@setkeys@a
410   \setkeys{#1}{#2}%
411   \let\KV@errx\bxjs@KV@errx}
412 \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。

```
413 \def\bxjs@set@keyval#1#2#3{%
414   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
415   \ifx\bxjs@next\relax
416     \bxjs@error@keyval{#1}{#2}%
```

```

417     #3%
418     \else \bxjs@next
419     \fi}
420 \@onlypreamble\bxjs@error@keyval
421 \def\bxjs@error@keyval#1#2{%
422     \ClassError\bxjs@clsname
423     {Invalid value '#2' for option #1}\@ehc}

\ifbxjs@scaleset 和文スケール値が指定されたか。
424 \newif\ifbxjs@scaleset

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

    base オプションの処理。
426 \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。
427 \@onlypreamble\bxjs@do@opt@jbase
428 \let\bxjs@do@opt@jbase\relax
429 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
430 \def\bxjs@setjbasefontsize#1{%
431     \def\bxjs@do@opt@jbase{%
432         \bxjs@setbasefontlength\@tempdima{#1}%
433         \bxjs@invscale\@tempdima\jsScale
434         \bxjs@setbasefontsize{\@tempdima}}

    scale オプションの処理。
435 \define@key{bxjs}{scale}{%
436     \bxjs@scalesettrue
437     \edef\jsScale{#1}}

    noscale オプションの処理。
438 \DeclareOption{noscale}{%
439     \bxjs@scalesettrue
440     \def\jsScale{1}}

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

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

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

\bxjs@jadriver 和文ドライバの名前。
444 \let\bxjs@jadriver\relax
445 %\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)。

```
446 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
447 \define@key{bxjs}{ja}[\relax]{%
448   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}
```

\jsJaFont 和文フォント設定の名前。

```
449 \let\jsJaFont\@empty
```

jafont オプションの処理。

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

\jsJaParam 和文ドライバパラメタの文字列。

```
451 \let\jsJaParam\@empty
```

japaram オプションの処理。

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

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

```
453 \let\bxjs@magstyle@mag=m
454 \let\bxjs@magstyle@real=r
455 \let\bxjs@magstyle@xreal=x
```

(新しい素敵なお名前。)

※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。

```
456 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
457 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
458 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

\bxjs@magstyle@default は既定の値を表す。

```
459 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
460 \ifx l\jsEngine \ifnum\luatexversion>86
461   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
462 \fi\fi
463 \ifjsWithpTeXng
464   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
465 \fi
466 \let\bxjs@magstyle\bxjs@magstyle@default
```

magstyle オプションの処理。

```
467 \define@key{bxjs}{magstyle}{%
468   \expandafter\let\expandafter\bxjs@magstyle\csname
469     bxjs@magstyle@#1\endcsname
470   \ifx\bxjs@magstyle\relax
471     \ClassError\bxjs@clsname
472     {Invalid value '#1' for option magstyle}\@ehc
```

```

473 \let\bxjs@magstyle\bxjs@magstyle@default
474 \fi}

```

`\bxjs@geometry` geometry オプションの値。

```

475 \let\bxjs@geometry@class=c
476 \let\bxjs@geometry@user=u
477 \let\bxjs@geometry\bxjs@geometry@class

geometry オプションの処理。
478 \define@key{bxjs}{geometry}{%
479 \expandafter\let\expandafter\bxjs@geometry\csname
480 bxjs@geometry@#1\endcsname
481 \ifx\bxjs@geometry\relax
482 \ClassError\bxjs@clsname
483 {Invalid value '#1' for option geometry}\@ehc
484 \let\bxjs@geometry\bxjs@geometry@class
485 \fi}

```

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

```

486 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

fancyhdr オプションの処理。
487 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
488 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
489 \define@key{bxjs}{fancyhdr}[true]{%
490 \bxjs@set@keyval{fancyhdr}{#1}{}}

```

`\ifbxjs@dvi@opt` dvi オプションが指定されたか。

```

491 \newif\ifbxjs@dvi@opt

DVI モードのドライバとドライバ種別との対応。
492 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
493 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
494 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvmode
495 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvmode
496 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

dvi オプションの処理。
497 \define@key{bxjs}{dvi}{%
498 \expandafter\let\expandafter\bxjs@tmpa\csname
499 bxjs@dvidriver@@#1\endcsname
500 \ifx\bxjs@tmpa\relax
501 \ClassError\bxjs@clsname
502 {Invalid value '#1' for option dvi}\@ehc
503 \else

\bxjs@driver@given を未定義にしていることに注意。
504 \def\bxjs@driver@opt{#1}%
505 \let\bxjs@driver@given\undefined
506 \bxjs@dvi@opttrue

```

507 \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。

508 \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。

※ただし 1.5 版では移行措置のため bxjsreport の既定値を真にする。

509 \newif\ifbxjs@force@chapterabstract

510 %<book|report>\bxjs@force@chapterabstracttrue

layout オプションの処理。

511 \@namedef{bxjs@kv@layout@v1}{%

512 %<book>\bxjs@layout@buggyhmargintrue

513 %<report>\bxjs@force@chapterabstracttrue

514 }

515 \@namedef{bxjs@kv@layout@v2}{%

516 %<book>\bxjs@layout@buggyhmarginfalse

517 %<report>\bxjs@force@chapterabstractfalse

518 }

519 \define@key{bxjs}{layout}{%

520 \bxjs@set@keyval{layout}{#1}{}}

\bxjs@textwidth@limit textwidth-limit の指定値。 \textwidth の上限。

521 %\let\bxjs@textwidth@limit\undefined

522 \define@key{bxjs}{textwidth-limit}{%

523 \edef\bxjs@textwidth@limit{#1}}

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

524 %\let\bxjs@paragraph@mark\undefined

525 \define@key{bxjs}{paragraph-mark}{%

526 \edef\bxjs@paragraph@mark{#1}}

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

527 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue

528 \let\bxjs@kv@wholezwlines>true\bxjs@whole@zw@linestrue

529 \let\bxjs@kv@wholezwlines>false\bxjs@whole@zw@linesfalse

530 \define@key{bxjs}{whole-zw-lines}[true]{\bxjs@set@keyval{wholezwlines}{#1}{}}

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

531 \newif\ifbxjs@xkanjiskip@cmd \bxjs@xkanjiskip@cmdtrue

532 \let\bxjs@kv@xkanjiskipcmd>true\bxjs@xkanjiskip@cmdtrue

533 \let\bxjs@kv@xkanjiskipcmd>false\bxjs@xkanjiskip@cmdfalse

534 \define@key{bxjs}{xkanjiskip-cmd}[true]{\bxjs@set@keyval{xkanjiskipcmd}{#1}{}}

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

535 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue

```

536 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
537 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
538 \define@key{bxjs}{hyperref-enc}[true]{\bxjs@set@keyval{hyperrefenc}{#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。

```

539 \let\bxjs@ltx@removeelement\@removeelement
540 \def\@removeelement#1#2#3{%
541   \def\reserved@a{#2}%
542   \ifx\reserved@a\@empty \let#3\@empty
543   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
544   \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。

```

545 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
546 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
547 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
548 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
549 \ProcessOptions\relax
550 \bxjs@composite@proc

```

グローバルオプションのトークン列に { } が含まれていると、やはり後のパッケージの読み込み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。

```

551 \@onlypreamble\bxjs@purge@brace@elts
552 \def\bxjs@purge@brace@elts{%
553   \def\bxjs@tmpa{\@gobble}%
554   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
555   \let\@classoptionslist\bxjs@tmpa}
556 \@onlypreamble\bxjs@purge@be@a
557 \def\bxjs@purge@be@a#1,{%
558   \ifx\@nil#1\relax\else
559     \bxjs@purge@be@b#1{\@nil}
560     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
561     \expandafter\bxjs@purge@be@a
562   \fi}

```

```

563 \@onlypreamble\bxjs@purge@be@b
564 \def\bxjs@purge@be@b#1#\{bxjs@purge@be@c}
565 \@onlypreamble\bxjs@purge@be@c
566 \def\bxjs@purge@be@c#1\@nil{%
567   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
568 \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。

```

569 \@expandtwoargs\@removeelement
570 {papersize}\@classoptionslist\@classoptionslist
571 \@expandtwoargs\@removeelement
572 {10pt}\@classoptionslist\@classoptionslist
573 \@expandtwoargs\@removeelement
574 {noscale}\@classoptionslist\@classoptionslist

```

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。

```

575 \ifx\bxjs@jadriver\relax\else
576   \let\bxjs@jadriver@given\bxjs@jadriver
577 \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。

```

578 \let\bxjs@tmpb\jsEngine
579 \ifx j\bxjs@tmpb\ifjsWithpTeXng
580   \let\bxjs@tmpb=g
581 \fi\fi
582 \ifx j\bxjs@tmpb\ifjsWithupTeX
583   \let\bxjs@tmpb=u
584 \fi\fi
585 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
586   \let\bxjs@tmpb=n
587 \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。）

```

588 \ifx *\bxjs@engine@given
589   \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。

```

590 \ifx j\bxjs@engine@given
591   \g@addto@macro\@classoptionslist{,latex}
592 \else\ifx u\bxjs@engine@given
593   \g@addto@macro\@classoptionslist{,uplatex}
594 \fi\fi
595 \fi
596 \ifx\bxjs@engine@given\@undefined\else
597 \ifx\bxjs@engine@given\bxjs@tmpb\else
598   \ClassError\bxjs@clsname

```

```

599     {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
600   \fi
601 \fi

```

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

```

602 \ifjsWithpTeXng
603   \g@addto@macro\@classoptionslist{,uplatex}
604 \fi

```

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。

```

605 \@tempwatrue
606 \ifx \bxjs@driver@given\undefined\else
607   \ifjsInPdfMode
608     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
609       \@tempwafalse
610     \fi
611   \else\ifx x\jsEngine
612     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
613       \@tempwafalse
614     \fi
615   \else
616     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
617       \@tempwafalse
618     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
619       \@tempwafalse
620     \fi\fi
621     \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
622       \@tempwafalse
623     \fi\fi
624   \fi\fi
625 \fi
626 \if@tempswa\else
627   \ClassError\bxjs@clsname
628   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
629 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

630 \ifjsInPdfMode \@tempwafalse
631 \else\ifx x\jsEngine \@tempwafalse
632 \else\ifjsWithpTeXng \@tempwafalse
633 \else \@tempwatrue
634 \fi\fi\fi
635 \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)。

```

636 \ifx\bxjs@driver@opt\undefined \ifx\bxjs@jadriver@given\undefined\else
637   \ClassWarningNoLine\bxjs@clsname
638   {No driver option is given}
639 \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。)

```
640 \ifbxjs@dvi@opt
641   \edef\bxjs@nxt{%
642     \let\noexpand\bxjs@driver@given
643     \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
644     \noexpand@g@addto@macro\noexpand@\classoptionslist
645     {,\bxjs@driver@opt}%
646   }\bxjs@nxt
647 \fi
648 \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。

```
649 \ifjswithpTeXng
650   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
651     \let\bxjs@platexng@nodrv@\undefined
652   \else\ifx t\bxjs@platexng@nodrv\else
653     \g@addto@macro@\classoptionslist{,dvipdfmx}
654   \fi\fi
655 \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。

```
656 \ifx\bxjs@driver@given\bxjs@driver@@none
657   \bxjs@papersizefalse
658 \fi
```

\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。

```
659 \def\bxjs@@minimal{minimal}
660 \ifx\bxjs@jadriver\relax
661   \ifx j\jsEngine
662     \def\bxjs@jadriver{standard}
663   \else
664     \let\bxjs@jadriver\bxjs@@minimal
665   \fi
666 \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。

```
667 \ifx\bxjs@jadriver@given@\undefined\else
668   \ifx\bxjs@engine@given@\undefined
669     \ClassError\bxjs@clsname
670     {An engine option must be explicitly given}%
671     {When you use a Japanese-driver you must specify a correct\MessageBreak
672     engine option.\MessageBreak\@ehc}
673 \fi\fi
```

新しい LuaTeX (0.87 版以降) では mag がアレなので、magstyle=usemag が指定されていた場合はエラーを出す。(この場合の既定値は nomag* であり、エラーの場合は既定値に置き換えられる。)

```
674 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
675   \ifx\bxjs@magstyle\bxjs@magstyle@mag
676     \let\bxjs@magstyle\bxjs@magstyle@default
677     \ClassError\bxjs@clsname
678     {The engine does not support 'magstyle=usemag'}%
679     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
680     The default value 'nomag*' is used instead.\MessageBreak \@ehc}
681   \fi
682 \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。

```
683 \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。

```
684 \let\Cjascale\jsScale
```

後処理

```
685 \if@slide
686   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}
687 \fi
688 \if@landscape
689   \setlength\@tempdima {\paperheight}
690   \setlength\paperheight{\paperwidth}
691   \setlength\paperwidth {\@tempdima}
692 \fi
```

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

```
693 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
694   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
695     \catcode\@tempcnta\active
696     \advance\@tempcnta\@ne
697   \repeat
698 \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。

```
699 \ifbxjs@disguise@js
700 %<book|report>\def\bxjs@js@clsname{jsbook}
701 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
702 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
703 \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。

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

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

```
705 \ifbxjs@oldfontcommands
```

```
706 \AtEndOfClass{\allowoldfontcommands}
```

```
707 \fi
```

■papersize スペシャルの出力

`geometry` パッケージが行う。

■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。

```
708 %<slide>\def\n@baseline{13}%
```

```
709 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
```

```
710 %<!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。

```
711 \ifx\bxjs@magstyle\bxjs@magstyle@mag
```

```
712 \jsc@magtrue
```

```
713 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
```

```
714 \jsc@mag@xrealtrue
```

```
715 \fi\fi
```

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

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

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

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

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

```
717 \@tempdima=\bxjs@param@basefontsize
```

```

718 \advance\@tempdima.001pt \multiply\@tempdima25
719 \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
720 \edef\bxjs@param@mag{\the\@tempcnta}
721 \else
722 % mag 値が直接指定された場合
723 \let\c@bxjs@cmta\@tempcnta
724 \setcounter{bxjs@cmta}{\bxjs@param@mag}
725 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
726 % 有効な mag 値の範囲は 1--32768
727 \edef\bxjs@param@mag{\the\@tempcnta}
728 \advance\@tempcnta100000
729 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
730 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
731 \edef\bxjs@param@basefontsize{\the\@tempdima}
732 \fi
733 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
734 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
735 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
736 \edef\jsc@magscale{\strip@pt\@tempdima}
737 \let\jsc@baseFontSize\bxjs@param@basefontsize
738 %\typeout{\string\jsc@DocClass: \meaning\jsc@DocClass}
739 %\typeout{\string\jsc@Engine: \meaning\jsc@Engine}
740 %\typeout{\string\jsc@baseFontSize: \jsc@baseFontSize}
741 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
742 %\typeout{\string\jsc@magscale: \jsc@magscale}
743 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
744 %\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。

```

745 \ifjsc@mag
746 \let\jsc@mpt\p@
747 \newdimen\jsc@mmm \jsc@mmm=1mm
748 \catcode\?=9 % \p@? read as \p@
749 \else
750 \newdimen\jsc@mpt \jsc@mpt=\jsc@magscale \p@
751 \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
752 \catcode\?=11 \let\p@?\jsc@mpt
753 \fi
754 \chardef\bxjs@qmc=\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 $\text{\jssScale} \times$ (指定フォントサイズ) に等しい。

`nozw` 非指定時は `\zw` を `\jsZw` と同義にする。

```
755 \newdimen\jsZw
756 \jsZw=10\jsc@empt \jsZw=\jsScale\jsZw
757 \ifbxjs@usezw
758 \providecommand*\zw{\jsZw}
759 \fi
```

`\zwspace` 全角幅の水平空き。

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

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

```
761 \ifjsc@mag@xreal
762 \RequirePackage{type1cm}
763 \let\jsc@invscale\bxjs@invscale
```

ムニヤムニヤムニヤ……。

```
764 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
765 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
766 \let\jsc@get@external@font\get@external@font
767 \def\get@external@font{%
768   \jsc@preadjust@extract@font
769   \jsc@get@external@font}
770 \def\jsc@fstrunc#1{%
771   \edef\jsc@tmpa{\strip@pt#1}%
772   \expandafter\jsc@fstrunc@a\jsc@tmpa.***\@nil}
773 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
774   \if#5*\else
775     \edef\jsc@tmpa{#1%
776       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
777   \fi}
778 \def\jsc@preadjust@extract@font{%
779   \let\jsc@req@size\f@size
780   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
781   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
782   \let\jsc@ref@size\jsc@tmpa
783   \let\f@size\jsc@ref@size}
784 \def\execute@size@function#1{%
785   \let\jsc@cref@size\f@size
786   \let\f@size\jsc@req@size
787   \csname s@fct@#1\endcsname}
788 \let\jsc@DeclareErrorFont\DeclareErrorFont
789 \def\DeclareErrorFont#1#2#3#4#5{%
790   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
791   \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
792   \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
793 \def\gen@sfcnt{%
794   \edef\mandatory@arg{\mandatory@arg\jsc@cdef@size}%

```

```

795 \empty@sfcnt}
796 \def\genb@sfcnt{%
797 \edef\mandatory@arg{%
798 \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@}%
799 \empty@sfcnt}
800 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
801 \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 802 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 803 %\def\jsc@medskip{\vspace\jsc@medskipamount}
804 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount 805 \newskip\jsc@smallskipamount
\jsc@bigskipamount 806 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
807 %\newskip\jsc@medskipamount
808 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
809 %\newskip\jsc@bigskipamount
810 %\jsc@bigskipamoun =12\jsc@mpt plus 4\jsc@mpt minus 4\jsc@mpt

```

■ pagesize スペシャルの出力

削除。

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、 $\text{L}^{\text{T}}\text{E}^{\text{X}}$ の `\def\@{\spacefactor\@m}` という定義 (`\@m` は 1000) では `I watch TV\@.` と書くと `V` とピリオドの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。

```

811 \chardef\bxjs@periodchar=\.
812 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcodes\bxjs@periodchar}
813 \def\@{\bxjs@SE{}}

```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

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

```
814 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
815 \expandafter\bxjs@tmpa\expandafter{%
816   \set@fontsize{#1}{#2}{#3}%
817 % 末尾にコードを追加
818   \expandafter\def\expandafter\size@update\expandafter{%
819     \size@update
820     \jsFontSizeChanged}%
821 }
```

`\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。

```
822 \newcommand*\jsFontSizeChanged{%
823   \jsZw=\f@size\p@
824   \jsZw=\jsScale \jsZw
825   \ifdim\parindent>\z@
826     \if@english \parindent=1em
827     \else       \parindent=1\jsZw
828     \fi
829   \fi\relax
830   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
831 \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。

```
832 \ifjsc@mag
833   \let\jsc@setfontsize\@setfontsize
834 \else
835   \def\jsc@setfontsize#1#2#3{%
836     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
837 \fi
```

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

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

```
838 \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

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

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

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

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

```

839 \newif\ifnarrowbaselines
840 \if@english
841   \narrowbaselinestrue
842 \fi
843 \def\narrowbaselines{%
844   \narrowbaselinestrue
845   \skip0=\abovedisplayskip
846   \skip2=\abovedisplayshortskip
847   \skip4=\belowdisplayskip
848   \skip6=\belowdisplayshortskip
849 % 一時的に警告を無効化する
850   \let\bxjs@ltx@nomath\@nomath
851   \let\@nomath@gobble
852   \@currsize\selectfont
853   \let\@nomath\bxjs@ltx@nomath
854   \abovedisplayskip=\skip0
855   \abovedisplayshortskip=\skip2
856   \belowdisplayskip=\skip4
857   \belowdisplayshortskip=\skip6\relax}
858 \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。

```

859 \def\bxjs@if@narrowbaselines{%
860   \ifnarrowbaselines\expandafter\@firstoftwo
861   \else \expandafter\@secondoftwo
862   \fi
863 }

```

`\normalsize` 標準のフォントサイズと行送りを選ぶコマンドです。

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

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

microtype 対策のため if 文を避ける。

```

864 \renewcommand{\normalsize}{%
865   \bxjs@if@narrowbaselines{%
866     \jsc@setfontsize\normalsize\@xpt\@xiipt

```

```

867 }{%else
868   \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
869 }%

```

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。

```

870 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
871 \abovedisplayshortskip \z@ \@plus3\p@?
872 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
873 \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。

```
874 \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。

```
875 \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ラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅 (`1zw`) です。

`\Cwd`

`\Cvs` まず `\Cwd` 等の変数を定義する。

```

\Cht 876 \ifx\Cht\@undefined \newdimen\Cht \fi
\Cdp 877 \ifx\Cdp\@undefined \newdimen\Cdp \fi
878 \ifx\Cwd\@undefined \newdimen\Cwd \fi
879 \ifx\Cvs\@undefined \newdimen\Cvs \fi
880 \ifx\Chs\@undefined \newdimen\Chs \fi

```

規約上, 現在の `\jsZw` の値が `\Cwd` である。`\Cht` と `\Cdp` は単純に `\Cwd` の 88% と 12% の値とする。

```

881 \setlength\Cht{0.88\jsZw}
882 \setlength\Cdp{0.12\jsZw}
883 \setlength\Cwd{1\jsZw}
884 \setlength\Cvs{\baselineskip}
885 \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。

```

886 \newcommand{\small}{%
887   \bxjs@if@narrowbaselines{%
888     <!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
889     <kiyou>    \jsc@setfontsize\small{8.8888}{11}%
890   }{%else
891     <!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
892     <kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
893   }%
894   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
895   \abovedisplayshortskip \z@ \@plus3\p@?
896   \belowdisplayskip \abovedisplayskip
897   \belowdisplayshortskip \belowdisplayskip
898   \def\@listi{\leftmargin\leftmargini
899             \topsep \z@
900             \parsep \z@
901             \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。

```

902 \newcommand{\footnotesize}{%
903   \bxjs@if@narrowbaselines{%
904     <!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
905     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
906   }{%else
907     <!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
908     <kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
909   }%
910   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
911   \abovedisplayshortskip \z@ \@plus2\p@?
912   \belowdisplayskip \abovedisplayskip
913   \belowdisplayshortskip \belowdisplayskip
914   \def\@listi{\leftmargin\leftmargini
915             \topsep \z@
916             \parsep \z@
917             \itemsep \parsep}}

```

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

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 918 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
919 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 920 \if@twocolumn
\HUGE 921 <!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
922 <kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
923 \else
924 <!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}

```

```

925 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
926 \fi
927 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
928 %<kiyou>\newcommand{\LARGE}{\jsc@setfontsize\Large{12.222}{21}}
929 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xvipt{25}}
930 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
931 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
932 \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り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。

```

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

```

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

見出し用のフォントは `\bfseries` 固定ではなく、`\headfont` という命令で定めることにします。これは太ゴシックが使えるときは `\sffamily \bfseries` でいいと思いますが、通常の中ゴシックでは単に `\sffamily` だけのほうがよさそうです。『`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。

```

934 % \newcommand{\headfont}{\bfseries}
935 \newcommand{\headfont}{\sffamily}
936 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

5 レイアウト

■二段組

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

```

937 %<kiyou>\setlength\columnsep{2\Cwd}
938 %<kiyou>\setlength\columnsep{28truebp}
939 \setlength\columnseprule{0\p@}

```

■段落

`\lineskip` 上下の行の文字が `\lineskiplimit` より接近したら、`\lineskip` より近づかないようにします。元は 0pt ですが 1pt に変更しました。normal... の付いた方は保存用です。
`\normallineskip`
`\lineskiplimit` 940 \setlength\lineskip{1\p@?}
941 \setlength\normallineskip{1\p@?}
`\normallineskiplimit` 942 \setlength\lineskiplimit{1\p@?}

```
943 \setlength\normallineskiplimit{1\p@?}
```

`\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。

```
944 \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。

```
945 \setlength\parskip{0\p@}
```

```
946 \if@slide
```

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

```
948 \else
```

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

```
950 \fi
```

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

```
\@highpenalty 951 \@lowpenalty 51
```

```
952 \@medpenalty 151
```

```
953 \@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。

```
954 % \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。

```
955 % \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。

```
956 \ifjsc@mag
```

```
957 \mag=\bxjs@param@mag
```

```
958 \fi
```

```
959 \setlength{\topskip}{10\p@?}
```

`\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。

```
960 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
961 \def\bxjs@read@b{%
962   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
963   \else \expandafter\bxjs@read@d \fi}
964 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
965 \def\bxjs@read@d#1\@nil{}
966 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
967 \edef\bxjs@layout@paper{%
968   \ifjsc@mag truedimen,\fi
969   \if@landscape landscape,\fi
970   \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。

```
971 %<*article|report>
972 \def\bxjs@layout{%
973   headheight=\topskip,footskip=0.03367\paperheight,%
974   headsep=\footskip-\topskip,includeheadfoot,%
975   hscale=0.76,hmarginratio=1:1,%
976   vscale=0.83,vmarginratio=1:1,%
977 }
978 %</article|report>
979 %<*book>
980 \ifbxjs@layout@buggyhmargin %---
981 % アレ
982 \def\bxjs@layout{%
983   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
984   hmargin=36\jsc@mmm,hmarginratio=1:1,%
985   vscale=0.83,vmarginratio=1:1,%
986 }
987 \else %---
988 % 非アレ
989 \def\bxjs@layout{%
990   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
991   hmargin=18\jsc@mmm,%
992   vscale=0.83,vmarginratio=1:1,%
993 }
994 \fi %---
995 %</book>
996 %<*slide>
997 \def\bxjs@layout{%
998   noheadfoot,%
999   hscale=0.9,hmarginratio=1:1,%
1000  vscale=0.944,vmarginratio=1:1,%
1001 }
1002 %</slide>
```

`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
1003 \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。

```
1004 %<*book>
1005 \newcommand\jsTextWidthLimit{40}
1006 \ifx\bxjs@textwidth@limit\undefined\else
1007   \let\c@bxjs@cnta\@tempcnta
1008   \setcounter{bxjs@cnta}{\bxjs@textwidth@limit}
1009   \long\edef\jsTextWidthLimit{\the\@tempcnta}
1010 \fi
1011 %</book>
```

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

```
1012 \def\bxjs@postproc@layout{%
1013 % ドライバ再設定
1014   \ifx\bxjs@geometry@driver\relax\else
1015     \let\Gm@driver\bxjs@geometry@driver
1016   \fi
1017 % textwidth 調整
1018   \ifbxjs@whole@zw@lines
1019     \@tempdimb=\textwidth
1020     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1021     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1022     \advance\@tempdimb-\textwidth
1023     \advance\oddsidemargin 0.5\@tempdimb
1024     \advance\evensidemargin 0.5\@tempdimb
1025   \fi
1026   \fullwidth=\textwidth
1027 %<*book>
1028   \ifdim\textwidth>\jsTextWidthLimit\Cwd
1029     \textwidth=\jsTextWidthLimit\Cwd
1030     \addtolength\evensidemargin{\fullwidth-\textwidth}
1031   \fi
1032 %</book>
1033 % textheight 調整
1034   \@tempdimb=\textheight
1035   \advance\textheight-\topskip
1036   \divide\textheight\baselineskip \multiply\textheight\baselineskip
1037   \advance\textheight\topskip
1038   \advance\@tempdimb-\textheight
1039   \advance\topmargin0.5\@tempdimb
1040 % headheight 調整
1041   \@tempdima=\topskip
1042   \advance\headheight\@tempdima
1043   \advance\topmargin-\@tempdima
1044 % marginpar 設定
1045   \setlength\marginparsep{\columnsep}
1046   \setlength\marginparpush{\baselineskip}
```

```

1047 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1048     -\textwidth-10\jsc@mmm-\marginparsep}
1049 \ifbxjs@whole@zw@lines
1050     \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1051     \fi
1052 % 連動する変数
1053 \maxdepth=.5\topskip
1054 \stockwidth=\paperwidth
1055 \stockheight=\paperheight
1056 }

```

\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。

```

1057 \edef\jsGeometryOptions{%
1058     \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージ読込 罫

geometry オプションの値に応じて分岐する。

まずは geometry=class の場合。

```

1059 \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。

```

1060 \@onlypreamble\bxjs@revert
1061 \let\bxjs@revert\@empty
1062 \edef\bxjs@tmpa{\the\ht\strutbox}
1063 \ht\strutbox=10\p@?
1064 \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。

```

1065 \ifbxjs@papersize
1066     \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1067         \PassOptionsToPackage{dvipdfm}{geometry}
1068     \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1069         \PassOptionsToPackage{dvipdfm}{geometry}
1070     \fi\fi
1071     \let\bxPapersizeSpecialDone=t
1072 \else
1073     \PassOptionsToPackage{driver=none}{geometry}
1074 \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。

```
1075 \edef\bxjs@nxt{%
1076   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1077 \AtBeginDocument{\bxjs@pre@geometry@hook}
1078 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1079 \bxjs@nxt \bxjs@revert
1080 \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。

```
1081 \let\bxjs@geometry@driver\Gm@driver
1082 \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。

```
1083 \@onlypreamble\bxjs@pre@geometry@hook
1084 \def\bxjs@pre@geometry@hook{%
1085   \@ifpackageloaded{geometry}{%
1086     \@ifpackagelater{geometry}{2010/02/12}{%}%else
1087     \PackageWarningNoLine\bxjs@clsname
1088       {The 'geometry' package installed\MessageBreak
1089        is too old (< v5.0)}%
1090     \if x\jsEngine \ifnum\mag=\@m\else
1091       \def\bxjs@Gm@driver{pdftex}
1092       \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1093       \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1094       \fi\fi
1095   }%
1096   \ifjsWithpTeXng
1097     \ifx\Gm@driver\@empty
1098       \def\Gm@driver{pdftex}%
1099     \fi
1100   \fi
1101 }{}}
```

`\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。

```
1102 \def\setpagelayout{\@ifstar
1103   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1104 \def\bxjs@modify@layout#1{%
1105   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1106   \bxjs@nxt\bxjs@postproc@layout}
```

```

1107 \def\bxjs@reset@layout#1{%
1108   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1109   \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。

```

1110 \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。

```

1111 \AtBeginDocument{\bxjs@check@page@layout}
1112 \@onlypreamble\bxjs@check@page@layout
1113 \def\bxjs@check@page@layout{%
1114   \ifdim\textwidth=.5\maxdimen
1115     \ClassError\bxjs@clsname
1116       {Page layout is not properly set}%
1117     {\@ehd}
1118   \fi}
1119 \def\jsUseMinimalPageLayout{%
1120   \setlength{\textwidth}{6.5in}%
1121   \setlength{\textheight}{8in}}

```

\setpagelayout はとりあえず無効にしておく。

```

1122 \let\bxjs@geometry@driver\relax
1123 \def\setpagelayout{\@ifstar
1124   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1125 \def\bxjs@pagelayout@a#1{%
1126   \ClassError\bxjs@clsname
1127   {Command '\string\setpagelayout' is not supported,\MessageBreak
1128     because 'geometry' value is not 'class'}\@eha}

```

geometry=user の場合の処理はここで終わり。

```

1129 \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。

■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。

1130 `\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。

1131 `\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。

1132 `\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。

1133 `\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。

1134 `\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。

1135 `\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。

1136 `\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。

1137 `\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。

1138 `\renewcommand{\floatpagefraction}{.8}`

`\c@dbltopnumber` 二段組のとき本文ページ上部に出力できる段抜きフロートの最大数です。
[2003-08-23] ちょっと増やしました。

1139 `\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。

```
1140 \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。

```
1141 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1142 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1143 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

```
1144 \setlength\intextsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

`\dblfloatsep` 二段組のときの段抜きのフロートについての値です。

```
\dbltextfloatsep 1145 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

```
1146 \setlength\dbltextfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

`\@fptop` フロートだけのページに入るグルーです。`\@fptop` はページ上部, `\@fpbot` はページ下部,

`\@fpsep` `\@fpsep` はフロート間に入ります。

```
\@fpbot 1147 \setlength\@fptop{0\p@? \@plus 1fil}
```

```
1148 \setlength\@fpsep{8\p@? \@plus 2fil}
```

```
1149 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
\@dblfpsep 1150 \setlength\@dblfpsep{0\p@? \@plus 1fil}
```

```
\@dblfpbot 1151 \setlength\@dblfpsep{8\p@? \@plus 2fil}
```

```
1152 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ (日本語 TeX 開発コミュニティ版のみ)

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

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

となっています。

```
1153 %\def\pltx@cleartorightpage{\clearpage\if@twoside
```

```
1154 % \ifodd\c@page
```

```
1155 % \iftdir
```

```
1156 % \hbox{\thispagestyle{empty}}\newpage
```

```
1157 % \if@twocolumn\hbox{\newpage}\fi
```

```

1158 % \fi
1159 % \else
1160 % \ifydir
1161 % \hbox{}\thispagestyle{empty}\newpage
1162 % \if@twocolumn\hbox{}\newpage\fi
1163 % \fi
1164 % \fi\fi}
1165 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1166 % \ifodd\c@page
1167 % \ifydir
1168 % \hbox{}\thispagestyle{empty}\newpage
1169 % \if@twocolumn\hbox{}\newpage\fi
1170 % \fi
1171 % \else
1172 % \iftdir
1173 % \hbox{}\thispagestyle{empty}\newpage
1174 % \if@twocolumn\hbox{}\newpage\fi
1175 % \fi
1176 % \fi\fi}
1177 \def\pltx@cleartooddpage{\clearpage\if@twoside
1178 \ifodd\c@page\else
1179 \hbox{}\thispagestyle{empty}\newpage
1180 \if@twocolumn\hbox{}\newpage\fi
1181 \fi\fi}
1182 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1183 \ifodd\c@page
1184 \hbox{}\thispagestyle{empty}\newpage
1185 \if@twocolumn\hbox{}\newpage\fi
1186 \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。

```

1187 \let\pltx@cleartorightpage\pltx@cleartooddpage
1188 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

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

```

1189 %<*book|report>
1190 \if@openleft
1191 \let\cleardoublepage\pltx@cleartoleftpage
1192 \else\if@openright
1193 \let\cleardoublepage\pltx@cleartorightpage
1194 \fi\fi
1195 %</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。

```
1196 % \def\ps@empty{%
1197 %   \let\@mkboth\@gobbletwo
1198 %   \let\@oddhead\@empty
1199 %   \let\@oddfoot\@empty
1200 %   \let\@evenhead\@empty
1201 %   \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。

```
1202 \def\ps@plainfoot{%
1203   \let\@mkboth\@gobbletwo
1204   \let\@oddhead\@empty
1205   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1206   \let\@evenhead\@empty
```

```

1207 \let\@evenfoot\@oddfoot}
1208 \def\ps@plainhead{%
1209 \let\@mkboth\@gobbletwo
1210 \let\@oddfoot\@empty
1211 \let\@evenfoot\@empty
1212 \def\@evenhead{%
1213 \if@mparswitch \hss \fi
1214 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1215 \if@mparswitch\else \hss \fi}%
1216 \def\@oddhead{%
1217 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1218 %<book>\let\ps@plain\ps@plainhead
1219 %<!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。

```

1220 %<*article|slide>
1221 \if@twoside
1222 \def\ps@headings{%
1223 \let\@oddfoot\@empty
1224 \let\@evenfoot\@empty
1225 \def\@evenhead{\if@mparswitch \hss \fi
1226 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1227 \if@mparswitch\else \hss \fi}%
1228 \def\@oddhead{%
1229 \underline{%
1230 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1231 \let\@mkboth\markboth
1232 \def\sectionmark##1{\markboth{%
1233 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
1234 ##1}{}}%
1235 \def\subsectionmark##1{\markright{%
1236 \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsw\fi
1237 ##1}}%
1238 }
1239 \else % if not twoside
1240 \def\ps@headings{%
1241 \let\@oddfoot\@empty
1242 \def\@oddhead{%
1243 \underline{%
1244 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1245 \let\@mkboth\markboth
1246 \def\sectionmark##1{\markright{%
1247 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
1248 ##1}}%
1249 \fi
1250 %</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。

```
1251 %<*book|report>
1252 \def\bxjs@maybe@autoxspacing{%
1253   \ifx\autoxspacing\undefined\else \autoxspacing \fi}
1254 \newif\ifomit@number
1255 \def\ps@headings{%
1256   \let\@oddfont\empty
1257   \let\@evenfont\empty
1258   \def\@evenhead{%
1259     \if@mparswitch \hss \fi
1260     \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1261       \textbf{\thepage}\hfil\leftmark}}}%
1262     \if@mparswitch\else \hss \fi}%
1263   \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1264     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1265   \let\@mkboth\markboth
1266   \def\chaptermark##1{\markboth{%
1267     \ifnum \c@secnumdepth >\m@ne
1268       \if@mainmatter
1269         \ifomit@number\else
1270           \@chapapp\thechapter\@chappos\hskip1\jsw
1271         \fi
1272       \fi
1273     \fi
1274     ##1}{}}}%
1275   \def\sectionmark##1{\markright{%
1276     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
1277     ##1}}}%
1278 %</book|report>
```

最後は学会誌の場合です。

```
1279 %<*jspf>
1280 \def\ps@headings{%
1281   \def\@oddfont{\normalfont\hfil\thepage\hfil}
1282   \def\@evenfont{\normalfont\hfil\thepage\hfil}
1283   \def\@oddhead{\normalfont\hfil \@title \hfil}
1284   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1285 %</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。

```

1286 \def\ps@myheadings{%
1287   \let\@oddfont\@empty\let\@evenfont\@empty
1288   \def\@evenhead{%
1289     \if@mparswitch \hss \fi%
1290     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1291     \if@mparswitch\else \hss \fi}%
1292   \def\@oddhead{%
1293     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1294   \let\@mkboth\@gobbletwo
1295   %<book|report> \let\chaptermark\@gobble
1296   \let\sectionmark\@gobble
1297   %<!book&!report> \let\subsectionmark\@gobble
1298 }

```

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 1299 % \newcommand*\title}[1]{\gdef\@title{#1}}
1300 % \newcommand*\author}[1]{\gdef\@author{#1}}
1301 % \newcommand*\date}[1]{\gdef\@date{#1}}
1302 % \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。

```

1303 \newcommand*\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1304 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1305 \renewcommand*\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1306 \AtBeginDocument{\bxjs@decl@subtitle}
1307 \def\bxjs@decl@subtitle{%
1308   \global\let\bxjs@decl@subtitle\relax
1309   \ifx\subtitle\@undefined
1310     \global\let\subtitle\jsSubtitle
1311   \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。

```

1312 \def\bxjs@annihilate@subtitle{%

```

```

1313 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1314 \global\let\jsSubtitle\relax}

```

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

```

\author 1315 %<*\jspf>
\keywords 1316 \newcommand*\etitle}[1]{\gdef\@etitle{#1}}
1317 \newcommand*\eauthor}[1]{\gdef\@eauthor{#1}}
1318 \newcommand*\keywords}[1]{\gdef\@keywords{#1}}
1319 \newcommand*\email}[1]{\gdef\authors@mail{#1}}
1320 \newcommand*\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1321 %</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。

```

1322 \def\plainifnotempty{%
1323 \ifx \@oddhead \@empty
1324 \ifx \@oddfoot \@empty
1325 \else
1326 \thispagestyle{plainfoot}%
1327 \fi
1328 \else
1329 \thispagestyle{plainhead}%
1330 \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。

```

1331 %<*\article|book|report|slide>
1332 \if@titlepage
1333 \newcommand{\maketitle}{%
1334 \begin{titlepage}%
1335 \let\footnotesize\small
1336 \let\footnoterule\relax
1337 \let\footnote\thanks
1338 \null\vfil
1339 \if@slide
1340 {\footnotesize \@date}%
1341 \begin{center}
1342 \mbox{} \\\[1\jsZw]
1343 \large

```

```

1344         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1345         \jsc@smallskip
1346         \@title
1347         \ifx\bxjs@subtitle\@undefined\else
1348             \par\vskip\z@
1349             {\small \bxjs@subtitle\par}
1350         \fi
1351         \jsc@smallskip
1352         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1353         \vfill
1354         {\small \@author}%
1355     \end{center}
1356 \else
1357 \vskip 60\p@?
1358 \begin{center}%
1359     {\LARGE \@title \par}%
1360     \ifx\bxjs@subtitle\@undefined\else
1361         \vskip5\p@?
1362         {\normalsize \bxjs@subtitle\par}
1363     \fi
1364     \vskip 3em%
1365     {\large
1366         \lineskip .75em
1367         \begin{tabular}[t]{c}%
1368             \@author
1369         \end{tabular}\par}%
1370     \vskip 1.5em
1371     {\large \@date \par}%
1372 \end{center}%
1373 \fi
1374 \par
1375 \@thanks\vfil\null
1376 \end{titlepage}%
1377 \setcounter{footnote}{0}%
1378 \global\let\thanks\relax
1379 \global\let\maketitle\relax
1380 \global\let\@thanks\@empty
1381 \global\let\@author\@empty
1382 \global\let\@date\@empty
1383 \global\let\@title\@empty
1384 \global\let\title\relax
1385 \global\let\author\relax
1386 \global\let\date\relax
1387 \global\let\and\relax
1388 \bxjs@annihilate@subtitle
1389 }%
1390 \else
1391 \newcommand{\maketitle}{\par
1392     \begin{group

```

```

1393 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1394 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1395 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1396 \parindent 1\jsZw\noindent
1397 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1398 \if@twocolumn
1399 \ifnum \col@number=\@ne
1400 \maketitle
1401 \else
1402 \twocolumn[\maketitle]%
1403 \fi
1404 \else
1405 \newpage
1406 \global\@topnum\z@ % Prevents figures from going at top of page.
1407 \maketitle
1408 \fi
1409 \plainifnotempty
1410 \@thanks
1411 \endgroup
1412 \setcounter{footnote}{0}%
1413 \global\let\thanks\relax
1414 \global\let\maketitle\relax
1415 \global\let\@thanks\@empty
1416 \global\let\@author\@empty
1417 \global\let\@date\@empty
1418 \global\let\@title\@empty
1419 \global\let\title\relax
1420 \global\let\author\relax
1421 \global\let\date\relax
1422 \global\let\and\relax
1423 \bxjs@annihilate@subtitle
1424 }

```

\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。

```

1425 \def\maketitle{%
1426 \newpage\null
1427 \vskip 2em
1428 \begin{center}%
1429 \let\footnote\thanks
1430 {\LARGE \@title \par}%
1431 \ifx\bxjs@subtitle\undefined\else
1432 \vskip3\p@?
1433 {\normalsize \bxjs@subtitle\par}
1434 \fi
1435 \vskip 1.5em
1436 {\large
1437 \lineskip .5em
1438 \begin{tabular}[t]{c}%
1439 \author

```

```

1440     \end{tabular}\par}%
1441     \vskip 1em
1442     {\large \@date}%
1443     \end{center}%
1444     \par\vskip 1.5em
1445 %<article|slide>     \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1446 }
1447 \fi
1448 %</article|book|report|slide>
1449 %<*jspf>
1450 \newcommand{\maketitle}{\par
1451   \begin{group}
1452     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1453     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1454     \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1455       \parindent 1\jsZw\noindent
1456       \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1457     \twocolumn[\@maketitle]%
1458     \plainifnotempty
1459     \@thanks
1460   \end{group}
1461   \setcounter{footnote}{0}%
1462   \global\let\thanks\relax
1463   \global\let\maketitle\relax
1464   \global\let\@thanks\@empty
1465   \global\let\@author\@empty
1466   \global\let\@date\@empty
1467 % \global\let\@title\@empty % \@title は柱に使う
1468   \global\let\title\relax
1469   \global\let\author\relax
1470   \global\let\date\relax
1471   \global\let\and\relax
1472   \ifx\authors@mail\@undefined\else{%
1473     \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1474     \footnotetext[0]{\itshape\authors@mail}%
1475   }\fi
1476   \global\let\authors@mail\@undefined}
1477 \def\@maketitle{%
1478   \newpage\null
1479   \vskip 6em % used to be 2em
1480   \begin{center}
1481     \let\footnote\thanks
1482     \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1483     \lineskip .5em
1484     \ifx\@author\@undefined\else
1485       \vskip 1em
1486       \begin{tabular}[t]{c}%
1487         \@author
1488       \end{tabular}\par

```

```

1489 \fi
1490 \ifx\@etitle\@undefined\else
1491 \vskip 1em
1492 {\large \@etitle \par}%
1493 \fi
1494 \ifx\@eauthor\@undefined\else
1495 \vskip 1em
1496 \begin{tabular}[t]{c}%
1497 \@eauthor
1498 \end{tabular}\par
1499 \fi
1500 \vskip 1em
1501 \@date
1502 \end{center}
1503 \vskip 1.5em
1504 \centerline{\box\@abstractbox}
1505 \ifx\@keywords\@undefined\else
1506 \vskip 1.5em
1507 \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1508 \fi
1509 \vskip 1.5em}
1510 %</jspf>

```

8.2 章・節

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

```

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

```

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

次は `\@startsection` の定義です。情報処理学会論文誌スタイルファイル (`ipsjcommon.sty`) を参考にさせていただきましたが、完全に行送り `\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。

```
1511 \def\@startsection#1#2#3#4#5#6{%
1512   \if@noskipsec \leavevmode \fi
1513   \par
1514 % 見出し上の空きを \@tempskipa にセットする
1515   \@tempskipa #4\relax
1516 % \@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
1517   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1518 % 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
1519   \ifdim \@tempskipa <\z@
1520     \@tempskipa -\@tempskipa \@afterindentfalse
1521   \fi
1522   \if@nobreak
1523 %   \everypar{\everyparhook}% これは間違い
1524     \everypar{}%
1525   \else
1526     \addpenalty\@secpenalty
1527 % 次の行は削除
1528 %   \addvspace\@tempskipa
1529 % 次の \noindent まで追加
1530     \ifdim \@tempskipa >\z@
1531       \if@slide\else
1532         \null
1533         \vspace*{-\baselineskip}%
1534       \fi
1535       \vskip\@tempskipa
1536     \fi
1537   \fi
1538   \noindent
1539 % 追加終わり
1540   \@ifstar
1541     {\@ssect{#3}{#4}{#5}{#6}}%
1542     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}
```

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

```
1543 \def\@sect#1#2#3#4#5#6[#7]#8{%
1544   \ifnum #2>\c@secnumdepth
1545     \let\@svsec\@empty
1546   \else
1547     \refstepcounter{#1}%
1548     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1549   \fi
1550 % 見出し後の空きを \@tempskipa にセット
1551   \@tempskipa #5\relax
```

```

1552 % 条件判断の順序を入れ替えました
1553 \ifdim \@tempskipa<\z@
1554 \def\@svsechd{%
1555 #6{\hskip #3\relax
1556 \@svsec #8}%
1557 \csname #1mark\endcsname{#7}%
1558 \addcontentsline{toc}{#1}{%
1559 \ifnum #2>\c@secnumdepth \else
1560 \protect\numberline{\csname the#1\endcsname}%
1561 \fi
1562 #7}% 目次にフルネームを載せるなら #8
1563 \else
1564 \begingroup
1565 \interlinepenalty \@M % 下から移動
1566 #6{%
1567 \@hangfrom{\hskip #3\relax\@svsec}%
1568 % \interlinepenalty \@M % 上に移動
1569 #8\@@par}%
1570 \endgroup
1571 \csname #1mark\endcsname{#7}%
1572 \addcontentsline{toc}{#1}{%
1573 \ifnum #2>\c@secnumdepth \else
1574 \protect\numberline{\csname the#1\endcsname}%
1575 \fi
1576 #7}% 目次にフルネームを載せるならここは #8
1577 \fi
1578 \@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。

```

1579 \def\@xsect#1{%
1580 % 見出しの後ろの空きを \@tempskipa にセット
1581 \@tempskipa #1\relax
1582 % 条件判断の順序を変えました
1583 \ifdim \@tempskipa<\z@
1584 \nobreakfalse
1585 \global\@noskipsectrue
1586 \everypar{%
1587 \if@noskipsec
1588 \global\@noskipsecfalse
1589 {\setbox\z@\lastbox}%
1590 \clubpenalty\@M
1591 \begingroup \@svsechd \endgroup
1592 \unskip
1593 \@tempskipa #1\relax
1594 \hskip -\@tempskipa

```

```

1595     \bxjs@ltj@inhibitglue
1596     \else
1597     \clubpenalty \@clubpenalty
1598     \everypar{\everyparhook}%
1599     \fi\everyparhook}%
1600 \else
1601     \par \nobreak
1602     \vskip \@tempskipa
1603     \@afterheading
1604 \fi
1605 \if@slide
1606     {\vskip\if@twocolumn-5\jsc@mpt\else-6\jsc@mpt\fi
1607     \maybeblue\hrule height0\jsc@mpt depth1\jsc@mpt
1608     \vskip\if@twocolumn 4\jsc@mpt\else 7\jsc@mpt\fi\relax}%
1609 \fi
1610 \par % 2000-12-18
1611 \ignorespaces}
1612 \def\@ssect#1#2#3#4#5{%
1613     \@tempskipa #3\relax
1614     \ifdim \@tempskipa<\z@
1615         \def\@svsechd{#4{\hskip #1\relax #5}}%
1616     \else
1617         \begingroup
1618             #4{%
1619                 \@hangfrom{\hskip #1}%
1620                 \interlinepenalty \@M #5\@@par}%
1621         \endgroup
1622     \fi
1623     \@xsect{#3}}

```

上記の定義中の `\bxjs@ltj@inhibitglue` は `LuaTeX-j` で用いられるフック。

```

1624 \let\bxjs@ltj@inhibitglue\@empty

```

■ 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 `LATEX` 本体で定義済みです。

```

\subsectionmark 1625 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1626 % \newcommand*\sectionmark[1]{}
1627 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1628 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1629 % \newcommand*\paragraphmark[1]{}
1630 % \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。

```
1631 %<!book&!report>\setcounter{secnumdepth}{3}
1632 %<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 1633 \newcounter{part}
1634 %<book|report>\newcounter{chapter}
\c@subsubsection 1635 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1636 %<!book&!report>\newcounter{section}
\c@subparagraph 1637 \newcounter{subsection}[section]
1638 \newcounter{subsubsection}[subsection]
1639 \newcounter{paragraph}[subsubsection]
1640 \newcounter{subparagraph}[paragraph]
```

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

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

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```
1641 \renewcommand{\thepart}{\@Roman\c@part}
1642 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1643 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1644 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1645 %<*book|report>
1646 \renewcommand{\thechapter}{\@arabic\c@chapter}
1647 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1648 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1649 %</book|report>
1650 \renewcommand{\thesubsubsection}{%
1651   \thesubsection.\@arabic\c@subsubsection}
1652 \renewcommand{\theparagraph}{%
1653   \thesubsubsection.\@arabic\c@paragraph}
1654 \renewcommand{\thesubparagraph}{%
1655   \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。

```
1656 %<book|report>\newcommand{\@chapapp}{\prechaptername}
```

```
1657 %<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` かどうかに依らず奇数ページまで繰るように修正することで、問題を解消しました。実は、`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 `pLATEX` の標準クラス 2017/03/05 も参照)。

```
1658 %<*book|report>
1659 \newcommand\frontmatter{%
1660   \pltx@cleartooddpage
1661   \@mainmatterfalse
1662   \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。

```
1663 \newcommand\mainmatter{%
1664   \pltx@cleartooddpage
1665   \@mainmattertrue
1666   \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。

```
1667 \newcommand\backmatter{%
1668   \if@openleft
1669     \cleardoublepage
1670   \else\if@openright
1671     \cleardoublepage
1672   \else
1673     \clearpage
1674   \fi\fi
1675   \@mainmatterfalse}
1676 %</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。

```

1677 %<!*book&!report>
1678 \newcommand\part{%
1679   \if@noskipsec \leavevmode \fi
1680   \par
1681   \addvspace{4ex}%
1682   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1683   \secdef\@part\@spart}
1684 %</!*book&!report>

```

book および report クラスの場合は、少し複雑です。

```

1685 %<*book|report>
1686 \newcommand\part{%
1687   \if@openleft
1688     \cleardoublepage
1689   \else\if@openright
1690     \cleardoublepage
1691   \else
1692     \clearpage
1693   \fi\fi
1694   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1695   \if@twocolumn
1696     \onecolumn
1697     \@restonecoltrue
1698   \else
1699     \@restonecolfalse
1700   \fi
1701   \null\vfil
1702   \secdef\@part\@spart}
1703 %</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。

```

1704 %<!*book&!report>
1705 \def\@part[#1]#2{%
1706   \ifnum \c@secnumdepth >\m@ne
1707     \refstepcounter{part}%
1708     \addcontentsline{toc}{part}{%
1709       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1710   \else
1711     \addcontentsline{toc}{part}{#1}%
1712   \fi
1713   \markboth{}{}%
1714   {\parindent\z@

```

```

1715 \raggedright
1716 \interlinepenalty \@M
1717 \normalfont
1718 \ifnum \c@secnumdepth >\m@ne
1719 \Large\headfont\prepartname\thepart\postpartname
1720 \par\nobreak
1721 \fi
1722 \huge \headfont #2%
1723 \markboth{}{}\par}%
1724 \nobreak
1725 \vskip 3ex
1726 \@afterheading}
1727 %</*!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。

```

1728 %<*book|report>
1729 \def\@part[#1]#2{%
1730 \ifnum \c@secnumdepth >-2\relax
1731 \refstepcounter{part}%
1732 \addcontentsline{toc}{part}{%
1733 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1734 \else
1735 \addcontentsline{toc}{part}{#1}%
1736 \fi
1737 \markboth{}{}\par}%
1738 {\centering
1739 \interlinepenalty \@M
1740 \normalfont
1741 \ifnum \c@secnumdepth >-2\relax
1742 \huge\headfont \prepartname\thepart\postpartname
1743 \par\vskip20\p@?
1744 \fi
1745 \Huge \headfont #2\par}%
1746 \@endpart}
1747 %</book|report>

```

\@spart 番号を付けない部です。

```

1748 %<*!book&!report>
1749 \def\@spart#1{%
1750 \parindent \z@ \raggedright
1751 \interlinepenalty \@M
1752 \normalfont
1753 \huge \headfont #1\par}%
1754 \nobreak
1755 \vskip 3ex
1756 \@afterheading}
1757 %</*!book&!report>
1758 %<*book|report>
1759 \def\@spart#1{%

```

```

1760 \centering
1761 \interlinepenalty \@M
1762 \normalfont
1763 \Huge \headfont #1\par}%
1764 \@endpart}
1765 %</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。

```

1766 %<*book|report>
1767 \def\@endpart{\vfil\newpage
1768 \if@twoside
1769 \if@openleft %% added (2017/02/24)
1770 \null\thispagestyle{empty}\newpage
1771 \else\if@openright %% added (2016/12/13)
1772 \null\thispagestyle{empty}\newpage
1773 \fi\fi %% added (2016/12/13, 2017/02/24)
1774 \fi
1775 \if@restonecol
1776 \twocolumn
1777 \fi}
1778 %</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。

```

1779 %<*book|report>
1780 \newcommand{\chapter}{%
1781 \if@openleft\cleardoublepage\else
1782 \if@openright\cleardoublepage\else\clearpage\fi\fi
1783 \plainifnotempty % 元: \thispagestyle{plain}
1784 \global\@topnum\z@
1785 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1786 \secdef
1787 {\@omit@numberfalse\@chapter}%
1788 {\@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。

```

1789 \def\@chapter[#1]#2{%
1790 \ifnum \c@secnumdepth >\m@ne
1791 \if@mainmatter
1792 \refstepcounter{chapter}%

```

```

1793 \typeout{\@chapapp\thechapter\@chappos}%
1794 \addcontentsline{toc}{chapter}%
1795 {\protect\numberline
1796 %   {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
1797   {\@chapapp\thechapter\@chappos}%
1798   #1}%
1799 \else\addcontentsline{toc}{chapter}{#1}\fi
1800 \else
1801 \addcontentsline{toc}{chapter}{#1}%
1802 \fi
1803 \chaptermark{#1}%
1804 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
1805 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
1806 \if@twocolumn
1807 \topnewpage[\@makechapterhead{#2}]%
1808 \else
1809 \makechapterhead{#2}%
1810 \afterheading
1811 \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。

```

1812 \def\@makechapterhead#1{%
1813 \vspace*{2\Cvs}% 欧文は 50pt
1814 {\parindent \z@ \raggedright \normalfont
1815 \ifnum \c@secnumdepth >\m@ne
1816 \if@mainmatter
1817 \huge\headfont \@chapapp\thechapter\@chappos
1818 \par\nobreak
1819 \vskip \Cvs % 欧文は 20pt
1820 \fi
1821 \fi
1822 \interlinepenalty\@M
1823 \Huge \headfont #1\par\nobreak
1824 \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。

```

1825 \def\@schapter#1{%
1826 \chaptermark{#1}%
1827 \if@twocolumn
1828 \topnewpage[\@makeschapterhead{#1}]%
1829 \else
1830 \makeschapterhead{#1}\@afterheading
1831 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

1832 \def\@makeschapterhead#1{%
1833 \vspace*{2\Cvs}% 欧文は 50pt
1834 {\parindent \z@ \raggedright
1835 \normalfont

```

```

1836 \interlinepenalty\@M
1837 \Huge \headfont #1\par\nobreak
1838 \vskip 3\Cvs}} % 欧文は 40pt
1839 %</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。

```

1840 \if@twocolumn
1841 \newcommand{\section}{%
1842 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
1843 \@startsection{section}{1}{\z@}%
1844 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%
1845 %<kiyou> {\Cvs}{0.5\Cvs}%
1846 % {\normalfont\large\headfont\@secapp}}
1847 {\normalfont\large\headfont\raggedright}}
1848 \else
1849 \newcommand{\section}{%
1850 \if@slide\clearpage\fi
1851 \@startsection{section}{1}{\z@}%
1852 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1853 {.5\Cvs \@plus.3\Cdp}% 後アキ
1854 % {\normalfont\Large\headfont\@secapp}}
1855 {\normalfont\Large\headfont\raggedright}}
1856 \fi

```

`\subsection` 同上です。

```

1857 \if@twocolumn
1858 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1859 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1860 {\normalfont\normalsize\headfont}}
1861 \else
1862 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1863 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1864 {.5\Cvs \@plus.3\Cdp}% 後アキ
1865 {\normalfont\large\headfont}}
1866 \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)。

```

1867 \if@twocolumn
1868 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1869 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1870 {\normalfont\normalsize\headfont}}
1871 \else
1872 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%

```

```

1873    {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1874    {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
1875    {\normalfont\normalsize\headfont}}
1876 \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。

```

1877 \newcommand\jsParagraphMark{\jsJaChar{■}}
1878 \ifx\bxjs@paragraph@mark\@empty
1879   \let\jsParagraphMark\@empty
1880 \else\ifx\bxjs@paragraph@mark\@undefined\else
1881   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
1882 \fi\fi
1883 \let\jsJaChar\@empty
1884 \if@twocolumn
1885   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1886     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1887 %<jspf>   {\normalfont\normalsize\headfont}}
1888 %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
1889 \else
1890   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1891     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1892     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1893 %<jspf>   {\normalfont\normalsize\headfont}}
1894 %<!jspf>   {\normalfont\normalsize\headfont\jsParagraphMark}}
1895 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

1896 \if@twocolumn
1897   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1898     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1899     {\normalfont\normalsize\headfont}}
1900 \else
1901   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%

```

```

1902     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1903     {\normalfont\normalsize\headfont}}
1904 \fi

```

8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

1905 \if@slide
1906   \setlength\leftmargini{1\jsZw}
1907 \else
1908   \if@twocolumn
1909     \setlength\leftmargini{2\jsZw}
1910   \else
1911     \setlength\leftmargini{3\jsZw}
1912   \fi
1913 \fi

```

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

```

\leftmarginiv 1914 \if@slide
\leftmarginv 1915   \setlength\leftmarginii {1\jsZw}
\leftmarginvi 1916   \setlength\leftmarginiii{1\jsZw}
1917   \setlength\leftmarginiv {1\jsZw}
1918   \setlength\leftmarginv  {1\jsZw}
1919   \setlength\leftmarginvi {1\jsZw}
1920 \else
1921   \setlength\leftmarginii {2\jsZw}
1922   \setlength\leftmarginiii{2\jsZw}
1923   \setlength\leftmarginiv {2\jsZw}
1924   \setlength\leftmarginv  {1\jsZw}
1925   \setlength\leftmarginvi {1\jsZw}
1926 \fi

```

`\labelsep` `\labelwidth` はラベルと本文の間の距離です。 `\labelwidth` はラベルの幅です。これは二分に変えました。

```

1927 \setlength \labelsep {0.5\jsZw} % .5em
1928 \setlength \labelwidth{\leftmargini}
1929 \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。

```
1930 \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 1931 \@beginparpenalty -\@lowpenalty
\@itempenalty 1932 \@endparpenalty -\@lowpenalty
1933 \@itempenalty -\@lowpenalty
```

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

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

```
1934 \def\@listi{\leftmargin\leftmarginI
1935 \parsep \z@
1936 \topsep 0.5\baselineskip
1937 \itemsep \z@ \relax}
1938 \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)。

```
1939 \@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 1940 \def\@listii{\leftmargin\leftmarginii
\@listiv 1941 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1942 \topsep \z@
\@listv 1943 \parsep \z@
\@listvi 1944 \itemsep\parsep}
1945 \def\@listiii{\leftmargin\leftmarginiii
1946 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1947 \topsep \z@
1948 \parsep \z@
1949 \itemsep\parsep}
1950 \def\@listiv {\leftmargin\leftmarginiv
1951 \labelwidth\leftmarginiv
1952 \advance\labelwidth-\labelsep}
1953 \def\@listv {\leftmargin\leftmarginv
1954 \labelwidth\leftmarginv
1955 \advance\labelwidth-\labelsep}
1956 \def\@listvi {\leftmargin\leftmarginvi
1957 \labelwidth\leftmarginvi
1958 \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
`\theenumii` 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
`\theenumiii` 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
`\theenumiv` 力する命令です。

```
1959 \renewcommand{\theenumi}{\@arabic\c@enumi}
1960 \renewcommand{\theenumii}{\@alph\c@enumii}
1961 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1962 \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
`\labelenumii` 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
`\labelenumiii` 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。

```
1963 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1964 \newcommand*\jsInJaParen[1]{%
1965   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1966 \newcommand{\labelenumi}{\theenumi.}
1967 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1968 \newcommand{\labelenumiii}{\theenumiii.}
1969 \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 1970 \renewcommand{\p@enumii}{\theenumi}
1971 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1972 \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 1973 \newcommand\labelitemi{\textbullet}
\labelitemiii 1974 \newcommand\labelitemii{\normalfont\bfseries \textendash}
1975 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 1976 \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。

```

1977 \newenvironment{description}{%
1978   \list{}{%
1979     \labelwidth=\leftmargin
1980     \labelsep=1\jsZw
1981     \advance \labelwidth by -\labelsep
1982     \let \makelabel=\descriptionlabel}}{\endlist}

```

`\descriptionlabel` `description` 環境のラベルを出力するコマンドです。好みに応じて #1 の前に適当な空き (たとえば `\hspace{1\jsZw}`) を入れるのもいいと思います。

```

1983 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}

```

■概要

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

bxjsreport クラスの abstract 環境は：

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

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

```

1984 %<*book|report>
1985 \newenvironment{chapterabstract}{%
1986   \begin{list}{}{%
1987     \listparindent=1\jsZw
1988     \itemindent=\listparindent
1989     \rightmargin=\z@
1990     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
1991 %</book|report>

```

“普通の” abstract 環境の定義。

```

1992 %<*article|report|slide>
1993 \newbox\@abstractbox
1994 \if@titlepage
1995   \newenvironment{abstract}{%
1996     \titlepage
1997     \null\vfil
1998     \@beginparpenalty\@lowpenalty
1999     \begin{center}%

```

```

2000     \headfont \abstractname
2001     \@endparpenalty\M
2002     \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。

```

2003     \par}%
2004     {\par\vfil\null\endtitlepage}
2005 \else
2006     \newenvironment{abstract}{%
2007         \if@twocolumn
2008             \ifx\maketitle\relax
2009                 \section*{\abstractname}%
2010             \else
2011                 \global\setbox\@abstractbox\hbox\bgroup
2012                 \begin{minipage}[b]{\textwidth}
2013                     \small\parindent1\jsZw
2014                     \begin{center}%
2015                         {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2016                     \end{center}%
2017                     \list{}{%
2018                         \listparindent\parindent
2019                         \itemindent \listparindent
2020                         \rightmargin \leftmargin}%
2021                     \item\relax
2022                 \fi
2023             \else
2024                 \small
2025                 \begin{center}%
2026                     {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2027                 \end{center}%
2028                 \list{}{%
2029                     \listparindent\parindent
2030                     \itemindent \listparindent
2031                     \rightmargin \leftmargin}%
2032                 \item\relax
2033             \fi}{\if@twocolumn
2034                 \ifx\maketitle\relax
2035                 \else
2036                     \endlist\end{minipage}\egroup
2037                 \fi
2038             \else
2039                 \endlist
2040             \fi}
2041 \fi
2042 %</article|report|slide>
2043 %<*.jspf>
2044 \newbox\@abstractbox
2045 \newenvironment{abstract}{%
2046     \global\setbox\@abstractbox\hbox\bgroup

```

```

2047 \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
2048 \small
2049 \if@english \parindent6mm \else \parindent1\jsZw \fi}%
2050 {\end{minipage}\egroup}
2051 %</jspf>

```

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

```

2052 %<*book|report>
2053 \ifbxjs@force@chapterabstract
2054 \let\abstract\chapterabstract
2055 \let\endabstract\endchapterabstract
2056 \fi
2057 %</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。

```

2058 %<*jspf>
2059 %\newbox\@keywordsbox
2060 %\newenvironment{keywords}{%
2061 % \global\setbox\@keywordsbox\hbox\bgroup
2062 % \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
2063 % \small\parindent0\jsZw}%
2064 % {\end{minipage}\egroup}
2065 %</jspf>

```

■verse 環境

verse 詩のための verse 環境です。

```

2066 \newenvironment{verse}{%
2067 \let \=\@centercr
2068 \list{}{%
2069 \itemsep \z@
2070 \itemindent -2\jsZw % 元: -1.5em
2071 \listparindent\itemindent
2072 \rightmargin \z@
2073 \advance\leftmargin 2\jsZw}% 元: 1.5em
2074 \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。

```

2075 \newenvironment{quotation}{%
2076 \list{}{%

```

```

2077 \listparindent\parindent
2078 \itemindent\listparindent
2079 \rightmargin \z0}%
2080 \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。

```

2081 \newenvironment{quote}%
2082 {\list}{\rightmargin\z0}\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。

```

2083 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2084 \item[\hskip \labelsep{\headfont #1\ #2}]}
2085 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2086 \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。

```

2087 \newenvironment{titlepage}{%
2088 %<book> \pltx@cleartooddpage %% 2017-02-24
2089 \if@twocolumn
2090 \@restonecoltrue\onecolumn
2091 \else
2092 \@restonecolfalse\newpage
2093 \fi
2094 \thispagestyle{empty}%
2095 \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-
24
2096 }%
2097 {\if@restonecol\twocolumn \else \newpage \fi
2098 \if@twoside\else

```

```
2099 \setcounter{page}\@ne
2100 \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2101 %<!*book&!report>
2102 \newcommand{\appendix}{\par
2103 \setcounter{section}{0}%
2104 \setcounter{subsection}{0}%
2105 \gdef\presectionname{\appendixname}%
2106 \gdef\postsectionname{}}%
2107 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2108 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2109 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
2110 %</!*book&!report>
2111 %<*book|report>
2112 \newcommand{\appendix}{\par
2113 \setcounter{chapter}{0}%
2114 \setcounter{section}{0}%
2115 \gdef\@chapapp{\appendixname}%
2116 \gdef\@chappos{}}%
2117 \gdef\thechapter{\@Alph\c@chapter}}
2118 %</book|report>
```

8.4 パラメータの設定

■array と tabular 環境

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

```
2119 \setlength\arraycolsep{5\p@?}
```

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

```
2120 \setlength\tabcolsep{6\p@?}
```

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

```
2121 \setlength\arrayrulewidth{.4\p@}
```

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

```
2122 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2123 \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。

```
2124 \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。

```
2125 \setlength\fboxsep{3\p@?}
2126 \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。

```
2127 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2128 %<*book|report>
2129 \@addtoreset{equation}{chapter}
2130 \renewcommand\theequation
2131   {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
2132 %</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。

```
2133 % \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。

```
2134 % \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。

```
2135 % \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。

```
2136 %<!*book&!report>
2137 \newcounter{figure}
2138 \renewcommand \thefigure {\@arabic\c@figure}
2139 %</!book&!report>
2140 %<*book|report>
2141 \newcounter{figure}[chapter]
2142 \renewcommand \thefigure
2143     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2144 %</book|report>
```

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

```
\ext@figure 2145 \def\fps@figure{tbp}
2146 \def\ftype@figure{1}
\fnum@figure 2147 \def\ext@figure{lof}
2148 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2149 \newenvironment{figure}%
2150     {\@float{figure}}%
2151     {\end@float}
2152 \newenvironment{figure*}%
2153     {\@dblfloat{figure}}%
2154     {\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。

```
2155 %<!*book&!report>
2156 \newcounter{table}
2157 \renewcommand\thetable{\@arabic\c@table}
2158 %</!book&!report>
2159 %<*book|report>
2160 \newcounter{table}[chapter]
2161 \renewcommand \thetable
2162     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2163 %</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 2164 \def\fps@table{tbp}
\fnun@table 2165 \def\ftype@table{2}
2166 \def\ext@table{lot}
2167 \def\fnun@table{\tablename\nobreak\thetable}
```

`table *` は段抜きのフロートです。

```
table* 2168 \newenvironment{table}%
2169         {\@float{table}}%
2170         {\end@float}
2171 \newenvironment{table*}%
2172         {\@dblfloat{table}}%
2173         {\end@dblfloat}
```

8.6 キャプション

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

`\abovecaptionskip` それぞれキャプションの前後に挿入されるスペースです。`\belowcaptionskip` が 0 になっ
`\belowcaptionskip` ていたもので、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。

```
2174 \newlength\abovecaptionskip
2175 \newlength\belowcaptionskip
2176 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2177 \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。

```
2178 %<*\jpsf>
2179 % \long\def\@makecaption#1#2{{\small
2180 %   \advance\leftskip1cm
2181 %   \advance\rightskip1cm
2182 %   \vskip\abovecaptionskip
2183 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2184 %   \ifdim \wd\@tempboxa >\hsize
2185 %     #1\hskip1\jsZw\relax #2\par
2186 %   \else
2187 %     \global \@minipagefalse
2188 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2189 %   \fi
2190 %   \vskip\belowcaptionskip}}
2191 % \long\def\@makecaption#1#2{{\small
2192 %   \advance\leftskip .0628\linewidth
```

```

2193 \advance\rightskip .0628\linewidth
2194 \vskip\abovecaptionskip
2195 \sbox\@tempboxa{#1\zwspace#2}%
2196 \ifdim \wd\@tempboxa <\hsize \centering \fi
2197 #1\zwspace#2\par
2198 \vskip\belowcaptionskip}}
2199 %</!jspf>
2200 %<*jspf>
2201 \long\def\@makecaption#1#2{%
2202 \vskip\abovecaptionskip
2203 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2204 \ifdim \wd\@tempboxa >\hsize
2205 {\small\sffamily
2206 \list{#1}{%
2207 \renewcommand{\makelabel}[1]{##1\hfil}
2208 \itemsep \z@
2209 \itemindent \z@
2210 \labelsep \z@
2211 \labelwidth 11mm
2212 \listparindent\z@
2213 \leftmargin 11mm}\item\relax #2\endlist}
2214 \else
2215 \global \@minipagefalse
2216 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2217 \fi
2218 \vskip\belowcaptionskip}
2219 %</jspf>

```

9 フォントコマンド

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

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

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

```

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

```

```

2220 \newif\if@jsc@warnoldfontcmd
2221 \@jsc@warnoldfontcmdtrue

```

```

2222 \newif\if@jsc@warnoldfontcmdexception
2223 \@jsc@warnoldfontcmdexceptionfalse

```

\jsc@DeclareOldFontCommand

```

2224 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2225   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2226   \DeclareOldFontCommand{#1}{%
2227     \bxjs@oldfontcmd{#1}#2%
2228   }{%
2229     \bxjs@oldfontcmd{#1}#3%
2230   }%
2231 }
2232 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2233   \ClassInfo\bxjs@clsname
2234   {Old font command '\string#1' is used!!\MessageBreak
2235     The first occurrence is}%
2236 }

```

\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。

```

2237 \newcommand*{\allowoldfontcommands}{%
2238   \@jsc@warnoldfontcmdexceptiontrue}
2239 \newcommand*{\disallowoldfontcommands}{%
2240   \@jsc@warnoldfontcmdexceptionfalse}

```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2241 \let\bxjs@oldfontcmd@list\@empty
2242 \def\bxjs@oldfontcmd#1{%
2243   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2244 \def\bxjs@oldfontcmd@a#1#2{%
2245   \if@jsc@warnoldfontcmdexception\else
2246     \global\@jsc@warnoldfontcmdfalse
2247     \ifx#1\relax
2248       \global\let#1=t%
2249       \jsc@warnoldfontcmd{#2}%
2250     \fi
2251   \fi}
2252 \def\bxjs@warnoldfontcmd@final{%
2253 % \par
2254   \global\let\bxjs@warnoldfontcmd@final\@empty
2255   \let\@tempa\@empty
2256   \def\do##1{%
2257     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2258       \edef\@tempa{\@tempa \space\string##1}\fi}
2259   \bxjs@oldfontcmd@list
2260   \ifx\@tempa\@empty\else

```

```

2261 \ClassWarningNoLine\bxjs@clsname
2262 {Some old font commands were used in text\MessageBreak
2263 (see the log file for detail)}%
2264 \ClassInfo\bxjs@clsname
2265 {Some old font commands were used in text:\MessageBreak
2266 \space\@tempa\MessageBreak
2267 You should note, that since 1994 LaTeX2e provides a\MessageBreak
2268 new font selection scheme called NFSS2 with several\MessageBreak
2269 new, combinable font commands. The
2270 class provides\MessageBreak
2271 the old font commands
2272 only for compatibility%
2273 \@gobble}%
2274 \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。

```

2275 \def\bxjs@warnoldfontcmd@kick@final{%
2276 \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2277 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2278 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2279 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 2280 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf 2281 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2282 \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。

```

2283 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2284 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2285 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2286 \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 2287 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2288 \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` \geq `\@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。

```
2289 \newcommand\@pnumwidth{1.55em}
2290 \newcommand\@tocrmarg{2.55em}
2291 \newcommand\@dotsep{4.5}
2292 %<!book&!report>\setcounter{tocdepth}{2}
2293 %<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)

```
2294 \newdimen\jsc@tocl@width
2295 \newcommand{\tableofcontents}{%
2296 %<*book|report>
2297   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2298   \settowidth\@tempdima{\headfont\appendixname}%
2299   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2300   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2301   \if@twocolumn
2302     \@restonecoltrue\onecolumn
2303   \else
2304     \@restonecolfalse
2305   \fi
2306   \chapter*{\contentsname}%
2307   \@mkboth{\contentsname}{}%
2308 %</book|report>
2309 %<!*book&!report>
2310   \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2311   \settowidth\@tempdima{\headfont\appendixname}%
2312   \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2313   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2314   \section*{\contentsname}%
2315   \@mkboth{\contentsname}{\contentsname}%
2316 %</!book&!report>
2317   \@starttoc{toc}%
2318 %<book|report> \if@restonecol\twocolumn\fi
2319 }
```

`\l@part` 部の目次です。

```
2320 \newcommand*{\l@part}[2]{%
2321   \ifnum \c@tocdepth >-2\relax
2322 %<!book&!report> \addpenalty\@secpenalty
2323 %<book|report> \addpenalty{-\@highpenalty}%
2324   \addvspace{2.25em \@plus\p@?}%
2325   \begingroup
2326     \parindent \z@
2327 % \@pnumwidth should be \@tocrmarg
2328 % \rightskip \@pnumwidth
2329 \rightskip \@tocrmarg
2330 \parfillskip -\rightskip
2331 {\leavevmode
2332   \large \headfont
2333   \setlength\@lnumwidth{4\jsZw}%
2334   #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2335   \nobreak
2336 %<book|report> \global\@nobreaktrue
2337 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
```

```
2338 \endgroup
2339 \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)

```
2340 %<*book|report>
2341 \newcommand*{\l@chapter}[2]{%
2342 \ifnum \c@tocdepth >\m@ne
2343 \addpenalty{-\@highpenalty}%
2344 \addvspace{1.0em \@plus\p@?}
2345 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2346 \begingroup
2347 \parindent\z@
2348 % \rightskip\@pnumwidth
2349 \rightskip\@tocrmarg
2350 \parfillskip-\rightskip
2351 \leavevmode\headfont
2352 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2353 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2354 \advance\leftskip\@lnumwidth \hskip-\leftskip
2355 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2356 \penalty\@highpenalty
2357 \endgroup
2358 \fi}
2359 %</book|report>
```

`\l@section` 節の目次です。

```
2360 %<!*book&!report>
2361 \newcommand*{\l@section}[2]{%
2362 \ifnum \c@tocdepth >\z@
2363 \addpenalty{\@secpenalty}%
2364 \addvspace{1.0em \@plus\p@?}%
2365 \begingroup
2366 \parindent\z@
2367 % \rightskip\@pnumwidth
2368 \rightskip\@tocrmarg
2369 \parfillskip-\rightskip
2370 \leavevmode\headfont
2371 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2372 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2373 \advance\leftskip\@lnumwidth \hskip-\leftskip
2374 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2375 \endgroup
2376 \fi}
2377 %</!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。

```
2378 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

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

\l@section さらに下位レベルの目次項目の体裁です。あまり使ったことがありませんので、要修正かも
\l@subsection 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@subsection 2379 %<!*book&!report>
2380 % \newcommand*\l@subsection {\@dottedtocline{2}{1.5em}{2.3em}}
2381 % \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
2382 % \newcommand*\l@paragraph {\@dottedtocline{4}{7.0em}{4.1em}}
2383 % \newcommand*\l@subparagraph {\@dottedtocline{5}{10em}{5em}}
2384 %
2385 % \newcommand*\l@subsubsection{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2386 % \newcommand*\l@paragraph {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2387 % \newcommand*\l@subparagraph {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2388 %
2389 \newcommand*\l@subsection{%
2390     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2391     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2392 \newcommand*\l@subsubsection{%
2393     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2394     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2395 \newcommand*\l@paragraph{%
2396     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2397     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2398 \newcommand*\l@subparagraph{%
2399     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2400     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2401 %</!*book&!report>
2402 %<*book|report>
2403 % \newcommand*\l@subsection {\@dottedtocline{2}{3.8em}{3.2em}}
2404 % \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
2405 % \newcommand*\l@paragraph {\@dottedtocline{4}{10em}{5em}}
2406 % \newcommand*\l@subparagraph {\@dottedtocline{5}{12em}{6em}}
2407 \newcommand*\l@section{%
2408     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2409     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2410 \newcommand*\l@subsection{%
2411     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2412     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2413 \newcommand*\l@subsubsection{%
2414     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2415     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2416 \newcommand*\l@paragraph{%
2417     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2418     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2419 \newcommand*\l@subparagraph{%
2420     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2421     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
```

2422 %</book|report>

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

2423 \newdimen\@lnumwidth

2424 \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。

2425 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else

2426 \vskip \z@ \@plus.2\p@?

2427 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip

2428 \parindent #2\relax\@afterindenttrue

2429 \interlinepenalty\@M

2430 \leavevmode

2431 \@lnumwidth #3\relax

2432 \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip

2433 {#4}\nobreak

2434 \leaders\hbox{\$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep

2435 mu\$}\hfill \nobreak\hb@xt@\@pnumwidth{%

2436 \hfil\normalfont \normalcolor #5}\par}\fi}

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

2437 \newcommand{\listoffigures}{%

2438 %<*book|report>

2439 \if@twocolumn\@restonecoltrue\onecolumn

2440 \else\@restonecolfalse\fi

2441 \chapter*{\listfigurename}%

2442 \@mkboth{\listfigurename}{}

2443 %</book|report>

2444 %<!*book&!report>

2445 \section*{\listfigurename}%

2446 \@mkboth{\listfigurename}{\listfigurename}%

2447 %</!book&!report>

2448 \@starttoc{lof}%

2449 %<book|report> \if@restonecol\twocolumn\fi

2450 }

`\l@figure` 図目次の項目を出力します。

2451 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

`\listoftables` 表目次を出力します。

2452 \newcommand{\listoftables}{%

2453 %<*book|report>

```

2454 \if@twocolumn\@restonecoltrue\onecolumn
2455 \else\@restonecolfalse\fi
2456 \chapter*{\listtablename}%
2457 \@mkboth{\listtablename}{}%
2458 %</book|report>
2459 %<!*book&!report>
2460 \section*{\listtablename}%
2461 \@mkboth{\listtablename}{\listtablename}%
2462 %</!book&!report>
2463 \@starttoc{lot}%
2464 %<book|report> \if@restonecol\twocolumn\fi
2465 }

```

`\l@table` 表目次は図目次と同じです。

```
2466 \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。

```

2467 \newdimen\bibindent
2468 \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。

```

2469 \newenvironment{thebibliography}[1]{%
2470 \@jsc@warnoldfontcmdexceptiontrue
2471 \global\let\presectionname\relax
2472 \global\let\postsectionname\relax
2473 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2474 %<*kiyou>
2475 \vspace{1.5\baselineskip}
2476 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2477 \vspace{0.5\baselineskip}
2478 %</kiyou>
2479 %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2480 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2481 \list{\@biblabel{\@arabic\c@enumiv}}%
2482 {\settowidth\labelwidth{\@biblabel{#1}}%
2483 \leftmargin\labelwidth
2484 \advance\leftmargin\labelsep
2485 \@openbib@code
2486 \usecounter{enumiv}%
2487 \let\p@enumiv\@empty
2488 \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2489 %<kiyou> \small

```

```

2490 \sloppy
2491 \clubpenalty4000
2492 \@clubpenalty\clubpenalty
2493 \widowpenalty4000%
2494 \sfcode`\.\@m}
2495 {\def\@noitemerr
2496   {\@latex@warning{Empty `thebibliography' environment}}}%
2497 \endlist}

```

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

```
2498 \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。

```
2499 \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。

```
2500 % \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。

```

2501 % \def\@citex[#1]#2{\leavevmode
2502 %   \let\@citea\@empty
2503 %   \@cite{\@for\@citeb:=#2\do
2504 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m} }%
2505 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2506 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2507 %     \ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2508 %     \G@refundefinedtrue
2509 %     \@latex@warning
2510 %       {Citation `@\@citeb' on page \thepage \space undefined}}%
2511 %     {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2512 % \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。

```

2513 % \DeclareRobustCommand\cite{\unskip
2514 %   \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}]
2515 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2516 %   , \jsInhibitGlue\ #2\fi) }}$}

```

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)。

```
2517 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2518     \if@twocolumn
2519         \onecolumn\@restonecolfalse
2520     \else
2521         \clearpage\@restonecoltrue
2522     \fi
2523     \columnseprule.4pt \columnsep 2\jsZw
2524     \ifx\multicols\@undefined
2525 %<book|report>         \twocolumn[\@makeschapterhead{\indexname}]%
2526 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}]%
2527 %<!book&!report>     \def\presectionname{}\def\postsectionname{}%
2528 %<!book&!report>     \twocolumn[\section*{\indexname}]%
2529     \else
2530         \ifdim\textwidth<\fullwidth
2531             \setlength{\evensidemargin}{\oddsidemargin}
2532             \setlength{\textwidth}{\fullwidth}
2533             \setlength{\linewidth}{\fullwidth}
2534 %<book|report>         \begin{multicols}{3}[\chapter*{\indexname}]%
2535 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}]%
2536 %<!book&!report>     \def\presectionname{}\def\postsectionname{}%
2537 %<!book&!report>     \begin{multicols}{3}[\section*{\indexname}]%
2538     \else
2539 %<book|report>         \begin{multicols}{2}[\chapter*{\indexname}]%
2540 %<book|report>         \addcontentsline{toc}{chapter}{\indexname}]%
2541 %<!book&!report>     \def\presectionname{}\def\postsectionname{}%
2542 %<!book&!report>     \begin{multicols}{2}[\section*{\indexname}]%
2543     \fi
2544     \fi
2545 %<book|report>     \@mkboth{\indexname}{}%
2546 %<!book&!report>     \@mkboth{\indexname}{\indexname}%
2547     \plainifnotempty % \thispagestyle{plain}
2548     \parindent\z@
2549     \parskip\z@ \@plus .3\p@?\relax
2550     \let\item\@idxitem
2551     \raggedright
2552     \footnotesize\narrowbaselines
2553 }{
2554     \ifx\multicols\@undefined
2555         \if@restonecol\onecolumn\fi
2556     \else
2557         \end{multicols}
2558     \fi
2559     \clearpage
```

2560 }

`\@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` 2561 `\newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt`

`\subsubitem` 2562 `\newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt`

2563 `\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。

2564 `\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) などもいいでしょう。

2565 `\newcommand\seename{\if@english see\else →\fi}`

2566 `\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。

2567 `\ifx\pltx@foot@penalty\@undefined`

2568 `\let\footnotes@ve=\footnote`

2569 `\def\footnote{\jsInhibitGlue\footnotes@ve}`

2570 `\let\footnotemarks@ve=\footnotemark`

2571 `\def\footnotemark{\jsInhibitGlue\footnotemarks@ve}`

2572 `\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 `newttext` や `newpertext` の使用時におかしくなっています。これらのパッケージは内部で `\thefootnote` を再定義していますので、気になる場合はパッケージを読み込むときに `defaultsups` オプションを付けてください (qa:57284, qa:57287)。

```
2573 \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。

```
2574 % \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。

```
2575 \renewcommand{\footnoterule}{%
2576   \kern-2.6\p@? \kern-.4\p@
2577   \hrule width .4\columnwidth
2578   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2579 %<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 pL_AT_EX の「閉じ括弧類の直後に `\footnotetext` が続く場合に改行が起きることがある問題に対処」と同等のコードを追加しました。

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

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

```
2580 \long\def\@footnotetext{%
2581   \insert\footins\bgroup
2582     \normalfont\footnotesize
2583     \interlinepenalty\interfootnotelinepenalty
2584     \splittopskip\footnotesep
2585     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2586     \hsize\columnwidth \@parboxrestore
2587     \protected@edef\@currentlabel{%
2588       \csname p@footnote\endcsname\@thefnmark
2589     }%
2590     \color@begingroup
2591     \@makefnmark%
2592     \rule{\z@\footnotesep\ignorespaces}%
2593     \futurelet\next\fo@t}
2594 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
2595           \else \let\next\fo@t\fi \next}
2596 \def\fo@t{\bgroup\aftergroup\@foot\let\next}
2597 \def\fo@t#1{#1\@foot}
```

```

2598 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2599 \ifx\pltx@foot@penalty\@undefined\else
2600 \ifhmode\null\fi
2601 \ifnum\pltx@foot@penalty=\z@\else
2602 \penalty\pltx@foot@penalty
2603 \pltx@foot@penalty\z@
2604 \fi
2605 \fi}

```

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

```

2606 \newcommand\@makefntext[1]{%
2607 \advance\leftskip 3\jsZw
2608 \parindent 1\jsZw
2609 \noindent
2610 \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。

```

2611 % \def\@xfootnotenext[#1]{%
2612 % \begingroup
2613 % \ifnum#1>\z@
2614 % \csname c@\@mpfn\endcsname #1\relax
2615 % \unrestored@protected@xdef\@thefnmark{\thempfn}%
2616 % \else
2617 % \unrestored@protected@xdef\@thefnmark{}%
2618 % \fi
2619 % \endgroup
2620 % \@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。

```

2621 \let\jsInhibitGlueAtParTop\@empty

```

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

```
2622 \def\everyparhook{\jsInhibitGlueAtParTop}
2623 \AtBeginDocument{\everypar{\everyparhook}}
```

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

`\@inhibitglue` JS クラスでの `\jsInhibitGlueAtParTop` の実装。(これは (u)pTeX 専用である。)

```
2624 \def\@inhibitglue{%
2625   \futurelet\@let@token\@@inhibitglue}
2626 \begingroup
2627 \let\GDEF=\gdef
2628 \let\CATCODE=\catcode
2629 \let\ENDGROUP=\endgroup
2630 \CATCODE`k=12
2631 \CATCODE`a=12
2632 \CATCODE`n=12
2633 \CATCODE`j=12
2634 \CATCODE`i=12
2635 \CATCODE`c=12
2636 \CATCODE`h=12
2637 \CATCODE`r=12
2638 \CATCODE`t=12
2639 \CATCODE`e=12
2640 \GDEF\KANJI@CHARACTER{kanji character }
2641 \ENDGROUP
2642 \def\@@inhibitglue{%
2643   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
2644   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@ig@temp#1}%
2645   \def\jsc@ig@temp{#1}%
2646   \ifx\jsc@ig@temp\@empty
2647     \ifnum\the\inhibitxspcode`#2=2\relax
2648       \inhibitglue
2649     \fi
2650   \fi}
```

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。

```
2651 \def\@doendpe{%
2652   \endpetrue
2653   \def\par{%
2654     \@restorepar\clubpenalty\clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2655   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}
```

`\item` 命令の直後です。

```
2656 \def\@item[#1]{%
2657   \if@noparitem
2658     \donoparitem
2659   \else
2660     \if@inlabel
2661       \indent \par
2662     \fi
2663     \ifhmode
2664       \unskip\unskip \par
2665     \fi
2666     \if@newlist
2667       \if@nobreak
2668         \nbitem
2669       \else
2670         \addpenalty\@beginparpenalty
2671         \addvspace\@topsep
2672         \addvspace{-\parskip}%
2673       \fi
2674     \else
2675       \addpenalty\@itempenalty
2676       \addvspace\itemsep
2677     \fi
2678     \global\@inlabeltrue
2679   \fi
2680   \everypar{%
2681     \@minipagefalse
2682     \global\@newlistfalse
2683     \if@inlabel
2684       \global\@inlabelfalse
2685       {\setbox\z@\lastbox
2686        \ifvoid\z@
2687          \kern-\itemindent
2688        \fi}%
2689     \box\@labels
2690     \penalty\z@
2691   \fi
2692   \if@nobreak
2693     \@nobreakfalse
```

```

2694     \clubpenalty \@M
2695     \else
2696     \clubpenalty \@clubpenalty
2697     \everypar{\everyparhook}%
2698     \fi
2699     \bxjs@ltj@inhibitglue
2700     \everyparhook}%
2701 \if@noitemarg
2702 \@noitemargfalse
2703 \if@nmbrrlist
2704     \refstepcounter\@listctr
2705     \fi
2706 \fi
2707 \sbox\@tempboxa{\makelabel{#1}}%
2708 \global\setbox\@labels\hbox{%
2709     \unhbox\@labels
2710     \hskip \itemindent
2711     \hskip -\labelwidth
2712     \hskip -\labelsep
2713     \ifdim \wd\@tempboxa >\labelwidth
2714         \box\@tempboxa
2715     \else
2716         \hbox to\labelwidth {\unhbox\@tempboxa}%
2717     \fi
2718     \hskip \labelsep}%
2719 \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。

```

2720 \def\@afterheading{%
2721     \@nobreaktrue
2722     \everypar{%
2723         \if@nobreak
2724             \@nobreakfalse
2725             \clubpenalty \@M
2726             \if@afterindent \else
2727                 {\setbox\z@\lastbox}%
2728             \fi
2729         \else
2730             \clubpenalty \@clubpenalty
2731             \everypar{\everyparhook}%
2732         \fi\everyparhook}}

```

`\@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。

```

2733 \def\@gnewline #1{%
2734   \ifvmode
2735     \@nolnerr
2736   \else
2737     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
2738     \jsInhibitGlue \ignorespaces
2739   \fi}

```

12 いろいろなロゴ

L^AT_EX 関連のロゴを作り直します。

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

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

※`\小`、`\上小` の制御綴は定義しない。

```

2740 \if@jslogo
2741   \IfFileExists{jslogo.sty}{%
2742     \RequirePackage{jslogo}%
2743   }{%
2744     \ClassWarningNoLine\bxjs@clsname
2745     {The package 'jslogo' is not installed.\MessageBreak
2746     It is included in the recent release of\MessageBreak
2747     the 'jsclasses' bundle}
2748   }
2749 \fi

```

13 amsmath との衝突の回避

最近の `\LaTeX` では該当の問題は対処されているので削除。

14 初期設定

■いろいろな語

```

\prepartname
\postpartname 2750 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2751 \newcommand{\postpartname}{\if@english\else 部\fi}
\postchaptername 2752 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\presectionname
\postsectionname

```

```

2753 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
2754 \newcommand{\presectionname}{}% 第
2755 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 2756 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2757 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
2758 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname

\bibName 2759 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2760 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
2761 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 2762 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
2763 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname

\abstractname 2764 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
2765 \newcommand{\appendixname}{\if@english \else 付録\fi}
2766 %<!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_TEX_PRIMITIVES が設定されている場合は“今日”が過去の日付になる可能性があるが、その場合、和暦表記は平成 2 年（1990 年）以降でのみサポートする。

※ “新元号” への対応？

```

\today

2767 \@tempwafalse
2768 \if p\jsEngine \@tempwattrue \fi
2769 \if n\jsEngine \@tempwattrue \fi
2770 \if@tempswa \expandafter\@firstoftwo
2771 \else \expandafter\@secondoftwo
2772 \fi
2773 {%
2774 % 欧文 8bitTeX の場合
2775 \newif\ifjsSeireki \jsSeirekitrue
2776 \def\bxjs@decl@Seireki@cmds{%
2777 \def\西暦{\jsSeirekitrue}%
2778 \def\和暦{\jsSeirekifalse}}
2779 \def\Seireki{\jsSeirekitrue}
2780 \def\Wareki{\jsSeirekifalse}

```

```

2781 \def\bxjs@if@use@seireki{%
2782   \ifjsSeireki \expandafter\@firstoftwo
2783   \else \expandafter\@secondoftwo \fi}
2784 }-%
2785 \newif\if 西曆 \西曆 true
2786 \def\bxjs@decl@Seireki@cmds{%
2787   \def\西曆{\西曆 true}%
2788   \def\和曆{\西曆 false}}
2789 \def\Seireki{\西曆 true}
2790 \def\Wareki{\西曆 false}
2791 \def\bxjs@if@use@seireki{%
2792   \if 西曆 \expandafter\@firstoftwo
2793   \else \expandafter\@secondoftwo \fi}
2794 }
2795 \bxjs@decl@Seireki@cmds
2796 % \bxjs@unxp
2797 \let\bxjs@unxp@firstofone
2798 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2799 % \bxjs@iai
2800 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2801   \def\bxjs@iai{\noexpand~}
2802 \else \def\bxjs@iai{}
2803 \fi
2804 % \heisei
2805 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2806 % \today
2807 \edef\bxjs@today{%
2808   \if@english
2809     \ifcase\month\or
2810       January\or February\or March\or April\or May\or June\or
2811       July\or August\or September\or October\or November\or December\fi
2812     \space\number\day, \number\year
2813   \else
2814     \ifnum\heisei>\@ne
2815       \expandafter\noexpand\expandafter\bxjs@if@use@seireki
2816     \else \expandafter\@firstoftwo
2817     \fi {%
2818       \number\year\bxjs@iai\bxjs@unxp{年}%
2819       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2820       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2821     }-%
2822     \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2823     \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2824     \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2825   }%
2826   \fi}
2827 \let\today\bxjs@today

```

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

```
2828 \AtBeginDocument{%
2829   \ifx\bbl@jpn@Seirekitrue\@undefined\else
2830     \bxjs@decl@Seireki@cmds
2831     \g@addto@macro\datejapanese{%
2832       \let\today\bxjs@today}%
2833   \fi}
```

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

```
2834 \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。

```
2835 %<slide>\pagestyle{empty}%
2836 %<article|report>\pagestyle{plain}%
2837 %<book>\pagestyle{headings}%
2838 \pagenumbering{arabic}
2839 \if@twocolumn
2840   \twocolumn
2841   \sloppy
2842   \flushbottom
2843 \else
2844   \onecolumn
2845   \raggedbottom
2846 \fi
2847 %<*/slide>
2848 \renewcommand\familydefault{\sfdefault}
2849 \raggedright
2850 %</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。

```
2851 \catcode\?=12
2852 \ifx\bxjs@jadriver\relax\else
2853 \input{bxjsja-\bxjs@jadriver.def}
2854 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
2855 \bxjs@restore@jltrcc
2856 %</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` `XqTeX`
 - `j` `pTeX` または `upTeX`
 - `n` 以上の何れでもない
- `\ifjsWithupTeX` [スイッチ] 使用されているエンジンが `upTeX` であるか。
- `\ifjsWitheTeX` [スイッチ] 使用されているエンジンが ϵ -`TeX` 拡張であるか。
- `\ifjsInPdfMode` [スイッチ] 使用されているエンジンが (`pdfTeX`・`LuaTeX` の) PDF モードであるか。
- `\jsUnusualPtSize` [整数定数を表す文字列のマクロ] 基底フォントサイズが `10pt`、`11pt`、`12pt` のいずれでもない場合の `\@ptsize` の値。 (`\@ptsize` 自体があまり有用でないと思われる。)
- `\jsScale` [実数を表す文字列のマクロ] 和文フォントサイズの要求サイズに対するスケール。クラスオプション `scale` で指定される。(既定値は `0.924715`。)
- `\jsJaFont` [マクロ] 和文フォント設定を表す文字列。クラスオプション `jafont` で指定された値。
- `\jsJaParam` [マクロ] 和文モジュールに渡すパラメタを表す文字列。この値が何を表すかは決まっておらず、各々の和文モジュールが独自に解釈する。クラスオプション `japaram` で指定された値。
- `\jsInhibitGlue` [マクロ] `\inhibitglue` という命令が定義されていればそれを実行し、そうでなければ何もしない。JS クラスで `\inhibitglue` を用いている箇所は全て `\jsInhibitGlue` に置き換えられている。従って、`\inhibitglue` は未定義でも動作するが、その実装がある場合は BXJS クラスはそれを活用する。
- `\jsInhibitGlueAtParTop` [マクロ] ★ 段落先頭におけるカギ括弧の位置調整を行うマクロ。全ての段落先頭で呼び出される。
- `\jsZw` [内部寸法値] 「現在の全角幅」を表す変数。JS クラスで `zw` 単位で設定されている長さパラメタはこの変数を単位として設定されている。この変数の値は実際

に用いられる「和文フォント」のメトリックに基づくのではなく、機械的に `\jsScale` × (フォントサイズ) であると定められている (フォントサイズ変更の度に再設定される)。従って、「和文コンポーネント」はこの設定と辻褃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。

```
2857 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

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

B.1 補助マクロ

```
2858 %<*minimal>
```

```
2859 %% 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.
```

```
2860 \def\DeclareJaTextFontCommand#1#2{%
```

```
2861   \DeclareRobustCommand#1[1]{%
```

```
2862     \relax
```

```
2863     \ifmmode \expandafter\nfss@text \fi
```

```
2864     {#2##1}}%
```

```
2865 }
```

```
\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。
```

```
2866 \def\DeclareJaMathFontCommand#1#2{%
```

```
2867   \DeclareRobustCommand#1[1]{%
```

```
2868     \relax
```

```
2869     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
2870     \nfss@text{\fontfamily\familydefault
```

```
2871       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
2872       #2##1}}%
```

```
2873 }%
2874 }
```

`\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。

```
2875 \long\def\bxjs@@CSsfdefault{\sfdefault}%
2876 \@onlypreamble\bxjs@if@sf@default
2877 \def\bxjs@if@sf@default#1{%
2878   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
2879   \AtBeginDocument{%
2880     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
2881 }
```

`\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。

```
2882 \@tempdima\p@ \bxjs@invscale\@tempdima\jsScale
2883 \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。

```
2884 \def\jsLetHeadChar#1#2{%
2885   \begingroup
2886     \escapechar=`\ %
2887     \let\bxjs@tmpa={% brace-match-hack
2888       \bxjs@let@hchar@exp#2}%
2889   \endgroup
2890   \let#1\bxjs@g@tmpa}
2891 \def\bxjs@let@hchar@exp{%
2892   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2893 \def\bxjs@let@hchar@exp@a{%
2894   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi% 波括弧
2895   \bxjs@let@hchar@out\let\relax
2896   }{\bxjs@cond@ifcat\noexpand\@let@token@sptoken\fi% 空白
2897   \bxjs@let@hchar@out\let\space%
2898   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi% バックスラッシュ
2899   \bxjs@let@hchar@out\let\@backslashchar
2900   }{\bxjs@let@hchar@exp@b}}
2901 \def\bxjs@let@hchar@exp@b#1{%
2902   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2903 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2904   %\message{<#1#2>}%
2905   \bxjs@cond@if#1\@backslashchar\fi% 制御綴
```

```

2906 \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2907 \bxjs@let@hchar@out\let\relax
2908 }{%else
2909 \expandafter\bxjs@let@hchar@exp
2910 }%
2911 }{%else
2912 \bxjs@let@hchar@chr#1%
2913 }}
2914 \def\bxjs@let@hchar@chr#1{%
2915 \bxjs@let@hchar@out\def{#{1}}}}
2916 \def\bxjs@let@hchar@out#1#2{%
2917 \global#1\bxjs@g@tmpa#2\relax
2918 \toks@\bgroup}% skip to right brace

```

UTF-8 のバイト列を扱うコード。

```

2919 \chardef\bxjs@let@hchar@csta=128
2920 \chardef\bxjs@let@hchar@cstb=192
2921 \chardef\bxjs@let@hchar@cstc=224
2922 \chardef\bxjs@let@hchar@cstd=240
2923 \chardef\bxjs@let@hchar@cste=248
2924 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2925 \def\bxjs@let@hchar@chr@ue#1{%
2926 \@tempcnta=#1\relax
2927 %\message{\the\@tempcnta}%
2928 \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2929 \bxjs@let@hchar@chr@ue@a#1%
2930 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2931 \bxjs@let@hchar@out\let\relax
2932 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2933 \bxjs@let@hchar@chr@ue@b
2934 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2935 \bxjs@let@hchar@chr@ue@c
2936 }{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2937 \bxjs@let@hchar@chr@ue@d
2938 }{%else
2939 \bxjs@let@hchar@out\let\relax
2940 }}}}}
2941 \def\bxjs@let@hchar@chr@ue@a#1{%
2942 \bxjs@let@hchar@out\def{#{1}}}}
2943 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2944 \bxjs@let@hchar@out\def{#{1#2}}}}
2945 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2946 \bxjs@let@hchar@out\def{#{1#2#3}}}}
2947 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2948 \bxjs@let@hchar@out\def{#{1#2#3#4}}}}

```

B.2 (u)pTeX 用の設定

```
2949 \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。

```
2950 \def\bxjs@let@hchar@chr@pp#1{%
2951   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
2952 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2953   %\message{(#1)}%
2954   \bxjs@cond\if#1t\fi{%
2955     \bxjs@let@hchar@chr@ue#3%
2956   }{%else
2957     \bxjs@let@hchar@out\def{#3}}%
2958   }%
2959 \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。

```
2960 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
2961 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
2962 \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。

```
2963 \@onlypreamble\bxjs@declarefontshape
2964 \ifjsWithupTeX
2965 \def\bxjs@declarefontshape{%
2966   \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
2967   \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
2968   \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
2969   \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
2970 }
2971 \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。

```
2972 \else
2973 \def\bxjs@declarefontshape{%
2974   \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
2975   \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
2976   \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{%
```

```

2977 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{%
2978 }
2979 \def\bxjs@sizereference{jis}
2980 \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。

```

2981 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
2982   \def\bxjs@y{#5}}
2983 \ifjswithptexng \def\bxjs@y{10}%
2984 \else
2985 \expandafter\expandafter\expandafter\bxjs@tmpa
2986 \expandafter\string\the\jfont\relax
2987 \fi
2988 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
2989               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
2990   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
2991    \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。

```

2992 \begingroup
2993 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2994 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2995 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2996 % 幅が丁度 10pt なら補正は不要
2997 \ifdim\wd\z@=10pt
2998   \global\let\bxjs@scale\jsScale
2999 \else
3000 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
3001   \edef\bxjs@tmpa{\strip@pt\wd\z@}
3002   \@tempdima=10pt \@tempdima=\jsScale\@tempdima
3003   \bxjs@invscale\@tempdima\bxjs@tmpa
3004   \xdef\bxjs@scale{\strip@pt\@tempdima}
3005 \fi
3006 \endgroup
3007 %\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。

```

3008 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3009 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3010 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
3011 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}

```

```

3012 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{}
3013 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{}
3014 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{}
3015 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{}
3016 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{}
3017 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{}
3018 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{}
3019 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{}
3020 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{}
3021 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{}
3022 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{}

```

欧文総称フォント命令で和文フォントが連動するように修正する。その他の和文フォント関係の定義を行う。

```

3023 \DeclareRobustCommand\rmfamily
3024   {\not@math@alphabet\rmfamily\mathrm
3025     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3026 \DeclareRobustCommand\sffamily
3027   {\not@math@alphabet\sffamily\mathsf
3028     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3029 \DeclareRobustCommand\ttfamily
3030   {\not@math@alphabet\ttfamily\mathtt
3031     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3032 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3033 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3034 \bxjs@if@sf@default{%
3035   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```
3036 \selectfont
```

■パラメタの設定

```

3037 \prebreakpenalty\jis"2147=10000
3038 \postbreakpenalty\jis"2148=10000
3039 \prebreakpenalty\jis"2149=10000
3040 \inhibitxspcode`! =1
3041 \inhibitxspcode`¯=2
3042 \xspcode`+=3
3043 \xspcode`%=3

```

"80~"FF の範囲の \xspcode を 3 に変更。

```

3044 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3045   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3046 \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。

```
3047 \begingroup
3048 \catcode`\!=0
```

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

```
3049 \gdef\bxjs@ptex@dir{%
3050 !iftdir t%
3051 !else!ifydir y%
3052 !else ?%
3053 !fi!fi}
```

新版の p_TE_X で脚注番号の周囲の空きが過大になる現象への対処。

※現在の p_LA_TE_X 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。

```
3054 % 古い \@makefnmark の定義
3055 \long\def\bxjs@tmpa{\hbox{%
3056 !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
3057 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3058 \ifx\@makefnmark\bxjs@tmpa
3059 \long\gdef\@makefnmark{%
3060 !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
3061 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3062 \fi
3063 \endgroup
```

B.3 pdf_TE_X 用の処理

```
3064 \else\ifx p\jsEngine
3065 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3066 \@onlypreamble\bxjs@cjk@loaded
3067 \def\bxjs@cjk@loaded{%
3068 \def\@footnotemark{%
3069 \leavevmode
3070 \ifhmode
3071 \edef\@x@sf{\the\spacefactor}%
3072 \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3073 \unkern\unkern
3074 \ifdim\lastskip>\z@ \unskip \fi
3075 \fi\fi
3076 \nobreak
3077 \fi
3078 \@makefnmark
3079 \ifhmode \spacefactor\@x@sf \fi
3080 \relax}%
3081 \let\bxjs@cjk@loaded\relax
3082 }
3083 \AtBeginDocument{%
```

```

3084 \ifpackageloaded{CJK}{%
3085   \bxjs@CJK@loaded
3086 }{}%
3087 }

```

B.4 XeTeX 用の処理

```
3088 \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。

```

3089 \def\bxjs@let@hchar@chr#1{%
3090   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3091   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3092     \bxjs@let@hchar@chr@xe
3093   }\bxjs@let@hchar@out\def{#{1}}}}
3094 \def\bxjs@let@hchar@chr@xe#1{%
3095   \lccode`0=`#1\relax
3096   \lowercase{\bxjs@let@hchar@out\def{0}}}}

```

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

```

3097 \ifx\XeTeXgenerateactualtext\@undefined\else
3098   \def\bxjs@do@precisetext{%
3099     \XeTeXgenerateactualtext=\@one}
3100 \fi

```

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

```

3101 \@onlypreamble\bxjs@do@simplejasetup
3102 \def\bxjs@do@simplejasetup{%
3103   \ifnum\XeTeXinterchartokenstate>\z@
3104   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3105     \jsSimpleJaSetup
3106     \ClassInfo\bxjs@clsname
3107     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3108   \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

3109 \newcommand*{\jsSimpleJaSetup}{%
3110   \XeTeXlinebreaklocale "ja"\relax
3111   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3112   \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理 (エンジン共通)

```
3113 \fi\fi\fi
```

simplejasetup オプションの処理。

```

3114 \ifx\bxjs@do@simplejasetup\@undefined\else
3115   \AtBeginDocument{%
3116     \ifbxjs@simplejasetup

```

```

3117     \bxjs@do@simplejasetup
3118     \fi}
3119 \fi

    precisetext オプションの処理。
3120 \ifbxjs@precisetext
3121     \ifx\bxjs@do@precisetext\undefined
3122     \ClassWarning\bxjs@clsname
3123     {The current engine does not supprt the\MessageBreak
3124     'precisetext' option\@gobble}
3125     \else
3126     \bxjs@do@precisetext
3127     \fi
3128 \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。

```

3129 \ifbxjs@fancyhdr

```

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

```

3130 \@onlypreamble\bxjs@adjust@fancyhdr
3131 \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。

```

3132 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3133 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3134 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3135 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3136 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3137 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3138 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3139 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3140 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3141 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3142 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3143 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3144 \def\bxjs@tmpa{\rm\thepage\strut}%
3145 \def\bxjs@tmpb{\thepage\strut}%
3146 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3147 \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。

```

3148 \ifx\fullwidth\undefined\else \ifdim\textwidth<\fullwidth
3149 \setlength{\@tempdima}{\fullwidth-\textwidth}%
3150 \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3151 }\bxjs@tmpa
3152 \fi\fi
3153 \PackageInfo\bxjs@clsname
3154 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3155 \def\bxjs@pagestyle@hook{%
3156 \ifpackageloaded{fancyhdr}{%
3157 \bxjs@adjust@fancyhdr
3158 \global\let\bxjs@adjust@fancyhdr\relax
3159 }{}}

```

\pagestyle にフックを入れ込む。

```

3160 \let\bxjs@org@pagestyle\pagestyle
3161 \def\pagestyle{%
3162 \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。

```

3163 \AtBeginDocument{%
3164 \bxjs@pagestyle@hook
3165 \global\let\bxjs@pagestyle@hook\relax}

```

終わり。

```
3166 \fi
```

以上で終わり。

```
3167 %</minimal>
```

付録 C 和文ドライバ : standard

標準のドライバ。

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

■和文フォント指定の扱い 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。

```
3168 %<*standard>
3169 %% このファイルは日本語文字を含みます
3170 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3171 \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。

`\bxjs@jp@jismmiv` 2004JIS 字形を優先させるか。

```
3172 \newif\ifbxjs@jp@jismmiv

jis2004 オプションの処理。
3173 \bxjs@cslet{bxjs@kv@jis2004@true}\bxjs@jp@jismmivtrue
3174 \bxjs@cslet{bxjs@kv@jis2004@false}\bxjs@jp@jismmivfalse
3175 \define@key{bxjsStd}{jis2004}[true]{%
3176 \bxjs@set@keyval{jis2004}{#1}{}}
```

`\bxjs@jp@font` フォントパッケージの追加オプション。

```
3177 \let\bxjs@jp@font\@empty

jis2004 オプションの処理。
3178 \define@key{bxjsStd}{font}{%
3179 \edef\bxjs@jp@font{#1}}

実際の japaram の値を適用する。
3180 \def\bxjs@next#1{\bxjs@safe@setkeys{bxjsStd}{#1}}
3181 \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。

```
3182 \ifbxjs@jp@jismmiv
3183 \g@addto@macro\@classoptionslist{,jis2004}
3184 % \@ifpackagewith 判定への対策
3185 \PassOptionsToPackage{jis2004}{otf}
3186 \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。

```
3187 \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。

```
\jH3188 \@tempdima=0.25mm
3189 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3190 \let\jH\jQ
```

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

```
\trueH3191 \ifjsc@mag
3192 \@tempdimb=\jsBaseFontSize\relax
3193 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3194 \@tempdima=2.5mm
3195 \bxjs@invscale\@tempdima\bxjs@tmpa
3196 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3197 \@tempdima=10pt
3198 \bxjs@invscale\@tempdima\bxjs@tmpa
3199 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3200 \else \let\trueQ\jQ \let\bxjs@truept\p@
3201 \fi
3202 \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。

```
3203 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3204 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3205 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3206 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3207 \fi
```

`\jsSetQHLength` `\jsSetQHLength\CS{〈長さ式〉}` : `\setlength` の変種で、通常の `calc` の長さ式の代わ
 りに、 $\langle \text{実数} \rangle Q \cdot \langle \text{実数} \rangle H \cdot \langle \text{実数} \rangle \text{trueQ} \cdot \langle \text{実数} \rangle \text{trueH}$ が記述できる。

```
3208 \def\jsSetQHLength#1#2{%
3209 \begingroup
3210 \bxjs@parse@qh{#2}%
3211 \ifx\bxjs@tmpb\relax \setlength\@tempdima{#2}%
3212 \else \@tempdima\bxjs@tmpb\relax
3213 \fi
3214 \xdef\bxjs@g@tmpa{\the\@tempdima}%
3215 \endgroup
3216 #1=\bxjs@g@tmpa\relax}
```

`\bxjs@parse@qh` `jsSetQHLength` 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。

```
3217 \if j\jsEngine
```

(u)pL^AT_EX ではこの処理は不要なので無効化する。

```
3218 \def\bxjs@parse@qh#1{%
3219   \let\bxjs@tmpb\relax}
3220 \else
3221 \def\bxjs@parse@qh#1{%
3222   \def\bxjs@tmpa{#1}\let\bxjs@tmpb\relax
3223   \bxjs@parse@qh@a{trueQ}\trueQ
3224   \bxjs@parse@qh@a{trueH}\trueH
3225   \bxjs@parse@qh@a{Q}\jQ
3226   \bxjs@parse@qh@a{H}\jH}
3227 \def\bxjs@parse@qh@a#1#2{%
3228   \ifx\bxjs@tmpb\relax
3229     \def\bxjs@next##1#1\@nil##2\@nnil{\bxjs@parse@qh@b{##1}{##2}}%
3230     \expandafter\bxjs@next\bxjs@tmpa\@nil#1\@nil\@nnil{#2}%
3231   \fi}
3232 \def\bxjs@parse@qh@b#1#2#3{%
3233   \ifx\@nnil#2\@nnil\else \def\bxjs@tmpb{#1#3}\fi}
3234 \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。

```
3235 \def\jafontsize#1#2{%
3236   \begingroup
3237     \bxjs@jafontsize@a{#1}%
3238     \@tempdimb\jsInverseScale\@tempdima
3239     \bxjs@jafontsize@a{#2}%
3240     \xdef\bxjs@g@tmpa{%
3241       \noexpand\fontsize{\the\@tempdimb}{\the\@tempdima}}%
3242   \endgroup\bxjs@g@tmpa}
3243 \def\bxjs@jafontsize@a#1{%
3244   \bxjs@parse@qh{#1}%
3245   \ifx\bxjs@tmpb\relax \def\bxjs@tmpb{#1}\fi
3246   \@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。

```
3247 \def\bxjs@kanjiskip{Opt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3248 \newcommand*\setkanjiskip[1]{%
3249   \edef\bxjs@kanjiskip{#1}%
3250   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3251 \newcommand*\getkanjiskip{%
3252   \bxjs@kanjiskip}
```

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

```
3253 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

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

```
\bxjs@disable@kanjiskip 3254 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3255   \bxjs@kanjiskip@enabledtrue
3256   \bxjs@reset@kanjiskip}
3257 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3258   \bxjs@kanjiskip@enabledfalse
3259   \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。

```
3260 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3261   \ifbxjs@kanjiskip@enabled
3262     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3263   \else \@tempskipa\z@
3264   \fi
3265   \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 3266 \def\bxjs@xkanjiskip{0pt}
\getxkanjiskip 3267 \newcommand*\setxkanjiskip[1]{%
```

```
\ifbxjs@xkanjiskip@enabled 3268   \edef\bxjs@xkanjiskip{#1}%
3269   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3270 \newcommand*\getxkanjiskip{%
```

```
3271   \bxjs@xkanjiskip}
```

```
\bxjs@disable@xkanjiskip 3272 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3273 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3274   \bxjs@xkanjiskip@enabledtrue
```

```
3275   \bxjs@reset@xkanjiskip}
```

```
3276 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3277   \bxjs@xkanjiskip@enabledfalse
```

```
3278   \bxjs@reset@xkanjiskip}
```

```
3279 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```
3280   \ifbxjs@xkanjiskip@enabled
```

```
3281     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
```

```
3282   \else \@tempskipa\z@
```

```
3283   \fi
```

```
3284   \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。

```
3285 \g@addto@macro\jsResetDimen{%
```

```

3286 \bxjs@reset@kanjiskip
3287 \bxjs@reset@xkanjiskip}
3288 \let\bxjs@apply@kanjiskip\relax
3289 \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。

```

3290 \@onlypreamble\bxjs@adjust@jafont
3291 \def\bxjs@adjust@jafont#1{%
3292   \ifx\jsJaFont\bxjs@@auto
3293     \bxjs@get@kanjiEmbed
3294     \ifx\bxjs@kanjiEmbed\relax
3295       \let\bxjs@tmpa\@empty
3296     \else
3297       \let\bxjs@tmpa\bxjs@kanjiEmbed
3298     \fi
3299   \else
3300     \let\bxjs@tmpa\jsJaFont
3301   \fi
3302   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3303     \ClassWarningNoLine\bxjs@clsname
3304     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3305     not available on the current situation}%
3306   \let\bxjs@tmpa\@empty
3307   \fi\fi
3308 }
3309 \def\bxjs@@auto{auto}
3310 \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。

```

3311 \let\bxjs@kanjiEmbed\relax

```

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

```

3312 \@onlypreamble\bxjs@get@kanjiEmbed
3313 \def\bxjs@get@kanjiEmbed{%
3314   \begingroup\setbox\z@=\hbox{%
3315     \global\let\bxjs@g@tmpa\relax
3316     \endlinechar\m@ne
3317     \let\do\@makeother\dospecials
3318     \catcode32=10 \catcode12=10 %form-feed
3319     \let\bxjs@tmpa\@empty
3320     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3321     \ifeof\@inputcheck\else
3322       \read\@inputcheck to\bxjs@tmpa

```

```

3323     \closein\@inputcheck
3324     \fi
3325     \ifx\bxjs@tmpa\@empty\else
3326         \openin\@inputcheck="\bxjs@tmpa"\relax
3327         \@tempwattrue
3328         \loop\if@tempswa
3329             \read\@inputcheck to\bxjs@tmpa
3330             \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3331             \ifx\bxjs@tmpb\relax\else
3332                 \global\let\bxjs@g@tmpa\bxjs@tmpb
3333                 \@tempwafalse
3334             \fi
3335             \expandafter\bxjs@get@ke@b\bxjs@tmpa\@nil jaEmbed \@nil\@nnil
3336             \ifx\bxjs@tmpb\relax\else
3337                 \global\let\bxjs@g@tmpa\bxjs@tmpb
3338                 \@tempwafalse
3339             \fi
3340             \ifeof\@inputcheck \@tempwafalse \fi
3341         \repeat
3342     \fi
3343 } \endgroup
3344 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3345 }
3346 \@onlypreamble\bxjs@get@ke@a
3347 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3348     \ifx$#1$\def\bxjs@tmpb{#2}%
3349     \else \let\bxjs@tmpb\relax
3350     \fi}
3351 \@onlypreamble\bxjs@get@ke@b
3352 \def\bxjs@get@ke@b#1jaEmbed #2\@nil#3\@nnil{%
3353     \ifx$#1$\def\bxjs@tmpb{#2}%
3354     \else \let\bxjs@tmpb\relax
3355     \fi}

```

\jachar \jachar{<文字>} : 和文文字として出力する。

```

3356 \newcommand*\jachar[1]{%
3357     \begingroup
3358         \jsLetHeadChar\bxjs@tmpa{#1}%
3359         \ifx\bxjs@tmpa\relax
3360             \ClassWarningNoLine\bxjs@clsname
3361             {Illegal argument given to \string\jachar}%
3362         \else
3363             \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3364         \fi
3365     \endgroup}

```

\jsJaChar を \jachar と等価にする。

```

3366 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
3367 \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。

```
3368 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3369 \@onlypreamble\bxjs@fix@hyperref@unicode
3370 \def\bxjs@fix@hyperref@unicode#1{%
3371   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3372   \@namedef{KV@Hyp@bxjs/hook}##1{%
3373     \KV@Hyp@unicode{##1}%
3374     \def\KV@Hyp@unicode####1{%
3375       \expandafter\ifx\csname if##1\expandafter\endcsname
3376       \csname if####1\endcsname\else
3377       \ClassWarningNoLine\bxjs@clsname
3378       {Blcked hyperref option 'unicode=####1'}%
3379       \fi
3380     }%
3381   }%
3382 }
```

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

```
3383 \@onlypreamble\bxjs@urgent@special
3384 \def\bxjs@urgent@special#1{%
3385   \AtBeginDvi{\special{#1}}%
3386   \AtBeginDocument{%
3387     \@ifpackageloaded{atbegshi}{%
3388       \begingroup
3389       \toks\z@{\special{#1}}%
3390       \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3391       \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3392     \endgroup
3393   }{%
3394   }%
3395 }
```

C.2 pTeX 用設定

```
3396 \if j\jsEngine
```

■共通命令の実装

```
3397 \def\bxjs@apply@kanjiskip{%
3398   \kanjiskip\@tempskipa}
3399 \def\bxjs@apply@xkanjiskip{%
3400   \xkanjiskip\@tempskipa}
```

`\jaJaChar` のサブマクロ。

```
3401 \def\bxjs@jachar#1{%
3402   \bxjs@jachar@a#1...\@nil}
3403 \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。

```
3404   \ifx.#2#1%
```

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

```
3405   \else\ifx.#3%
3406     \@tempcnta`#1 \multiply\@tempcnta64
3407     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3408     \bxjs@jachar@b
3409   \else\ifx.#4%
3410     \@tempcnta`#1 \multiply\@tempcnta64
3411     \advance\@tempcnta`#2 \multiply\@tempcnta64
3412     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3413     \bxjs@jachar@b
3414   \else
3415     \@tempcnta`#1 \multiply\@tempcnta64
3416     \advance\@tempcnta`#2 \multiply\@tempcnta64
3417     \advance\@tempcnta`#3 \multiply\@tempcnta64
3418     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3419     \bxjs@jachar@b
3420   \fi\fi\fi}
```

符号値が `\@tempcnta` の和文文字を出力する処理。

```
3421 \ifjswithupteX
3422   \def\bxjs@jachar@b{\kchar\@tempcnta}
3423 \else
3424   \def\bxjs@jachar@b{%
3425     \ifx\bxUInt\@undefined\else
3426       \bxUInt{\@tempcnta}%
3427     \fi}
3428 \fi
```

和欧文間空白の命令 `\>` の実装。

```
3429 \ifbxjs@xkanjiskip@cmd
3430   \def\bxjs@put@xkanjiskip{%
3431     \relax\ifmmode \mskip\medmuskip
3432     \else \hskip\xkanjiskip
3433     \fi}
3434   \ifjswithteX \protected\def\>{\bxjs@put@xkanjiskip}
3435   \else \def\>{\protect\bxjs@put@xkanjiskip}
3436   \fi
3437 \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。

```
3438 \ifbxjs@jp@jismmiv
3439   \PassOptionsToPackage{prefer2004jis}{pxchfon}
3440   \PassOptionsToPackage{prefer2004jis}{pxbabel}
3441 \fi
```

■和文フォント指定の扱い pTeX は既定で kanji-config-updmap の設定に従うため、\jsJaFont が auto の場合は何もする必要がない。無指定でも auto でもない場合は、\jsJaFont をオプションにして pxchfon パッケージを読み込む。ここで、和文ドライバパラメタ font が指定されている場合は、その値を pxchfon のオプションに追加する。

```
3442 \let\bxjs@tmpa\jsJaFont
3443 \ifx\bxjs@tmpa\bxjs@@auto
3444   \let\bxjs@tmpa@empty
3445 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3446   \def\bxjs@tmpa{noembed}
3447 \fi\fi
3448 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3449 \ifx\bxjs@tmpa@empty\else
3450   \edef\bxjs@next{%
3451     \noexpand\RequirePackage[\bxjs@tmpa]{pxchfon}[2010/05/12]% v0.5
3452   }\bxjs@next
3453 \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……。)

```
3454 \begingroup
3455   \global\let\@gtempa\relax
3456   \catcode`\|=0 \catcode`\|=12
3457   |def|bxjs@check#a#1|@nil{%
3458     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3459   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3460     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3461   |catcode`\|=0 \catcode`\|=12
3462   \def\bxjs@check@b#1keyval#2@nnil{%
3463     \ifx$#2$\else
3464       \xdef\@gtempa{%
3465         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3466     \fi}
3467 \@firstofone{%
3468   \catcode10=12 \endlinechar\m@ne
3469   \let\do\@makeother \dospecials \catcode32=10
3470   \openin\@inputcheck=otf.sty\relax
3471   \@tempwattrue
```

```

3472 \loop\if@tempswa
3473 \ifeof\@inputcheck \@tempswafalse \fi
3474 \if@tempswa
3475 \read\@inputcheck to\bxjs@line
3476 \expandafter\bxjs@check\bxjs@line\@nil
3477 \fi
3478 \repeat
3479 \closein\@inputcheck
3480 \endgroup}
3481 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3482 \ifbxjs@hyperref@enc
3483 \bxjs@fix@hyperref@unicode{false}
3484 \fi

```

tounicode special 命令を出力する。

```

3485 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3486 \else\ifjsWithpTeXng T\else F\fi\fi T%
3487 \ifnum\jis"2121="A1A1 %euc
3488 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3489 \else\ifnum\jis"2121="8140 %sjis
3490 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3491 \else\ifnum\jis"2121="3000 %uptex
3492 \ifbxjs@bigcode
3493 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3494 \PassOptionsToPackage{bigcode}{pxjahyper}
3495 \else
3496 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3497 \fi
3498 \fi\fi\fi
3499 \let\bxToUnicodeSpecialDone=t
3500 \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。

```

3501 \ifx f\bxjs@enablejfam\else
3502 \@enablejfamtrue
3503 \fi

```

実際に和文用の数式ファミリの設定を行う。

```

3504 \if@enablejfam
3505 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3506 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3507 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3508 \jfam\symmincho
3509 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3510 \AtBeginDocument{%
3511 \ifx\reDeclareMathAlphabet\@undefined\else

```

```

3512     \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3513     \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3514     \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3515     \fi}
3516 \fi

```

C.3 pdfTeX 用設定：CJK + bxcjkatype

```
3517 \else\if p\jsEngine
```

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

```

3518 \bxjs@adjust@jafont{f}
3519 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3520 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa{whole,autotilde}}
3521 \edef\bxjs@next{%
3522   \noexpand\RequirePackage[\bxjs@tmpa]{bxcjkatype}[2013/10/15]% v0.2c
3523 }\bxjs@next
3524 \bxjs@cjk@loaded

```

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

※取りあえず固定はしない。

```

3525 \ifbxjs@hyperref@enc
3526   \PassOptionsToPackage{unicode}{hyperref}
3527 \fi

```

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

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

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

```

3528 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3529 \begingroup
3530   \CJK@input{UTF8.bdg}
3531 \endgroup
3532 \g@addto@macro\pdfstringdefPreHook{%
3533   \@nameuse{CJK@UTF8Binding}%
3534 }
3535 \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。

```

3536 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3537 \g@addto@macro\pdfstringdefPreHook{%
3538   \ifx~\bxjs@@CJKtilde
3539     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace

```

```

3540 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3541 \let~\@empty
3542 \fi
3543 }
3544 \def\bxjs@CJKtilde{\CJKecglue\ignorespaces}
3545 \def\bxjs@tildecmd{~}
3546 \def\bxjs@LetUnexpandableSpace#1{%
3547 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@tildecmd\else
3548 \bxjs@org@LetUnexpandableSpace#1%
3549 \fi}
3550 \fi

```

■ 共通命令の実装

```

3551 \newskip\jsKanjiSkip
3552 \newskip\jsXKanjiSkip
3553 \ifx\CJKecglue\@undefined
3554 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3555 \fi
3556 \let\autospacing\bxjs@enable@kanjiskip
3557 \let\noautospacing\bxjs@disable@kanjiskip
3558 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3559 \def\bxjs@apply@kanjiskip{%
3560 \jsKanjiSkip\@tempskipa
3561 \let\CJKglue\bxjs@CJKglue}
3562 \let\autoxspacing\bxjs@enable@xkanjiskip
3563 \let\noautoxspacing\bxjs@disable@xkanjiskip
3564 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3565 \def\bxjs@apply@xkanjiskip{%
3566 \jsXKanjiSkip\@tempskipa
3567 \let\CJKecglue\bxjs@CJKecglue}

```

\jchar のサブマクロの実装。

```

3568 \def\bxjs@jchar#1{%
3569 \CJKforced{#1}}

```

\> は bxcjkatype で定義されているが、保護付のマクロに変換する。

```

3570 \ifbxjs@xkanjiskip@cmd
3571 \protected\def\>{%
3572 \relax\ifmmode \mskip\medmuskip \else \CJKecglue\ignorespaces \fi}
3573 \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。

```

3574 \ifx t\bxjs@enablejfam
3575 \ClassWarningNoLine\bxjs@clsname
3576 {You cannot use 'enablejfam=true', since the\MessageBreak
3577 CJK package does not support Japanese math}
3578 \fi

```

C.4 X_YTeX 用設定 : xeCJK + zxjatype

```
3579 \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。

```
3580 \RequirePackage{zxjatype}
3581 \PassOptionsToPackage{no-math}{fontspec}%!
3582 \PassOptionsToPackage{xetex}{graphicx}%!
3583 \PassOptionsToPackage{xetex}{graphics}%!
3584 \ifx\zxJaFamilyName\@undefined
3585   \ClassError\bxjs@clsname
3586   {xeCJK or zxjatype is too old}\@ehc
3587 \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。

```
3588 \bxjs@adjust@jafont{f}
3589 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3590 \ifx\bxjs@tmpa\@empty
3591   \setCJKmainfont [BoldFont=IPAexGothic]{IPAexMincho}
3592   \setCJKsansfont [BoldFont=IPAexGothic]{IPAexGothic}
3593 \else
3594   \edef\bxjs@next{%
3595     \noexpand\RequirePackage [\bxjs@tmpa]{zxjafont}[2013/01/28]% v0.2a
3596   }\bxjs@next
3597 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X_YTeX 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_YTeX 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_YTeX 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。

```
3598 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3599   \ifbxjs@hyperref@enc
3600     \PassOptionsToPackage{unicode}{hyperref}
3601   \fi
3602 \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。

```
3603 \let\jsInhibitGlueAtParTop\@empty
```

■ 共通命令の実装

```
3604 \newskip\jsKanjiSkip
3605 \newskip\jsXKanjiSkip
3606 \ifx\CJKecglue\@undefined
3607   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3608 \fi
3609 \let\autospacing\bxjs@enable@kanjiskip
3610 \let\noautospacing\bxjs@disable@kanjiskip
3611 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3612 \def\bxjs@apply@kanjiskip{%
3613   \jsKanjiSkip\@tempskipa
3614   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3615 \let\autoxspacing\bxjs@enable@xkanjiskip
3616 \let\noautoxspacing\bxjs@disable@xkanjiskip
3617 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3618 \def\bxjs@apply@xkanjiskip{%
3619   \jsXKanjiSkip\@tempskipa
3620   \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。

```
3621 \ifx\mcfamily\@undefined
3622   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3623   \protected\def\gtfamily{\CJKfamily{\JKsfdefault}}
3624 \fi
```

`\jachar` のサブマクロの実装。

```
3625 \def\bxjs@jachar#1{%
3626   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3627   #1}
```

`\>` は `zxjatype` で定義されているが、`standard` の仕様に合わせて変更する。

※元は `\ignorespaces` でなく `\relax` (`\scan_stop:`) だった。

```
3628 \ifbxjs@xkanjiskip@cmd
3629   \protected\def\>{%
3630     \relax\ifmmode \mskip\medmuskip \else \CJKecglue\ignorespaces \fi}
3631 \fi
```

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

```
3632 \ifx t\bxjs@enablejfam
3633   \@enablejfamtrue
3634 \fi
```

実際に和文用の数式ファミリの設定を行う。

※ FIXME: 要検討。

```
3635 \if@enablejfam
3636 \xeCJKsetup{CJKmath=true}
3637 \fi
```

C.5 LuaTeX 用設定 : LuaTeX-ja

```
3638 \else\if 1\jsEngine
```

■ LuaTeX-ja パッケージの読み込み `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。

```
3639 \let\zw\undefined
3640 \RequirePackage{luatexja}
3641 \RequirePackage{luatexja-fontspec}
```

■ 和文フォント定義 `luatexja-fontspec` で使用する和文スケール値を `\jsScale` と合致させたいのだが……もっと良い方法はないのか？

```
3642 \ExplSyntaxOn
3643 \fp_gset:Nn \g_ljtj_fontspec_scale_fp { \jsScale }
3644 \ExplSyntaxOff
```

`\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)。

```
3645 \bxjs@adjust@jafont{t}
3646 \ifx\bxjs@tmpa\bxjs@noEmbed
3647 \def\bxjs@tmpa{noembed}
3648 \fi
3649 \edef\bxjs@tmpa{\bxjs@catopt\bxjs@tmpa\bxjs@jp@font}
3650 \ifx\bxjs@tmpa\empty
3651 \defaultjfontfeatures{ Kerning=Off }
3652 \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3653 \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3654 \else
3655 \edef\bxjs@next{%
3656 \noexpand\RequirePackage[\bxjs@tmpa]{luatexja-preset}%
3657 }\bxjs@next
3658 \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。

```

3659 \DeclareRobustCommand\rmfamily
3660   {\not@math@alphabet\rmfamily\mathrm}
3661   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3662 \DeclareRobustCommand\sffamily
3663   {\not@math@alphabet\sffamily\mathsf}
3664   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3665 \DeclareRobustCommand\ttfamily
3666   {\not@math@alphabet\ttfamily\mathtt}
3667   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3668 \AtBeginDocument{%
3669   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3670   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
3671   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
3672 \bxjs@if@sf@default{%
3673   \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

3674 % 次の3つは既定値の通り
3675 \ltjsetParameter{prebreakpenalty={` ,10000}}
3676 \ltjsetParameter{postbreakpenalty={` ",10000}}
3677 \ltjsetParameter{prebreakpenalty={` ” ,10000}}
3678 \ltjsetParameter{jaxspmode={` ! ,1}}
3679 \ltjsetParameter{jaxspmode={` 〒 ,2}}
3680 \ltjsetParameter{alxspmode={` + ,3}}
3681 \ltjsetParameter{alxspmode={` \% ,3}}

```

■段落頭でのグルー挿入禁止

```

3682 \protected\def\@inhibitglue{%
3683   \directlua{%
3684     luatexja.jfmglue.create_beginpar_node()}}
3685 \let\bxjs@ltj@inhibitglue\@inhibitglue
3686 \let\@inhibitglue\undefined

```

■hyperref 対策 unicode にするべき。

```

3687 \ifbxjs@hyperref@enc
3688   \bxjs@fix@hyperref@unicode{true}
3689 \fi

```

■共通命令の実装

```

3690 \protected\def\autospacing{%
3691   \ltjsetParameter{autospacing=true}}
3692 \protected\def\noautospacing{%
3693   \ltjsetParameter{autospacing=false}}
3694 \protected\def\autoxspacing{%
3695   \ltjsetParameter{autoxspacing=true}}
3696 \protected\def\noautoxspacing{%
3697   \ltjsetParameter{autoxspacing=false}}
3698 \def\bxjs@apply@kanjiskip{%

```

```

3699 \ltjsetparameter{kanjiskip={\@tempskipa}}
3700 \def\bxjs@apply@xkanjiskip{%
3701 \ltjsetparameter{xkanjiskip={\@tempskipa}}
      \jachar のサブマクロの実装。
3702 \def\bxjs@jachar#1{%
3703 \ltjjachar`#1\relax}
      \> の実装。
3704 \ifbxjs@xkanjiskip@cmd
3705 \protected\def\>{%
3706 \relax\ifmode \mskip\medmuskip
3707 \else \hskip\ltjgetparameter{xkanjiskip}\ignorespaces
3708 \fi}
3709 \fi

```

■和文数式ファミリー LuaTeX-ja では和文数式ファミリーは常に有効で、既にこの時点で必要な設定は済んでいる。従って @enablejfam は常に真になる。

```

3710 \ifx f\bxjs@enablejfam
3711 \ClassWarningNoLine\bxjs@clsname
3712 {You cannot use 'enablejfam=false', since the\MessageBreak
3713 LuaTeX-ja always provides Japanese math families}
3714 \fi

```

C.6 共通処理 (2)

```
3715 \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 3716 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3717 \DeclareJaTextFontCommand{\textgt}{\gtfamily}

```

\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。

```

3718 \ifx\mathmc\@undefined
3719 \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
3720 \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
3721 \fi

```

■和文・和欧文間空白の初期値

```

3722 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3723 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3724 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3725 \fi

```

以上で終わり。

```
3726 %</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。

```
3727 %<*modern>
3728 \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。

```
3729 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3730 \def\encodingdefault{T1}%
3731 \input{t1enc.def}%
3732 \fontencoding\encodingdefault\selectfont
3733 \fi
```

基本フォントを Latin Modern フォントファミリーに変更する。

※以下は `\usepackage[noamth]{lmodern}` と同じ。ユーザは後で `lmodern` を好きなオプションを付けて読み込むことができる。

```
3734 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3735 \renewcommand{\rmdefault}{lmr}
3736 \renewcommand{\sfdefault}{lmss}
3737 \renewcommand{\ttdefault}{lmtt}
3738 \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。

```
3739 \DeclareFontShape{OMX}{cmex}{m}{n}{%
3740   <-7.5>cmex7<7.5-8.5>cmex8%
3741   <8.5-9.5>cmex9<9.5->cmex10}{}%
3742 \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。

```
3743 \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。

```
3744 \ifx\@IncludeInRelease\@undefined
3745 \RequirePackage{fixltx2e}
3746 \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。

```
3747 \RequirePackage{bxjscjkcat}
```

D.4 完了

おしまい。

```
3748 %</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。

```
3749 %<*pandoc>
```

```
3750 \input{bxjsja-standard.def}
```

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。
```

```
3751 \@onlypreamble\bxjs@set@dupload@proc
```

```
3752 \def\bxjs@set@dupload@proc#1{%
```

```
3753 \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
3754 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
3755 \def\bxjs@set@dupload@proc@a#1{%
```

```
3756 \@onlypreamble#1\def#1##1}
```

```
\@ifoptions \@ifoptions の再定義。
```

```
3757 \@onlypreamble\bxjs@org@ifoptions
```

```
3758 \let\bxjs@org@ifoptions\@ifoptions
```

```
3759 \newif\ifbxjs@dlp
```

```
3760 \def\@ifoptions#1#2#3{%
```

```
3761 \bxjs@dlpfalse
```

```
3762 \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
```

```
3763 \ifx\bxjs@tmpa\bxjs@tmpb
```

```
3764 \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
```

```
3765 \bxjs@dlptrue \fi
```

```
3766 \fi
```

```

3767 \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
3768 \else \expandafter\bxjs@org@if@ptions
3769 \fi {#1}{#2}{#3}}
3770 \AtBeginDocument{%
3771 \let\@if@ptions\bxjs@org@if@ptions}
3772 \@onlypreamble\bxjs@do@dupload@proc
3773 \def\bxjs@do@dupload@proc#1#2#3{%
3774 \csname bxjs@dlp/#2.#1\endcsname{#3}%
3775 \@firstoftwo}

```

\bxjs@mark@as@loaded \bxjs@mark@as@loaded{ファイル名} : 特定のファイルに対して、
(\@filewithoptions の処理に関して) 読込済であるとマークする。

```

3776 \def\bxjs@mark@as@loaded#1{%
3777 \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3778 \def\bxjs@mal@a#1#2{%
3779 \ifx#1\relax
3780 \def#1{2001/01/01}%
3781 \ClassInfo\bxjs@clsname
3782 {File '#2' marked as loaded\@gobble}%
3783 \fi}

```

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。

```

3784 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0

```

Polyglossia について。

```

3785 \bxjs@mark@as@loaded{polyglossia.sty}
3786 \bxjs@set@dupload@proc{polyglossia.sty}{%
3787 \ClassWarning\bxjs@clsname
3788 {Loading of polyglossia is blocked}}
3789 \ifx\setmainlanguage\@undefined
3790 \newcommand*\setmainlanguage[2] [] {}
3791 \newcommand*\setotherlanguage[2] [] {}
3792 \ifcat_#2_\else
3793 \expandafter\let\csname #2\endcsname\@empty
3794 \expandafter\let\csname end#2\endcsname\@empty
3795 \expandafter\let\csname text#2\endcsname\@firstofone
3796 \fi}
3797 \newcommand*\setotherlanguages[2] [] {}
3798 \@for\bxjs@tmpa:=#2\do{%
3799 \setotherlangauge{\bxjs@tmpa}}
3800 \fi
3801 \else

```

Babel について。

```

3802 \bxjs@mark@as@loaded{babel.sty}
3803 \bxjs@set@dupload@proc{babel.sty}{%
3804   \ClassWarning\bxjs@clsname
3805     {Loading of babel is blocked}}
3806 \let\foreignlanguage\@secondoftwo
3807 \let\otherlanguage\@gobble
3808 \let\endotherlanguage\@empty
3809 \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。

```

3810 \bxjs@set@dupload@proc{geometry.sty}{%
3811   \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。

```

3812 \if 1\jsEngine
3813   \bxjs@mark@as@loaded{xeCJK.sty}
3814   \providecommand*\setCJKmainfont{\setmainjfont}
3815 \fi

```

E.5 fixltx2e パッケージ

テンプレートでは `fixltx2e` パッケージを読み込むが、最近 (2015 年版以降) の L^AT_EX ではこれで警告が出る。これを抑止する。

L^AT_EX カーネルが新しい場合は `fixltx2e` を読込済にする。

```

3816 \ifx\@IncludeInRelease\@undefined\else
3817   \bxjs@mark@as@loaded{fixltx2e.sty}
3818 \fi

```

E.6 cmap パッケージ

エンジンが (u)pL^AT_EX のときに `cmap` パッケージが読み込まれるのを阻止する。(実際は警告が出るだけで無害であるが。)

```

3819 \if j\jsEngine
3820   \bxjs@mark@as@loaded{cmap.sty}
3821 \fi

```

E.7 microtype パッケージ

警告が多すぎなので消す。

```
3822 \if j\jsEngine \else
3823   \PassOptionsToPackage{verbose=silent}{microtype}
3824 \fi
```

エンジンが (u)pL^AT_EX のときに microtype パッケージが読み込まれるのを阻止し、さらにテンプレートで使われている命令を通すためにダミーの定義を行う。

※昔は standard ドライバでこの処理を行っていたが、元来は Pandoc 用の処理なので、1.5 版で pandoc に移動。

```
3825 \if j\jsEngine
3826   \bxjs@mark@as@loaded{microtype.sty}
3827   \newcommand*\UseMicrotypeSet [2] [] {}
3828 \fi
```

E.8 完了

おしまい。

```
3829 %</pandoc>
```

和文ドライバ実装はここまで。

```
3830 %</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。

- bxjscjkat : modern ドライバ用の和文カテゴリを適用する。

```
3831 %<*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

```
3832 %<*compat>
3833 \def\bxac@pkgname{bxjscompat}
```

`\bxjx@engine` エンジンの種別。

```
3834 \let\bxac@engine=n
3835 \def\bxac@do#1#2{%
```

```

3836 \edef\bxac@tmpa{\string#1}%
3837 \edef\bxac@tmpb{\meaning#1}%
3838 \ifx\bxac@tmpa\bxac@tmpb #2\fi}
3839 \bxac@do\XeTeXversion{\let\bxac@engine=x}
3840 \bxac@do\luatexversion{\let\bxac@engine=l}

```

`\bxac@delayed@if@bxjs` もし BXJS クラスの読込中でこのパッケージが読み込まれているならば、BXJS のクラスの
終わりまで実行を遅延する。

```

3841 \ifx\jsAtEndOfClass\@undefined
3842 \let\bxac@delayed@if@bxjs\@firstofone
3843 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
3844 \fi

```

`\ImposeOldLuaTeXBehavior` ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 3845 \newif\ifbxac@in@old@behavior
3846 \let\ImposeOldLuaTeXBehavior\relax
3847 \let\RevokeOldLuaTeXBehavior\relax

```

G.2 XeTeX 部分

```

3848 \ifx x\bxac@engine
    XeTeX 文字クラスのムニヤムニヤ。
3849 \@onlypreamble\bxac@adjust@charclass
3850 \bxac@delayed@if@bxjs{%
3851 \ifpackageloaded{xCJK}{-}{%else
3852 \ifx\Xe@alloc@intercharclass\@undefined\else
3853 \ifnum\Xe@alloc@intercharclass=\z@
3854 \PackageInfo\bxac@pkgname
3855 {Setting up interchar class for CJK...\@gobble}%
3856 \InputIfFileExists{load-unicode-xetex-classes.tex}{%
3857 \Xe@alloc@intercharclass=3
3858 }{%else
3859 \PackageWarning\bxac@pkgname
3860 {Cannot find file 'load-unicode-xetex-classes.tex'%
3861 \@gobble}%
3862 }%
3863 \fi\fi
3864 \ifnum\XeTeXcharclass"3041=\z@
3865 \PackageInfo\bxac@pkgname
3866 {Adjusting interchar class for CJK...\@gobble}%
3867 \@for\bxac@x:={%
3868 3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3869 3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3870 30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3871 31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3872 31FF%
3873 }do{\XeTeXcharclass"\bxac@x=\@ne}%
3874 \fi

```

```

3875 }%
3876 }

```

以上。

```

3877 \fi

```

G.3 LuaTeX 部分

```

3878 \ifx l\bxac@engine

```

ムニヤムニヤ。

```

3879 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
3880 \chardef\pdfTeXversion=200
3881 \def\pdfTeXrevision{0}
3882 \let\pdfTeXbanner\luatexbanner
3883 \fi\fi

```

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

```

\RevokeOldLuaTeXBehavior
3884 \begingroup\expandafter\expandafter\expandafter\endgroup
3885 \expandafter\ifx\csname outputmode\endcsname\relax\else
3886 \def\bxac@ob@list{%
3887 \do{\let}\pdfoutput{\outputmode}%
3888 \do{\let}\pdfpagewidth{\pagewidth}%
3889 \do{\let}\pdfpageheight{\pageheight}%
3890 \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
3891 \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}%
3892 \def\bxac@ob@do#1#2{\begingroup
3893 \expandafter\bxac@ob@do@a#1#2#3#4\endcsname{#1}#2}
3894 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3895 \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3896 \else \let#3#1\relax \let#1\@undefined
3897 \fi}
3898 \protected\def\ImposeOldLuaTeXBehavior{%
3899 \unless\ifbxac@in@old@behavior
3900 \bxac@in@old@behaviortrue
3901 \let\do\bxac@ob@do \bxac@ob@list
3902 \fi}
3903 \protected\def\RevokeOldLuaTeXBehavior{%
3904 \ifbxac@in@old@behavior
3905 \bxac@in@old@behaviorfalse
3906 \let\do\bxac@ob@do \bxac@ob@list
3907 \fi}
3908 \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。

```

3909 \ifnum\luatexversion>64 \directlua{
3910 local function range(cs, ce, cc, ff)
3911 if ff or not tex.getcatcode(cs) == cc then
3912 local setcc = tex.setcatcode
3913 for c = cs, ce do setcc(c, cc) end
3914 end

```

```

3915     end
3916     range(0x3400, 0x4DB5, 11, false)
3917     range(0x4DB5, 0x4DBF, 11, true)
3918     range(0x4E00, 0x9FCC, 11, false)
3919     range(0x9FCD, 0x9FFF, 11, true)
3920     range(0xAC00, 0xD7A3, 11, false)
3921     range(0x20000, 0x2A6D6, 11, false)
3922     range(0x2A6D7, 0x2A6FF, 11, true)
3923     range(0x2A700, 0x2B734, 11, false)
3924     range(0x2B735, 0x2B73F, 11, true)
3925     range(0x2B740, 0x2B81D, 11, false)
3926     range(0x2B81E, 0x2B81F, 11, true)
3927     range(0x2B820, 0x2CEA1, 11, false)
3928     range(0x2CEA2, 0x2FFFD, 11, true)
3929 } \fi

```

以上。

```
3930 \fi
```

G.4 完了

おしまい。

```
3931 %</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

```

3932 %<*cjkcat>
3933 \def\bxjx@pkgname{bxjscjkat}
3934 \newcount\bxjx@cnta

\bxjx@engine エンジンの種別。

3935 \let\bxjx@engine=n
3936 \def\bxjx@do#1#2{%
3937   \edef\bxjx@tmpa{\string#1}%
3938   \edef\bxjx@tmpb{\meaning#1}%
3939   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3940 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3941 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3942 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3943 \bxjx@do\pdftexversion{\let\bxjx@engine=p}
3944 \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。

```

3945 \def\bxjx@do#1#2{%
3946   \if#1\bxjx@engine
3947     \@ifpackageloaded{#2}{-}{%else
3948       \PackageError\bxjx@pkgname
3949         {Package '#2' must be loaded}%
3950         {Package loading is aborted.\MessageBreak\@ehc}%
3951     \endinput}
3952 \fi}
3953 \bxjx@do{p}{bxcjkjatype}
3954 \bxjx@do{x}{xeCJK}
3955 \bxjx@do{l}{luatexja}

```

古い L^AT_EX の場合、`\TextOrMath` は `fixltx2e` パッケージで提供される。

```

3956 \ifx\TextOrMath\undefined
3957   \RequirePackage{fixltx2e}
3958 \fi

```

H.2 和文カテゴリコードの設定

upL^AT_EX の場合、和文カテゴリコードの設定を LuaT_EX-ja と（ほぼ）等価なものに変更する。

※ LuaT_EX-ja との相違点：A830、A960、1B000。

```

3959 \if u\bxjx@engine
3960 \@for\bxjx@x:={%
3961 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3962 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3963 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3964 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3965 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3966 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
3967 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3968 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
3969 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3970 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
3971 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3972 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3973 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3974 10480,10500,10530,10600,10800,10840,10860,10880,%
3975 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
3976 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3977 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3978 11280,112B0,11300,11480,11580,11600,11680,11700,%
3979 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3980 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3981 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3982 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
3983 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3984 }\do{\kcatcode"\bxjx@x=15 }

```

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。

3986	\def\bjx@grkcyr@list{%	
3987	\do{0391}{LGR}{\textAlpha}{A}%	% GR. C. L. ALPHA
3988	\do{0392}{LGR}{\textBeta}{B}%	% GR. C. L. BETA
3989	\do{0393}{LGR}{\textGamma}{\Gamma}%	% GR. C. L. GAMMA
3990	\do{0394}{LGR}{\textDelta}{\Delta}%	% GR. C. L. DELTA
3991	\do{0395}{LGR}{\textEpsilon}{E}%	% GR. C. L. EPSILON
3992	\do{0396}{LGR}{\textZeta}{Z}%	% GR. C. L. ZETA
3993	\do{0397}{LGR}{\textEta}{H}%	% GR. C. L. ETA
3994	\do{0398}{LGR}{\textTheta}{\Theta}%	% GR. C. L. THETA
3995	\do{0399}{LGR}{\textIota}{I}%	% GR. C. L. IOTA
3996	\do{039A}{LGR}{\textKappa}{K}%	% GR. C. L. KAPPA
3997	\do{039B}{LGR}{\textLambda}{\Lambda}%	% GR. C. L. LAMDA
3998	\do{039C}{LGR}{\textMu}{M}%	% GR. C. L. MU
3999	\do{039D}{LGR}{\textNu}{N}%	% GR. C. L. NU
4000	\do{039E}{LGR}{\textXi}{\Xi}%	% GR. C. L. XI
4001	\do{039F}{LGR}{\textOmicron}{O}%	% GR. C. L. OMICRON
4002	\do{03A0}{LGR}{\textPi}{\Pi}%	% GR. C. L. PI
4003	\do{03A1}{LGR}{\textRho}{P}%	% GR. C. L. RHO
4004	\do{03A3}{LGR}{\textSigma}{\Sigma}%	% GR. C. L. SIGMA
4005	\do{03A4}{LGR}{\textTau}{T}%	% GR. C. L. TAU
4006	\do{03A5}{LGR}{\textUpsilon}{\Upsilon}%	% GR. C. L. UPSILON
4007	\do{03A6}{LGR}{\textPhi}{\Phi}%	% GR. C. L. PHI
4008	\do{03A7}{LGR}{\textChi}{X}%	% GR. C. L. CHI
4009	\do{03A8}{LGR}{\textPsi}{\Psi}%	% GR. C. L. PSI
4010	\do{03A9}{LGR}{\textOmega}{\Omega}%	% GR. C. L. OMEGA
4011	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
4012	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
4013	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
4014	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
4015	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
4016	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
4017	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
4018	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
4019	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
4020	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
4021	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
4022	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
4023	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
4024	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
4025	\do{03BF}{LGR}{\textomicron}{o}%	% GR. S. L. OMICRON

4026 \do{03C0}{LGR}{\textpi}{\pi}% % GR. S. L. PI
4027 \do{03C1}{LGR}{\textrho}{\rho}% % GR. S. L. RHO
4028 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
4029 \do{03C3}{LGR}{\textsigma}{\sigma}% % GR. S. L. SIGMA
4030 \do{03C4}{LGR}{\texttau}{\tau}% % GR. S. L. TAU
4031 \do{03C5}{LGR}{\textupsilon}{\upsilon}% % GR. S. L. UPSILON
4032 \do{03C6}{LGR}{\textphi}{\phi}% % GR. S. L. PHI
4033 \do{03C7}{LGR}{\textchi}{\chi}% % GR. S. L. CHI
4034 \do{03C8}{LGR}{\textpsi}{\psi}% % GR. S. L. PSI
4035 \do{03C9}{LGR}{\textomega}{\omega}% % GR. S. L. OMEGA
4036 \do{0401}{T2A}{\CYRYO}{-}% % CY. C. L. IO
4037 \do{0410}{T2A}{\CYRA}{-}% % CY. C. L. A
4038 \do{0411}{T2A}{\CYRB}{-}% % CY. C. L. BE
4039 \do{0412}{T2A}{\CYRV}{-}% % CY. C. L. VE
4040 \do{0413}{T2A}{\CYRG}{-}% % CY. C. L. GHE
4041 \do{0414}{T2A}{\CYRD}{-}% % CY. C. L. DE
4042 \do{0415}{T2A}{\CYRE}{-}% % CY. C. L. IE
4043 \do{0416}{T2A}{\CYRZH}{-}% % CY. C. L. ZHE
4044 \do{0417}{T2A}{\CYRZ}{-}% % CY. C. L. ZE
4045 \do{0418}{T2A}{\CYRI}{-}% % CY. C. L. I
4046 \do{0419}{T2A}{\CYRISHRT}{-}% % CY. C. L. SHORT I
4047 \do{041A}{T2A}{\CYRK}{-}% % CY. C. L. KA
4048 \do{041B}{T2A}{\CYRL}{-}% % CY. C. L. EL
4049 \do{041C}{T2A}{\CYRM}{-}% % CY. C. L. EM
4050 \do{041D}{T2A}{\CYRN}{-}% % CY. C. L. EN
4051 \do{041E}{T2A}{\CYRO}{-}% % CY. C. L. O
4052 \do{041F}{T2A}{\CYRP}{-}% % CY. C. L. PE
4053 \do{0420}{T2A}{\CYRR}{-}% % CY. C. L. ER
4054 \do{0421}{T2A}{\CYRS}{-}% % CY. C. L. ES
4055 \do{0422}{T2A}{\CYRT}{-}% % CY. C. L. TE
4056 \do{0423}{T2A}{\CYRU}{-}% % CY. C. L. U
4057 \do{0424}{T2A}{\CYRF}{-}% % CY. C. L. EF
4058 \do{0425}{T2A}{\CYRH}{-}% % CY. C. L. HA
4059 \do{0426}{T2A}{\CYRC}{-}% % CY. C. L. TSE
4060 \do{0427}{T2A}{\CYRCH}{-}% % CY. C. L. CHE
4061 \do{0428}{T2A}{\CYRSH}{-}% % CY. C. L. SHA
4062 \do{0429}{T2A}{\CYRSHCH}{-}% % CY. C. L. SHCHA
4063 \do{042A}{T2A}{\CYRHRDSN}{-}% % CY. C. L. HARD SIGN
4064 \do{042B}{T2A}{\CYRERY}{-}% % CY. C. L. YERU
4065 \do{042C}{T2A}{\CYRSFTSN}{-}% % CY. C. L. SOFT SIGN
4066 \do{042D}{T2A}{\CYREREV}{-}% % CY. C. L. E
4067 \do{042E}{T2A}{\CYRYU}{-}% % CY. C. L. YU
4068 \do{042F}{T2A}{\CYRYA}{-}% % CY. C. L. YA
4069 \do{0430}{T2A}{\cyra}{-}% % CY. S. L. A
4070 \do{0431}{T2A}{\cyrb}{-}% % CY. S. L. BE
4071 \do{0432}{T2A}{\cyrv}{-}% % CY. S. L. VE
4072 \do{0433}{T2A}{\cyrg}{-}% % CY. S. L. GHE
4073 \do{0434}{T2A}{\cyrd}{-}% % CY. S. L. DE
4074 \do{0435}{T2A}{\cyre}{-}% % CY. S. L. IE

```

4075 \do{0436}{T2A}{\cyrzh}{}%           % CY. S. L. ZHE
4076 \do{0437}{T2A}{\cyrz}{}%           % CY. S. L. ZE
4077 \do{0438}{T2A}{\cyri}{}%           % CY. S. L. I
4078 \do{0439}{T2A}{\cyrishrt}{}%       % CY. S. L. SHORT I
4079 \do{043A}{T2A}{\cyrk}{}%           % CY. S. L. KA
4080 \do{043B}{T2A}{\cyr1}{}%           % CY. S. L. EL
4081 \do{043C}{T2A}{\cyrm}{}%           % CY. S. L. EM
4082 \do{043D}{T2A}{\cyrn}{}%           % CY. S. L. EN
4083 \do{043E}{T2A}{\cyro}{}%           % CY. S. L. O
4084 \do{043F}{T2A}{\cyrp}{}%           % CY. S. L. PE
4085 \do{0440}{T2A}{\cyr r}{}%          % CY. S. L. ER
4086 \do{0441}{T2A}{\cyr s}{}%          % CY. S. L. ES
4087 \do{0442}{T2A}{\cyr t}{}%          % CY. S. L. TE
4088 \do{0443}{T2A}{\cyr u}{}%          % CY. S. L. U
4089 \do{0444}{T2A}{\cyr f}{}%          % CY. S. L. EF
4090 \do{0445}{T2A}{\cyr h}{}%          % CY. S. L. HA
4091 \do{0446}{T2A}{\cyr c}{}%          % CY. S. L. TSE
4092 \do{0447}{T2A}{\cyr ch}{}%         % CY. S. L. CHE
4093 \do{0448}{T2A}{\cyr sh}{}%         % CY. S. L. SHA
4094 \do{0449}{T2A}{\cyrshch}{}%        % CY. S. L. SHCHA
4095 \do{044A}{T2A}{\cyrhrdsn}{}%       % CY. S. L. HARD SIGN
4096 \do{044B}{T2A}{\cyr rery}{}%       % CY. S. L. YERU
4097 \do{044C}{T2A}{\cyr sftsn}{}%      % CY. S. L. SOFT SIGN
4098 \do{044D}{T2A}{\cyr rerev}{}%      % CY. S. L. E
4099 \do{044E}{T2A}{\cyr yu}{}%         % CY. S. L. YU
4100 \do{044F}{T2A}{\cyr ya}{}%         % CY. S. L. YA
4101 \do{0451}{T2A}{\cyr yo}{}%         % CY. S. L. IO
4102 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4103 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4104 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4105 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
4106 \do{00B4}{TS1}{\textasciicute}{}%  % ACUTE ACCENT
4107 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4108 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4109 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
4110 }

4111 \providecommand*{\mathdegree}{\circ}

\ifbxjx@gcc@cj k 〔スイッチ〕ギリシャ・キリル文字を和文扱いにするか。
4112 \newif\ifbxjx@gcc@cj k

\greekasCJ K ギリシャ・キリル文字を和文扱いにする。

\nogreekasCJ K ギリシャ・キリル文字を欧文扱いにする。
4113 \newcommand*\greekasCJ K{%
4114   \bxjx@gcc@cj ktrue}
4115 \newcommand*\nogreekasCJ K{%
4116   \bxjx@gcc@cj kfalse}

```

`\bx@fake@grk` `\bx@fake@grk{(出力文字)}{(基準文字)}` :

```
4117 \def\bxjx@do#1\relax{%
4118   \def\bxjx@fake@grk##1##2{%
4119     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4120   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4121     \ifx\##1\%
4122       \bxjx@cnta##4\divide\bxjx@cnta\@cclvi
4123       \multiply\bxjx@cnta\@cclvi \advance\bxjx@cnta`##3\relax
4124       \mathchar\bxjx@cnta
4125     \else ##3\fi}
4126 } \expandafter\bxjx@do\string\mathchar\relax
```

■pdfLaTeX・upLaTeX の場合

```
4127 \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。

```
4128 \@ifpackageloaded{inputenc}{}{%else
4129   \RequirePackage[utf8]{inputenc}}
4130 \def\bxjx@tmpa{utf8}
4131 \ifx\bxjx@tmpa\inputencdoingname
4132   \PackageWarningNoLine\bxjx@pkgname
4133     {Input encoding changed to utf8}%
4134   \inputencoding{utf8}%
4135 \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。

```
4136 \if u\bxjx@engine
4137 \kcatcode"0370=15
4138 \kcatcode"0400=15
4139 \kcatcode"0500=15
4140 \fi
```

各文字について `\DeclareUnicodeCharacter` を実行する。

```
4141 \def\do#1{%
4142   \@tempcnta=#1\relax
4143   \@tempcntb=\@tempcnta \divide\@tempcntb256
4144   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4145   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4146 \def\bxjx@do@a#1#2#3#4#5{%
4147   \ifx\##5\%
4148     \def\bxjx@tmpa{\@inmathwarn#4}%
4149   \else\ifcat A\noexpand#5%
4150     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4151       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4152   \else \def\bxjx@tmpa{#5}%
4153   \fi\fi
4154   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4155   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
4156 \if u\bxjx@engine
```

```

4157 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
4158 \def\bxjx@do@b#1#2#3#4#5{%
4159   \kchardef#1=\@tempcnta
4160   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4161   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4162 \else\if p\bxjx@engine
4163 \def\bxjx@do@b#1#2#3#4#5{%
4164   \mathchardef#1=\@tempcnta
4165   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
4166   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4167 \fi\fi
4168 \bxjx@grkcyr@list
4169 \let\bxjx@do@a\undefined
4170 \let\bxjx@do@b\undefined

```

`\bxjx@DeclareUnicodeCharacter` `\bxjx@DeclareUnicodeCharacter` を変更して、ギリシャ・キリル文字の場合に再定義を抑制したもの。

```

4171 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4172 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4173 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4174 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4175   \count@=#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
4176   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
4177     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4178   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4179     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4180   \else
4181     \wlog{ \space\space skipped defining Unicode char U+#1}%
4182   \fi\fi}

```

`\bxjx@ja@or@not` `\bxjx@ja@or@not`

```

4183 \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。
4184   \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。
4185   \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4186   \else \UseTextSymbol{#2}{#3}%
4187   \fi\fi}

```

`\DeclareFontEncoding@` `\DeclareFontEncoding@` にパッチを当てて、`\DeclareFontEncoding` の実行中だけ変更後の `\DeclareUnicodeCharacter` が使われるようにする。

```

4188 \begingroup
4189 \toks@\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}
4190 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4191   \noexpand\bxjx@swap@DUC@cmd
4192   \the\toks@

```

```

4193 \noexpand\bxjx@swap@DUC@cmd}}
4194 \endgroup\next
4195 \def\bxjx@swap@DUC@cmd{%
4196 \let\bxjx@tmpa\DeclareUnicodeCharacter
4197 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4198 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■ Xe_{La}TeX・Lua_{La}TeX の場合

```
4199 \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。

```

4200 \def\do#1{%
4201 \bxjx@canta="#1\relax
4202 \begingroup
4203 \lccode`~=\bxjx@canta
4204 \lowercase{\endgroup
4205 \bxjx@do@a{~}}{#1}}
4206 \def\bxjx@do@a#1#2#3#4#5{%
4207 \ifx\#5\\\let\bxjx@tmpa\relax
4208 \else\ifcat A\noexpand#5%
4209 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4210 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4211 \else \def\bxjx@tmpa{#5}%
4212 \fi\fi
4213 \ifx\bxjx@tmpa\relax\else
4214 \mathcode\bxjx@canta"8000 \let#1\bxjx@tmpa
4215 \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。

```

4216 \mathchardef\bxjx@tmpa="119
4217 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4218 \let\bxjx@do@a\undefined

```

Lua_{TeX} における \(\no)greekasCJK の定義。jacharrange の設定を変更する。

```

4219 \if 1\bxjx@engine
4220 \protected\def\greekasCJK{%
4221 \bxjx@gcc@cjctrue
4222 \ltjsetparameter{jacharrange={+2, +8}}
4223 \protected\def\nogreekasCJK{%
4224 \bxjx@gcc@cjcfalse
4225 \ltjsetparameter{jacharrange={-2, -8}}
4226 \fi

```

X_eTeX における \(\no)greekasCJK の定義。

```

4227 \if x\bxjx@engine
4228 \protected\def\greekasCJK{%
4229 \bxjx@gcc@cjctrue

```

```
4230 \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4231 \bxjx@grkcyr@list}
4232 \protected\def\nogreekasCJK{%
4233 \bxjx@gcc@cjkgfalse
4234 \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4235 \bxjx@grkcyr@list}
4236 \fi
    以上。
4237 \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。

```
4238 \nogreekasCJK
```

H.5 完了

おしまい。

```
4239 %</cjkgcat>
```

補助パッケージ実装はここまで。

```
4240 %</anc>
```