

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”)

v1.3a [2017/01/28]

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	31
4	フォントサイズ	32
5	レイアウト	37
5.1	ページレイアウト	38
6	ページスタイル	45
7	文書のマークアップ	49
7.1	表題	49
7.2	章・節	54
7.3	リスト環境	65
7.4	パラメータの設定	72
7.5	フロート	73
7.6	キャプション	75
8	フォントコマンド	76
9	相互参照	78

9.1	目次の類	78
9.2	参考文献	83
9.3	索引	85
9.4	脚注	86
10	段落の頭へのグルー挿入禁止	89
11	いろいろなロゴ	92
12	amsmath との衝突の回避	93
13	初期設定	93
付録 A	和文ドライバの仕様 ㊦	97
付録 B	和文ドライバ : minimal ㊦	98
B.1	補助マクロ	98
B.2	(u)pTeX 用の設定	100
B.3	pdfTeX 用の処理	104
B.4	X _Y TeX 用の処理	104
B.5	後処理 (エンジン共通)	105
付録 C	和文ドライバ : standard ㊦	107
C.1	共通処理 (1)	107
C.2	pTeX 用設定	112
C.3	pdfTeX 用設定 : CJK + bxcjkatype	115
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	116
C.5	LuaTeX 用設定 : LuaTeX-ja	118
C.6	共通処理 (2)	120
付録 D	和文ドライバ : modern ㊦	120
D.1	フォント設定	120
D.2	fixltx2e 読込	121
D.3	和文カテゴリコード	121
D.4	完了	121
付録 E	和文ドライバ : pandoc ㊦	121
E.1	dupload システム	122
E.2	lang 変数	123
E.3	geometry 変数	123
E.4	CJKmainfont 変数	124
E.5	fixltx2e パッケージ	124
E.6	cmap パッケージ	124

E.7	microtype パッケージ	124
E.8	完了	124
付録 F	補助パッケージ一覧 𐄂	124
付録 G	補助パッケージ : bxjscompat 𐄂	125
G.1	準備	125
G.2	X _Y TeX 部分	125
G.3	LuaTeX 部分	126
G.4	完了	127
付録 H	補助パッケージ : bxjscjkat 𐄂	127
H.1	準備	127
H.2	和文カテゴリコードの設定	128
H.3	ギリシャ・キリル文字の扱い	129
H.4	初期設定	135
H.5	完了	135

1 はじめに

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

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

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

これは L^AT_EX3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいて奥村が改変したものです。権利については両者のものに従います。奥村は何の権利も主張しません。

[2009-02-22] 田中琢爾氏による up_YTeX 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 (TEX の種類) の種別：j = pTEX 系、x = XYTEX、p = pdfTEX (含 DVI モード)、l = LuaTEX、J = NTT jTEX、O = Omega 系、n = 以上の何れでもない。
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=O}
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) upTEX 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\eTeXversion{\jsWitheTeXtrue}

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

`\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。)

```
71 \def\jsAtEndOfClass{%
72   \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。

```
73 \@onlypreamble\bxjs@restore@jltrcc
74 \let\bxjs@restore@jltrcc\@empty
75 \if 1\jsEngine
76 \def\bxjs@change@jltrcc#1{%
77   \xdef\bxjs@restore@jltrcc{%
78     \bxjs@restore@jltrcc
79     \catcode`#1=\the\catcode`#1\relax}%
80   \catcode`#1=11\relax}
81 \@tfor\bxjs@x:=西暦\do
82   {\expandafter\bxjs@change@jltrcc\bxjs@x}
83 \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。

```
84 \bxjs@robust\def\jsInhibitGlue{%
85   \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。

```
86 \if@compatibility
87   \ClassError\bxjs@clsname
88     {Something went chaotic!\MessageBreak
89     (How come '\string\documentstyle' is there?)\MessageBreak
90     I cannot go a single step further...}
91     {If the chant of '\string\documentstyle' was just a blunder of yours,\MessageBreak
92     then there'll still be hope....}
93   \expandafter\@firstofone
94 \else \expandafter\@gobble
95 \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。
96 \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。
97 \newif\if@titlepage

\if@openright \chapter, \part を奇数ページ起こしにするかどうかです。書籍では真が標準です。
98 %<book|report>\newif\if@openright

\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。
99 %<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。
100 \newif\if@enablejfam \@enablejfamtrue
```

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

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

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

```
101 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
102 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
103 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
104 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
105 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
106 \DeclareOption{b4paper}{\bxjs@setpaper{{257true mm}{364true mm}}}
107 \DeclareOption{b5paper}{\bxjs@setpaper{{182true mm}{257true mm}}}
```

```

108 \DeclareOption{b6paper}{\bxjs@setpaper{{128truemm}{182truemm}}}
109 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
110 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
111 \DeclareOption{b4j}{\bxjs@setpaper{{257truemm}{364truemm}}}
112 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
113 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
114 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
115 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
116 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
117 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

118 \newif\if@landscape
119 \@landscapefalse
120 \DeclareOption{landscape}{\@landscapetrue}

```

■slide オプション slide を新設しました。

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

```

121 \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。

```

122 %<!slide>\@slidefalse
123 %<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。

```
124 \newcommand{\@ptsize}{0}
125 \def\bxjs@param@basefontsize{10pt}
126 \def\jsUnusualPtSize{-20}
```

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

```
127 \def\bxjs@setbasefontsize#1{%
128   \bxjs@setbasefontlength\@tempdima{#1}%
129   \edef\bxjs@param@basefontsize{\the\@tempdima}%
130   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
131   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
132   \else\ifdim\@tempdima=12pt  \long\def\@ptsize{2}%
133   \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。

```
134 \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。

```
135   \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。

```
136   \ifx j\jsEngine \setlength#1{#2}%
137   \else
138     \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
139     \ifx\bxjs@tmpa\relax \setlength#1{#2}%
140     \else \@tempdimc0.25mm #1=\bxjs@tmpa\@tempdimc
141     \fi
142   \fi}
143 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
144   \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
145   \else \def\bxjs@tmpa{#1}%
146   \fi}
147 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
148   \ifx\@nnil#2\@nnil\else
149     \ClassWarningNoLine\bxjs@clsname
150     {You should not use 'true' lengths here}%
151   \fi}
```

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

```
152 \newif\ifjsc@mag
153 \newif\ifjsc@mag@xreal
154 %\let\jsc@magscale\@undefined
155 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
```

```

156 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
157 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
158 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
159 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
160 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
161 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
162 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
163 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
164 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
165 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
166 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
167 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
168 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
169 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
170 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
171 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
172 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
173 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

174 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
175 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
176 \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。

```

177 \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。

```

178 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
179 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
180 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 twocolumn で二段組になります。

```
181 \DeclareOption{onecolumn}{\@twocolumnfalse}
182 \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。

```
183 \DeclareOption{titlepage}{\@titlepagetrue}
184 \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, openany で偶数ページからでも始まるようになります。

```
185 %<book|report>\DeclareOption{openright}{\@openrighttrue}
186 %<book|report>\DeclareOption{openany}{\@openrightfalse}
```

■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。

```
187 \def\eqnarray{%
188   \stepcounter{equation}%
189   \def\@currentlabel{\p@equation\theequation}%
190   \global\@eqnswtrue
191   \m@th
192   \global\@eqcnt\z@
193   \tabskip\@centering
194   \let\\\@eqnocr
195   $$\everycr{}\halign to\displaywidth\bgroup
196     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
197     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
198     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
199     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
200     \tabskip\z@skip
201   \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。

```
202 \DeclareOption{leqno}{\input{leqno.clo}}
203 \DeclareOption{fleqn}{\input{fleqn.clo}}
204 % fleqn 用の eqnarray 環境の再定義
205 \def\eqnarray{%
206   \stepcounter{equation}%
207   \def\@currentlabel{\p@equation\theequation}%
208   \global\@eqnswtrue\m@th
209   \global\@eqcnt\z@
210   \tabskip\mathindent
211   \let\=\@eqnocr
212   \setlength\abovedisplayskip{\topsep}%
213   \ifvmode
```

```

214 \addtolength\abovedisplayskip{\partopsep}%
215 \fi
216 \addtolength\abovedisplayskip{\parskip}%
217 \setlength\belowdisplayskip{\abovedisplayskip}%
218 \setlength\belowdisplayshortskip{\abovedisplayskip}%
219 \setlength\abovedisplayshortskip{\abovedisplayskip}%
220 $$\everycr{}\halign to\linewidth% $$
221 \bgroup
222 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
223 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
224 &\global\@eqcnt\tw@
225 $\displaystyle{##}$\hfil \tabskip\@centering
226 &\global\@eqcnt\thr@@ \hbext\z@\bgroup\hss##\egroup
227 \tabskip\z@skip\cr
228 }}

```

■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。

```

229 % \DeclareOption{openbib}{%
230 % \AtEndOfPackage{%
231 % \renewcommand\@openbib@code{%
232 % \advance\leftmargin\bibindent
233 % \itemindent -\bibindent
234 % \listparindent \itemindent
235 % \parsep \z@}%
236 % \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

ここは和文ドライバの管轄。

■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。

```

237 \let\ifjsDraft\iffalse
238 \@onlypreamble\bxjs@draft
239 \def\bxjs@draft#1{%
240 \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
241 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
242 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
243 \AtBeginDocument{%

```

```

244 \expandafter\ifx\csname ifdraft\endcsname\relax
245 \expandafter\let\csname ifdraft\endcsname\expandafter\endcsname
246 \csname ifjsDraft\endcsname
247 \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。

```

248 \newif\ifbxjs@papersize
249 \bxjs@papersizetrue
250 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
251 \DeclareOption{papersize}{\bxjs@papersizetrue}

```

■英語化 オプション english を新設しました。

```

252 \newif\if@english
253 \@englishfalse
254 \DeclareOption{english}{\@englishttrue}

```

■jsreport 相当 オプション report を新設しました。

BXJS では ‘report’ 相当のものは別に `bxjsreport` クラスとして用意する。

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

BXJS クラスでは、`nojslogo` を既定とする。

```

255 \newif\if@jslogo \@jslogofalse
256 \DeclareOption{jslogo}{\@jslogotrue}
257 \DeclareOption{nojslogo}{\@jslogofalse}

```

-
- 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 10 pt 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 (= 13 Q/10 pt)。
 - 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：（予定）

- dummy-microtype : (予定)
- hyperref-enc : (予定)

`\bxjs@invscale` `\bxjs@invscale` は $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ における「長さのスケール」の逆関数を求めるもの。例えば `\bxjs@invscale\dimX{1.3}` は `\dimX=1.3\dimX` の逆の演算を行う。
 ※局所化の `\begingroup~\endgroup` について、以前は `\group~\egroup` を使っていたが、これだと数詞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。

```

258 \mathchardef\bxjs@csta=259
259 \def\bxjs@invscale#1#2{%
260   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
261     \@tempcnta\@tempdima \multiply\@tempcnta\ccclvi
262     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\ccclvi
263     \@tempcntb\p@ \divide\@tempcntb\@tempdimb
264     \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
265     \@tempdimb\@tempcnta\@ne
266     \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
267     \advance\@tempcnta\bxjs@csta \@tempdimc\@tempcnta\@ne
268     \@whiledim\@tempdimb<\@tempdimc\do{%
269       \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
270       \advance\@tempcntb\@ne \divide\@tempcntb\tw@
271       \ifdim #2\@tempcntb>\@tempdima
272         \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
273       \else \@tempdimb=\@tempcntb\@ne \fi}%
274     \xdef\bxjs@gtmpa{\the\@tempdimb}%
275   \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。

```

276 \let\bxjs@composite@proc\relax

```

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

```

277 \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。

```

278   \def\bxjs@composite@proc{%
279     \bxjs@oldfontcommandstrue
280     \setkeys{bxjs}{ja=pandoc}%
281     \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…。

```
282 \def\bxjs@driver@opt{dvipdfmx}%
283 \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。

```
284 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
285 %\let\bxjs@engine@opt\@undefined
```

エンジン明示指定のオプションの処理。

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

```
286 \DeclareOption{autodetect-engine}{%
287   \let\bxjs@engine@given=*}
288 \DeclareOption{latex}{%
289   \def\bxjs@engine@opt{latex}%
290   \let\bxjs@engine@given=n}
291 \DeclareOption{platex}{%
292   \def\bxjs@engine@opt{platex}%
293   \let\bxjs@engine@given=j}
294 \DeclareOption{uplatex}{%
295   \def\bxjs@engine@opt{uplatex}%
296   \let\bxjs@engine@given=u}
297 \DeclareOption{xelatex}{%
298   \def\bxjs@engine@opt{xelatex}%
299   \let\bxjs@engine@given=x}
300 \DeclareOption{pdflatex}{%
301   \def\bxjs@engine@opt{pdflatex}%
302   \let\bxjs@engine@given=p}
303 \DeclareOption{lualatex}{%
304   \def\bxjs@engine@opt{lualatex}%
305   \let\bxjs@engine@given=l}
306 \DeclareOption{platex-ng}{%
307   \def\bxjs@engine@opt{platex-ng}%
308   \let\bxjs@engine@given=g}
309 \DeclareOption{platex-ng*}{%
310   \def\bxjs@engine@opt{platex-ng*}%
311   \let\bxjs@platexng@nodrv=t%
312   \let\bxjs@engine@given=g}
```

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


```

313 %\let\bxjs@driver@given\@undefined
314 \let\bxjs@driver@@dvimode=0
315 \let\bxjs@driver@@dvipdfmx=1
316 \let\bxjs@driver@@pdfmode=2
317 \let\bxjs@driver@@xetex=3

```

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

```

318 %\let\bxjs@driver@opt\@undefined

319 \DeclareOption{dvips}{%
320   \def\bxjs@driver@opt{dvips}%
321   \let\bxjs@driver@given\bxjs@driver@@dvimode}
322 \DeclareOption{dviout}{%
323   \def\bxjs@driver@opt{dviout}%
324   \let\bxjs@driver@given\bxjs@driver@@dvimode}
325 \DeclareOption{xdvi}{%
326   \def\bxjs@driver@opt{xdvi}%
327   \let\bxjs@driver@given\bxjs@driver@@dvimode}
328 \DeclareOption{dvipdfmx}{%
329   \def\bxjs@driver@opt{dvipdfmx}%
330   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
331 \DeclareOption{pdftex}{%
332   \def\bxjs@driver@opt{pdftex}%
333   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
334 \DeclareOption{luatex}{%
335   \def\bxjs@driver@opt{luatex}%
336   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
337 \DeclareOption{xetex}{%
338   \def\bxjs@driver@opt{xetex}%
339   \let\bxjs@driver@given\bxjs@driver@@xetex}

```

「もし DVI モードであればドライバを `dvipdfmx` にする」というオプション。

※ 1.2 版で `dvi` オプションが新設されたが、互換性のためこのオプションも残す。

```

340 \DeclareOption{dvipdfmx-if-dvi}{%
341   \setkeys{bxjs}{dvi=dvipdfmx}}

```

■その他の BXJS 独自オプション

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

```

342 \newif\ifbxjs@usezw \bxjs@usezwtrue

```

`zw`、`noz` オプションの定義。

```

343 \DeclareOption{noz}{%
344   \bxjs@usezwfalse}
345 \DeclareOption{zw}{%
346   \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。

```

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

    nojs、js オプションの定義。
348 \DeclareOption{nojs}{%
349   \bxjs@disguise@jsfalse}
350 \DeclareOption{js}{%
351   \bxjs@disguise@jstrue}

```

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

```

352 \newif\ifbxjs@precisetext

    noprecisetext / precisetext オプションの定義。
353 \DeclareOption{noprecisetext}{%
354   \bxjs@precisetextfalse}
355 \DeclareOption{precisetext}{%
356   \bxjs@precisetexttrue}

```

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

```

357 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

    nosimplejasetup / simplejasetup オプションの定義。
358 \DeclareOption{nosimplejasetup}{%
359   \bxjs@simplejasetupfalse}
360 \DeclareOption{simplejasetup}{%
361   \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。

```

362 \newif\ifbxjs@bigcode \bxjs@bigcodefalse

```

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

```

363 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
364 \def\bxjs@tmpb#1 #2#3\@nil{%
365   \ifx1#2\bxjs@bigcodetrue \fi}
366 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil

    nobigcode / bigcode オプションの定義。
367 \DeclareOption{nobigcode}{%
368   \bxjs@bigcodefalse}
369 \DeclareOption{bigcode}{%
370   \bxjs@bigcodetrue}

```

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

```

371 \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。

```
372 \DeclareOption{nooldfontcommands}{%
373   \bxjs@oldfontcommandsfalse}
374 \DeclareOption{oldfontcommands}{%
375   \bxjs@oldfontcommandstrue}
```

■keyval 型のオプション

```
376 \def\bxjs@setkey{%
377   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
378 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
379 \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。

```
380 \def\bxjs@safe@setkeys#1#2{%
381   \let\bxjs@KV@errx\KV@errx
382   \let\KV@errx\bxjs@safe@setkeys@a
383   \setkeys{#1}{#2}%
384   \let\KV@errx\bxjs@KV@errx}
385 \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。

```
386 \def\bxjs@set@keyval#1#2#3{%
387   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
388   \ifx\bxjs@next\relax
389     \bxjs@error@keyval{#1}{#2}%
390     #3%
391   \else \bxjs@next
392   \fi}
393 \@onlypreamble\bxjs@error@keyval
394 \def\bxjs@error@keyval#1#2{%
395   \ClassError\bxjs@clsname
396   {Invalid value '#2' for option #1}\@ehc}
```

\ifbxjs@scaleset 和文スケール値が指定されたか。

```
397 \newif\ifbxjs@scaleset
```

\jsScale 〔実数値マクロ〕和文スケール値。

```
398 \def\jsScale{0.924715}
```

base オプションの処理。

```
399 \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。

```
400 \onlypreamble\bxjs@do@opt@jbase
401 \let\bxjs@do@opt@jbase\relax
402 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
403 \def\bxjs@setjbasefontsize#1{%
404   \def\bxjs@do@opt@jbase{%
405     \bxjs@setbasefontlength\@tempdima{#1}%
406     \bxjs@invscale\@tempdima\jsScale
407     \bxjs@setbasefontsize{\@tempdima}}
```

scale オプションの処理。

```
408 \define@key{bxjs}{scale}{%
409   \bxjs@scalesettrue
410   \edef\jsScale{#1}}
```

noscale オプションの処理。

```
411 \DeclareOption{noscale}{%
412   \bxjs@scalesettrue
413   \def\jsScale{1}}
```

\bxjs@param@mag mag オプションの値。

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

mag オプションの処理。

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

paper オプションの処理。

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

\bxjs@jadriver 和文ドライバの名前。

```
417 \let\bxjs@jadriver\relax
418 %\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)。

```
419 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
420 \define@key{bxjs}{ja}[\relax]{%
421   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}
```

\jsJaFont 和文フォント設定の名前。

```
422 \let\jsJaFont\@empty
```

jafont オプションの処理。

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

\jsJaParam 和文ドライバパラメタの文字列。

```
424 \let\jsJaParam\@empty
```

japaram オプションの処理。

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

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

```
426 \let\bxjs@magstyle@mag=m
```

```
427 \let\bxjs@magstyle@real=r
```

```
428 \let\bxjs@magstyle@xreal=x
```

(新しい素敵な名前。)

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

```
429 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
```

```
430 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
```

```
431 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

\bxjs@magstyle@default は既定の値を表す。

```
432 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
```

```
433 \ifx l\jsEngine \ifnum\luatexversion>86
```

```
434 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
```

```
435 \fi\fi
```

```
436 \ifjsWithpTeXng
```

```
437 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
```

```
438 \fi
```

```
439 \let\bxjs@magstyle\bxjs@magstyle@default
```

magstyle オプションの処理。

```
440 \define@key{bxjs}{magstyle}{%
```

```
441 \expandafter\let\expandafter\bxjs@magstyle\csname
```

```
442 bxjs@magstyle@#1\endcsname
```

```
443 \ifx\bxjs@magstyle\relax
```

```
444 \ClassError\bxjs@clsname
```

```
445 {Invalid value '#1' for option magstyle}\@ehc
```

```
446 \let\bxjs@magstyle\bxjs@magstyle@default
```

```
447 \fi}
```

\bxjs@geometry geometry オプションの値。

```
448 \let\bxjs@geometry@class=c
```

```
449 \let\bxjs@geometry@user=u
```

```
450 \let\bxjs@geometry\bxjs@geometry@class
```

geometry オプションの処理。

```
451 \define@key{bxjs}{geometry}{%
```

```
452 \expandafter\let\expandafter\bxjs@geometry\csname
```

```
453 bxjs@geometry@#1\endcsname
```

```
454 \ifx\bxjs@geometry\relax
```

```
455 \ClassError\bxjs@clsname
```

```
456 {Invalid value '#1' for option geometry}\@ehc
```

```
457 \let\bxjs@geometry\bxjs@geometry@class
```

```
458 \fi}
```

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

```
459 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

fancyhdr オプションの処理。

460 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
461 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
462 \define@key{bxjs}{fancyhdr}{%
463   \bxjs@set@keyval{fancyhdr}{#1}{}}
```

`\ifbxjs@dvi@opt` dvi オプションが指定されたか。

```
464 \newif\ifbxjs@dvi@opt

DVI モードのドライバとドライバ種別との対応。

465 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
466 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvimode
467 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
468 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode

dvi オプションの処理。

469 \define@key{bxjs}{dvi}{%
470   \expandafter\let\expandafter\bxjs@tmpa\csname
471     bxjs@dvidriver@@#1\endcsname
472   \ifx\bxjs@tmpa\relax
473     \ClassError\bxjs@clsname
474       {Invalid value '#1' for option dvi}\@ehc
475   \else

\bxjs@driver@given を未定義にしていることに注意。

476   \def\bxjs@driver@opt{#1}%
477   \let\bxjs@driver@given\@undefined
478   \bxjs@dvi@opttrue
479 \fi}
```

`\ifbxjs@layout@buggyhmargin` [スイッチ] bxjsbook の左右マージンがアレか。

※既定はアレだが 1.3 版で非アレになる予定。

```
480 \newif\ifbxjs@layout@buggyhmargin \bxjs@layout@buggyhmarginfalse

layout オプションの処理。

481 \@namedef{bxjs@kv@layout@v1}{%
482   \bxjs@layout@buggyhmargintrue}
483 \@namedef{bxjs@kv@layout@v2}{%
484   \bxjs@layout@buggyhmarginfalse}
485 \define@key{bxjs}{layout}{%
486   \bxjs@set@keyval{layout}{#1}{}}
```

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

```
487 %\let\bxjs@textwidth@limit\@undefined
488 \define@key{bxjs}{textwidth-limit}{%
489   \edef\bxjs@textwidth@limit{#1}}
```

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

```
490 %\let\bxjs@paragraph@mark\@undefined
491 \define@key{bxjs}{paragraph-mark}{%
492   \edef\bxjs@paragraph@mark{#1}}
```

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

```
493 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
494 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
495 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
496 \define@key{bxjs}{whole-zw-lines}{\bxjs@set@keyval{wholezwlines}{#1}{}}
```

`\ifbxjs@dummy@microtype` [スイッチ] dummy-microtype の指定値。

```
497 \newif\ifbxjs@dummy@microtype \bxjs@dummy@microtypetrue
498 \let\bxjs@kv@dummymicrotype@true\bxjs@dummy@microtypetrue
499 \let\bxjs@kv@dummymicrotype@false\bxjs@dummy@microtypefalse
500 \define@key{bxjs}{dummy-microtype}{\bxjs@set@keyval{dummymicrotype}{#1}{}}
```

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

```
501 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
502 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
503 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
504 \define@key{bxjs}{hyperref-enc}{\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。

```
505 \let\bxjs@ltx@removeelement\@removeelement
506 \def\@removeelement#1#2#3{%
507   \def\reserved@a{#2}%
508   \ifx\reserved@a\@empty \let#3\@empty
509   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
510   \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。

```
511 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
512 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
513 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
514 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
```

```

515 \ProcessOptions\relax
516 \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処理で不具合を起こすようである (\ProcessOptions* がエラーになる)。従って、このようなオプションは除外することにする。

```

517 \@onlypreamble\bxjs@purge@brace@elts
518 \def\bxjs@purge@brace@elts{%
519   \def\bxjs@tmpa{\@gobble}%
520   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
521   \let\@classoptionslist\bxjs@tmpa}
522 \@onlypreamble\bxjs@purge@be@a
523 \def\bxjs@purge@be@a#1,{%
524   \ifx\@nil#1\relax\else
525     \bxjs@purge@be@b#1}\@nil
526     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
527     \expandafter\bxjs@purge@be@a
528   \fi}
529 \@onlypreamble\bxjs@purge@be@b
530 \def\bxjs@purge@be@b#1#{\bxjs@purge@be@c}
531 \@onlypreamble\bxjs@purge@be@c
532 \def\bxjs@purge@be@c#1\@nil{%
533   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
534 \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。

```

535 \@expandtwoargs\@removeelement
536 {papersize}\@classoptionslist\@classoptionslist
537 \@expandtwoargs\@removeelement
538 {10pt}\@classoptionslist\@classoptionslist
539 \@expandtwoargs\@removeelement
540 {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。

```

541 \ifx\bxjs@jadriver\relax\else
542   \let\bxjs@jadriver@given\bxjs@jadriver
543 \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。

```

544 \let\bxjs@tmpb\jsEngine
545 \ifx j\bxjs@tmpb\ifjsWithpTeXng
546   \let\bxjs@tmpb=g
547 \fi\fi
548 \ifx j\bxjs@tmpb\ifjsWithupTeX
549   \let\bxjs@tmpb=u
550 \fi\fi

```



```

551 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
552   \let\bxjs@tmpb=n
553 \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。)

```

554 \ifx *\bxjs@engine@given
555   \let\bxjs@engine@given\bxjs@tmpb

```

エンジン指定が autotect-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。

```

556   \ifx j\bxjs@engine@given
557     \g@addto@macro\@classoptionslist{,latex}
558   \else\ifx u\bxjs@engine@given
559     \g@addto@macro\@classoptionslist{,uplatex}
560   \fi\fi
561 \fi
562 \ifx\bxjs@engine@given\@undefined\else
563   \ifx\bxjs@engine@given\bxjs@tmpb\else
564     \ClassError\bxjs@clsname
565       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
566   \fi
567 \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。

```

568 \ifjsWithpTeXng
569   \g@addto@macro\@classoptionslist{,uplatex}
570 \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。

```

571 \@tempswatrue
572 \ifx \bxjs@driver@given\@undefined\else
573   \ifjsInPdfMode
574     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
575       \@tempswafalse
576     \fi
577   \else\ifx x\jsEngine
578     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
579       \@tempswafalse
580     \fi
581   \else
582     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
583       \@tempswafalse
584     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
585       \@tempswafalse
586     \fi\fi
587   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
588     \@tempswafalse
589   \fi\fi
590 \fi\fi

```

```

591 \fi
592 \if@tempswa\else
593   \ClassError\bxjs@clsname
594   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
595 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

596 \ifjsInPdfMode \@tempswafalse
597 \else\ifx x\jsEngine \@tempswafalse
598 \else\ifjsWithpTeXng \@tempswafalse
599 \else \@tempwatrue
600 \fi\fi\fi
601 \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)。

```

602   \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
603     \ClassWarningNoLine\bxjs@clsname
604     {No driver option is given}
605   \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。)

```

606   \ifbxjs@dvi@opt
607     \edef\bxjs@nxt{%
608       \let\noexpand\bxjs@driver@given
609       \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
610       \noexpand\g@addto@macro\noexpand\@classoptionslist
611       {,\bxjs@driver@opt}%
612     }\bxjs@nxt
613   \fi
614 \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。

```

615 \ifjsWithpTeXng
616   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
617     \let\bxjs@platexng@nodrv\@undefined
618   \else\ifx t\bxjs@platexng@nodrv\else
619     \g@addto@macro\@classoptionslist{,dvipdfmx}
620   \fi\fi
621 \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。

```

622 \def\bxjs@@minimal{minimal}
623 \ifx\bxjs@jadriver\relax
624   \ifx j\jsEngine

```

```

625 \def\bxjs@jadriver{standard}
626 \else
627 \let\bxjs@jadriver\bxjs@@minimal
628 \fi
629 \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。

```

630 \ifx\bxjs@jadriver@given\@undefined\else
631 \ifx\bxjs@engine@given\@undefined
632 \ClassError\bxjs@clsname
633 {An engine option must be explicitly given}%
634 {When you use a Japanese-driver you must specify a correct\MessageBreak
635 engine option.\MessageBreak\@ehc}
636 \fi\fi

```

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

```

637 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
638 \ifx\bxjs@magstyle\bxjs@magstyle@mag
639 \let\bxjs@magstyle\bxjs@magstyle@default
640 \ClassError\bxjs@clsname
641 {The engine does not support 'magstyle=usemag'}%
642 {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
643 The default value 'nomag*' is used instead.\MessageBreak \@ehc}
644 \fi
645 \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。

```

646 \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。

```

647 \let\Cjascale\jsScale

```

後処理

```

648 \if@slide
649 \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
650 \fi
651 \if@landscape
652 \setlength\@tempdima {\paperheight}
653 \setlength\paperheight{\paperwidth}
654 \setlength\paperwidth {\@tempdima}
655 \fi

```

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

```

656 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
657 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
658 \catcode\@tempcnta\active
659 \advance\@tempcnta\@ne
660 \repeat
661 \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。

```

662 \ifbxjs@disguise@js
663 %<book|report>\def\bxjs@js@clsname{jsbook}
664 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
665 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
666 \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。

```

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

```

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

```

668 \ifbxjs@oldfontcommands
669 \AtEndOfClass{\allowoldfontcommands}
670 \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。

```

671 %<slide>\def\n@baseline{13}%
672 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
673 %<!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。

```

674 \ifx\bxjs@magstyle\bxjs@magstyle@mag
675 \jsc@magtrue
676 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
677 \jsc@mag@xrealtrue
678 \fi\fi

```

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

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

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

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

```
679 \ifx\bxjs@param@mag\relax
680   \@tempdima=\bxjs@param@basefontsize
681   \advance\@tempdima.001pt \multiply\@tempdima25
682   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
683   \edef\bxjs@param@mag{\the\@tempcnta}
684 \else
685 % mag 値が直接指定された場合
686   \let\c@bxjs@cnta\@tempcnta
687   \setcounter{bxjs@cnta}{\bxjs@param@mag}
688   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
689 % 有効な mag 値の範囲は 1--32768
690   \edef\bxjs@param@mag{\the\@tempcnta}
691   \advance\@tempcnta100000
692   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
693   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
694   \edef\bxjs@param@basefontsize{\the\@tempdima}
695 \fi
696 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
697 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
698 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
699 \edef\jsc@magscale{\strip@pt\@tempdima}
700 \let\jsBaseFontSize\bxjs@param@basefontsize
701 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
702 %\typeout{\string\jsEngine: \meaning\jsEngine}
703 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
704 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
705 %\typeout{\string\jsc@magscale: \jsc@magscale}
706 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
707 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}
```

[2016-07-08] `\jsc@empt` 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@mp` 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@mp` 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。

```
708 \ifjsc@mag
709   \let\jsc@mp\p@
710   \newdimen\jsc@mmm \jsc@mmm=1mm
711   \catcode`\?=9 % \p@? read as \p@
712 \else
713   \newdimen\jsc@mp \jsc@mp=\jsc@magscale \p@
714   \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
715   \catcode`\?=11 \let\p@?\jsc@mp
716 \fi
717 \chardef\bxjs@qmc=\catcode`\?\relax
```

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

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

```
718 \newdimen\jsZw
719 \jsZw=10\jsc@mp \jsZw=\jsScale\jsZw
720 \ifbxjs@usezw
721   \providecommand*\zw{\jsZw}
722 \fi
```

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

```
723 \ifjsc@mag@xreal
724   \RequirePackage{type1cm}
725   \let\jsc@invscale\bxjs@invscale
```

ムニャムニャムニャ……。

```
726 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
727 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
728 \let\jsc@get@external@font\get@external@font
729 \def\get@external@font{%
730   \jsc@preadjust@extract@font
731   \jsc@get@external@font}
732 \def\jsc@fstrunc#1{%
733   \edef\jsc@tmpa{\strip@pt#1}%
734   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
735 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
736   \if#5*\else
737     \edef\jsc@tmpa{#1%
738       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
739   \fi}
740 \def\jsc@preadjust@extract@font{%
741   \let\jsc@req@size\f@size
742   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
743   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
744   \let\jsc@ref@size\jsc@tmpa
```

```

745 \let\fontsize\jsc@ref@size}
746 \def\execute@size@function#1{%
747 \let\jsc@cref@size\fontsize
748 \let\fontsize\jsc@req@size
749 \csname s@fct@#1\endcsname}
750 \let\jsc@DeclareErrorFont\DeclareErrorFont
751 \def\DeclareErrorFont#1#2#3#4#5{%
752 \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
753 \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
754 \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
755 \def\gen@sfcnt{%
756 \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
757 \empty@sfcnt}
758 \def\genb@sfcnt{%
759 \edef\mandatory@arg{%
760 \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@}%
761 \empty@sfcnt}
762 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
763 \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 764 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 765 %\def\jsc@medskip{\vspace\jsc@medskipamount}
766 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

`\jsc@smallskipamount`

```

\jsc@medskipamount 767 \newskip\jsc@smallskipamount
\jsc@bigskipamount 768 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
769 %\newskip\jsc@medskipamount
770 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
771 %\newskip\jsc@bigskipamount
772 %\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， \LaTeX の $\text{\def\@{\spacefactor\@m}}$ という定義（ \@m は 1000）では I watch TV\@. と書くと V とピリオドのペアカーニングが効かなくなります。そこで，次のような定義に直し， I watch TV.\@ と書くことにします。

```
773 \chardef\bxjs@periodchar=`\.  
774 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcode\bxjs@periodchar}  
775 \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 \LaTeX の内部命令 \@xpt を使っています。この \@xpt の類は次のものがあり， \LaTeX 本体で定義されています。

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

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

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

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

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

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

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

```
776 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
```



```

777 \expandafter\bxjs@tmpa\expandafter{%
778   \set@fontsize{#1}{#2}{#3}%
779 % 末尾にコードを追加
780   \expandafter\def\expandafter\size@update\expandafter{%
781     \size@update
782     \jsFontSizeChanged}%
783 }

```

`\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。

```

784 \newcommand*\jsFontSizeChanged{%
785   \jsZw=\f@size\p@
786   \jsZw=\jsScale \jsZw
787   \ifdim\parindent>\z@
788     \if@english \parindent=1em
789     \else       \parindent=1\jsZw
790   \fi
791   \fi\relax
792   \jsResetDimen}

```

`\jsResetDimen` ユーザ定義用のフック。

```

793 \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。

```

794 \ifjsc@mag
795   \let\jsc@setfontsize\@setfontsize
796 \else
797   \def\jsc@setfontsize#1#2#3{%
798     \@setfontsize#1{#2\jsc@mpt}{#3\jsc@mpt}}
799 \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。

```

800 \emergencystretch 3\jsZw

```

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

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

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

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

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

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

```
801 \newif\ifnarrowbaselines
802 \if@english
803   \narrowbaselinestrue
804 \fi
805 \def\narrowbaselines{%
806   \narrowbaselinestrue
807   \skip0=\abovedisplayskip
808   \skip2=\abovedisplayshortskip
809   \skip4=\belowdisplayskip
810   \skip6=\belowdisplayshortskip
811 % 一時的に警告を無効化する
812   \let\bxjs@ltx@nomath\@nomath
813   \let\@nomath\@gobble
814   \@currsize\selectfont
815   \let\@nomath\bxjs@ltx@nomath
816   \abovedisplayskip=\skip0
817   \abovedisplayshortskip=\skip2
818   \belowdisplayskip=\skip4
819   \belowdisplayshortskip=\skip6\relax}
820 \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。

```
821 \def\bxjs@ifnarrowbaselines{%
822   \ifnarrowbaselines\expandafter\@firstoftwo
823   \else \expandafter\@secondoftwo
824   \fi
825 }
```

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

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

公称 10 ポイントの和文フォントが約 9.25 ポイント（アスキーのものの 0.961 倍）である

こともあり、行送りがかなりゆったりとしたと思います。実際、 $16/9.25 \approx 1.73$ であり、和文の推奨値の一つ「二分四分」(1.75) に近づきました。

microtype 対策のため if 文を避ける。

```
826 \renewcommand{\normalsize}{%
827   \bxjs@if@narrowbaselines{%
828     \jsc@setfontsize\normalsize\@xpt\@xipt
829   }{%else
830     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
831   }%
```

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。

```
832 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
833 \abovedisplayshortskip \z@ \@plus3\p@?
834 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
835 \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。

```
836 \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。

```
837 \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。

```
\Chs 838 \ifx\Cht\@undefined \newdimen\Cht \fi
839 \ifx\Cdp\@undefined \newdimen\Cdp \fi
840 \ifx\Cwd\@undefined \newdimen\Cwd \fi
841 \ifx\Cvs\@undefined \newdimen\Cvs \fi
842 \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。

```
843 \setlength\Cht{0.88\jsZw}
844 \setlength\Cdp{0.12\jsZw}
845 \setlength\Cwd{1\jsZw}
846 \setlength\Cvs{\baselineskip}
847 \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。

```

848 \newcommand{\small}{%
849   \bxjs@if@narrowbaselines{%
850     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
851     %<kiyou>    \jsc@setfontsize\small{8.8888}{11}%
852   }{%else
853     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
854     %<kiyou>    \jsc@setfontsize\small{8.8888}{13.2418}%
855   }%
856   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
857   \abovedisplayshortskip \z@ \@plus3\p@?
858   \belowdisplayskip \abovedisplayskip
859   \belowdisplayshortskip \belowdisplayskip
860   \def\@listi{\leftmargin\leftmargini
861             \topsep \z@
862             \parsep \z@
863             \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。

```

864 \newcommand{\footnotesize}{%
865   \bxjs@if@narrowbaselines{%
866     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{9.5}%
867     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{11}%
868   }{%else
869     %<!kiyou>   \jsc@setfontsize\footnotesize\@viipt{11}%
870     %<kiyou>    \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
871   }%
872   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
873   \abovedisplayshortskip \z@ \@plus2\p@?
874   \belowdisplayskip \abovedisplayskip
875   \belowdisplayshortskip \belowdisplayskip
876   \def\@listi{\leftmargin\leftmargini
877             \topsep \z@
878             \parsep \z@
879             \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`
`\huge`
`\Huge`

[2004-11-03] \HUGE を追加。

```
880 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
881 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
882 \if@twocolumn
883 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
884 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
885 \else
886 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
887 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
888 \fi
889 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
890 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
891 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
892 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
893 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
894 \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。

```
895 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}
```

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

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

```
896 % \newcommand{\headfont}{\bfseries}
897 \newcommand{\headfont}{\sffamily}
898 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}
```

5 レイアウト

■二段組

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

```
899 %<!kiyou>\setlength\columnsep{2\Cwd}
900 %<kiyou>\setlength\columnsep{28truebp}
901 \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`

```
902 \setlength\lineskip{1\p@?}
903 \setlength\normallineskip{1\p@?}
904 \setlength\lineskiplimit{1\p@?}
905 \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。

```
906 \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。

```
907 \setlength\parskip{0\p@}
908 \if@slide
909   \setlength\parindent{0\p@}
910 \else
911   \setlength\parindent{1\Cwd}
912 \fi
```

`\@lowpenalty` `\nopagebreak`, `\nolinebreak` は引数に応じて次のペナルティ値のうちどれかを選ぶよう

`\@medpenalty` になっています。ここはオリジナル通りです。

```
913 \@lowpenalty 51
914 \@medpenalty 151
915 \@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。

```
916 % \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。

```
917 % \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。

```
918 \ifjsc@mag
919 \mag=\bxjs@param@mag
920 \fi
921 \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。

```
922 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
923 \def\bxjs@read@b{%
924   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
925   \else \expandafter\bxjs@read@d \fi}
926 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
927 \def\bxjs@read@d#1\@nil{}
928 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
929 \edef\bxjs@layout@paper{%
930   \ifjsc@mag truedimen,\fi
931   \if@landscape landscape,\fi
932   \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。

```
933 %<*article|report>
934 \def\bxjs@layout{%
935   headheight=\topskip,footskip=0.03367\paperheight,%
936   headsep=\footskip-\topskip,includeheadfoot,%
937   hscale=0.76,hmarginratio=1:1,%
938   vscale=0.83,vmarginratio=1:1,%
939 }
940 %</article|report>
941 %<*book>
942 \ifbxjs@layout@buggyhmargin %---
943 % アレ
944 \def\bxjs@layout{%
945   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
946   hmargin=36\jsc@mmm,hmarginratio=1:1,%
947   vscale=0.83,vmarginratio=1:1,%
948 }
949 \else %---
950 % 非アレ
951 \def\bxjs@layout{%
952   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
953   hmargin=18\jsc@mmm,%
954   vscale=0.83,vmarginratio=1:1,%
```

```

955 }
956 \fi %---
957 %</book>
958 %<*slide>
959 \def\bxjs@layout{%
960   noheadfoot,%
961   hscale=0.9,hmarginratio=1:1,%
962   vscale=0.944,vmarginratio=1:1,%
963 }
964 %</slide>

```

\fullwidth 〔寸法レジスタ〕 ヘッダ・フッタ領域の横幅。

```

965 \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。

```

966 %<*book>
967 \newcommand\jsTextWidthLimit{40}
968 \ifx\bxjs@textwidth@limit\@undefined\else
969   \let\c@bxjs@cmta\@tempcmta
970   \setcounter{bxjs@cmta}{\bxjs@textwidth@limit}
971   \long\edef\jsTextWidthLimit{\the\@tempcmta}
972 \fi
973 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

974 \def\bxjs@postproc@layout{%
975 % ドライバ再設定
976   \ifx\bxjs@geometry@driver\relax\else
977     \let\Gm@driver\bxjs@geometry@driver
978   \fi
979 % textwidth 調整
980   \@tempdima=\textwidth
981   \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
982   \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
983   \advance\@tempdima-\textwidth
984   \advance\oddsidemargin 0.5\@tempdima
985   \advance\evensidemargin 0.5\@tempdima
986   \fullwidth=\textwidth
987 %<*book>
988   \ifdim\textwidth>\jsTextWidthLimit\Cwd
989     \textwidth=\jsTextWidthLimit\Cwd
990     \addtolength\evensidemargin{\fullwidth-\textwidth}
991   \fi
992 %</book>
993 % textheight 調整
994   \@tempdima=\textheight
995   \advance\textheight-\topskip
996   \divide\textheight\baselineskip \multiply\textheight\baselineskip
997   \advance\textheight\topskip

```



```

998 \advance\@tempdima-\textheight
999 \advance\topmargin0.5\@tempdima
1000 % headheight 調整
1001 \@tempdima=\topskip
1002 \advance\headheight\@tempdima
1003 \advance\topmargin-\@tempdima
1004 % marginpar 設定
1005 \setlength\marginparsep{\columnsep}
1006 \setlength\marginparpush{\baselineskip}
1007 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1008     -\textwidth-10\jsc@mmm-\marginparsep}
1009 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1010 % 連動する変数
1011 \maxdepth=.5\topskip
1012 \stockwidth=\paperwidth
1013 \stockheight=\paperheight
1014 }

```

`\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。

```

1015 \edef\jsGeometryOptions{%
1016     \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージ読み込み

geometry オプションの値に応じて分岐する。

まずは `geometry=class` の場合。

```

1017 \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。

```

1018 \@onlypreamble\bxjs@revert
1019 \let\bxjs@revert\empty
1020 \edef\bxjs@tmpa{\the\ht\strutbox}
1021 \ht\strutbox=10\p@?
1022 \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。

```

1023 \ifbxjs@papersize
1024     \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1025         \PassOptionsToPackage{dvipdfm}{geometry}
1026     \fi

```

```

1027 \let\bxPapersizeSpecialDone=t
1028 \else
1029 \PassOptionsToPackage{driver=none}{geometry}
1030 \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。

```

1031 \edef\bxjs@nxt{%
1032 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1033 \AtBeginDocument{\bxjs@pre@geometry@hook}
1034 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1035 \bxjs@nxt \bxjs@revert
1036 \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。

```

1037 \let\bxjs@geometry@driver\Gm@driver
1038 \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。

```

1039 \@onlypreamble\bxjs@pre@geometry@hook
1040 \def\bxjs@pre@geometry@hook{%
1041 \ifpackageloaded{geometry}{%
1042 \ifpackagelater{geometry}{2010/02/12}{%}%else
1043 \PackageWarningNoLine\bxjs@clsname
1044 {The 'geometry' package installed\MessageBreak
1045 is too old (< v5.0)}%
1046 \if x\jsEngine \ifnum\mag=\@m\else
1047 \def\bxjs@Gm@driver{pdftex}
1048 \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1049 \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1050 \fi\fi
1051 }%
1052 \ifjsWithpTeXng
1053 \ifx\Gm@driver\@empty
1054 \def\Gm@driver{pdftex}%
1055 \fi
1056 \fi
1057 }{}}

```

`\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。

```

1058 \def\setpagelayout{\@ifstar
1059   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1060 \def\bxjs@modify@layout#1{%
1061   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1062   \bxjs@nxt\bxjs@postproc@layout}
1063 \def\bxjs@reset@layout#1{%
1064   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1065   \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。

```

1066 \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。

```

1067 \AtBeginDocument{\bxjs@check@page@layout}
1068 \@onlypreamble\bxjs@check@page@layout
1069 \def\bxjs@check@page@layout{%
1070   \ifdim\textwidth=.5\maxdimen
1071     \ClassError\bxjs@clsname
1072       {Page layout is not properly set}%
1073     {\@ehd}
1074   \fi}
1075 \def\jsUseMinimalPageLayout{%
1076   \setlength{\textwidth}{6.5in}%
1077   \setlength{\textheight}{8in}}

```

\setpagelayout はとりあえず無効にしておく。

```

1078 \let\bxjs@geometry@driver\relax
1079 \def\setpagelayout{\@ifstar
1080   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1081 \def\bxjs@pagelayout@a#1{%
1082   \ClassError\bxjs@clsname
1083     {Command '\string\setpagelayout' is not supported,\MessageBreak
1084     because 'geometry' value is not 'class'}\@eha}

```

geometry=user の場合の処理はここで終わり。

```

1085 \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。

`\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。

```
1086 \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。

```
1087 \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。

```
1088 \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。

```
1089 \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。

```
1090 \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。

```
1091 \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。

```
1092 \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。

```
1093 \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。

```
1094 \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。

```
1095 \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。

```
1096 \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。

```
1097 \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。

```
1098 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1099 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1100 \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 1101 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1102 \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 1103 \setlength\@fptop{0\p@? \@plus 1fil}
1104 \setlength\@fpsep{8\p@? \@plus 2fil}
1105 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
\@dblfpsep 1106 \setlength\@dblfpsep{0\p@? \@plus 1fil}
\@dblfpbot 1107 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1108 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 ページスタイル

ページスタイルとして, $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$ (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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$ 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` で変更する場合がこれにあたります。しかし、同一ページに複数の `\markboth` があると、おかしい結果になることがあります。

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

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

```
1109 % \def\ps@empty{%
1110 %   \let\mkboth\gobbletwo
1111 %   \let\@oddhead\@empty
1112 %   \let\@oddfoot\@empty
1113 %   \let\@evenhead\@empty
1114 %   \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。

```
1115 \def\ps@plainfoot{%
1116   \let\mkboth\gobbletwo
1117   \let\@oddhead\@empty
1118   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1119   \let\@evenhead\@empty
1120   \let\@evenfoot\@oddfoot}
1121 \def\ps@plainhead{%
1122   \let\mkboth\gobbletwo
1123   \let\@oddfoot\@empty
1124   \let\@evenfoot\@empty
1125   \def\@evenhead{%
1126     \if@mparswitch \hss \fi
1127     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1128     \if@mparswitch\else \hss \fi}%
1129   \def\@oddhead{%
1130     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1131 %<book>\let\ps@plain\ps@plainhead
```

```
1132 %<!book>\let\ps@plain\ps@plainfoot
```

`\ps@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。

```
1133 %<*article|slide>
1134 \if@twoside
1135   \def\ps@headings{%
1136     \let\@oddfoot\@empty
1137     \let\@evenfoot\@empty
1138     \def\@evenhead{\if@mparswitch \hss \fi
1139       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1140     \if@mparswitch\else \hss \fi}%
1141     \def\@oddhead{%
1142       \underline{%
1143         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1144     \let\@mkboth\markboth
1145     \def\sectionmark##1{\markboth{%
1146       \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1147       ##1}{}}%
1148     \def\subsectionmark##1{\markright{%
1149       \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
1150       ##1}}%
1151   }
1152 \else % if not twoside
1153   \def\ps@headings{%
1154     \let\@oddfoot\@empty
1155     \def\@oddhead{%
1156       \underline{%
1157         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1158     \let\@mkboth\markboth
1159     \def\sectionmark##1{\markright{%
1160       \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1161       ##1}}%
1162   \fi
1163 %</article|slide>
```

次は `book` 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）。

```
1164 %<*book|report>
1165 \newif\if@omit@number
1166 \def\ps@headings{%
1167   \let\@oddfoot\@empty
1168   \let\@evenfoot\@empty
1169   \def\@evenhead{%
1170     \if@mparswitch \hss \fi
1171     \underline{\hbox to \fullwidth{\autoxspacing
1172       \textbf{\thepage}\hfil\leftmark}}}%
1173   \if@mparswitch\else \hss \fi}%

```

```

1174 \def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
1175     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1176 \let\@mkboth\markboth
1177 \def\chaptermark##1{\markboth{%
1178     \ifnum \c@secnumdepth >\m@ne
1179         \if@mainmatter
1180             \if@omit@number\else
1181                 \@chapapp\thechapter\@chappos\hskip1\jsZw
1182             \fi
1183         \fi
1184     \fi
1185     ##1}}}%
1186 \def\sectionmark##1{\markright{%
1187     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1188     ##1}}}%
1189 %</book|report>

```

最後は学会誌の場合です。

```

1190 %<*jspf>
1191 \def\ps@headings{%
1192     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1193     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1194     \def\@oddhead{\normalfont\hfil \@title \hfil}
1195     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1196 %</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。

```

1197 \def\ps@myheadings{%
1198     \let\@oddfoot\@empty\let\@evenfoot\@empty
1199     \def\@evenhead{%
1200         \if@mparswitch \hss \fi%
1201         \hbox to \fullwidth{\thepage\hfil\leftmark}%
1202         \if@mparswitch\else \hss \fi}%
1203     \def\@oddhead{%
1204         \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1205     \let\@mkboth\@gobbletwo
1206 %<book|report> \let\chaptermark\@gobble
1207 \let\sectionmark\@gobble
1208 %<!book&!report> \let\subsectionmark\@gobble
1209 }

```


7 文書のマークアップ

7.1 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 1210 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1211 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1212 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1213 % \date{\today}
```

`\subtitle` 副題を設定する。

`\jsSubtitle` ※プレアンプルにおいて `\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。

```
1214 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1215 %\let\bxjs@subtitle\undefined

      \title にフックを入れる。

1216 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1217 \AtBeginDocument{\bxjs@decl@subtitle}
1218 \def\bxjs@decl@subtitle{%
1219   \global\let\bxjs@decl@subtitle\relax
1220   \ifx\subtitle\undefined
1221     \global\let\subtitle\jsSubtitle
1222   \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。

```
1223 \def\bxjs@annihilate@subtitle{%
1224   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1225   \global\let\jsSubtitle\relax}
```

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

```
\eauthor 1226 %<*jspf>
\keywords 1227 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
          1228 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
          1229 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
          1230 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
          1231 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
          1232 %</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。

```

1233 \def\plainifnotempty{%
1234   \ifx \@oddhead \@empty
1235     \ifx \@oddfoot \@empty
1236       \else
1237         \thispagestyle{plainfoot}%
1238       \fi
1239     \else
1240       \thispagestyle{plainhead}%
1241     \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。

```

1242 %<article|book|report|slide>
1243 \if@titlepage
1244   \newcommand{\maketitle}{%
1245     \begin{titlepage}%
1246       \let\footnotesize\small
1247       \let\footnoterule\relax
1248       \let\footnote\thanks
1249       \null\vfil
1250       \if@slide
1251         {\footnotesize \@date}%
1252         \begin{center}
1253           \mbox{} \[\!1\jsZw]
1254           \large
1255           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1256           \jsc@smallskip
1257           \@title
1258           \ifx\bxjs@subtitle\@undefined\else
1259             \par\vskip\z@
1260             {\small \bxjs@subtitle\par}
1261           \fi
1262           \jsc@smallskip
1263           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1264           \vfill
1265           {\small \@author}%
1266         \end{center}
1267       \else
1268         \vskip 60\p@?

```

```

1269 \begin{center}%
1270 {\LARGE \@title \par}%
1271 \ifx\bxjs@subtitle\@undefined\else
1272 \vskip5\p@?
1273 {\normalsize \bxjs@subtitle\par}
1274 \fi
1275 \vskip 3em%
1276 {\large
1277 \lineskip .75em
1278 \begin{tabular}[t]{c}%
1279 \author
1280 \end{tabular}\par}%
1281 \vskip 1.5em
1282 {\large \@date \par}%
1283 \end{center}%
1284 \fi
1285 \par
1286 \@thanks\vfil\null
1287 \end{titlepage}%
1288 \setcounter{footnote}{0}%
1289 \global\let\thanks\relax
1290 \global\let\maketitle\relax
1291 \global\let\@thanks\@empty
1292 \global\let\@author\@empty
1293 \global\let\@date\@empty
1294 \global\let\@title\@empty
1295 \global\let\title\relax
1296 \global\let\author\relax
1297 \global\let\date\relax
1298 \global\let\and\relax
1299 \bxjs@annihilate@subtitle
1300 }%
1301 \else
1302 \newcommand{\maketitle}{\par
1303 \begin{group
1304 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1305 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1306 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1307 \parindent 1\jsZw\noindent
1308 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1309 \if@twocolumn
1310 \ifnum \col@number=\@ne
1311 \maketitle
1312 \else
1313 \twocolumn[\maketitle]%
1314 \fi
1315 \else
1316 \newpage
1317 \global\@topnum\z@ % Prevents figures from going at top of page.

```

```

1318     \@maketitle
1319     \fi
1320     \plainifnotempty
1321     \@thanks
1322 \endgroup
1323 \setcounter{footnote}{0}%
1324 \global\let\thanks\relax
1325 \global\let\maketitle\relax
1326 \global\let\@thanks\@empty
1327 \global\let\@author\@empty
1328 \global\let\@date\@empty
1329 \global\let\@title\@empty
1330 \global\let\title\relax
1331 \global\let\author\relax
1332 \global\let\date\relax
1333 \global\let\and\relax
1334 \bxjs@annihilate@subtitle
1335 }

```

`\@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。

```

1336 \def\@maketitle{%
1337   \newpage\null
1338   \vskip 2em
1339   \begin{center}%
1340     \let\footnote\thanks
1341     {\LARGE \@title \par}%
1342     \ifx\bxjs@subtitle\@undefined\else
1343       \vskip3\p@?
1344       {\normalsize \bxjs@subtitle\par}
1345     \fi
1346     \vskip 1.5em
1347     {\large
1348       \lineskip .5em
1349       \begin{tabular}[t]{c}%
1350         \@author
1351       \end{tabular}\par}%
1352     \vskip 1em
1353     {\large \@date}%
1354   \end{center}%
1355   \par\vskip 1.5em
1356 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1357 }
1358 \fi
1359 %</article|book|report|slide>
1360 %<*jspf>
1361 \newcommand{\maketitle}{\par
1362   \begingroup
1363     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1364     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%

```

```

1365 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1366 \parindent 1\jsZw\noindent
1367 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1368 \twocolumn[\@maketitle]%
1369 \plainifnotempty
1370 \@thanks
1371 \endgroup
1372 \setcounter{footnote}{0}%
1373 \global\let\thanks\relax
1374 \global\let\maketitle\relax
1375 \global\let\@thanks\@empty
1376 \global\let\@author\@empty
1377 \global\let\@date\@empty
1378 % \global\let\@title\@empty % \@title は柱に使う
1379 \global\let\title\relax
1380 \global\let\author\relax
1381 \global\let\date\relax
1382 \global\let\and\relax
1383 \ifx\authors@mail\@undefined\else{%
1384 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1385 \footnotetext[0]{\itshape\authors@mail}%
1386 }\fi
1387 \global\let\authors@mail\@undefined}
1388 \def\@maketitle{%
1389 \newpage\null
1390 \vskip 6em % used to be 2em
1391 \begin{center}
1392 \let\footnote\thanks
1393 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1394 \lineskip .5em
1395 \ifx\@author\@undefined\else
1396 \vskip 1em
1397 \begin{tabular}[t]{c}%
1398 \@author
1399 \end{tabular}\par
1400 \fi
1401 \ifx\@etitle\@undefined\else
1402 \vskip 1em
1403 {\large \@etitle \par}%
1404 \fi
1405 \ifx\@eauthor\@undefined\else
1406 \vskip 1em
1407 \begin{tabular}[t]{c}%
1408 \@eauthor
1409 \end{tabular}\par
1410 \fi
1411 \vskip 1em
1412 \@date
1413 \end{center}

```

```

1414 \vskip 1.5em
1415 \centerline{\box\@abstractbox}
1416 \ifx\@keywords\@undefined\else
1417     \vskip 1.5em
1418     \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1419 \fi
1420 \vskip 1.5em}
1421 %</jspf>

```

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

```

1422 \def\@startsection#1#2#3#4#5#6{%
1423     \if@noskipsec \leavevmode \fi
1424     \par
1425 % 見出し上の空きを \@tempskipa にセットする
1426     \@tempskipa #4\relax
1427 % \@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
1428     \if@english \@afterindentfalse \else \@afterindenttrue \fi
1429 % 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
1430     \ifdim \@tempskipa <\z@

```

```

1431 \@tempskipa -\@tempskipa \@afterindentfalse
1432 \fi
1433 \if@nobreak
1434 % \everypar{\everyparhook}% これは間違い
1435 \everypar{}%
1436 \else
1437 \addpenalty\@secpenalty
1438 % 次の行は削除
1439 % \addvspace\@tempskipa
1440 % 次の \noindent まで追加
1441 \ifdim \@tempskipa >\z@
1442 \if@slide\else
1443 \null
1444 \vspace*{-\baselineskip}%
1445 \fi
1446 \vskip\@tempskipa
1447 \fi
1448 \fi
1449 \noindent
1450 % 追加終わり
1451 \@ifstar
1452 {\@ssect{#3}{#4}{#5}{#6}}%
1453 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

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

1454 \def\@sect#1#2#3#4#5#6[#7]#8{%
1455 \ifnum #2>\c@secnumdepth
1456 \let\@svsec\@empty
1457 \else
1458 \refstepcounter{#1}%
1459 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1460 \fi
1461 % 見出し後の空きを \@tempskipa にセット
1462 \@tempskipa #5\relax
1463 % 条件判断の順序を入れ替えました
1464 \ifdim \@tempskipa<\z@
1465 \def\@svsechd{%
1466 #6{\hskip #3\relax
1467 \@svsec #8}%
1468 \csname #1mark\endcsname{#7}}%
1469 \addcontentsline{toc}{#1}{%
1470 \ifnum #2>\c@secnumdepth \else
1471 \protect\numberline{\csname the#1\endcsname}%
1472 \fi
1473 #7}}% 目次にフルネームを載せるなら #8
1474 \else
1475 \begingroup
1476 \interlinepenalty \@M % 下から移動

```

```

1477      #6{%
1478      \@hangfrom{\hskip #3\relax\@svsec}%
1479 %      \interlinepenalty \@M % 上に移動
1480      #8\@par}%
1481      \endgroup
1482      \csname #1mark\endcsname{#7}%
1483      \addcontentsline{toc}{#1}{%
1484      \ifnum #2>\c@secnumdepth \else
1485      \protect\numberline{\csname the#1\endcsname}%
1486      \fi
1487      #7}% 目次にフルネームを載せるならここは #8
1488      \fi
1489      \@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。

```

1490 \def\@xsect#1{%
1491 % 見出しの後ろの空きを \@tempskipa にセット
1492 \@tempskipa #1\relax
1493 % 条件判断の順序を変えました
1494 \ifdim \@tempskipa<\z@
1495 \@nbreakfalse
1496 \global\@noskipsectrue
1497 \everypar{%
1498 \if@noskipsec
1499 \global\@noskipsecfalse
1500 {\setbox\z@\lastbox}%
1501 \clubpenalty\@M
1502 \begingroup \@svsechd \endgroup
1503 \unskip
1504 \@tempskipa #1\relax
1505 \hskip -\@tempskipa
1506 \bxjs<ltj@inhibitglue
1507 \else
1508 \clubpenalty \@clubpenalty
1509 \everypar{\everyparhook}%
1510 \fi\everyparhook}%
1511 \else
1512 \par \nbreak
1513 \vskip \@tempskipa
1514 \@afterheading
1515 \fi
1516 \if@slide
1517 {\vskip\if@twocolumn-5\jsc@mpt\else-6\jsc@mpt\fi
1518 \maybeblue\hrule height0\jsc@mpt depth1\jsc@mpt
1519 \vskip\if@twocolumn 4\jsc@mpt\else 7\jsc@mpt\fi\relax}%

```



```

1520 \fi
1521 \par % 2000-12-18
1522 \ignorespaces}
1523 \def\@ssect#1#2#3#4#5{%
1524 \@tempskipa #3\relax
1525 \ifdim \@tempskipa<\z@
1526 \def\@svsechd{#4{\hskip #1\relax #5}}%
1527 \else
1528 \begingroup
1529 #4{%
1530 \@hangfrom{\hskip #1}%
1531 \interlinepenalty \@M #5\@@par}%
1532 \endgroup
1533 \fi
1534 \@xsect{#3}}

```

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

```

1535 \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 6 節参照）。`\chaptermark` 以外は L^AT_EX 本体で定義済みです。

```

\subsectionmark 1536 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1537 % \newcommand*\sectionmark[1]{}
1538 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1539 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1540 % \newcommand*\paragraphmark[1]{}
1541 % \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決めるカウンタです。

```

1542 %<!book&!report>\setcounter{secnumdepth}{3}
1543 %<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 1544 \newcounter{part}
1545 %<book|report>\newcounter{chapter}
\c@subsubsection 1546 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1547 %<!book&!report>\newcounter{section}
\c@subparagraph 1548 \newcounter{subsection}[section]
1549 \newcounter{subsubsection}[subsection]
1550 \newcounter{paragraph}[subsubsection]
1551 \newcounter{subparagraph}[paragraph]

```

<code>\thepart</code>	カウンタの値を出力する命令 <code>\the</code> 何々 を定義します。
<code>\thechapter</code>	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。

```

1552 \renewcommand{\thepart}{\@Roman\c@part}
1553 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1554 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1555 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1556 %<*book|report>
1557 \renewcommand{\thechapter}{\@arabic\c@chapter}
1558 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1559 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1560 %</book|report>
1561 \renewcommand{\thesubsubsection}{\%
1562   \thesubsection.\@arabic\c@subsubsection}
1563 \renewcommand{\theparagraph}{\%
1564   \thesubsubsection.\@arabic\c@paragraph}
1565 \renewcommand{\thesubparagraph}{\%
1566   \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。

```

1567 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1568 %<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。

```

1569 %<*book|report>
1570 \newcommand\frontmatter{%
1571   \if@openright
1572     \cleardoublepage
1573   \else
1574     \clearpage
1575   \fi
1576   \@mainmatterfalse
1577   \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。

```
1578 \newcommand\mainmatter{%
1579 % \if@openright
1580     \cleardoublepage
1581 % \else
1582 %     \clearpage
1583 % \fi
1584 \@mainmattertrue
1585 \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。

```
1586 \newcommand\backmatter{%
1587     \if@openright
1588         \cleardoublepage
1589     \else
1590         \clearpage
1591     \fi
1592 \@mainmatterfalse}
1593 %</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 クラス以外です。

```
1594 %<*&!book&!report>
1595 \newcommand\part{%
1596     \if@noskipsec \leavevmode \fi
1597     \par
1598     \addvspace{4ex}%
1599     \if@english \@afterindentfalse \else \@afterindenttrue \fi
1600     \secdef\@part\@spart}
1601 %</!book&!report>
```

book スタイルの場合は，少し複雑です。

```
1602 %<*&book|report>
1603 \newcommand\part{%
```

```

1604 \if@openright
1605     \cleardoublepage
1606 \else
1607     \clearpage
1608 \fi
1609 \thispagestyle{empty}% 欧文用標準スタイルでは plain
1610 \if@twocolumn
1611     \onecolumn
1612     \@restonecoltrue
1613 \else
1614     \@restonecolfalse
1615 \fi
1616 \null\vfil
1617 \secdef\@part\@spart}
1618 %</book|report>

```

\@part 部の見出しを出力します。 \bfseries を \headfont に変えました。

book 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。

```

1619 %<*&!book&!report>
1620 \def\@part[#1]#2{%
1621     \ifnum \c@secnumdepth >\m@ne
1622         \refstepcounter{part}%
1623         \addcontentsline{toc}{part}{%
1624             \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1625     \else
1626         \addcontentsline{toc}{part}{#1}%
1627     \fi
1628     \markboth{}{}%
1629     {\parindent\z@
1630         \raggedright
1631         \interlinepenalty \@M
1632         \normalfont
1633         \ifnum \c@secnumdepth >\m@ne
1634             \Large\headfont\prepartname\thepart\postpartname
1635             \par\nobreak
1636         \fi
1637         \huge \headfont #2%
1638         \markboth{}{}\par}%
1639     \nobreak
1640     \vskip 3ex
1641     \@afterheading}
1642 %</!book&!report>

```

book クラスでは secnumdepth が -2 より大きいとき部番号を付けます。

```

1643 %<*&book|report>
1644 \def\@part[#1]#2{%
1645     \ifnum \c@secnumdepth >-2\relax
1646         \refstepcounter{part}%
1647         \addcontentsline{toc}{part}{%

```

```

1648     \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1649 \else
1650     \addcontentsline{toc}{part}{#1}%
1651 \fi
1652 \markboth{}{}%
1653 {\centering
1654     \interlinepenalty \@M
1655     \normalfont
1656     \ifnum \c@secnumdepth >-2\relax
1657         \huge\headfont \prepartname\thepart\postpartname
1658         \par\vskip20\p@?
1659     \fi
1660     \Huge \headfont #2\par}%
1661 \@endpart}
1662 %</book|report>

```

\@spart 番号を付けない部です。

```

1663 %<*&!book&!report>
1664 \def\@spart#1{%
1665     \parindent \z@ \raggedright
1666     \interlinepenalty \@M
1667     \normalfont
1668     \huge \headfont #1\par}%
1669 \nobreak
1670 \vskip 3ex
1671 \@afterheading}
1672 %</!book&!report>
1673 %<*&book|report>
1674 \def\@spart#1{%
1675     \centering
1676     \interlinepenalty \@M
1677     \normalfont
1678     \Huge \headfont #1\par}%
1679 \@endpart}
1680 %</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。

```

1681 %<*&book|report>
1682 \def\@endpart{\vfil\newpage
1683     \if@twoside
1684         \if@openright %% added (2016/12/13)
1685             \null
1686             \thispagestyle{empty}%
1687         \newpage

```

```

1688 \fi %% added (2016/12/13)
1689 \fi
1690 \if@restonecol
1691 \twocolumn
1692 \fi}
1693 %</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。

```

1694 %<*book|report>
1695 \newcommand{\chapter}{%
1696 \if@openright\cleardoublepage\else\clearpage\fi
1697 \plainifnotempty % 元: \thispagestyle{plain}
1698 \global\@topnum\z@
1699 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1700 \secdef
1701 {\@omit@numberfalse\@chapter}%
1702 {\@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。

```

1703 \def\@chapter[#1]#2{%
1704 \ifnum \c@secnumdepth >\m@ne
1705 \if@mainmatter
1706 \refstepcounter{chapter}%
1707 \typeout{\@chapapp\thechapter\@chappos}%
1708 \addcontentsline{toc}{chapter}%
1709 {\protect\numberline
1710 % {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
1711 {\@chapapp\thechapter\@chappos}%
1712 #1}%
1713 \else\addcontentsline{toc}{chapter}{#1}\fi
1714 \else
1715 \addcontentsline{toc}{chapter}{#1}%
1716 \fi
1717 \chaptermark{#1}%
1718 \addtocontents{lof}{\protect\addvspace{10\jsc@mp}}%
1719 \addtocontents{lot}{\protect\addvspace{10\jsc@mp}}%
1720 \if@twocolumn
1721 \@topnewpage[\@makechapterhead{#2}]%
1722 \else
1723 \@makechapterhead{#2}%
1724 \@afterheading
1725 \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。

```

1726 \def\@makechapterhead#1{%
1727   \vspace*{2\Cvs}% 欧文は 50pt
1728   {\parindent \z@ \raggedright \normalfont
1729     \ifnum \c@secnumdepth >\m@ne
1730       \if@mainmatter
1731         \huge\headfont \@chapapp\thechapter\@chappos
1732         \par\nobreak
1733         \vskip \Cvs % 欧文は 20pt
1734       \fi
1735     \fi
1736     \interlinepenalty\@M
1737     \Huge \headfont #1\par\nobreak
1738     \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。

```

1739 \def\@schapter#1{%
1740   \chaptermark{#1}%
1741   \if@twocolumn
1742     \@topnewpage[\@makeschapterhead{#1}]%
1743   \else
1744     \@makeschapterhead{#1}\@afterheading
1745   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

1746 \def\@makeschapterhead#1{%
1747   \vspace*{2\Cvs}% 欧文は 50pt
1748   {\parindent \z@ \raggedright
1749     \normalfont
1750     \interlinepenalty\@M
1751     \Huge \headfont #1\par\nobreak
1752     \vskip 3\Cvs}} % 欧文は 40pt
1753 %</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。

```

1754 \if@twocolumn
1755   \newcommand{\section}{%
1756     <jspf>\ifx\maketitle\relax\else\maketitle\fi
1757     \@startsection{section}{1}{\z@}%
1758     <kiyou> {0.6\Cvs}{0.4\Cvs}%
1759     <kiyou> {\Cvs}{0.5\Cvs}%
1760     % {\normalfont\large\headfont\@secapp}}
1761     {\normalfont\large\headfont\raggedright}}
1762 \else
1763   \newcommand{\section}{%

```

```

1764 \if@slide\clearpage\fi
1765 \@startsection{section}{1}{\z@}%
1766 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1767 {.5\Cvs \@plus.3\Cdp}% 後アキ
1768 % {\normalfont\Large\headfont\@secapp}%
1769 {\normalfont\Large\headfont\raggedright}}
1770 \fi

```

\subsection 同上です。

```

1771 \if@twocolumn
1772 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1773 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1774 {\normalfont\normalsize\headfont}}
1775 \else
1776 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1777 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1778 {.5\Cvs \@plus.3\Cdp}% 後アキ
1779 {\normalfont\large\headfont}}
1780 \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)。

```

1781 \if@twocolumn
1782 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1783 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1784 {\normalfont\normalsize\headfont}}
1785 \else
1786 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1787 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1788 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
1789 {\normalfont\normalsize\headfont}}
1790 \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 \jchar と同義になるが、それ以外は何もしない。

```

1791 \newcommand\jsParagraphMark{\jsJaChar{■}}
1792 \ifx\bxjs@paragraph@mark\undefined\else
1793   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
1794 \fi
1795 \let\jsJaChar\empty
1796 \if@twocolumn
1797   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1798     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1799   %<jspf> {\normalfont\normalsize\headfont}}
1800   %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1801 \else
1802   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1803     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1804     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1805   %<jspf> {\normalfont\normalsize\headfont}}
1806   %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1807 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

1808 \if@twocolumn
1809   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1810     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1811     {\normalfont\normalsize\headfont}}
1812 \else
1813   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1814     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1815     {\normalfont\normalsize\headfont}}
1816 \fi

```

7.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```

1817 \if@slide
1818   \setlength\leftmargini{1\jsZw}
1819 \else
1820   \if@twocolumn
1821     \setlength\leftmargini{2\jsZw}
1822   \else
1823     \setlength\leftmargini{3\jsZw}
1824   \fi

```

1825 \fi

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

```
\leftmarginiv 1826 \if@slide
\leftmarginv 1827 \setlength\leftmarginii {1\jsZw}
1828 \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 1829 \setlength\leftmarginiv {1\jsZw}
1830 \setlength\leftmarginv {1\jsZw}
1831 \setlength\leftmarginvi {1\jsZw}
1832 \else
1833 \setlength\leftmarginii {2\jsZw}
1834 \setlength\leftmarginiii{2\jsZw}
1835 \setlength\leftmarginiv {2\jsZw}
1836 \setlength\leftmarginv {1\jsZw}
1837 \setlength\leftmarginvi {1\jsZw}
1838 \fi
```

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

```
1839 \setlength \labelsep {0.5\jsZw} % .5em
1840 \setlength \labelwidth{\leftmargini}
1841 \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。

```
1842 \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 1843 \@beginparpenalty -\@lowpenalty
\@itempenalty 1844 \@endparpenalty -\@lowpenalty
1845 \@itempenalty -\@lowpenalty
```

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

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

```
1846 \def\@listi{\leftmargin\leftmargini
1847 \parsep \z@
1848 \topsep 0.5\baselineskip
1849 \itemsep \z@ \relax}
1850 \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）。

```
1851 \@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 1852 \def\@listii{\leftmargin\leftmarginii
\@listiv 1853 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1854 \topsep \z@
\@listv 1855 \parsep \z@
\@listvi 1856 \itemsep\parsep}
1857 \def\@listiii{\leftmargin\leftmarginiii
1858 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1859 \topsep \z@
1860 \parsep \z@
1861 \itemsep\parsep}
1862 \def\@listiv {\leftmargin\leftmarginiv
1863 \labelwidth\leftmarginiv
1864 \advance\labelwidth-\labelsep}
1865 \def\@listv {\leftmargin\leftmarginv
1866 \labelwidth\leftmarginv
1867 \advance\labelwidth-\labelsep}
1868 \def\@listvi {\leftmargin\leftmarginvi
1869 \labelwidth\leftmarginvi
1870 \advance\labelwidth-\labelsep}
```

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

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

```
1871 \renewcommand{\theenumi}{\@arabic\c@enumi}
1872 \renewcommand{\theenumii}{\@alph\c@enumii}
1873 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1874 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

\labelenumi enumerate 環境の番号を出力する命令です。第 2 レベル以外は最後に欧文のピリオドが付きますが、これは好みに応じて取り払ってください。第 2 レベルの番号のかっこは和文用に換え、その両側に入る余分なグルーを `\inhibitglue` で取り除いています。

\labelenumiv

和文の括弧で囲むための補助命令 `\jsInJaParen` を定義して `\labelenumii` でそれを用いている。

※現状の `zxjatype` の `\inhibitglue` の実装には「前後のグルーを消してしまう」という不備があって、そのため `enumii` の出力が異常になるという不具合があった。`zxjatype` を修正するまでの回避策として、サイズがゼロの罫 (`\bxjs@dust`) でガードしておく。

```

1875 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1876 \newcommand*\{jsInJaParen}[1]{%
1877   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1878 \newcommand{\labelenumii}{\theenumi.}
1879 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1880 \newcommand{\labelenumiii}{\theenumiii.}
1881 \newcommand{\labelenumiv}{\theenumiv.}

```

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

```

\p@enumiv 1882 \renewcommand{\p@enumii}{\theenumi}
1883 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1884 \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 1885 \newcommand\labelitemi{\textbullet}
\labelitemiii 1886 \newcommand\labelitemii{\normalfont\bfseries \textendash}
1887 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 1888 \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。

```

1889 \newenvironment{description}{%
1890   \list{}{%
1891     \labelwidth=\leftmargin
1892     \labelsep=1\jsZw
1893     \advance \labelwidth by -\labelsep
1894     \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。

```

1895 \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。

```

1896 %<*book|report>
1897 \newenvironment{abstract}{%

```

```

1898 \begin{list}{}{%
1899     \listparindent=1\jsZw
1900     \itemindent=\listparindent
1901     \rightmargin=\z@
1902     \leftmargin=5\jsZw}\item[]{}\end{list}\vspace{\baselineskip}}
1903 %</book|report>
1904 %<*article|slide>
1905 \newbox\@abstractbox
1906 \if@titlepage
1907     \newenvironment{abstract} {%
1908         \titlepage
1909         \null\vfil
1910         \@beginparpenalty\@lowpenalty
1911         \begin{center}%
1912             \headfont \abstractname
1913             \@endparpenalty\@M
1914         \end{center}}%
1915     {\par\vfil\null\endtitlepage}
1916 \else
1917     \newenvironment{abstract} {%
1918         \if@twocolumn
1919             \ifx\maketitle\relax
1920                 \section*{\abstractname}%
1921             \else
1922                 \global\setbox\@abstractbox\hbox\bgroup
1923                 \begin{minipage}[b]{\textwidth}
1924                     \small\parindent1\jsZw
1925                     \begin{center}%
1926                         {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1927                     \end{center}%
1928                     \list{}{%
1929                         \listparindent\parindent
1930                         \itemindent \listparindent
1931                         \rightmargin \leftmargin}%
1932                     \item\relax
1933                 \fi
1934             \else
1935                 \small
1936                 \begin{center}%
1937                     {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1938                 \end{center}%
1939                 \list{}{%
1940                     \listparindent\parindent
1941                     \itemindent \listparindent
1942                     \rightmargin \leftmargin}%
1943                 \item\relax
1944             \fi}\if@twocolumn
1945             \ifx\maketitle\relax
1946             \else

```

```

1947         \endlist\end{minipage}\egroup
1948     \fi
1949     \else
1950         \endlist
1951     \fi}
1952 \fi
1953 %</article|slide>
1954 %<*jspf>
1955 \newbox\@abstractbox
1956 \newenvironment{abstract}{%
1957     \global\setbox\@abstractbox\hbox\bgroup
1958     \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
1959         \small
1960         \if@english \parindent6mm \else \parindent1\jsZw \fi}%
1961     {\end{minipage}\egroup}
1962 %</jspf>

```

■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。

```

1963 %<*jspf>
1964 %\newbox\@keywordsbox
1965 %\newenvironment{keywords}{%
1966 %     \global\setbox\@keywordsbox\hbox\bgroup
1967 %     \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
1968 %         \small\parindent0\jsZw}%
1969 %     {\end{minipage}\egroup}
1970 %</jspf>

```

■verse 環境

verse 詩のための `verse` 環境です。

```

1971 \newenvironment{verse}{%
1972     \let \\\=\@centercr
1973     \list{}{%
1974         \itemsep \z@
1975         \itemindent -2\jsZw % 元: -1.5em
1976         \listparindent\itemindent
1977         \rightmargin \z@
1978         \advance\leftmargin 2\jsZw}% 元: 1.5em
1979     \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。

```

1980 \newenvironment{quotation}{%
1981     \list{}{%

```

```

1982 \listparindent\parindent
1983 \itemindent\listparindent
1984 \rightmargin \z0}%
1985 \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。

```

1986 \newenvironment{quote}%
1987 {\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。

```

1988 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
1989 \item[\hskip \labelsep{\headfont #1\ #2}]}
1990 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
1991 \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。

```

1992 \newenvironment{titlepage}{%
1993 %<book|report> \cleardoublepage
1994 \if@twocolumn
1995 \@restonecoltrue\onecolumn
1996 \else
1997 \@restonecolfalse\newpage
1998 \fi
1999 \thispagestyle{empty}%
2000 \setcounter{page}\@ne
2001 }%
2002 {\if@restonecol\twocolumn \else \newpage \fi
2003 \if@twoside\else
2004 \setcounter{page}\@ne
2005 \fi}

```

■付録

\appendix 本文と付録を分離するコマンドです。

```

2006 %<*&!book&!report>
2007 \newcommand{\appendix}{\par

```

```

2008 \setcounter{section}{0}%
2009 \setcounter{subsection}{0}%
2010 \gdef\presectionname{\appendixname}%
2011 \gdef\postsectionname{}%
2012 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2013 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2014 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}%
2015 %<!/book&!report>
2016 %<*book|report>
2017 \newcommand{\appendix}{\par
2018 \setcounter{chapter}{0}%
2019 \setcounter{section}{0}%
2020 \gdef\@chapapp{\appendixname}%
2021 \gdef\@chappos{}%
2022 \gdef\thechapter{\@Alph\c@chapter}}
2023 %</book|report>

```

7.4 パラメータの設定

■array と tabular 環境

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

```
2024 \setlength\arraycolsep{5\p@?}
```

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

```
2025 \setlength\tabcolsep{6\p@?}
```

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

```
2026 \setlength\arrayrulewidth{.4\p@}
```

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

```
2027 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2028 \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。

```
2029 \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。

```
2030 \setlength\fboxsep{3\p@?}
2031 \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。

```
2032 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2033 %<*book|report>
2034 \@addtoreset{equation}{chapter}
2035 \renewcommand\theequation
2036 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2037 %</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。

```
2038 % \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。

```
2039 % \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。

```
2040 % \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip\@italiccorr)}}
```

7.5 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。

```
2041 %<*!book&!report>
2042 \newcounter{figure}
2043 \renewcommand \thefigure {\@arabic\c@figure}
```

```

2044 %</*!book&!report>
2045 %<*book|report>
2046 \newcounter{figure}[chapter]
2047 \renewcommand \thefigure
2048     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2049 %</book|report>

```

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

```

\ext@figure 2050 \def\fps@figure{tbp}
2051 \def\ftype@figure{1}
\fnun@figure 2052 \def\ext@figure{lof}
2053 \def\fnun@figure{\figurename\nobreak\thefigure}

```

figure * 形式は段抜きのフロートです。

```

figure* 2054 \newenvironment{figure}%
2055     {\@float{figure}}%
2056     {\end@float}
2057 \newenvironment{figure*}%
2058     {\@dblfloat{figure}}%
2059     {\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。

```

2060 %</*!book&!report>
2061 \newcounter{table}
2062 \renewcommand\thetable{\@arabic\c@table}
2063 %</*!book&!report>
2064 %<*book|report>
2065 \newcounter{table}[chapter]
2066 \renewcommand \thetable
2067     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2068 %</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 2069 \def\fps@table{tbp}
2070 \def\ftype@table{2}
\fnun@table 2071 \def\ext@table{lot}
2072 \def\fnun@table{\tablename\nobreak\thetable}

```

table * は段抜きのフロートです。

```

table* 2073 \newenvironment{table}%
2074     {\@float{table}}%
2075     {\end@float}
2076 \newenvironment{table*}%

```

```

2077          {\@dblfloat{table}}}%
2078          {\end@dblfloat}

```

7.6 キャプション

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

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

```

2079 \newlength\abovecaptionskip
2080 \newlength\belowcaptionskip
2081 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2082 \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。

```

2083 %<*.jspf>
2084 % \long\def\@makecaption#1#2{{\small
2085 %   \advance\leftskip1cm
2086 %   \advance\rightskip1cm
2087 %   \vskip\abovecaptionskip
2088 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2089 %   \ifdim \wd\@tempboxa >\hsize
2090 %     #1\hskip1\jsZw\relax #2\par
2091 %   \else
2092 %     \global \@minipagefalse
2093 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2094 %   \fi
2095 %   \vskip\belowcaptionskip}}
2096 \long\def\@makecaption#1#2{{\small
2097   \advance\leftskip .0628\linewidth
2098   \advance\rightskip .0628\linewidth
2099   \vskip\abovecaptionskip
2100   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2101   \ifdim \wd\@tempboxa <\hsize \centering \fi
2102   #1\hskip1\jsZw\relax #2\par
2103   \vskip\belowcaptionskip}}
2104 %</!.jspf>
2105 %<*.jspf>
2106 \long\def\@makecaption#1#2{%
2107   \vskip\abovecaptionskip
2108   \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2109   \ifdim \wd\@tempboxa >\hsize
2110     {\small\sffamily

```

```

2111 \list{#1}{%
2112 \renewcommand{\makelabel}[1]{##1\hfil}
2113 \itemsep \z@
2114 \itemindent \z@
2115 \labelsep \z@
2116 \labelwidth 11mm
2117 \listparindent\z@
2118 \leftmargin 11mm}\item\relax #2\endlist}
2119 \else
2120 \global \@minipagefalse
2121 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2122 \fi
2123 \vskip\belowcaptionskip}
2124 %</jspf>

```

8 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。

```

```

2125 \newif\if@jsc@warnoldfontcmd
2126 \@jsc@warnoldfontcmdtrue
2127 \newif\if@jsc@warnoldfontcmdexception
2128 \@jsc@warnoldfontcmdexceptionfalse

```

`\jsc@DeclareOldFontCommand`

```

2129 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2130 \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2131 \DeclareOldFontCommand{#1}{%
2132 \bxjs@oldfontcmd{#1}#2%
2133 }{%
2134 \bxjs@oldfontcmd{#1}#3%
2135 }%
2136 }
2137 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%

```

```

2138 \ClassInfo\bxjs@clsname
2139 {Old font command '\string#1' is used!!\MessageBreak
2140 The first occurrence is}%
2141 }

```

`\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。

```

2142 \newcommand*{\allowoldfontcommands}{%
2143 \@jsc@warnoldfontcmdexceptiontrue}
2144 \newcommand*{\disallowoldfontcommands}{%
2145 \@jsc@warnoldfontcmdexceptionfalse}

```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2146 \let\bxjs@oldfontcmd@list\@empty
2147 \def\bxjs@oldfontcmd#1{%
2148 \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2149 \def\bxjs@oldfontcmd@a#1#2{%
2150 \if@jsc@warnoldfontcmdexception\else
2151 \global\@jsc@warnoldfontcmdfalse
2152 \ifx#1\relax
2153 \global\let#1=t%
2154 \jsc@warnoldfontcmd{#2}%
2155 \fi
2156 \fi}
2157 \def\bxjs@warnoldfontcmd@final{%
2158 \par
2159 \let\@tempa\@empty
2160 \def\do##1{%
2161 \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2162 \edef\@tempa{\@tempa \space\string##1}\fi}
2163 \bxjs@oldfontcmd@list
2164 \ifx\@tempa\@empty\else
2165 \ClassWarningNoLine\bxjs@clsname
2166 {Some old font commands were used in text\MessageBreak
2167 (see the log file for detail)}%
2168 \ClassInfo\bxjs@clsname
2169 {Some old font commands were used in text:\MessageBreak
2170 \space\@tempa\MessageBreak
2171 You should note, that since 1994 LaTeX2e provides a\MessageBreak
2172 new font selection scheme called NFSS2 with several\MessageBreak
2173 new, combinable font commands. The
2174 class provides\MessageBreak
2175 the old font commands
2176 only for compatibility%
2177 \@gobble}%
2178 \fi}
2179 \AtEndDocument{\bxjs@warnoldfontcmd@final}

```

`\mc` フォントファミリーを変更します。

```
\gt 2180 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2181 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2182 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2183 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2184 \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るコマンドは `\mdseries` です。

```
2185 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

`\it` フォントシェイプを変えるコマンドです。斜体とスモールキャップスは数式中では何もしません（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るコマンドは `\upshape` です。

```
2186 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2187 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2188 \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 2189 \DeclareRobustCommand*{\cal}{\@fontswitch\relax\mathcal}
2190 \DeclareRobustCommand*{\mit}{\@fontswitch\relax\mathnormal}
```

9 相互参照

9.1 目次の類

`\section` コマンドは `.toc` ファイルに次のような行を出力します。

```
\contentsline{section}{タイトル}{ページ}
```

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

となります。この「番号」は `\thesection` コマンドで生成された見出し番号です。

`figure` 環境の `\caption` コマンドは `.lof` ファイルに次のような行を出力します。

```
\contentsline{figure}{\numberline{番号}{キャプション}{ページ}}
```

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

`\contentsline{...}` は `\l@...` というコマンドを実行するので、あらかじめ `\l@chapter`, `\l@section`, `\l@figure` などを定義しておかなければなりません。これらの多くは `\@dottedtocline` コマンドを使って定義します。これは

`\@dottedtocline{レベル}{インデント}{幅}{タイトル}{ページ}`

という書式です。

レベル この値が `tocdepth` 以下のときだけ出力されます。`\chapter` はレベル 0, `\section` はレベル 1, 等々です。

インデント 左側の字下げ量です。

幅 「タイトル」に `\numberline` コマンドが含まれる場合, 節番号が入る箱の幅です。

`\@pnumwidth` ページ番号の入る箱の幅です。

`\@tocrmarg` 右マージンです。`\@tocrmarg ≥ \@pnumwidth` とします。

`\@dotsep` 点の間隔です (単位 mu)。

`\c@tocdepth` 目次ページに出力する見出しレベルです。元は `article` で 3, その他で 2 でしたが, ここでは一つずつ減らしています。

```
2191 \newcommand\@pnumwidth{1.55em}
2192 \newcommand\@tocrmarg{2.55em}
2193 \newcommand\@dotsep{4.5}
2194 %<!book&!report>\setcounter{tocdepth}{2}
2195 %<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)

```
2196 \newdimen\jsc@tocl@width
2197 \newcommand{\tableofcontents}{%
2198 %<*book|report>
2199   \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2200   \settowidth\@tempdima{\headfont\appendixname}%
2201   \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2202   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2203   \if@twocolumn
2204     \@restonecoltrue\onecolumn
2205   \else
2206     \@restonecolfalse
2207   \fi
2208   \chapter*{\contentsname}%
2209   \@mkboth{\contentsname}{}%
2210 %</book|report>
2211 %<*!book&!report>
2212   \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2213   \settowidth\@tempdima{\headfont\appendixname}%
2214   \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2215   \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2216   \section*{\contentsname}%

```

```

2217 \mkboth{\contentsname}{\contentsname}%
2218 %</!book&!report>
2219 \@starttoc{toc}%
2220 %<book|report> \if@restonecol\twocolumn\fi
2221 }

```

\l@part 部の目次です。

```

2222 \newcommand*{\l@part}[2]{%
2223 \ifnum \c@tocdepth >-2\relax
2224 %<!book&!report> \addpenalty\@secpenalty
2225 %<book|report> \addpenalty{-\@highpenalty}%
2226 \addvspace{2.25em \@plus\p@}%
2227 \begingroup
2228 \parindent \z@
2229 % \@pnumwidth should be \@tocrmarg
2230 % \rightskip \@pnumwidth
2231 \rightskip \@tocrmarg
2232 \parfillskip -\rightskip
2233 {\leavevmode
2234 \large \headfont
2235 \setlength\@lnumwidth{4\jsZw}%
2236 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2237 \nobreak
2238 %<book|report> \global\@nobreaktrue
2239 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2240 \endgroup
2241 \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)

```

2242 %<*book|report>
2243 \newcommand*{\l@chapter}[2]{%
2244 \ifnum \c@tocdepth >\m@ne
2245 \addpenalty{-\@highpenalty}%
2246 \addvspace{1.0em \@plus\p@}
2247 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2248 \begingroup
2249 \parindent \z@
2250 % \rightskip \@pnumwidth
2251 \rightskip \@tocrmarg
2252 \parfillskip -\rightskip
2253 \leavevmode\headfont
2254 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2255 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2256 \advance\leftskip\@lnumwidth \hskip-\leftskip
2257 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2258 \penalty\@highpenalty
2259 \endgroup

```



```

2260 \fi}
2261 %</book|report>

```

\l@section 節の目次です。

```

2262 %<!*book&!report>
2263 \newcommand*{\l@section}[2]{%
2264 \ifnum \c@tocdepth >\z@
2265 \addpenalty{\@secpenalty}%
2266 \addvspace{1.0em \@plus\p@?}%
2267 \begingroup
2268 \parindent\z@
2269 % \rightskip\@pnumwidth
2270 \rightskip\@tocrmarg
2271 \parfillskip-\rightskip
2272 \leavevmode\headfont
2273 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2274 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2275 \advance\leftskip\@lnumwidth \hskip-\leftskip
2276 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2277 \endgroup
2278 \fi}
2279 %</!*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。

```

2280 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

```

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

\l@subsubsection しれません。

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

```

\l@subparagraph 2281 %<!*book&!report>
2282 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2283 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2284 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2285 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2286 %
2287 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2288 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2289 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2290 %
2291 \newcommand*{\l@subsection}{%
2292 \tempdima\jsc@tocl@width \advance\tempdima -1\jsZw
2293 \@dottedtocline{2}{\tempdima}{3\jsZw}}
2294 \newcommand*{\l@subsubsection}{%
2295 \tempdima\jsc@tocl@width \advance\tempdima 0\jsZw
2296 \@dottedtocline{3}{\tempdima}{4\jsZw}}
2297 \newcommand*{\l@paragraph}{%
2298 \tempdima\jsc@tocl@width \advance\tempdima 1\jsZw

```

```

2299          \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2300 \newcommand*{\l@subparagraph}{%
2301          \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2302          \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2303 %<!/book&!report>
2304 %<*book|report>
2305 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2306 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2307 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2308 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2309 \newcommand*{\l@section}{%
2310          \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2311          \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2312 \newcommand*{\l@subsubsection}{%
2313          \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2314          \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2315 \newcommand*{\l@subsubsection}{%
2316          \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2317          \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2318 \newcommand*{\l@paragraph}{%
2319          \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2320          \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2321 \newcommand*{\l@subparagraph}{%
2322          \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2323          \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2324 %</book|report>

```

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

```

2325 \newdimen\@lnumwidth
2326 \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。

```

2327 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2328   \vskip \z@ \@plus.2\p@?
2329   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2330    \parindent #2\relax\@afterindenttrue
2331    \interlinepenalty\@M
2332    \leavevmode
2333    \@lnumwidth #3\relax
2334    \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2335    {#4}\nobreak
2336    \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2337     mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2338     \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```
2339 \newcommand{\listoffigures}{%
2340 %<*book|report>
2341 \if@twocolumn\@restonecoltrue\onecolumn
2342 \else\@restonecolfalse\fi
2343 \chapter*{\listfigurename}%
2344 \@mkboth{\listfigurename}{}%
2345 %</book|report>
2346 %<*&!book&!report>
2347 \section*{\listfigurename}%
2348 \@mkboth{\listfigurename}{\listfigurename}%
2349 %</&!book&!report>
2350 \@starttoc{lof}%
2351 %<book|report> \if@restonecol\twocolumn\fi
2352 }
```

`\l@figure` 図目次の項目を出力します。

```
2353 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```
2354 \newcommand{\listoftables}{%
2355 %<*book|report>
2356 \if@twocolumn\@restonecoltrue\onecolumn
2357 \else\@restonecolfalse\fi
2358 \chapter*{\listtablename}%
2359 \@mkboth{\listtablename}{}%
2360 %</book|report>
2361 %<*&!book&!report>
2362 \section*{\listtablename}%
2363 \@mkboth{\listtablename}{\listtablename}%
2364 %</&!book&!report>
2365 \@starttoc{lot}%
2366 %<book|report> \if@restonecol\twocolumn\fi
2367 }
```

`\l@table` 表目次は図目次と同じです。

```
2368 \let\l@table\l@figure
```

9.2 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。

```
2369 \newdimen\bibindent
2370 \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。

```

2371 \newenvironment{thebibliography}[1]{%
2372   \jsc@warnoldfontcmdexceptiontrue
2373   \global\let\presectionname\relax
2374   \global\let\postsectionname\relax
2375   %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2376   %<*kiyou>
2377   \vspace{1.5\baselineskip}
2378   \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2379   \vspace{0.5\baselineskip}
2380 %</kiyou>
2381 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2382 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2383   \list{\@biblabel{\@arabic{c@enumiv}}}%
2384     {\settowidth\labelwidth{\@biblabel{#1}}%
2385      \leftmargin\labelwidth
2386      \advance\leftmargin\labelsep
2387      \@openbib@code
2388      \usecounter{enumiv}%
2389      \let\p@enumiv\@empty
2390      \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2391 %<kiyou> \small
2392   \sloppy
2393   \clubpenalty4000
2394   \@clubpenalty\clubpenalty
2395   \widowpenalty4000%
2396   \sfcode`.\@m}
2397 {\def\@noitemerr
2398   {\@latex@warning{Empty `thebibliography' environment}}}%
2399 \endlist}

```

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

```

2400 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

```

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

```

2401 \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。

```

2402 % \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。かつこの前後に入るグルーを `\jsInhibitGlue` で取っていますので、オリジナル同様、`Knuth~\cite{knu}` のように半角空白で囲んでください。

```

2403 % \def\@citex[#1]#2{\leavevmode
2404 %   \let\@citea\@empty
2405 %   \@cite{\@for\@citeb:=#2\do
2406 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2407 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2408 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2409 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2410 %       \G@refundefinedtrue
2411 %       \@latex@warning
2412 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2413 %       {\@citeofmt{\csname b@\@citeb\endcsname}}}{#1}}
2414 % \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。

```

2415 % \DeclareRobustCommand\cite{\unskip
2416 %   \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}]
2417 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2418 %   , \jsInhibitGlue\ #2\fi}})}$}

```

9.3 索引

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

```

2419 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2420   \if@twocolumn
2421     \onecolumn\@restonecolfalse
2422   \else
2423     \clearpage\@restonecoltrue
2424   \fi
2425   \columnseprule.4pt \columnsep 2\jsZw
2426   \ifx\multicols\@undefined
2427 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2428 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2429 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2430 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2431   \else
2432     \ifdim\textwidth<\fullwidth
2433       \setlength{\evensidemargin}{\oddsidemargin}
2434       \setlength{\textwidth}{\fullwidth}
2435       \setlength{\linewidth}{\fullwidth}
2436 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
2437 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2438 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2439 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%

```

```

2440     \else
2441 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}]%
2442 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2443 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2444 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2445     \fi
2446   \fi
2447 %<book|report>      \@mkboth{\indexname}{}%
2448 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2449   \plainifnotempty % \thispagestyle{plain}
2450   \parindent\z@
2451   \parskip\z@ \@plus .3\p@?\relax
2452   \let\item\@idxitem
2453   \raggedright
2454   \footnotesize\narrowbaselines
2455 }{
2456   \ifx\multicols\@undefined
2457     \if@restonecol\onecolumn\fi
2458   \else
2459     \end{multicols}
2460   \fi
2461   \clearpage
2462 }

```

`\@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 2463 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2464 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2465 \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。

```

2466 \newcommand{\indexspace}{\par \vskip 10\p@? \@plus5\p@? \@minus3\p@?\relax}

```

`\seename` 索引の `\see`, `\seealso` コマンドで出力されるものです。デフォルトはそれぞれ `see`, `see also`

`\alsoname` という英語ですが、ここではとりあえず両方とも「→」に変えました。⇒ (`\Rightarrow`)
などでもいいでしょう。

```

2467 \newcommand\seename{\if@english see\else →\fi}
2468 \newcommand\alsoname{\if@english see also\else →\fi}

```

9.4 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。

```

2469 \ifx\pltx@foot@penalty\undefined
2470   \let\footnotes@ve=\footnote
2471   \def\footnote{\jsInhibitGlue\footnotes@ve}
2472   \let\footnotemarks@ve=\footnotemark
2473   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2474 \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 pTeX 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 pLaTeX の変更に従いました (Thanks: 角藤さん)。pLaTeX 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。

pTeX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

[2016-10-08] TODO: 脚注番号が `newtxtext` や `newpxtext` の使用時におかしくなっています。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)。

```

2475 \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。

```

2476 % \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。

```

2477 \renewcommand{\footnoterule}{%
2478   \kern-2.6\p@? \kern-.4\p@
2479   \hrule width .4\columnwidth
2480   \kern 2.6\p@?}

```

\c@footnote 脚注番号は章ごとにリセットされます。

```

2481 %<book|report>\@addtoreset{footnote}{chapter}

```

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

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

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

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

```
2482 \long\def\@footnotetext{%
2483   \insert\footins\bgroup
2484     \normalfont\footnotesize
2485     \interlinepenalty\interfootnotelinepenalty
2486     \splittopskip\footnotesep
2487     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2488     \hsize\columnwidth \@parboxrestore
2489     \protected@edef\@currentlabel{%
2490       \csname p@footnote\endcsname\@thefnmark
2491     }%
2492     \color@begingroup
2493       \@makefnmark
2494       \rule{z@\footnotesep}{\ignorespaces}%
2495       \futurelet\next\fo@t}
2496 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@@
2497                                     \else \let\next\fo@t\fi \next}
2498 \def\fo@@{\bgroup\aftergroup\@foot\let\next}
2499 \def\fo@t#1{#1\@foot}
2500 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2501   \ifx\pltx@foot@penalty\@undefined\else
2502     \ifhmode\null\fi
2503     \ifnum\pltx@foot@penalty=z@\else
2504       \penalty\pltx@foot@penalty
2505       \pltx@foot@penalty\z@
2506     \fi
2507   \fi}
```

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

```
2508 \newcommand\@makefnmark[1]{%
2509   \advance\leftskip 3\jsZw
2510   \parindent 1\jsZw
2511   \noindent
2512   \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。

```
2513 % \def\@xfootnotenext[#1]{%
2514 %   \begingroup
```



```

2515 %      \ifnum#1>\z@
2516 %      \csname c@\@mpfn\endcsname #1\relax
2517 %      \unrestored@protected@xdef\@thefnmark{\thempfn}%
2518 %      \else
2519 %      \unrestored@protected@xdef\@thefnmark{}%
2520 %      \fi
2521 %  \endgroup
2522 %  \@footnotetext}

```

ここまでのコードは JS クラスを踏襲する。

10 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。

```
2523 \let\jsInhibitGlueAtParTop\@empty
```

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

```

2524 \def\everyparhook{\jsInhibitGlueAtParTop}
2525 \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)。

`\@inhibitglue` JS クラスでの `\jsInhibitGlueAtParTop` の実装。(これは (u)pTeX 専用である。)
 ※`\jsc@tempa` は実はテンポラリではなく「この処理専用のユニーク制御綴」である（でないと不正である）が、間違って別の箇所で使う危険性が高いので `\bxjs@ig@temp` に置き換えた。

```

2526 \def\@inhibitglue{%
2527   \futurelet\@let@token\@@inhibitglue}
2528 \begingroup
2529 \let\GDEF=\gdef
2530 \let\CATCODE=\catcode
2531 \let\ENDGROUP=\endgroup
2532 \CATCODE`k=12
2533 \CATCODE`a=12

```

```

2534 \CATCODE`n=12
2535 \CATCODE`j=12
2536 \CATCODE`i=12
2537 \CATCODE`c=12
2538 \CATCODE`h=12
2539 \CATCODE`r=12
2540 \CATCODE`t=12
2541 \CATCODE`e=12
2542 \GDEF\KANJI@CHARACTER{kanji character }
2543 \ENDGROUP
2544 \def\@@inhibitglue{%
2545   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
2546 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
2547 \def\bxjs@ig@temp{#1}%
2548 \ifx\bxjs@ig@temp\@empty
2549   \ifnum\the\inhibitxspcode`#2=2\relax
2550     \inhibitglue
2551   \fi
2552 \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。

```

2553 \def\@doendpe{%
2554   \@endpetrue
2555   \def\par{%
2556     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2557   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

`\item` 命令の直後です。

```

2558 \def\@item[#1]{%
2559   \if@noperitem
2560     \@donoperitem
2561   \else
2562     \if@inlabel
2563       \indent \par
2564     \fi
2565     \ifhmode
2566       \unskip\unskip \par
2567     \fi
2568     \if@newlist
2569       \if@nobreak
2570         \@nbitem
2571       \else
2572         \addpenalty\@beginparpenalty

```

```

2573         \addvspace\@topsep
2574         \addvspace{-\parskip}%
2575     \fi
2576 \else
2577     \addpenalty\@itempenalty
2578     \addvspace\itemsep
2579 \fi
2580 \global\@inlabeltrue
2581 \fi
2582 \everypar{%
2583     \@minipagefalse
2584     \global\@newlistfalse
2585     \if@inlabel
2586         \global\@inlabelfalse
2587         {\setbox\z@\lastbox
2588         \ifvoid\z@
2589             \kern-\itemindent
2590         \fi}%
2591     \box\@labels
2592     \penalty\z@
2593 \fi
2594 \if@nobreak
2595     \@nobreakfalse
2596     \clubpenalty \@M
2597 \else
2598     \clubpenalty \@clubpenalty
2599     \everypar{\everyparhook}%
2600 \fi
2601 \bxjs@ltj@inhibitglue
2602 \everyparhook}%
2603 \if@noitemarg
2604     \@noitemargfalse
2605     \if@nmbrlist
2606         \refstepcounter\@listctr
2607     \fi
2608 \fi
2609 \sbox\@tempboxa{\makelabel{#1}}%
2610 \global\setbox\@labels\hbox{%
2611     \unhbox\@labels
2612     \hskip \itemindent
2613     \hskip -\labelwidth
2614     \hskip -\labelsep
2615     \ifdim \wd\@tempboxa >\labelwidth
2616         \box\@tempboxa
2617     \else
2618         \hbox to\labelwidth {\unhbox\@tempboxa}%
2619     \fi
2620     \hskip \labelsep}%
2621 \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。

```
2622 \def\@afterheading{%
2623   \@nobreaktrue
2624   \everypar{%
2625     \if@nobreak
2626       \@nobreakfalse
2627       \clubpenalty \@M
2628       \if@afterindent \else
2629         {\setbox\z@\lastbox}%
2630       \fi
2631     \else
2632       \clubpenalty \@clubpenalty
2633       \everypar{\everyparhook}%
2634     \fi\everyparhook}}
```

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

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

```
2635 \def\@gnewline #1{%
2636   \ifvmode
2637     \@nolnerr
2638   \else
2639     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
2640     \jsInhibitGlue \ignorespaces
2641   \fi}
```

11 いろいろなロゴ

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

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

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

※`\小`、`\上小` の制御綴は定義しない。

```
2642 \if@jslogo
2643   \IfFileExists{jslogo.sty}{%
2644     \RequirePackage{jslogo}%
2645   }{%
2646     \ClassWarningNoLine{bxjs}{clsname
```

```

2647 {The package 'jslogo' is not installed.\MessageBreak
2648 It is included in the recent release of\MessageBreak
2649 the 'jsclasses' bundle}
2650 }
2651 \fi

```

12 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。

13 初期設定

■いろいろな語

```

\prepartname
\postpartname 2652 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2653 \newcommand{\postpartname}{\if@english\else 部\fi}
2654 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2655 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2656 \newcommand{\presectionname}{}% 第
\postsectionname 2657 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 2658 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2659 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
2660 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 2661 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2662 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
2663 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 2664 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
2665 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname
\abstractname 2666 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
2667 \newcommand{\appendixname}{\if@english \else 付録\fi}
2668 %<!book&!report>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

```
2669 \@tempwafalse
2670 \if p\jsEngine \@tempwattrue \fi
2671 \if n\jsEngine \@tempwattrue \fi
2672 \if@tempswa \expandafter\@firstoftwo
2673 \else \expandafter\@secondoftwo
2674 \fi
2675 {%
2676 % 欧文 8bitTeX の場合
2677 \newif\ifjsSeireki \jsSeirekitrue
2678 \def\bxjs@decl@Seireki@cmds{%
2679   \def\西暦{\jsSeirekitrue}%
2680   \def\和暦{\jsSeirekifalse}}
2681 \def\Seireki{\jsSeirekitrue}
2682 \def\Wareki{\jsSeirekifalse}
2683 \def\bxjs@if@use@seireki{%
2684   \ifjsSeireki \expandafter\@firstoftwo
2685   \else \expandafter\@secondoftwo \fi}
2686 }{%
2687 \newif\if 西暦 \西暦 true
2688 \def\bxjs@decl@Seireki@cmds{%
2689   \def\西暦{\西暦 true}%
2690   \def\和暦{\西暦 false}}
2691 \def\Seireki{\西暦 true}
2692 \def\Wareki{\西暦 false}
2693 \def\bxjs@if@use@seireki{%
2694   \if 西暦 \expandafter\@firstoftwo
2695   \else \expandafter\@secondoftwo \fi}
2696 }
2697 \bxjs@decl@Seireki@cmds
2698 % \bxjs@unxp
2699 \let\bxjs@unxp\@firstofone
2700 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2701 % \bxjs@iai
2702 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2703   \def\bxjs@iai{\noexpand~}
2704 \else \def\bxjs@iai{}
2705 \fi
2706 % \heisei
2707 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2708 % \today
2709 \edef\bxjs@today{%
```

```

2710 \if@english
2711 \ifcase\month\or
2712     January\or February\or March\or April\or May\or June\or
2713     July\or August\or September\or October\or November\or December\fi
2714     \space\number\day, \number\year
2715 \else
2716     \ifnum\heisei>\@ne
2717         \expandafter\noexpand\expandafter\bxjs@if@use@seireki
2718     \else \expandafter\@firstoftwo
2719     \fi {%
2720         \number\year\bxjs@iai\bxjs@unxp{年}%
2721         \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2722         \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2723     }{%
2724         \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2725         \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2726         \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2727     }%
2728 \fi}
2729 \let\today\bxjs@today

```

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

```

2730 \AtBeginDocument{%
2731     \ifx\bbl@jpn@Seirekitrue\@undefined\else
2732         \bxjs@decl@Seireki@cmds
2733         \g@addto@macro\datejapanese{%
2734             \let\today\bxjs@today}%
2735     \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: eng-lish)

```

2736 \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。

```

2737 %<slide>\pagestyle{empty}%
2738 %<article|report>\pagestyle{plain}%
2739 %<book>\pagestyle{headings}%
2740 \pagenumbering{arabic}
2741 \if@twocolumn
2742     \twocolumn
2743     \sloppy
2744     \flushbottom
2745 \else
2746     \onecolumn

```

```
2747 \raggedbottom
2748 \fi
2749 %<*slide>
2750 \renewcommand\familydefault{\sfdefault}
2751 \raggedright
2752 %</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。

```
2753 \catcode\?=12
2754 \ifx\bxjs@jadriver\relax\else
2755 \input{bxjsja-\bxjs@jadriver.def}
2756 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
2757 \bxjs@restore@jltrcc
2758 %</cls>
```

以上です。

付録 A 和文ドライバの仕様

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

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

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

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

以下で標準で用意されている和文ドライバの実装を示す。

```
2759 %<*drv>
```

付録 B 和文ドライバ : minimal

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

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

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

B.1 補助マクロ

```
2760 %<*minimal>
```

```
2761 %% 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.
```

```
2762 \def\DeclareJaTextFontCommand#1#2{%
```

```
2763   \DeclareRobustCommand#1[1]{%
```

```
2764     \relax
```

```
2765     \ifmmode \expandafter\nfss@text \fi
```

```
2766     {#2##1}}%
```

```
2767 }
```

```
\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。
```

```
2768 \long\def\bxjs@@CSsfdefault{\sfdefault}%
```

```
2769 \@onlypreamble\bxjs@if@sf@default
```

```
2770 \def\bxjs@if@sf@default#1{%
```

```
2771   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
```

```
2772   \AtBeginDocument{%
```

```
2773     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
```

```
2774 }
```

`\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。

```
2775 \def\jsLetHeadChar#1#2{%
2776   \begingroup
2777   \escapechar=`\ %
2778   \let\bxjs@tmpa={% brace-match-hack
2779   \bxjs@let@hchar@exp#2}%
2780   \endgroup
2781   \let#1\bxjs@g@tmpa}
2782 \def\bxjs@let@hchar@exp{%
2783   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2784 \def\bxjs@let@hchar@exp@a{%
2785   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
2786     \bxjs@let@hchar@out\let\relax
2787   }{\bxjs@cond@ifcat\noexpand\@let@token\sptoken\fi{% 空白
2788     \bxjs@let@hchar@out\let\space%
2789   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
2790     \bxjs@let@hchar@out\let\@backslashchar
2791   }{\bxjs@let@hchar@exp@b}}}%
2792 \def\bxjs@let@hchar@exp@b#1{%
2793   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2794 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2795   %\message{<#1#2>}%
2796   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴
2797     \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2798       \bxjs@let@hchar@out\let\relax
2799     }{%else
2800       \expandafter\bxjs@let@hchar@exp
2801     }%
2802   }{%else
2803     \bxjs@let@hchar@chr#1%
2804   }}
2805 \def\bxjs@let@hchar@chr#1{%
2806   \bxjs@let@hchar@out\def{#{1}}}%
2807 \def\bxjs@let@hchar@out#1#2{%
2808   \global#1\bxjs@g@tmpa#2\relax
2809   \toks@\bgroup}% skip to right brace
```

UTF-8 のバイト列を扱うコード。

```
2810 \chardef\bxjs@let@hchar@csta=128
2811 \chardef\bxjs@let@hchar@cstb=192
2812 \chardef\bxjs@let@hchar@cstc=224
```

```

2813 \chardef\bxjs@let@hchar@cstd=240
2814 \chardef\bxjs@let@hchar@cste=248
2815 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2816 \def\bxjs@let@hchar@chr@ue#1{%
2817   \@tempcnta=#1\relax
2818   \%message{\the\@tempcnta}%
2819   \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2820     \bxjs@let@hchar@chr@ue@a#1%
2821   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2822     \bxjs@let@hchar@out\let\relax
2823   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2824     \bxjs@let@hchar@chr@ue@b
2825   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2826     \bxjs@let@hchar@chr@ue@c
2827   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2828     \bxjs@let@hchar@chr@ue@d
2829   }{%else
2830     \bxjs@let@hchar@out\let\relax
2831   }}}}
2832 \def\bxjs@let@hchar@chr@ue@a#1{%
2833   \bxjs@let@hchar@out\def{{#1}}}
2834 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2835   \bxjs@let@hchar@out\def{{#1#2}}}
2836 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2837   \bxjs@let@hchar@out\def{{#1#2#3}}}
2838 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2839   \bxjs@let@hchar@out\def{{#1#2#3#4}}}

```

B.2 (u)pTeX 用の設定

```
2840 \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。

```

2841 \def\bxjs@let@hchar@chr@pp#1{%
2842   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
2843 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2844   \%message{(#1)}%
2845   \bxjs@cond\if#1t\fi{%
2846     \bxjs@let@hchar@chr@ue#3%
2847   }{%else
2848     \bxjs@let@hchar@out\def{{#3}}}%
2849   }}
2850 \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。

```
2851 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
2852 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
2853 \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。

```
2854 \@onlypreamble\bxjs@declarefontshape
2855 \ifjsWithupTeX
2856 \def\bxjs@declarefontshape{%
2857 \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
2858 \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpgnt-h}{}%
2859 \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
2860 \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpgnt-v}{}%
2861 }
2862 \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。

```
2863 \else
2864 \def\bxjs@declarefontshape{%
2865 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
2866 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
2867 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
2868 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
2869 }
2870 \def\bxjs@sizereference{jis}
2871 \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。

```
2872 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
2873   \def\bxjs@y{#5}}
2874 \ifjsWithpTeXng \def\bxjs@y{10}%
2875 \else
2876 \expandafter\expandafter\expandafter\bxjs@tmpa
2877 \expandafter\string\the\jfont\relax
2878 \fi
2879 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
2880               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
2881   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
2882    \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。

```
2883 \begingroup
2884 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2885 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2886 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2887 % 幅が丁度 10pt なら補正は不要
2888 \ifdim\wd\z@=10pt
2889 \global\let\bxjs@scale\jsScale
2890 \else
2891 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
2892 \edef\bxjs@tmpa{\strip@pt\wd\z@}
2893 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
2894 \bxjs@invscale\@tempdima\bxjs@tmpa
2895 \xdef\bxjs@scale{\strip@pt\@tempdima}
2896 \fi
2897 \endgroup
2898 %\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。

```
2899 \bxjs@declarefontshape

    フォント代替の明示的定義。

2900 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2901 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
2902 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
2903 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
2904 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
2905 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
2906 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
2907 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2908 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
2909 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
2910 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
2911 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
2912 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
2913 \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。

```
2914 \DeclareRobustCommand\rmfamily
2915 {\not@math@alphabet\rmfamily\mathrm
2916 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
```

```

2917 \DeclareRobustCommand\sffamily
2918   {\not@math@alphabet\sffamily\mathsf
2919     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
2920 \DeclareRobustCommand\ttfamily
2921   {\not@math@alphabet\ttfamily\mathtt
2922     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
2923 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
2924 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
2925 \bxjs@if@sf@default{%
2926   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

2927 \selectfont

```

■パラメタの設定

```

2928 \prebreakpenalty\jis"2147=10000
2929 \postbreakpenalty\jis"2148=10000
2930 \prebreakpenalty\jis"2149=10000
2931 \inhibitxspcode`!=1
2932 \inhibitxspcode`〒=2
2933 \xspcode`+=3
2934 \xspcode`%=3

```

"80～"FF の範囲の \spcode を 3 に変更。

```

2935 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
2936   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

2937 \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。

```

2938 \begingroup
2939 \catcode`\!=0

```

\bxjs@ptex@dir 現在の組方向： t=縦、y=横、?=その他。

```

2940 \gdef\bxjs@ptex@dir{%
2941   !ifttdir t%
2942   !else!ifydir y%
2943   !else ?%
2944   !fi!fi}

```

新版の pTeX で脚注番号の周囲の空きが過大になる現象への対処。

※現在の pTeX 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。

```

2945 % 古い \@makefnmark の定義

```

```

2946 \long\def\bxjs@tmpa{\hbox{%
2947   !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
2948   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
2949 \ifx\@makefnmark\bxjs@tmpa
2950 \long\gdef\@makefnmark{%
2951   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
2952   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
2953 \fi
2954 \endgroup

```

B.3 pdfTeX 用の処理

```

2955 \else\ifx p\jsEngine
2956 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
2957 \@onlypreamble\bxjs@cjk@loaded
2958 \def\bxjs@cjk@loaded{%
2959   \def\@footnotemark{%
2960     \leavevmode
2961     \ifhmode
2962       \edef\x@sf{\the\spacefactor}%
2963       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
2964         \unkern\unkern
2965         \ifdim\lastskip>\z@ \unskip \fi
2966         \fi\fi
2967         \nobreak
2968       \fi
2969       \@makefnmark
2970       \ifhmode \spacefactor\x@sf \fi
2971       \relax}%
2972   \let\bxjs@cjk@loaded\relax
2973 }
2974 \AtBeginDocument{%
2975   \@ifpackageloaded{CJK}{%
2976     \bxjs@cjk@loaded
2977   }{}%
2978 }

```

B.4 XeTeX 用の処理

```

2979 \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。

2980 \def\bxjs@let@hchar@chr#1{%
2981   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
2982   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
2983     \bxjs@let@hchar@chr@xe
2984   }\bxjs@let@hchar@out\def{{#1}}}}
2985 \def\bxjs@let@hchar@chr@xe#1{%

```



```

2986 \lccode`0=`#1\relax
2987 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

2988 \ifx\XeTeXgenerateactualtext\@undefined\else
2989 \def\bxjs@do@precisetext{%
2990 \XeTeXgenerateactualtext=\@ne}
2991 \fi

```

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

```

2992 \@onlypreamble\bxjs@do@simplejasetup
2993 \def\bxjs@do@simplejasetup{%
2994 \ifnum\XeTeXinterchartokenstate>\z@
2995 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
2996 \jsSimpleJaSetup
2997 \ClassInfo\bxjs@clsname
2998 {'\string\jsSimpleJaSetup' is applied\@gobble}%
2999 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```

3000 \newcommand*{\jsSimpleJaSetup}{%
3001 \XeTeXlinebreaklocale "ja"\relax
3002 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3003 \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理 (エンジン共通)

```

3004 \fi\fi\fi

```

`simplejasetup` オプションの処理。

```

3005 \ifx\bxjs@do@simplejasetup\@undefined\else
3006 \AtBeginDocument{%
3007 \ifbxjs@simplejasetup
3008 \bxjs@do@simplejasetup
3009 \fi}
3010 \fi

```

`precisetext` オプションの処理。

```

3011 \ifbxjs@precisetext
3012 \ifx\bxjs@do@precisetext\@undefined
3013 \ClassWarning\bxjs@clsname
3014 {The current engine does not support the\MessageBreak
3015 'precisetext' option\@gobble}
3016 \else
3017 \bxjs@do@precisetext
3018 \fi
3019 \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。

3020 `\ifbxjs@fancyhdr`

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

3021 `\@onlypreamble\bxjs@adjust@fancyhdr`

3022 `\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。

3023 `\def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%`

3024 `\def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%`

3025 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

3026 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

3027 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

3028 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

3029 `\def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%`

3030 `\def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%`

3031 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

3032 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

3033 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

3034 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

3035 `\def\bxjs@tmpa{\rm\thepage\strut}%`

3036 `\def\bxjs@tmpb{\thepage\strut}%`

3037 `\ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi`

3038 `\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。

3039 `\ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth`

3040 `\setlength{\@tempdima}{\fullwidth-\textwidth}%`

3041 `\edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%`

3042 `}\bxjs@tmpa`

3043 `\fi\fi`

3044 `\PackageInfo\bxjs{clsname`

3045 `{Patch to fancyhdr is applied\@gobble}}`

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

3046 `\def\bxjs@pagestyle@hook{%`

3047 `\@ifpackageloaded{fancyhdr}{%`

3048 `\bxjs@adjust@fancyhdr`

3049 `\global\let\bxjs@adjust@fancyhdr\relax`

3050 `}{}}`

`\pagestyle` にフックを入れ込む。

3051 `\let\bxjs@org@pagestyle\pagestyle`

```

3052 \def\pagestyle{%
3053   \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。
3054 \AtBeginDocument{%
3055   \bxjs@pagestyle@hook
3056   \global\let\bxjs@pagestyle@hook\relax}

終わり。
3057 \fi

以上で終わり。
3058 %</minimal>

```

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

C.1 共通処理 (1)

まず minimal ドライバを読み込む。

```

3059 %<*standard>
3060 %% このファイルは日本語文字を含みます
3061 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3062 \bxjs@simplejasetupfalse

```

■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。

```
3063 \ifjsWitheTeX
```

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

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

```
\jH 3064 \@tempdima=0.25mm
      3065 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
      3066 \let\jH\jQ
```

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

```
\trueH 3067 \ifjsc@mag
      3068 \@tempdimb=\jsBaseFontSize\relax
      3069 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
      3070 \@tempdima=2.5mm
      3071 \bxjs@invscale\@tempdima\bxjs@tmpa
      3072 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
      3073 \@tempdima=10pt
      3074 \bxjs@invscale\@tempdima\bxjs@tmpa
      3075 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
      3076 \else \let\trueQ\jQ \let\bxjs@truept\p@
      3077 \fi
      3078 \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。

```
3079 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3080 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3081 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3082 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3083 \fi
```

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。

```
3084 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3085 \newcommand*\setkanjiskip[1]{%
3086 \edef\bxjs@kanjiskip{#1}%
3087 \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3088 \newcommand*\getkanjiskip{%
3089 \bxjs@kanjiskip}
```

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

```
3090 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

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

```
\bxjs@disable@kanjiskip 3091 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3092   \bxjs@kanjiskip@enabledtrue
3093   \bxjs@reset@kanjiskip}
3094 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3095   \bxjs@kanjiskip@enabledfalse
3096   \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。

```
3097 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3098   \ifbxjs@kanjiskip@enabled
3099     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3100   \else \@tempskipa\z@
3101   \fi
3102   \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 3103 \def\bxjs@xkanjiskip{Opt}
\getxkanjiskip 3104 \newcommand*\setxkanjiskip[1]{%
3105   \edef\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3106   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3107 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3108   \bxjs@xkanjiskip}
```

```
3109 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3110 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3111   \bxjs@xkanjiskip@enabledtrue
```

```
3112   \bxjs@reset@xkanjiskip}
```

```
3113 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3114   \bxjs@xkanjiskip@enabledfalse
```

```
3115   \bxjs@reset@xkanjiskip}
```

```
3116 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```
3117   \ifbxjs@xkanjiskip@enabled
```

```
3118     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
```

```
3119   \else \@tempskipa\z@
```

```
3120   \fi
```

```
3121   \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。

```
3122 \g@addto@macro\jsResetDimen{%
```

```
3123   \bxjs@reset@kanjiskip
```

```
3124   \bxjs@reset@xkanjiskip}
```

```
3125 \let\bxjs@apply@kanjiskip\relax
```

```
3126 \let\bxjs@apply@xkanjiskip\relax
```

■和文フォント指定の扱い

\bxjs@adjust@jafont ムニャムニャ…。

```

3127 \@onlypreamble\bxjs@adjust@jafont
3128 \def\bxjs@adjust@jafont#1{%
3129   \ifx\jsJaFont\bxjs@@auto
3130     \bxjs@get@kanjiEmbed
3131     \ifx\bxjs@kanjiEmbed\relax
3132       \let\bxjs@tmpa\@empty
3133     \else
3134       \let\bxjs@tmpa\bxjs@kanjiEmbed
3135     \fi
3136   \else
3137     \let\bxjs@tmpa\jsJaFont
3138   \fi
3139   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3140     \ClassWarningNoLine\bxjs@clsname
3141     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3142     not available on the current situation}%
3143   \let\bxjs@tmpa\@empty
3144   \fi\fi
3145 }
3146 \def\bxjs@@auto{auto}
3147 \def\bxjs@@noEmbed{noEmbed}

```

\bxjs@kanjiEmbed 現在の updmap の kanjiEmbed パラメタの値。 \bxjs@get@kanjiEmbed により設定される。

```

3148 \let\bxjs@kanjiEmbed\relax

```

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

```

3149 \@onlypreamble\bxjs@get@kanjiEmbed
3150 \def\bxjs@get@kanjiEmbed{%
3151   \begingroup\setbox\z@=\hbox{%
3152     \global\let\bxjs@g@tmpa\relax
3153     \endlinechar\m@ne
3154     \let\do\@makeother\dospecials
3155     \catcode32=10 \catcode12=10 %form-feed
3156     \let\bxjs@tmpa\@empty
3157     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3158     \ifeof\@inputcheck\else
3159       \read\@inputcheck to\bxjs@tmpa
3160       \closein\@inputcheck
3161     \fi
3162     \ifx\bxjs@tmpa\@empty\else
3163       \openin\@inputcheck="\bxjs@tmpa"\relax
3164       \@tempwattrue
3165       \loop\if@tempwa
3166         \read\@inputcheck to\bxjs@tmpa
3167         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil

```

```

3168         \ifx\bxjs@tmpa\relax\else
3169         \global\let\bxjs@g@tmpa\bxjs@tmpa
3170         \@tempswafalse
3171         \fi
3172         \ifeof\@inputcheck \@tempswafalse \fi
3173     \repeat
3174 \fi
3175 }\endgroup
3176 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3177 }
3178 \@onlypreamble\bxjs@get@ke@a
3179 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3180     \ifx$#1$\def\bxjs@tmpa{#2}%
3181     \else \let\bxjs@tmpa\relax
3182     \fi}

```

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

```

3183 \newcommand*\jachar[1]{%
3184     \begingroup
3185         \jsLetHeadChar\bxjs@tmpa{#1}%
3186         \ifx\bxjs@tmpa\relax
3187             \ClassWarningNoLine\bxjs@clsname
3188             {Illegal argument given to \string\jachar}%
3189         \else
3190             \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3191         \fi
3192     \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3193 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3194 \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。

```

3195 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3196 \@onlypreamble\bxjs@fix@hyperref@unicode
3197 \def\bxjs@fix@hyperref@unicode#1{%
3198     \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3199     \@namedef{KV@Hyp@bxjs/hook}##1{%
3200         \KV@Hyp@unicode{##1}%
3201         \def\KV@Hyp@unicode####1{%
3202             \expandafter\ifx\csname if##1\expandafter\endcsname
3203             \csname if####1\endcsname\else

```

```

3204         \ClassWarningNoLine\bxjs@clsname
3205         {Blcoked hyperref option 'unicode=###1'}%
3206     \fi
3207 }%
3208 }%
3209 }

```

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

```

3210 \onlypreamble\bxjs@urgent@special
3211 \def\bxjs@urgent@special#1{%
3212     \AtBeginDvi{\special{#1}}%
3213     \AtBeginDocument{%
3214         \@ifpackageloaded{atbegshi}{%
3215             \begingroup
3216                 \toks\z@\special{#1}}%
3217                 \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3218                 \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3219             \endgroup
3220         }{}%
3221     }%
3222 }

```

C.2 pTeX 用設定

```

3223 \if j\jsEngine

```

■ 共通命令の実装

```

3224 \def\bxjs@apply@kanjiskip{%
3225     \kanjiskip\@tempkipa}
3226 \def\bxjs@apply@xkanjiskip{%
3227     \xkanjiskip\@tempkipa}

```

`\jaJaChar` のサブマクロ。

```

3228 \def\bxjs@jachar#1{%
3229     \bxjs@jachar@a#1...\@nil}
3230 \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。

```

3231     \ifx.#2#1%

```

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

```

3232     \else\ifx.#3%
3233         \@tempcnta`#1 \multiply\@tempcnta64
3234         \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3235         \bxjs@jachar@b
3236     \else\ifx.#4%
3237         \@tempcnta`#1 \multiply\@tempcnta64
3238         \advance\@tempcnta`#2 \multiply\@tempcnta64

```



```

3239 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3240 \bxjs@jachar@b
3241 \else
3242 \@tempcnta`#1 \multiply\@tempcnta64
3243 \advance\@tempcnta`#2 \multiply\@tempcnta64
3244 \advance\@tempcnta`#3 \multiply\@tempcnta64
3245 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3246 \bxjs@jachar@b
3247 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3248 \ifjsWithupTeX
3249 \def\bxjs@jachar@b{\kchar\@tempcnta}
3250 \else
3251 \def\bxjs@jachar@b{%
3252 \ifx\bxUInt\@undefined\else
3253 \bxUInt{\@tempcnta}%
3254 \fi}
3255 \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。

```

3256 \let\bxjs@tmpa\jsJaFont
3257 \ifx\bxjs@tmpa\bxjs@@auto
3258 \let\bxjs@tmpa\@empty
3259 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3260 \def\bxjs@tmpa{noembed}
3261 \fi\fi
3262 \ifx\jsJaFont\@empty\else
3263 \edef\bxjs@nxt{%
3264 \noexpand\RequirePackage[\jsJaFont]
3265 {pxchfon}[2010/05/12]}% v0.5
3266 \bxjs@nxt
3267 \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……。)

```

3268 \begingroup
3269 \global\let\@gtempa\relax
3270 \catcode`\|=0 \catcode`\|=12
3271 |def|bxjs@check#1|@nil{%
3272 |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3273 |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3274 |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%

```

```

3275 |catcode`\|=0 \catcode`\|=12
3276 \def\bxjs@check@b#1keyval#2\@nnil{%
3277   \ifx$#2$\else
3278     \xdef\@gtempa{%
3279       \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3280     \fi}
3281 \@firstofone{%
3282   \catcode10=12 \endlinechar\m@ne
3283   \let\do\@makeother \dospecials \catcode32=10
3284   \openin\@inputcheck=otf.sty\relax
3285   \@tempswatrue
3286   \loop\if@tempswa
3287     \ifeof\@inputcheck \@tempswafalse \fi
3288     \if@tempswa
3289       \read\@inputcheck to\bxjs@line
3290       \expandafter\bxjs@check\bxjs@line\@nil
3291     \fi
3292   \repeat
3293   \closein\@inputcheck
3294 \endgroup}
3295 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3296 \bxjs@fix@hyperref@unicode{false}

tounicode special 命令を出力する。

3297 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3298   \else\ifjsWithpTeXng T\else F\fi\fi T%
3299   \ifnum\jis"2121="A1A1 %euc
3300     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3301   \else\ifnum\jis"2121="8140 %sjis
3302     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3303   \else\ifnum\jis"2121="3000 %uptex
3304     \ifbxjs@bigcode
3305       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3306       \PassOptionsToPackage{bigcode}{pxjahyper}
3307     \else
3308       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3309     \fi
3310   \fi\fi\fi
3311   \let\bxToUnicodeSpecialDone=t
3312 \fi

```

■microtype 対策

```

3313 \@namedef{ver@microtype.sty}{2000/01/01}
3314 \newcommand*\UseMicrotypeSet[2][{}]{

```

C.3 pdfTeX 用設定：CJK + bxcjkatype

```
3315 \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。

```
3316 \bxjs@adjust@jafont{f}
3317 \edef\bxjs@nxt{%
3318   \noexpand\RequirePackage[%
3319     \ifx\bxjs@tmpa\@empty\else \bxjs@tmpa,\fi
3320     whole,autotilde]{bxcjkatype}[2013/10/15]}% v0.2c
3321 \bxjs@nxt
3322 \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。

```
3323 \PassOptionsToPackage{unicode}{hyperref}
```

\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。

```
3324 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3325 \begingroup
3326   \CJK@input{UTF8.bdg}
3327 \endgroup
3328 \g@addto@macro\pdfstringdefPreHook{%
3329   \@nameuse{CJK@UTF8Binding}%
3330 }
3331 \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。

3332 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3333 \g@addto@macro\pdfstringdefPreHook{%
3334   \ifx~\bxjs@@CJKtilde
3335     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3336     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3337     \let~\@empty
3338   \fi
3339 }
3340 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3341 \def\bxjs@@tildecmd{~}
3342 \def\bxjs@LetUnexpandableSpace#1{%
3343   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
```

```

3344 \bxjs@org@LetUnexpandableSpace#1%
3345 \fi}
3346 \fi

```

■共通命令の実装

```

3347 \newskip\jsKanjiSkip
3348 \newskip\jsXKanjiSkip
3349 \ifx\CJKecglue\@undefined
3350 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3351 \fi
3352 \let\autospacing\bxjs@enable@kanjiskip
3353 \let\noautospacing\bxjs@disable@kanjiskip
3354 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3355 \def\bxjs@apply@kanjiskip{%
3356 \jsKanjiSkip\@tempskipa
3357 \let\CJKglue\bxjs@CJKglue}
3358 \let\autoxspacing\bxjs@enable@xkanjiskip
3359 \let\noautoxspacing\bxjs@disable@xkanjiskip
3360 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3361 \def\bxjs@apply@xkanjiskip{%
3362 \jsXKanjiSkip\@tempskipa
3363 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

3364 \def\bxjs@jachar#1{%
3365 \CJKforced{#1}}

```

C.4 Xe_{La}TeX 用設定：xeCJK + zxjatype

```

3366 \else\if x\jsEngine

```

■zxjatype パッケージの読み込み スケール値 (\jsScale) の反映は zxjatype の側で行われる。

```

3367 \RequirePackage{zxjatype}
3368 \PassOptionsToPackage{no-math}{fontspec}%!
3369 \PassOptionsToPackage{xetex}{graphicx}%!
3370 \PassOptionsToPackage{xetex}{graphics}%!
3371 \ifx\zxJaFamilyName\@undefined
3372 \ClassError\bxjs@clsname
3373 {xeCJK or zxjatype is too old}\@ehc
3374 \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。

```

3375 \bxjs@adjust@jafont{f}
3376 \ifx\bxjs@tmpa\@empty
3377 \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3378 \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3379 \else

```

```

3380 \edef\bxjs@nxt{%
3381     \noexpand\RequirePackage[\bxjs@tmpa]%
3382         {zxjafont}[2013/01/28]}% v0.2a
3383 \bxjs@nxt
3384 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X_ƎT_ƎX 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_ƎT_ƎX 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_ƎT_ƎX 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。

```

3385 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3386     \PassOptionsToPackage{unicode}{hyperref}
3387 \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。

```

3388 \let\jsInhibitGlueAtParTop\@empty

```

■共通命令の実装

```

3389 \newskip\jsKanjiSkip
3390 \newskip\jsXKanjiSkip
3391 \ifx\CJKecglue\@undefined
3392     \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3393 \fi
3394 \let\autospaceing\bxjs@enable@kanjiskip
3395 \let\noautospaceing\bxjs@disable@kanjiskip
3396 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3397 \def\bxjs@apply@kanjiskip{%
3398     \jsKanjiSkip\@tempskipa
3399     \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3400 \let\autoxspaceing\bxjs@enable@xkanjiskip
3401 \let\noautoxspaceing\bxjs@disable@xkanjiskip
3402 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3403 \def\bxjs@apply@xkanjiskip{%
3404     \jsXKanjiSkip\@tempskipa
3405     \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。

```
3406 \ifx\mcfamily\@undefined
3407   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3408   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3409 \fi
```

`\jachar` のサブマクロの実装。

```
3410 \def\bxjs@jachar#1{%
3411   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3412   #1}
```

C.5 LuaTeX 用設定：LuaTeX-ja

```
3413 \else\if 1\jsEngine
```

■LuaTeX-ja パッケージの読込 `luatexja` とともに `luatexja-fontspec` パッケージを読み込む。

`luatexja` は自前の `\zw` (これは実際の現在和文フォントに基づく値を返す) を定義するので、`\zw` の定義を消しておく。なお、レイアウト定義の「全角幅」は「規定」に基づく `\jsZw` であることに注意が必要。

```
3414 \let\zw\@undefined
3415 \RequirePackage{luatexja}
3416 \RequirePackage{luatexja-fontspec}
3417 \PassOptionsToPackage{pdftex}{graphicx}%!
3418 \PassOptionsToPackage{pdftex}{graphics}%!
```

■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？

```
3419 \ExplSyntaxOn
3420 \fp_gset:Nn \g_ltx_fonts_spec_scale_fp { \jsScale }
3421 \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)。

```
3422 \bxjs@adjust@jafont{t}
3423 \ifx\bxjs@tmpa\bxjs@noEmbed
3424   \def\bxjs@tmpa{noembed}
3425 \fi
3426 \ifx\bxjs@tmpa\@empty
3427   \defaultfontfeatures{ Kerning=Off }
3428   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3429   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3430 \else
3431   \edef\bxjs@nxt{%
3432     \noexpand\RequirePackage[\bxjs@tmpa]
```

```

3433     {\luatexja-preset}}}%
3434     \bxjs@nxt
3435 \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。

```

3436 \DeclareRobustCommand\rmfamily
3437   {\not@math@alphabet\rmfamily\mathrm
3438     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3439 \DeclareRobustCommand\sffamily
3440   {\not@math@alphabet\sffamily\mathsf
3441     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3442 \DeclareRobustCommand\ttfamily
3443   {\not@math@alphabet\ttfamily\mathtt
3444     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3445 \AtBeginDocument{%
3446   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3447   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}}%
3448 \bxjs@if@sf@default{%
3449   \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

3450 % 次の3つは既定値の通り
3451 %\ltjsetparameter{prebreakpenalty={` ,10000}}
3452 %\ltjsetparameter{postbreakpenalty={` “,10000}}
3453 %\ltjsetparameter{prebreakpenalty={` ”,10000}}
3454 \ltjsetparameter{jaxspmode={` ! ,1}}
3455 \ltjsetparameter{jaxspmode={` 〒,2}}
3456 \ltjsetparameter{alxspmode={` +,3}}
3457 \ltjsetparameter{alxspmode={` \%,3}}

```

■段落頭でのグルー挿入禁止

```

3458 \protected\def\@inhibitglue{%
3459   \directlua{%
3460     luatexja.jfmglue.create_beginpar_node()}}
3461 \let\bxjs@ltj@inhibitglue\@inhibitglue
3462 \let\@inhibitglue\undefined

```

■hyperref 対策 unicode にするべき。

```

3463 \bxjs@fix@hyperref@unicode{true}

```

■共通命令の実装

```

3464 \protected\def\autospacing{%
3465   \ltjsetparameter{autospacing=true}}
3466 \protected\def\noautospacing{%
3467   \ltjsetparameter{autospacing=false}}
3468 \protected\def\autoxspacing{%
3469   \ltjsetparameter{autoxspacing=true}}

```

```

3470 \protected\def\noautoxspacing{%
3471   \ltjsetparameter{autoxspacing=false}}
3472 \def\bxjs@apply@kanjiskip{%
3473   \ltjsetparameter{kanjiskip={\@tempskipa}}}}
3474 \def\bxjs@apply@xkanjiskip{%
3475   \ltjsetparameter{xkanjiskip={\@tempskipa}}}}

\jachar のサブマクロの実装。
3476 \def\bxjs@jachar#1{%
3477   \ltjjachar`#1\relax}

```

C.6 共通処理 (2)

```

3478 \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 3479 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3480 \DeclareJaTextFontCommand{\textgt}{\gtfamily}

```

■和文・和欧文間空白の初期値

```

3481 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3482 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3483 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3484 \fi

```

以上で終わり。

```

3485 %</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。

```

3486 %<*modern>
3487 \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。

```

3488 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3489 \def\encodingdefault{T1}%
3490 \input{t1enc.def}%
3491 \fontencoding\encodingdefault\selectfont
3492 \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。

```
3493 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
3494 \renewcommand{\rmdefault}{\lmr}
3495 \renewcommand{\sfdefault}{\lms}
3496 \renewcommand{\ttdefault}{\lmtt}
3497 \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。

```
3498 \DeclareFontShape{OMX}{cmex}{m}{n}{%
3499   <-7.5>cmex7<7.5-8.5>cmex8%
3500   <8.5-9.5>cmex9<9.5->cmex10}{}%
3501 \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。
3502 \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。

```
3503 \ifx\@IncludeInRelease\@undefined
3504 \RequirePackage{fixltx2e}
3505 \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。

```
3506 \RequirePackage{bxjsjckcat}
```

D.4 完了

おしまい。

```
3507 %</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。

```
3508 %<*pandoc>
3509 \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。

```
3510 \@onlypreamble\bxjs@set@dupload@proc
3511 \def\bxjs@set@dupload@proc#1{%
3512   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
3513 \@onlypreamble\bxjs@set@dupload@proc@a
3514 \def\bxjs@set@dupload@proc@a#1{%
3515   \@onlypreamble#1\def#1##1}
```

`\@if@options` `\@if@options` の再定義。

```
3516 \@onlypreamble\bxjs@org@if@options
3517 \let\bxjs@org@if@options\@if@options
3518 \newif\ifbxjs@dlp
3519 \def\@if@options#1#2#3{%
3520   \bxjs@dlpfalse
3521   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
3522   \ifx\bxjs@tmpa\bxjs@tmpb
3523     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
3524     \bxjs@dlptrue \fi
3525   \fi
3526   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
3527   \else \expandafter\bxjs@org@if@options
3528   \fi {#1}{#2}{#3}}
3529 \AtBeginDocument{%
3530   \let\@if@options\bxjs@org@if@options}
3531 \@onlypreamble\bxjs@do@dupload@proc
3532 \def\bxjs@do@dupload@proc#1#2#3{%
3533   \csname bxjs@dlp/#2.#1\endcsname{#3}%
3534   \@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。

```
3535 \def\bxjs@mark@as@loaded#1{%
3536   \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3537 \def\bxjs@mal@a#1#2{%
3538   \ifx#1\relax
3539     \def#1{2001/01/01}%
3540     \ClassInfo\bxjs@clsname
3541     {File '#2' marked as loaded\@gobble}%
3542   \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。

```
3543 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0
```

Polyglossia について。

```
3544 \bxjs@mark@as@loaded{polyglossia.sty}
3545 \bxjs@set@dupload@proc{polyglossia.sty}{%
3546   \ClassWarning\bxjs@clsname
3547   {Loading of polyglossia is blocked}}
3548 \ifx\setmainlanguage\@undefined
3549 \newcommand*\setmainlanguage[2][]{%
3550 \newcommand*\setotherlanguage[2][]{%
3551   \ifcat_#2_\else
3552     \expandafter\let\csname #2\endcsname\@empty
3553     \expandafter\let\csname end#2\endcsname\@empty
3554     \expandafter\let\csname text#2\endcsname\@firstofone
3555   \fi}
3556 \newcommand*\setotherlanguages[2][]{%
3557   \@for\bxjs@tmpa:={#2}\do{%
3558     \setotherlanguage{\bxjs@tmpa}}}
3559 \fi
3560 \else
```

Babel について。

```
3561 \bxjs@mark@as@loaded{babel.sty}
3562 \bxjs@set@dupload@proc{babel.sty}{%
3563   \ClassWarning\bxjs@clsname
3564   {Loading of babel is blocked}}
3565 \let\foreignlanguage\@secondoftwo
3566 \let\otherlanguage\@gobble
3567 \let\endotherlanguage\@empty
3568 \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。

```
3569 \bxjs@set@dupload@proc{geometry.sty}{%
3570   \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 `\setmainfont` の指定にまわす。

```
3571 \if 1\jsEngine
3572   \bxjs@mark@as@loaded{xeCJK.sty}
3573   \providecommand*\setCJKmainfont{\setmainfont}
3574 \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 \LaTeX ではこれで警告が出る。これを抑止する。

\LaTeX カーネルが新しい場合は `fixltx2e` を読込済にする。

```
3575 \ifx\@IncludeInRelease\@undefined\else
3576   \bxjs@mark@as@loaded{fixltx2e.sty}
3577 \fi
```

E.6 cmap パッケージ

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

```
3578 \if j\jsEngine
3579   \bxjs@mark@as@loaded{cmap.sty}
3580 \fi
```

E.7 microtype パッケージ

警告が多すぎなので消す。

```
3581 \PassOptionsToPackage{verbose=silent}{microtype}
```

E.8 完了

おしまい。

```
3582 %</pandoc>
```

和文ドライバ実装はここまで。

```
3583 %</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。

3584 %<*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

3585 %<*compat>

3586 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

3587 \let\bxac@engine=n

3588 \def\bxac@do#1#2{%

3589 \edef\bxac@tmpa{\string#1}%

3590 \edef\bxac@tmpb{\meaning#1}%

3591 \ifx\bxac@tmpa\bxac@tmpb #2\fi}

3592 \bxac@do\XeTeXversion{\let\bxac@engine=x}

3593 \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。

3594 \ifx\jsAtEndOfClass\@undefined

3595 \let\bxac@delayed@if@bxjs\@firstofone

3596 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass

3597 \fi

\ImposeOldLuaTeXBehavior ムニャムニャ。

\RevokeOldLuaTeXBehavior 3598 \newif\ifbxac@in@old@behavior

3599 \let\ImposeOldLuaTeXBehavior\relax

3600 \let\RevokeOldLuaTeXBehavior\relax

G.2 X_YTEX 部分

3601 \ifx x\bxac@engine

X_YTEX 文字クラスのムニャムニャ。

3602 \@onlypreamble\bxac@adjust@charclass

3603 \bxac@delayed@if@bxjs{%

3604 \@ifpackageloaded{xeCJK}{-}{%else

3605 \ifx\xe@alloc@intercharclass\@undefined\else

3606 \ifnum\xe@alloc@intercharclass=\z@

3607 \PackageInfo\bxac@pkgname

3608 {Setting up interchar class for CJK...\@gobble}%

3609 \InputIfFileExists{load-unicode-xetex-classes.tex}{%

3610 \xe@alloc@intercharclass=3

```

3611     }{%else
3612         \PackageWarning\bxac@pkgname
3613         {Cannot find file 'load-unicode-xetex-classes.tex'%
3614         \@gobble}%
3615     }%
3616 \fi\fi
3617 \ifnum\XeTeXcharclass"3041=\z@
3618     \PackageInfo\bxac@pkgname
3619     {Adjusting interchar class for CJK...\@gobble}%
3620     \@for\bxac@x={%
3621         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3622         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3623         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3624         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3625         31FF%
3626     }\do{\XeTeXcharclass"\bxac@x=\@ne}%
3627 \fi
3628 }%
3629 }

```

以上。

```

3630 \fi

```

G.3 LuaTeX 部分

```

3631 \ifx l\bxac@engine

```

ムニヤムニヤ。

```

3632 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
3633     \chardef\pdfTeXversion=200
3634     \def\pdfTeXrevision{0}
3635     \let\pdfTeXbanner\luatexbanner
3636 \fi\fi

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 3637 \begingroup\expandafter\expandafter\expandafter\endgroup
3638 \expandafter\ifx\csgname outputmode\endcsname\relax\else
3639 \def\bxac@ob@list{%
3640     \do{\let}\pdfoutput{\outputmode}%
3641     \do{\let}\pdfpagewidth{\pagewidth}%
3642     \do{\let}\pdfpageheight{\pageheight}%
3643     \do{\protected\edef}\pdfhorigin{{\pdfvariable horigin}}%
3644     \do{\protected\edef}\pdfvorigin{{\pdfvariable vorigin}}}
3645 \def\bxac@ob@do#1#2{\begingroup
3646     \expandafter\bxac@ob@do@a\csgname bxac@\string#2\endcsname{#1}#2}
3647 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3648     \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3649     \else \let#3#1\relax \let#1\@undefined
3650     \fi}
3651 \protected\def\ImposeOldLuaTeXBehavior{%
3652     \unless\ifbxac@in@old@behavior

```

```

3653 \bxac@in@old@behaviortrue
3654 \let\do\bxac@ob@do \bxac@ob@list
3655 \fi}
3656 \protected\def\RevokeOldLuaTeXBehavior{%
3657 \ifbxac@in@old@behavior
3658 \bxac@in@old@behaviorfalse
3659 \let\do\bxac@ob@do \bxac@ob@list
3660 \fi}
3661 \fi

```

漢字および完成形ハングルのカテゴリコードのムニャムニャ。

```

3662 \ifnum\luatexversion>64 \directlua{
3663   local function range(cs, ce, cc, ff)
3664     if ff or not tex.getcatcode(cs) == cc then
3665       local setcc = tex.setcatcode
3666       for c = cs, ce do setcc(c, cc) end
3667     end
3668   end
3669   range(0x3400, 0x4DB5, 11, false)
3670   range(0x4DB5, 0x4DBF, 11, true)
3671   range(0x4E00, 0x9FCC, 11, false)
3672   range(0x9FCD, 0x9FFF, 11, true)
3673   range(0xAC00, 0xD7A3, 11, false)
3674   range(0x20000, 0x2A6D6, 11, false)
3675   range(0x2A6D7, 0x2A6FF, 11, true)
3676   range(0x2A700, 0x2B734, 11, false)
3677   range(0x2B735, 0x2B73F, 11, true)
3678   range(0x2B740, 0x2B81D, 11, false)
3679   range(0x2B81E, 0x2B81F, 11, true)
3680   range(0x2B820, 0x2CEA1, 11, false)
3681   range(0x2CEA2, 0x2FFFD, 11, true)
3682 } \fi

```

以上。

```
3683 \fi
```

G.4 完了

おしまい。

```
3684 %</compat>
```

付録 H 補助パッケージ : bxjscjkcat

modern ドライバ用の和文カテゴリを適用する。

H.1 準備

```

3685 %<*cjkcat>
3686 \def\bxjx@pkgname{bxjscjkcat}
3687 \newcount\bxjx@canta

\bxjx@engine エンジンの種別。

3688 \let\bxjx@engine=n
3689 \def\bxjx@do#1#2{%
3690   \edef\bxjx@tmpa{\string#1}%
3691   \edef\bxjx@tmpb{\meaning#1}%
3692   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3693 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3694 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3695 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3696 \bxjx@do\pdfTeXversion{\let\bxjx@engine=p}
3697 \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。

```

3698 \def\bxjx@do#1#2{%
3699   \if#1\bxjx@engine
3700     \@ifpackageloaded{#2}{\}%else
3701     \PackageError\bxjx@pkgname
3702       {Package '#2' must be loaded}%
3703       {Package loading is aborted.\MessageBreak\@ehc}%
3704     \endinput}
3705   \fi}
3706 \bxjx@do{p}{bxcjkatype}
3707 \bxjx@do{x}{xeCJK}
3708 \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。

```

3709 \ifx\TextOrMath\undefined
3710   \RequirePackage{fixltx2e}
3711 \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。

```

3712 \if u\bxjx@engine
3713   \@for\bxjx@x:={%
3714     0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3715     0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3716     0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3717     1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3718     1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3719     1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%

```



```

3720 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3721 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
3722 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3723 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABCO,%
3724 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3725 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3726 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3727 10480,10500,10530,10600,10800,10840,10860,10880,%
3728 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
3729 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3730 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3731 11280,112B0,11300,11480,11580,11600,11680,11700,%
3732 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3733 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3734 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3735 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
3736 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3737 }\do{\kcatcode"\bxjx\x=15 }
3738 \fi

```

H.3 ギリシャ・キリル文字の扱い

※ここで「ギリシャ・キリル文字」は Unicode と JIS X 0213 に共通して含まれるもののみを指すことにする。

\bxjx@grkcyr@list 対象のギリシャ・キリル文字に関するデータ。

```

3739 \def\bxjx@grkcyr@list{%
3740 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
3741 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
3742 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
3743 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
3744 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
3745 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
3746 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
3747 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
3748 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
3749 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
3750 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
3751 \do{039C}{LGR}{\textMu}{M}%            % GR. C. L. MU
3752 \do{039D}{LGR}{\textNu}{N}%            % GR. C. L. NU
3753 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
3754 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
3755 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
3756 \do{03A1}{LGR}{\textRho}{P}%           % GR. C. L. RHO
3757 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
3758 \do{03A4}{LGR}{\textTau}{T}%           % GR. C. L. TAU
3759 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
3760 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI

```

3761	\do{03A7}{LGR}{\textChi}{X}%	% GR. C. L. CHI
3762	\do{03A8}{LGR}{\textPsi}{Psi}%	% GR. C. L. PSI
3763	\do{03A9}{LGR}{\textOmega}{\Omega}%	% GR. C. L. OMEGA
3764	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
3765	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
3766	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
3767	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
3768	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
3769	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
3770	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
3771	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
3772	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
3773	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
3774	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
3775	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
3776	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
3777	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
3778	\do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
3779	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
3780	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
3781	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
3782	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
3783	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
3784	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
3785	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
3786	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
3787	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
3788	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
3789	\do{0401}{T2A}{\CYRYO}{}	% CY. C. L. IO
3790	\do{0410}{T2A}{\CYRA}{}	% CY. C. L. A
3791	\do{0411}{T2A}{\CYRB}{}	% CY. C. L. BE
3792	\do{0412}{T2A}{\CYRV}{}	% CY. C. L. VE
3793	\do{0413}{T2A}{\CYRG}{}	% CY. C. L. GHE
3794	\do{0414}{T2A}{\CYRD}{}	% CY. C. L. DE
3795	\do{0415}{T2A}{\CYRE}{}	% CY. C. L. IE
3796	\do{0416}{T2A}{\CYRZH}{}	% CY. C. L. ZHE
3797	\do{0417}{T2A}{\CYRZ}{}	% CY. C. L. ZE
3798	\do{0418}{T2A}{\CYRI}{}	% CY. C. L. I
3799	\do{0419}{T2A}{\CYRISHRT}{}	% CY. C. L. SHORT I
3800	\do{041A}{T2A}{\CYRK}{}	% CY. C. L. KA
3801	\do{041B}{T2A}{\CYRL}{}	% CY. C. L. EL
3802	\do{041C}{T2A}{\CYRM}{}	% CY. C. L. EM
3803	\do{041D}{T2A}{\CYRN}{}	% CY. C. L. EN
3804	\do{041E}{T2A}{\CYRO}{}	% CY. C. L. O
3805	\do{041F}{T2A}{\CYRP}{}	% CY. C. L. PE
3806	\do{0420}{T2A}{\CYRR}{}	% CY. C. L. ER
3807	\do{0421}{T2A}{\CYRS}{}	% CY. C. L. ES
3808	\do{0422}{T2A}{\CYRT}{}	% CY. C. L. TE
3809	\do{0423}{T2A}{\CYRU}{}	% CY. C. L. U

3810	\do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
3811	\do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
3812	\do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
3813	\do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
3814	\do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
3815	\do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
3816	\do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
3817	\do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
3818	\do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
3819	\do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
3820	\do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
3821	\do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
3822	\do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
3823	\do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
3824	\do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
3825	\do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
3826	\do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
3827	\do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
3828	\do{0436}{T2A}{\cyrz}{}%	% CY. S. L. ZHE
3829	\do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
3830	\do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
3831	\do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
3832	\do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
3833	\do{043B}{T2A}{\cyr1}{}%	% CY. S. L. EL
3834	\do{043C}{T2A}{\cyr1}{}%	% CY. S. L. EM
3835	\do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
3836	\do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
3837	\do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
3838	\do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
3839	\do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES
3840	\do{0442}{T2A}{\cyrt}{}%	% CY. S. L. TE
3841	\do{0443}{T2A}{\cyru}{}%	% CY. S. L. U
3842	\do{0444}{T2A}{\cyrf}{}%	% CY. S. L. EF
3843	\do{0445}{T2A}{\cyrh}{}%	% CY. S. L. HA
3844	\do{0446}{T2A}{\cyrc}{}%	% CY. S. L. TSE
3845	\do{0447}{T2A}{\cyrch}{}%	% CY. S. L. CHE
3846	\do{0448}{T2A}{\cyrsh}{}%	% CY. S. L. SHA
3847	\do{0449}{T2A}{\cyrshch}{}%	% CY. S. L. SHCHA
3848	\do{044A}{T2A}{\cyrhrdsn}{}%	% CY. S. L. HARD SIGN
3849	\do{044B}{T2A}{\cyrrery}{}%	% CY. S. L. YERU
3850	\do{044C}{T2A}{\cyrsftsn}{}%	% CY. S. L. SOFT SIGN
3851	\do{044D}{T2A}{\cyrrerev}{}%	% CY. S. L. E
3852	\do{044E}{T2A}{\cyryu}{}%	% CY. S. L. YU
3853	\do{044F}{T2A}{\cyrya}{}%	% CY. S. L. YA
3854	\do{0451}{T2A}{\cyryo}{}%	% CY. S. L. IO
3855	\do{00A7}{TS1}{\textsection}{\mathsection}%	SECTION SYMBOL
3856	\do{00A8}{TS1}{\textasciidieresis}{}%	% DIAERESIS
3857	\do{00B0}{TS1}{\textdegree}{\mathdegree}%	% DEGREE SIGN
3858	\do{00B1}{TS1}{\textpm}{\pm}%	% PLUS-MINUS SIGN

```

3859 \do{00B4}{TS1}{\textasciicute}{}%          % ACUTE ACCENT
3860 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
3861 \do{00D7}{TS1}{\texttimes}{\times}%          % MULTIPLICATION SIGN
3862 \do{00F7}{TS1}{\textdiv}{\div}%             % DIVISION SIGN
3863 }

3864 \providecommand*\mathdegree{{}^{\circ}}

\ifbxjx@gcc@cjk 〔スイッチ〕ギリシャ・キリル文字を和文扱いにするか。
3865 \newif\ifbxjx@gcc@cjk

\greekasCJK   ギリシャ・キリル文字を和文扱いにする。

\nogreekasCJK ギリシャ・キリル文字を欧文扱いにする。

3866 \newcommand*\greekasCJK{%
3867   \bxjx@gcc@cjktrue}
3868 \newcommand*\nogreekasCJK{%
3869   \bxjx@gcc@cjkfalse}

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

3870 \def\bxjx@do#1\relax{%
3871   \def\bxjx@fake@grk##1##2{%
3872     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
3873   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
3874     \ifx\\##1\\%
3875       \bxjx@canta##4\divide\bxjx@canta\@cclvi
3876       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
3877       \mathchar\bxjx@canta
3878     \else ##3\fi}
3879 }\expandafter\bxjx@do\string\mathchar\relax

■pdfLaTeX・upLaTeX の場合

3880 \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。

3881 \@ifpackageloaded{inputenc}{}{%else
3882   \RequirePackage[utf8]{inputenc}}
3883 \def\bxjx@tmpa{utf8}
3884 \ifx\bxjx@tmpa\inputencdoingname
3885   \PackageWarningNoLine\bxjx@pkgname
3886     {Input encoding changed to utf8}%
3887   \inputencoding{utf8}%
3888 \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。

3889 \if u\bxjx@engine
3890 \kcatcode"0370=15
3891 \kcatcode"0400=15
3892 \kcatcode"0500=15
3893 \fi

```

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

```

3894 \def\do#1{%
3895   \@tempcnta="#1\relax
3896   \@tempcntb=\@tempcnta \divide\@tempcntb256
3897   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
3898   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
3899 \def\bxjx@do@a#1#2#3#4#5{%
3900   \ifx\#5\%
3901     \def\bxjx@tmpa{\@inmathwarn#4}%
3902   \else\ifcat A\noexpand#5%
3903     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3904       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
3905   \else \def\bxjx@tmpa{#5}%
3906   \fi\fi
3907   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
3908   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
3909 \if u\bxjx@engine
3910 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
3911 \def\bxjx@do@b#1#2#3#4#5{%
3912   \kchardef#1=\@tempcnta
3913   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
3914   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
3915 \else\if p\bxjx@engine
3916 \def\bxjx@do@b#1#2#3#4#5{%
3917   \mathchardef#1=\@tempcnta
3918   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
3919   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
3920 \fi\fi
3921 \bxjx@grkcyr@list
3922 \let\bxjx@do@a\undefined
3923 \let\bxjx@do@b\undefined

```

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

```

3924 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
3925 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
3926 \@onlypreamble\bxjx@DeclareUnicodeCharacter
3927 \def\bxjx@DeclareUnicodeCharacter#1#2{%
3928   \count@="#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
3929   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
3930     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3931   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
3932     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3933   \else
3934     \wlog{ \space\space skipped defining Unicode char U+#1}%
3935   \fi\fi}

```

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

```
3936 \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。

```
3937 \ifbxjx@gcc@cj k #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。

```
3938 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
```

```
3939 \else \UseTextSymbol{#2}{#3}%
```

```
3940 \fi\fi}
```

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

```
3941 \begingroup
```

```
3942 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
```

```
3943 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
```

```
3944 \noexpand\bxjx@swap@DUC@cmd
```

```
3945 \the\toks@
```

```
3946 \noexpand\bxjx@swap@DUC@cmd}}}
```

```
3947 \endgroup\next
```

```
3948 \def\bxjx@swap@DUC@cmd{%
```

```
3949 \let\bxjx@tmpa\DeclareUnicodeCharacter
```

```
3950 \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
```

```
3951 \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}
```

以上。

■ Xe_{La}TeX・Lua_{La}TeX の場合

```
3952 \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。

```
3953 \def\do#1{%
```

```
3954 \bxjx@cmta="#1\relax
```

```
3955 \begingroup
```

```
3956 \lccode`~=\bxjx@cmta
```

```
3957 \lowercase{\endgroup
```

```
3958 \bxjx@do@a{~}}{#1}}
```

```
3959 \def\bxjx@do@a#1#2#3#4#5{%
```

```
3960 \ifx\\#5\\let\bxjx@tmpa\relax
```

```
3961 \else\ifcat A\noexpand#5%
```

```
3962 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
```

```
3963 {\ifnum\uccode`#5=`#5\noexpand\pi\else\noexpand\pi\fi}}%
```

```
3964 \else \def\bxjx@tmpa{#5}%
```

```
3965 \fi\fi
```

```
3966 \ifx\bxjx@tmpa\relax\else
```

```
3967 \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
```

```
3968 \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。

```

3969 \mathchardef\bxjx@tmpa="119
3970 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
3971 \let\bxjx@do@a\undefined

```

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

```

3972 \if 1\bxjx@engine
3973   \protected\def\greekasCJK{%
3974     \bxjx@gcc@cjktrue
3975     \ltjsetparameter{jacharrange={+2, +8}}
3976   \protected\def\nogreekasCJK{%
3977     \bxjx@gcc@cjkfalse
3978     \ltjsetparameter{jacharrange={-2, -8}}
3979 \fi

```

X_YTeX における \(\text{no}\)greekasCJK の定義。

```

3980 \if x\bxjx@engine
3981   \protected\def\greekasCJK{%
3982     \bxjx@gcc@cjktrue
3983     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
3984     \bxjx@grkcyr@list}
3985   \protected\def\nogreekasCJK{%
3986     \bxjx@gcc@cjkfalse
3987     \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
3988     \bxjx@grkcyr@list}
3989 \fi

```

以上。

```

3990 \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。

```

3991 \nogreekasCJK

```

H.5 完了

おしまい。

```

3992 %</cjkat>

```

補助パッケージ実装はここまで。

```

3993 %</anc>

```