

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”)

v1.3 [2016/11/01]

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	31
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	48
7.1	表題	48
7.2	章・節	53
7.3	リスト環境	64
7.4	パラメータの設定	71
7.5	フロート	72
7.6	キャプション	74
8	フォントコマンド	75
9	相互参照	77

9.1	目次の類	77
9.2	参考文献	83
9.3	索引	84
9.4	脚注	86
10	段落の頭へのグルー挿入禁止	88
11	いろいろなロゴ	91
12	amsmath との衝突の回避	91
13	初期設定	92
付録 A	和文ドライバの仕様 ㊦	95
付録 B	和文ドライバ : minimal ㊦	96
B.1	補助マクロ	96
B.2	(u)pTeX 用の設定	98
B.3	pdfTeX 用の処理	102
B.4	X _Y TeX 用の処理	102
B.5	後処理 (エンジン共通)	103
付録 C	和文ドライバ : standard ㊦	105
C.1	共通処理 (1)	105
C.2	pTeX 用設定	110
C.3	pdfTeX 用設定 : CJK + bxcjkatype	113
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	114
C.5	LuaTeX 用設定 : LuaTeX-ja	116
C.6	共通処理 (2)	118
付録 D	和文ドライバ : modern ㊦	118
D.1	フォント設定	118
D.2	fixltx2e 読込	119
D.3	和文カテゴリコード	119
D.4	完了	119
付録 E	和文ドライバ : pandoc ㊦	119
E.1	dupload システム	119
E.2	lang 変数	120
E.3	geometry 変数	121
E.4	CJKmainfont 変数	121
E.5	fixltx2e パッケージ	122
E.6	cmap パッケージ	122

E.7	microtype パッケージ	122
E.8	完了	122
付録 F	補助パッケージ一覧 	122
付録 G	補助パッケージ : bxjscompat 	122
G.1	準備	123
G.2	X _Y TeX 部分	123
G.3	LuaTeX 部分	124
G.4	完了	125
付録 H	補助パッケージ : bxjscjkat 	125
H.1	準備	125
H.2	和文カテゴリコードの設定	126
H.3	ギリシャ・キリル文字の扱い	127
H.4	初期設定	133
H.5	完了	133

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

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

[2009-02-22] 田中琢爾氏による up_{La}TeX 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=bxjsar-
ticle、\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。

```

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。

[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。

`\@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。

```

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\@eqncr
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\@eqncr
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$\displaystyle\tabskip\z@skip{##}$\@eqnse
223 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
224 &\global\@eqcnt\tw@
225 $\displaystyle{##}$\hfil \tabskip\@centering
226 &\global\@eqcnt\thr@@ \hb@xt@\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}
```

■BXJS 特有のオプションの一覧

- エンジンオプション： `xelatex` 等。
- ドライバオプション： `dvipdfmx` 等。

- 複合設定オプション：pandoc 等。
- nopapersize：papersize（既定で有効）の否定。
- zw / nozw：\jsZw と等価な命令として \zw を定義する／しない。
- js / nojs：JS クラスを読込済として扱う／扱わない。
- precisetext / noprecisetext：Xe_{La}TeX の “generateactualtext” を有効／無効にする。
- simplejasetup / nosimplejasetup：Xe_{La}TeX の “linebreaklocale” を有効／無効にする。
- bigcode / nobigcode：upTeX で CMap として UTF8-UCS2 の代わりに UTF8-UTF16 を使う／使わない。
- oldfontcommands / nooldfontcommands：古い“二文字フォント命令”に対する警告を抑止する／しない。
- base=*<dimen>*：基底フォントサイズを直接指定する。（xxpt オプションの代用なので、既定値は 10pt である。）
- jbase=*<dimen>*：基底フォントサイズを“和文規準で”直接指定する。
- scale=*<real>*：和文フォントのスケールを表すマクロ \jsScale の値を設定する。もちろんこの値を何らかの方法で和文処理モジュールに渡さないと意味を成さない。既定値は 0.924715 (= 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 T_EX における「長さのスケール」の逆関数を求めるもの。例えば
\bxjs@invscale\dimX{1.3} は \dimX=1.3\dimX の逆の演算を行う。

※局所化の `\begingroup~\endgroup` について、以前は `\group~\egroup` を使っていたが、これだと数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\@cclvi
262     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
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 \LaTeX 用の既定テンプレートを用いて他形式から \LaTeX (および PDF) 形式に変換する用途に最適化した設定を与える。

```

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@@dvi pdfmx=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 XeΛTeX 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 XeΛTeX 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#2{\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 `autodetect-engine` であり、かつ実際のエンジンが (u)pL^AT_EX だった場合は、本来のエンジンオプションをグローバルオプションに加える。

```
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 pT_EX-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 \@tempwattrue
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_{TeX} (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 \TeX の場合は、高位バイトをアクティブ化しておく。(和文を含むマクロ定義を通用させるため。)

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

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

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

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

```

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@empt 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@empt 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@mpt\p@
710   \newdimen\jsc@mmm \jsc@mmm=1mm
711   \catcode`\?=9 % \p@? read as \p@
712 \else
713   \newdimen\jsc@mpt \jsc@mpt=\jsc@magscale \p@
714   \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
715   \catcode`\?=11 \let\p@?\jsc@mpt
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_w 非指定時は `\zw` を `\jsZw` と同義にする。

```

718 \newdimen\jsZw
719 \jsZw=10\jsc@mpt \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\f@size\jsc@ref@size}
746 \def\execute@size@function#1{%

```

```

747 \let\jsc@cref@size\f@size
748 \let\f@size\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

```

■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。

```

764 \chardef\bxjs@periodchar=`\ .
765 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcodes\bxjs@periodchar}
766 \def\@{\bxjs@SE{}}

```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

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

```

767 \def\bxjs@tmpa{\def\setfontsize##1##2##3{
768 \expandafter\bxjs@tmpa\expandafter{%
769   \setfontsize{#1}{#2}{#3}%
770 % 末尾にコードを追加
771 \expandafter\def\expandafter\size@update\expandafter{%
772   \size@update
773   \jsFontSizeChanged}%
774 }
```

`\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。

```

775 \newcommand*\jsFontSizeChanged{%
776   \jsZw=\f@size\p@
777   \jsZw=\jsScale \jsZw
778   \ifdim\parindent>\z@
779     \if@english \parindent=1em
780     \else       \parindent=1\jsZw
781   \fi
782   \fi\relax
783   \jsResetDimen}
```


`\jsResetDimen` ユーザ定義用のフック。

```
784 \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。

```
785 \ifjsc@mag
786   \let\jsc@setfontsize\@setfontsize
787 \else
788   \def\jsc@setfontsize#1#2#3{%
789     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
790 \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。

```
791 \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、

[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 (`\@currsz` 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。

```
792 \newif\ifnarrowbaselines
793 \if@english
794   \narrowbaselinestrue
795 \fi
796 \def\narrowbaselines{%
797   \narrowbaselinestrue
798   \skip0=\abovedisplayskip
799   \skip2=\abovedisplayshortskip
800   \skip4=\belowdisplayskip
801   \skip6=\belowdisplayshortskip
```

```

802 % 一時的に警告を無効化する
803 \let\bxjs@ltx@nomath\@nomath
804 \let\@nomath\@gobble
805 \@currsize\selectfont
806 \let\@nomath\bxjs@ltx@nomath
807 \abovedisplayskip=\skip0
808 \abovedisplayshortskip=\skip2
809 \belowdisplayskip=\skip4
810 \belowdisplayshortskip=\skip6\relax}
811 \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。

```

812 \def\bxjs@if@narrowbaselines{%
813   \ifnarrowbaselines\expandafter\@firstoftwo
814   \else \expandafter\@secondoftwo
815   \fi
816 }

```

`\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。

```

817 \renewcommand{\normalsize}{%
818   \bxjs@if@narrowbaselines{%
819     \jsc@setfontsize\normalsize\@xpt\@xiipt
820   }{%else
821     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
822   }%

```

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なりました。

```

823 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?

```

```

824 \abovedisplayskip \z@ \@plus3\p@?
825 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
826 \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。

```
827 \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。

```
828 \normalsize
```

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

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

```
\Cwd
```

`\Cvs` まず `\Cwd` 等の変数を定義する。

```

\Cht 829 \ifx\Cht\@undefined \newdimen\Cht \fi
\Cdp 830 \ifx\Cdp\@undefined \newdimen\Cdp \fi
831 \ifx\Cwd\@undefined \newdimen\Cwd \fi
832 \ifx\Cvs\@undefined \newdimen\Cvs \fi
833 \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。

```

834 \setlength\Cht{0.88\jsZw}
835 \setlength\Cdp{0.12\jsZw}
836 \setlength\Cwd{1\jsZw}
837 \setlength\Cvs{\baselineskip}
838 \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。

```

839 \newcommand{\small}{%
840   \bxjs@if@narrowbaselines{%
841     %<!kiyou>   \jsc@setfontsize\small\@ixpt{11}%
842     %<kiyou>   \jsc@setfontsize\small{8.8888}{11}%
843   }{%else
844     %<!kiyou>   \jsc@setfontsize\small\@ixpt{13}%
845     %<kiyou>   \jsc@setfontsize\small{8.8888}{13.2418}%
846   }%
847   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
848   \abovedisplayshortskip \z@ \@plus3\p@?

```

```

849 \belowdisplayskip \abovedisplayskip
850 \belowdisplayshortskip \belowdisplayskip
851 \def\@listi{\leftmargin\leftmargini
852         \topsep \z@
853         \parsep \z@
854         \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³, ここではゼロ (`\z@`) にしました。

```

855 \newcommand{\footnotesize}{%
856   \bxjs@if@narrowbaselines{%
857     %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
858     %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
859   }{%else
860     %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
861     %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
862   }%
863   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
864   \abovedisplayshortskip \z@ \@plus2\p@?
865   \belowdisplayskip \abovedisplayskip
866   \belowdisplayshortskip \belowdisplayskip
867   \def\@listi{\leftmargin\leftmargini
868         \topsep \z@
869         \parsep \z@
870         \itemsep \parsep}}

```

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

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 871 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
872 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge 873 \if@twocolumn
\HUGE 874 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
875 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
876 \else
877 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
878 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
879 \fi
880 %<!kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xivpt{21}}
881 %<kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE{12.222}{21}}
882 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xivpt{25}}
883 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
884 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
885 \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。

```
886 \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。『*LaTeX 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。

```
887 % \newcommand{\headfont}{\bfseries}
```

```
888 \newcommand{\headfont}{\sffamily}
```

```
889 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}
```

5 レイアウト

■二段組

`\columnsep` `\columnsep` は二段組のときの左右の段間の幅です。元は 10pt ですが，2zw にしました。

`\columnseprule` このスペースの中央に `\columnseprule` の幅の罫線が引かれます。

```
890 %!kiyou>\setlength\columnsep{2\Cwd}
```

```
891 %!kiyou>\setlength\columnsep{28truebp}
```

```
892 \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

`\normallineskip` ます。元は 0pt ですが 1pt に変更しました。`normal...` の付いた方は保存用です。

```
\lineskiplimit 893 \setlength\lineskip{1\p@?}
```

```
894 \setlength\normallineskip{1\p@?}
```

```
\normallineskiplimit 895 \setlength\lineskiplimit{1\p@?}
```

```
896 \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。

```
897 \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。

```
898 \setlength\parskip{0\p@}  
899 \if@slide  
900 \setlength\parindent{0\p@}  
901 \else  
902 \setlength\parindent{1\Cwd}  
903 \fi
```

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

```
\@highpenalty 904 \@lowpenalty 51  
905 \@medpenalty 151  
906 \@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。

```
907 % \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。

```
908 % \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。

```
909 \ifjsc@mag  
910 \mag=\bxjs@param@mag  
911 \fi  
912 \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。

```
913 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}  
914 \def\bxjs@read@b{%  
915 \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c  
916 \else \expandafter\bxjs@read@d \fi}  
917 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}  
918 \def\bxjs@read@d#1\@nil{}  
919 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
920 \edef\bxjs@layout@paper{%
921   \ifjsc@mag truedimen,\fi
922   \if@landscape landscape,\fi
923   \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。

```
924 %<*article|report>
925 \def\bxjs@layout{%
926   headheight=\topskip,footskip=0.03367\paperheight,%
927   headsep=\footskip-\topskip,includeheadfoot,%
928   hscale=0.76,hmarginratio=1:1,%
929   vscale=0.83,vmarginratio=1:1,%
930 }
931 %</article|report>
932 %<*book>
933 \ifbxjs@layout@buggyhmargin      %---
934 % アレ
935 \def\bxjs@layout{%
936   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
937   hmargin=36\jsc@mmm,hmarginratio=1:1,%
938   vscale=0.83,vmarginratio=1:1,%
939 }
940 \else                               %---
941 % 非アレ
942 \def\bxjs@layout{%
943   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
944   hmargin=18\jsc@mmm,%
945   vscale=0.83,vmarginratio=1:1,%
946 }
947 \fi                               %---
948 %</book>
949 %<*slide>
950 \def\bxjs@layout{%
951   noheadfoot,%
952   hscale=0.9,hmarginratio=1:1,%
953   vscale=0.944,vmarginratio=1:1,%
954 }
955 %</slide>
```

`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
956 \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。

```
957 %<*book>
958 \newcommand\jsTextWidthLimit{40}
959 \ifx\bxjs@textwidth@limit\@undefined\else
960   \let\c@bxjs@canta\@tempcanta
961   \setcounter{bxjs@canta}{\bxjs@textwidth@limit}
```

```

962 \long\edef\jsTextWidthLimit{\the\@tempcnta}
963 \fi
964 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

965 \def\bxjs@postproc@layout{%
966 % ドライバ再設定
967 \ifx\bxjs@geometry@driver\relax\else
968 \let\Gm@driver\bxjs@geometry@driver
969 \fi
970 % textwidth 調整
971 \@tempdimb=\textwidth
972 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
973 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
974 \advance\@tempdimb-\textwidth
975 \advance\oddsidemargin 0.5\@tempdimb
976 \advance\evensidemargin 0.5\@tempdimb
977 \fullwidth=\textwidth
978 %<*book>
979 \ifdim\textwidth>\jsTextWidthLimit\Cwd
980 \textwidth=\jsTextWidthLimit\Cwd
981 \addtolength\evensidemargin{\fullwidth-\textwidth}
982 \fi
983 %</book>
984 % textheight 調整
985 \@tempdimb=\textheight
986 \advance\textheight-\topskip
987 \divide\textheight\baselineskip \multiply\textheight\baselineskip
988 \advance\textheight\topskip
989 \advance\@tempdimb-\textheight
990 \advance\topmargin0.5\@tempdimb
991 % headheight 調整
992 \@tempdima=\topskip
993 \advance\headheight\@tempdima
994 \advance\topmargin-\@tempdima
995 % marginpar 設定
996 \setlength\marginparsep{\columnsep}
997 \setlength\marginparpush{\baselineskip}
998 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
999 -\textwidth-10\jscmmm-\marginparsep}
1000 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1001 % 連動する変数
1002 \maxdepth=.5\topskip
1003 \stockwidth=\paperwidth
1004 \stockheight=\paperheight
1005 }

```

\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。


```

1006 \edef\jsGeometryOptions{%
1007   \bxjs@layout@paper,\bxjs@layout}

```

■geometry パッケージ読込

geomentry オプションの値に応じて分岐する。

まずは geometry=class の場合。

```

1008 \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。

```

1009 \@onlypreamble\bxjs@revert
1010 \let\bxjs@revert\empty
1011 \edef\bxjs@tmpa{\the\ht\strutbox}
1012 \ht\strutbox=10\p@?
1013 \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。

```

1014 \ifbxjs@papersize
1015   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1016     \PassOptionsToPackage{dvipdfm}{geometry}
1017   \fi
1018   \let\bxPapersizeSpecialDone=t
1019 \else
1020   \PassOptionsToPackage{driver=none}{geometry}
1021 \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。

```

1022 \edef\bxjs@nxt{%
1023   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1024 \AtBeginDocument{\bxjs@pre@geometry@hook}
1025 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1026 \bxjs@nxt \bxjs@revert
1027 \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。

```

1028 \let\bxjs@geometry@driver\Gm@driver
1029 \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。

```
1030 \onlypreamble\bxjs@pre@geometry@hook
1031 \def\bxjs@pre@geometry@hook{%
1032   \@ifpackageloaded{geometry}{}%
1033   \@ifpackagelater{geometry}{2010/02/12}{}%\else
1034     \PackageWarningNoLine\bxjs@clsname
1035     {The 'geometry' package installed\MessageBreak
1036      is too old (< v5.0)}%
1037     \if x\jsEngine \ifnum\mag=\@m\else
1038       \def\bxjs@Gm@driver{pdftex}
1039       \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1040       \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1041       \fi\fi
1042   }%
1043   \ifjsWithpTeXng
1044     \ifx\Gm@driver\@empty
1045       \def\Gm@driver{pdftex}%
1046     \fi
1047   \fi
1048 }{}}
```

`\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。

```
1049 \def\setpagelayout{\@ifstar
1050   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1051 \def\bxjs@modify@layout#1{%
1052   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1053   \bxjs@nxt\bxjs@postproc@layout}
1054 \def\bxjs@reset@layout#1{%
1055   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1056   \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。

```
1057 \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。

```
1058 \AtBeginDocument{\bxjs@check@page@layout}
1059 \onlypreamble\bxjs@check@page@layout
1060 \def\bxjs@check@page@layout{%
```

```

1061 \ifdim\textwidth=.5\maxdimen
1062 \ClassError\bxjs@clsname
1063 {Page layout is not properly set}%
1064 {\@ehd}
1065 \fi}
1066 \def\jsUseMinimalPageLayout{%
1067 \setlength{\textwidth}{6.5in}%
1068 \setlength{\textheight}{8in}}

\setpagelayout はとりあえず無効にしておく。
1069 \let\bxjs@geometry@driver\relax
1070 \def\setpagelayout{\@ifstar
1071 {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1072 \def\bxjs@pagelayout@a#1{%
1073 \ClassError\bxjs@clsname
1074 {Command '\string\setpagelayout' is not supported,\MessageBreak
1075 because 'geometry' value is not 'class'}\@eha}

geometry=user の場合の処理はここで終わり。
1076 \fi\fi

```

ここからのコードは以下の点を除いて JS クラスのものを踏襲する。

- zw の代わりに \jsZw を用いる。
- article/report/book/slide の切り分けの処理が異なる。

■脚注

`\footnotesep` 各脚注の頭に入る支柱 (strut) の高さです。脚注間に余分のアキが入らないように、`\footnotesize` の支柱の高さ (行送りの 0.7 倍) に等しくします。

ここは元々は

```
{\footnotesize\global\setlength\footnotesep{\baselineskip}}
```

としていたが、そもそも `\global\setlength~` は calc 使用時には有意義な動作をしない。`\global\footnotesep` だと所望の値が得られるが、同時に `\footnotesize` のフォントを固定させてしまうという副作用をもつ。なので、実際の設定値を直接使うことにする。

```
1077 \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。

```
1078 \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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 本体で定義されていますが，ここで設定変更します。本文ページ（本文とフロートが共存するページ）とフロートだけのページで設定が異なります。ちなみに，カウンタは内部では `\c@` を名前に冠したマクロになっています。

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

[2003-08-23] ちょっと増やしました。

1079 `\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。

1080 `\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。

1081 `\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。

1082 `\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。

1083 `\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。

1084 `\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。

1085 `\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。

1086 `\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。

1087 `\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。

1088 `\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。

```

1089 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1090 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1091 \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 1092 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1093 \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 1094 \setlength\@fptop{0\p@? \@plus 1fil}
1095 \setlength\@fpsep{8\p@? \@plus 2fil}
1096 \setlength\@fpbot{0\p@? \@plus 1fil}

```

`\@dblfpsep` 段抜きフロートについての値です。

```

\@dblfpsep 1097 \setlength\@dblfpsep{0\p@? \@plus 1fil}
1098 \setlength\@dblfpsep{8\p@? \@plus 2fil}
\@dblfpbot 1099 \setlength\@dblfpbot{0\p@? \@plus 1fil}

```

6 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

```

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

```

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

```
1100 % \def\ps@empty{%
1101 %   \let\@mkboth\@gobbletwo
1102 %   \let\@oddhead\@empty
1103 %   \let\@oddfoot\@empty
1104 %   \let\@evenhead\@empty
1105 %   \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。

```
1106 \def\ps@plainfoot{%
1107   \let\@mkboth\@gobbletwo
1108   \let\@oddhead\@empty
1109   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1110   \let\@evenhead\@empty
1111   \let\@evenfoot\@oddfoot}
1112 \def\ps@plainhead{%
1113   \let\@mkboth\@gobbletwo
1114   \let\@oddfoot\@empty
1115   \let\@evenfoot\@empty
1116   \def\@evenhead{%
1117     \if@mparswitch \hss \fi
1118     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1119     \if@mparswitch\else \hss \fi}%
1120   \def\@oddhead{%
1121     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1122 %<book>\let\ps@plain\ps@plainhead
1123 %<!book>\let\ps@plain\ps@plainfoot
```

`\ps@headings` `headings` スタイルはヘッダに見出しとページ番号を出力します。ここではヘッダにアンダーラインを引くようにしてみました。

まず `article` の場合です。

```
1124 %<*article|slide>
1125 \if@twoside
1126   \def\ps@headings{%
1127     \let\@oddfoot\@empty
1128     \let\@evenfoot\@empty
1129     \def\@evenhead{\if@mparswitch \hss \fi
1130       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1131     \if@mparswitch\else \hss \fi}%
1132     \def\@oddhead{%
1133       \underline{%
1134         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1135     \let\@mkboth\markboth
1136     \def\sectionmark##1{\markboth{%
1137       \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
```

```

1138     ##1}{}}%
1139     \def\subsectionmark##1{\markright{%
1140         \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
1141         ##1}}%
1142     }
1143 \else % if not twoside
1144     \def\ps@headings{%
1145         \let\@oddfoot\@empty
1146         \def\@oddhead{%
1147             \underline{%
1148                 \hbox to \fullwidth{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1149         \let\@mkboth\markboth
1150         \def\sectionmark##1{\markright{%
1151             \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1152             ##1}}}%
1153 \fi
1154 %</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）。

```

1155 %<*book|report>
1156 \newif\if@omit@number
1157 \def\ps@headings{%
1158     \let\@oddfoot\@empty
1159     \let\@evenfoot\@empty
1160     \def\@evenhead{%
1161         \if@mparswitch \hss \fi
1162         \underline{\hbox to \fullwidth{\autoxspacing
1163             \textbf{\thepage}\hfil\leftmark}}}%
1164         \if@mparswitch\else \hss \fi}%
1165     \def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
1166         {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1167     \let\@mkboth\markboth
1168     \def\chaptermark##1{\markboth{%
1169         \ifnum \c@secnumdepth >\m@ne
1170             \if@mainmatter
1171                 \if@omit@number\else
1172                     \@chapapp\thechapter\@chappos\hskip1\jsZw
1173                 \fi
1174             \fi
1175             \fi
1176             ##1}{}}%
1177     \def\sectionmark##1{\markright{%
1178         \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1179         ##1}}}%
1180 %</book|report>

```

最後は学会誌の場合です。

```

1181 %<*jspf>

```

```

1182 \def\ps@headings{%
1183   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1184   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1185   \def\@oddhead{\normalfont\hfil \@title \hfil}
1186   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1187 %</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。

```

1188 \def\ps@myheadings{%
1189   \let\@oddfoot\empty\let\@evenfoot\empty
1190   \def\@evenhead{%
1191     \if@mparswitch \hss \fi%
1192     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1193     \if@mparswitch\else \hss \fi}%
1194   \def\@oddhead{%
1195     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1196   \let\@mkboth\@gobbletwo
1197 %<book|report> \let\chaptermark\@gobble
1198   \let\sectionmark\@gobble
1199 %<!\book&!report> \let\subsectionmark\@gobble
1200 }

```

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 1201 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1202 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1203 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1204 % \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。

```

1205 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1206 %\let\bxjs@subtitle\undefined

```

`\title` にフックを入れる。


```

1207 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1208 \AtBeginDocument{\bxjs@decl@subtitle}
1209 \def\bxjs@decl@subtitle{%
1210   \global\let\bxjs@decl@subtitle\relax
1211   \ifx\subtitle\@undefined
1212     \global\let\subtitle\jsSubtitle
1213   \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。

```

1214 \def\bxjs@annihilate@subtitle{%
1215   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1216   \global\let\jsSubtitle\relax}

```

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

```

\author 1217 %<*jspf>
\keywords 1218 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1219 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
1220 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1221 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1222 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1223 %</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。

```

1224 \def\plainifnotempty{%
1225   \ifx \@oddhead \@empty
1226     \ifx \@oddfoot \@empty
1227       \else
1228         \thispagestyle{plainfoot}%
1229       \fi
1230     \else
1231       \thispagestyle{plainhead}%
1232     \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。

```

1233 %<*article|book|report|slide>
1234 \if@titlepage
1235   \newcommand{\maketitle}{%
1236     \begin{titlepage}%
1237       \let\footnotesize\small
1238       \let\footnoterule\relax

```

```

1239 \let\footnote\thanks
1240 \null\vfil
1241 \if@slide
1242   {\footnotesize \@date}%
1243   \begin{center}
1244     \mbox{} \[\!1\!j\!s\!Z\!w\]
1245     \large
1246     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1247     \smallskip
1248     \@title
1249     \ifx\bxjs@subtitle\@undefined\else
1250       \par\vskip\z@
1251       {\small \bxjs@subtitle\par}
1252     \fi
1253     \smallskip
1254     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1255     \vfill
1256     {\small \@author}%
1257   \end{center}
1258 \else
1259 \vskip 60\p@?
1260 \begin{center}%
1261   {\LARGE \@title \par}%
1262   \ifx\bxjs@subtitle\@undefined\else
1263     \vskip5\p@?
1264     {\normalsize \bxjs@subtitle\par}
1265   \fi
1266   \vskip 3em%
1267   {\large
1268     \lineskip .75em
1269     \begin{tabular}[t]{c}%
1270       \@author
1271     \end{tabular}\par}%
1272   \vskip 1.5em
1273   {\large \@date \par}%
1274 \end{center}%
1275 \fi
1276 \par
1277 \@thanks\vfil\null
1278 \end{titlepage}%
1279 \setcounter{footnote}{0}%
1280 \global\let\thanks\relax
1281 \global\let\maketitle\relax
1282 \global\let\@thanks\@empty
1283 \global\let\@author\@empty
1284 \global\let\@date\@empty
1285 \global\let\@title\@empty
1286 \global\let\title\relax
1287 \global\let\author\relax

```

```

1288 \global\let\date\relax
1289 \global\let\and\relax
1290 \bxjs@annihilate@subtitle
1291 }%
1292 \else
1293 \newcommand{\maketitle}{\par
1294 \begin{group}
1295 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1296 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1297 \long\def\@makefnmark##1{\advance\leftskip 3\jsZw
1298 \parindent 1\jsZw\noindent
1299 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1300 \if@twocolumn
1301 \ifnum \col@number=\@ne
1302 \maketitle
1303 \else
1304 \twocolumn[\maketitle]%
1305 \fi
1306 \else
1307 \newpage
1308 \global\@topnum\z@ % Prevents figures from going at top of page.
1309 \maketitle
1310 \fi
1311 \plainifnotempty
1312 \@thanks
1313 \endgroup
1314 \setcounter{footnote}{0}%
1315 \global\let\thanks\relax
1316 \global\let\maketitle\relax
1317 \global\let\@thanks\@empty
1318 \global\let\@author\@empty
1319 \global\let\@date\@empty
1320 \global\let\@title\@empty
1321 \global\let\title\relax
1322 \global\let\author\relax
1323 \global\let\date\relax
1324 \global\let\and\relax
1325 \bxjs@annihilate@subtitle
1326 }

```

`\@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。

```

1327 \def\@maketitle{%
1328 \newpage\null
1329 \vskip 2em
1330 \begin{center}%
1331 \let\footnote\thanks
1332 {\LARGE \@title \par}%
1333 \ifx\bxjs@subtitle\@undefined\else
1334 \vskip3\p@?

```

```

1335     {\normalsize \bxjs@subtitle\par}
1336     \fi
1337     \vskip 1.5em
1338     {\large
1339     \lineskip .5em
1340     \begin{tabular}[t]{c}%
1341     \@author
1342     \end{tabular}\par}%
1343     \vskip 1em
1344     {\large \@date}%
1345     \end{center}%
1346     \par\vskip 1.5em
1347 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1348 }
1349 \fi
1350 %</article|book|report|slide>
1351 %<*jspf>
1352 \newcommand{\maketitle}{\par
1353 \begingroup
1354 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1355 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1356 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1357 \parindent 1\jsZw\noindent
1358 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1359 \twocolumn[\@maketitle]%
1360 \plainifnotempty
1361 \@thanks
1362 \endgroup
1363 \setcounter{footnote}{0}%
1364 \global\let\thanks\relax
1365 \global\let\maketitle\relax
1366 \global\let\@thanks\@empty
1367 \global\let\@author\@empty
1368 \global\let\@date\@empty
1369 % \global\let\@title\@empty % \@title は柱に使う
1370 \global\let\title\relax
1371 \global\let\author\relax
1372 \global\let\date\relax
1373 \global\let\and\relax
1374 \ifx\authors@mail\@undefined\else{%
1375 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1376 \footnotetext[0]{\itshape\authors@mail}%
1377 }\fi
1378 \global\let\authors@mail\@undefined}
1379 \def\@maketitle{%
1380 \newpage\null
1381 \vskip 6em % used to be 2em
1382 \begin{center}
1383 \let\footnote\thanks

```

```

1384 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1385 \lineskip .5em
1386 \ifx\@author\@undefined\else
1387 \vskip 1em
1388 \begin{tabular}[t]{c}%
1389 \@author
1390 \end{tabular}\par
1391 \fi
1392 \ifx\@etitle\@undefined\else
1393 \vskip 1em
1394 {\large \@etitle \par}%
1395 \fi
1396 \ifx\@eauthor\@undefined\else
1397 \vskip 1em
1398 \begin{tabular}[t]{c}%
1399 \@eauthor
1400 \end{tabular}\par
1401 \fi
1402 \vskip 1em
1403 \@date
1404 \end{center}
1405 \vskip 1.5em
1406 \centerline{\box\@abstractbox}
1407 \ifx\@keywords\@undefined\else
1408 \vskip 1.5em
1409 \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1410 \fi
1411 \vskip 1.5em}
1412 %</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 `\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。

```
1413 \def\@startsection#1#2#3#4#5#6{%
1414   \if@noskipsec \leavevmode \fi
1415   \par
1416 % 見出し上の空きを \@tempskipa にセットする
1417   \@tempskipa #4\relax
1418 % \@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
1419   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1420 % 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
1421   \ifdim \@tempskipa <\z@
1422     \@tempskipa -\@tempskipa \@afterindentfalse
1423   \fi
1424   \if@nobreak
1425 %   \everypar{\everyparhook}% これは間違い
1426     \everypar{}%
1427   \else
1428     \addpenalty\@secpenalty
1429 % 次の行は削除
1430 %   \addvspace\@tempskipa
1431 % 次の \noindent まで追加
1432     \ifdim \@tempskipa >\z@
1433       \if@slide\else
1434         \null
1435         \vspace*{-\baselineskip}%
1436       \fi
1437       \vskip\@tempskipa
1438     \fi
1439   \fi
1440   \noindent
1441 % 追加終わり
1442   \@ifstar
1443     {\@ssect{#3}{#4}{#5}{#6}}%
1444     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}
```

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

```
1445 \def\@sect#1#2#3#4#5#6[#7]#8{%
```

```

1446 \ifnum #2>\c@secnumdepth
1447 \let\@svsec\@empty
1448 \else
1449 \refstepcounter{#1}%
1450 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1451 \fi
1452 % 見出し後の空きを \@tempskipa にセット
1453 \@tempskipa #5\relax
1454 % 条件判断の順序を入れ替えました
1455 \ifdim \@tempskipa<\z@
1456 \def\@svsechd{%
1457 #6{\hskip #3\relax
1458 \@svsec #8}%
1459 \csname #1mark\endcsname{#7}}%
1460 \addcontentsline{toc}{#1}{%
1461 \ifnum #2>\c@secnumdepth \else
1462 \protect\numberline{\csname the#1\endcsname}%
1463 \fi
1464 #7}}% 目次にフルネームを載せるなら #8
1465 \else
1466 \begingroup
1467 \interlinepenalty \@M % 下から移動
1468 #6{%
1469 \@hangfrom{\hskip #3\relax\@svsec}%
1470 % \interlinepenalty \@M % 上に移動
1471 #8\@@par}%
1472 \endgroup
1473 \csname #1mark\endcsname{#7}%
1474 \addcontentsline{toc}{#1}{%
1475 \ifnum #2>\c@secnumdepth \else
1476 \protect\numberline{\csname the#1\endcsname}%
1477 \fi
1478 #7}}% 目次にフルネームを載せるならここは #8
1479 \fi
1480 \@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。

```

1481 \def\@xsect#1{%
1482 % 見出しの後ろの空きを \@tempskipa にセット
1483 \@tempskipa #1\relax
1484 % 条件判断の順序を変えました
1485 \ifdim \@tempskipa<\z@
1486 \@nbreakfalse
1487 \global\@noskipsecttrue
1488 \everypar{%

```

```

1489 \if@noskipsec
1490 \global\@noskipsecfalse
1491 {\setbox\z@\lastbox}%
1492 \clubpenalty\@M
1493 \begingroup \@svsechd \endgroup
1494 \unskip
1495 \@tempskipa #1\relax
1496 \hskip -\@tempskipa
1497 \bxjs@ltj@inhibitglue
1498 \else
1499 \clubpenalty \@clubpenalty
1500 \everypar{\everyparhook}%
1501 \fi\everyparhook}%
1502 \else
1503 \par \nobreak
1504 \vskip \@tempskipa
1505 \@afterheading
1506 \fi
1507 \if@slide
1508 {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1509 \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1510 \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1511 \fi
1512 \par % 2000-12-18
1513 \ignorespaces}
1514 \def\@ssect#1#2#3#4#5{%
1515 \@tempskipa #3\relax
1516 \ifdim \@tempskipa<\z@
1517 \def\@svsechd{#4{\hskip #1\relax #5}}%
1518 \else
1519 \begingroup
1520 #4{%
1521 \hangfrom{\hskip #1}%
1522 \interlinepenalty \@M #5\@par}%
1523 \endgroup
1524 \fi
1525 \@xsect{#3}}

```

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

```

1526 \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 1527 \newcommand*\chaptermark[1]{}
                  1528 % \newcommand*\sectionmark[1]{}
\subsubsectionmark
\paragraphmark
\subparagraphmark

```



```

1529 % \newcommand*\subsectionmark}[1]{}
1530 % \newcommand*\subsubsectionmark}[1]{}
1531 % \newcommand*\paragraphmark}[1]{}
1532 % \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。

```

1533 %<!book&!report>\setcounter{secnumdepth}{3}
1534 %<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 1535 \newcounter{part}
1536 %<book|report>\newcounter{chapter}
\c@subsubsection 1537 %<book|report>\newcounter{section}[chapter]
1538 %<!book&!report>\newcounter{section}
\c@paragraph 1539 \newcounter{subsection}[section]
1540 \newcounter{subsubsection}[subsection]
1541 \newcounter{paragraph}[subsubsection]
1542 \newcounter{subparagraph}[paragraph]

```

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

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

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

1543 \renewcommand{\thepart}{\@Roman\c@part}
1544 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1545 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1546 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1547 %<*book|report>
1548 \renewcommand{\thechapter}{\@arabic\c@chapter}
1549 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1550 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1551 %</book|report>
1552 \renewcommand{\thesubsubsection}{%
1553   \thesubsection.\@arabic\c@subsubsection}
1554 \renewcommand{\theparagraph}{%
1555   \thesubsubsection.\@arabic\c@paragraph}
1556 \renewcommand{\thesubparagraph}{%
1557   \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。

```

1558 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1559 %<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。

```

1560 %<*book|report>
1561 \newcommand\frontmatter{%
1562   \if@openright
1563     \cleardoublepage
1564   \else
1565     \clearpage
1566   \fi
1567   \@mainmatterfalse
1568   \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。

```

1569 \newcommand\mainmatter{%
1570 % \if@openright
1571   \cleardoublepage
1572 % \else
1573 %   \clearpage
1574 % \fi
1575   \@mainmattertrue
1576   \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。

```

1577 \newcommand\backmatter{%
1578   \if@openright
1579     \cleardoublepage
1580   \else
1581     \clearpage
1582   \fi
1583   \@mainmatterfalse}
1584 %</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。

```
1585 %<!*book&!report>
1586 \newcommand\part{%
1587   \if@noskipsec \leavevmode \fi
1588   \par
1589   \addvspace{4ex}%
1590   \if@english \afterindentfalse \else \afterindenttrue \fi
1591   \secdef\@part\@spart}
1592 %</!*book&!report>
```

book スタイルの場合は、少し複雑です。

```
1593 %<*book|report>
1594 \newcommand\part{%
1595   \if@openright
1596     \cleardoublepage
1597   \else
1598     \clearpage
1599   \fi
1600   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1601   \if@twocolumn
1602     \onecolumn
1603     \@restonecoltrue
1604   \else
1605     \@restonecolfalse
1606   \fi
1607   \null\vfil
1608   \secdef\@part\@spart}
1609 %</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。

```
1610 %<!*book&!report>
1611 \def\@part[#1]#2{%
1612   \ifnum \c@secnumdepth >\m@ne
1613     \refstepcounter{part}%
1614     \addcontentsline{toc}{part}{%
1615       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1616   \else
1617     \addcontentsline{toc}{part}{#1}%
1618   \fi
1619   \markboth{}{}%
1620   {\parindent\z@
```

```

1621 \raggedright
1622 \interlinepenalty \@M
1623 \normalfont
1624 \ifnum \c@secnumdepth >\m@ne
1625 \Large\headfont\prepartname\thepart\postpartname
1626 \par\nobreak
1627 \fi
1628 \huge \headfont #2%
1629 \markboth{}{}\par}%
1630 \nobreak
1631 \vskip 3ex
1632 \@afterheading}
1633 %</!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。

```

1634 %<*book|report>
1635 \def\@part[#1]#2{%
1636 \ifnum \c@secnumdepth >-2\relax
1637 \refstepcounter{part}%
1638 \addcontentsline{toc}{part}{%
1639 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1640 \else
1641 \addcontentsline{toc}{part}{#1}%
1642 \fi
1643 \markboth{}{}\par}%
1644 {\centering
1645 \interlinepenalty \@M
1646 \normalfont
1647 \ifnum \c@secnumdepth >-2\relax
1648 \huge\headfont \prepartname\thepart\postpartname
1649 \par\vskip20\p@?
1650 \fi
1651 \Huge \headfont #2\par}%
1652 \@endpart}
1653 %</book|report>

```

\@spart 番号を付けない部です。

```

1654 %<*!book&!report>
1655 \def\@spart#1{%
1656 \parindent \z@ \raggedright
1657 \interlinepenalty \@M
1658 \normalfont
1659 \huge \headfont #1\par}%
1660 \nobreak
1661 \vskip 3ex
1662 \@afterheading}
1663 %</!book&!report>
1664 %<*book|report>
1665 \def\@spart#1{%

```

```

1666 \centering
1667 \interlinepenalty \@M
1668 \normalfont
1669 \Huge \headfont #1\par}%
1670 \@endpart}
1671 %</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。

```

1672 %<*book|report>
1673 \def\@endpart{\vfil\newpage
1674 \if@twoside
1675 \null
1676 \thispagestyle{empty}%
1677 \newpage
1678 \fi
1679 \if@restonecol
1680 \twocolumn
1681 \fi}
1682 %</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。

```

1683 %<*book|report>
1684 \newcommand{\chapter}{%
1685 \if@openright\cleardoublepage\else\clearpage\fi
1686 \plainifnotempty % 元: \thispagestyle{plain}
1687 \global\@topnum\z@
1688 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1689 \secdef
1690 {\@omit@numberfalse\@chapter}%
1691 {\@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。

```

1692 \def\@chapter[#1]#2{%
1693 \ifnum \c@secnumdepth >\m@ne
1694 \if@mainmatter
1695 \refstepcounter{chapter}%
1696 \typeout{\@chapapp\thechapter\@chappos}%
1697 \addcontentsline{toc}{chapter}%
1698 {\protect\numberline
1699 % {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
1700 {\@chapapp\thechapter\@chappos}%
1701 #1}%
1702 \else\addcontentsline{toc}{chapter}{#1}\fi

```

```

1703 \else
1704   \addcontentsline{toc}{chapter}{#1}%
1705 \fi
1706 \chaptermark{#1}%
1707 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
1708 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
1709 \if@twocolumn
1710   \@topnewpage[\@makechapterhead{#2}]%
1711 \else
1712   \@makechapterhead{#2}%
1713   \@afterheading
1714 \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。

```

1715 \def\@makechapterhead#1{%
1716   \vspace*{2\Cvs}% 欧文は 50pt
1717   {\parindent \z@ \raggedright \normalfont
1718     \ifnum \c@secnumdepth >\m@ne
1719       \if@mainmatter
1720         \huge\headfont \@chapapp\thechapter\@chappos
1721         \par\nobreak
1722         \vskip \Cvs % 欧文は 20pt
1723       \fi
1724     \fi
1725     \interlinepenalty\@M
1726     \Huge \headfont #1\par\nobreak
1727     \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。

```

1728 \def\@schapter#1{%
1729   \chaptermark{#1}%
1730   \if@twocolumn
1731     \@topnewpage[\@makeschapterhead{#1}]%
1732   \else
1733     \@makeschapterhead{#1}\@afterheading
1734   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

1735 \def\@makeschapterhead#1{%
1736   \vspace*{2\Cvs}% 欧文は 50pt
1737   {\parindent \z@ \raggedright
1738     \normalfont
1739     \interlinepenalty\@M
1740     \Huge \headfont #1\par\nobreak
1741     \vskip 3\Cvs}} % 欧文は 40pt
1742 %</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。

```
1743 \if@twocolumn
1744   \newcommand{\section}{%
1745     %<jspf>\ifx\maketitle\relax\else\maketitle\fi
1746     \@startsection{section}{1}{\z@}%
1747     %<kiyou>    {0.6\Cvs}{0.4\Cvs}%
1748     %<kiyou>    {\Cvs}{0.5\Cvs}%
1749     %   {\normalfont\large\headfont\@secapp}%
1750     {\normalfont\large\headfont\raggedright}}
1751 \else
1752   \newcommand{\section}{%
1753     \if@slide\clearpage\fi
1754     \@startsection{section}{1}{\z@}%
1755     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1756     {.5\Cvs \@plus.3\Cdp}% 後アキ
1757     %   {\normalfont\Large\headfont\@secapp}%
1758     {\normalfont\Large\headfont\raggedright}}
1759 \fi
```

`\subsection` 同上です。

```
1760 \if@twocolumn
1761   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1762     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1763     {\normalfont\normalsize\headfont}}
1764 \else
1765   \newcommand{\subsection}{\@startsection{subsection}{2}{\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}}
1769 \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)。

```
1770 \if@twocolumn
1771   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1772     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1773     {\normalfont\normalsize\headfont}}
1774 \else
1775   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1776     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1777     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
1778     {\normalfont\normalsize\headfont}}
1779 \fi
```

`\paragraph` 見出しの後ろで改行されません。

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。さらに、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た。

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

```
1780 \newcommand\jsParagraphMark{\jsJaChar{■}}
1781 \ifx\bxjs@paragraph@mark\@undefined\else
1782   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
1783 \fi
1784 \let\jsJaChar\@empty
1785 \if@twocolumn
1786   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1787     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1788 %<jspf>    {\normalfont\normalsize\headfont}}
1789 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
1790 \else
1791   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1792     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1793     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1794 %<jspf>    {\normalfont\normalsize\headfont}}
1795 %<!jspf>    {\normalfont\normalsize\headfont\jsParagraphMark}}
1796 \fi
```

`\subparagraph` 見出しの後ろで改行されません。

```
1797 \if@twocolumn
1798   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1799     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1800     {\normalfont\normalsize\headfont}}
1801 \else
1802   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1803     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1804     {\normalfont\normalsize\headfont}}
1805 \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。

```
1806 \if@slide
1807   \setlength\leftmargini{1\jsZw}
1808 \else
```



```

1809 \if@twocolumn
1810 \setlength\leftmargini{2\jsZw}
1811 \else
1812 \setlength\leftmargini{3\jsZw}
1813 \fi
1814 \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 1815 \if@slide
\leftmarginv 1816 \setlength\leftmarginii {1\jsZw}
1817 \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 1818 \setlength\leftmarginiv {1\jsZw}
1819 \setlength\leftmarginv {1\jsZw}
1820 \setlength\leftmarginvi {1\jsZw}
1821 \else
1822 \setlength\leftmarginii {2\jsZw}
1823 \setlength\leftmarginiii{2\jsZw}
1824 \setlength\leftmarginiv {2\jsZw}
1825 \setlength\leftmarginv {1\jsZw}
1826 \setlength\leftmarginvi {1\jsZw}
1827 \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。

```

1828 \setlength \labelsep {0.5\jsZw} % .5em
1829 \setlength \labelwidth{\leftmargini}
1830 \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。

```

1831 \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 1832 \@beginparpenalty -\@lowpenalty
\@itempenalty 1833 \@endparpenalty -\@lowpenalty
1834 \@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っていました。

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

```

1835 \def\@listi{\leftmargin\leftmargini

```

```

1836 \parsep \z@
1837 \topsep 0.5\baselineskip
1838 \itemsep \z@ \relax}
1839 \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）。

```
1840 \@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 1841 \def\@listii{\leftmargin\leftmarginii
\@listiv 1842 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1843 \topsep \z@
\@listv 1844 \parsep \z@
\@listvi 1845 \itemsep\parsep}
1846 \def\@listiii{\leftmargin\leftmarginiii
1847 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1848 \topsep \z@
1849 \parsep \z@
1850 \itemsep\parsep}
1851 \def\@listiv {\leftmargin\leftmarginiv
1852 \labelwidth\leftmarginiv
1853 \advance\labelwidth-\labelsep}
1854 \def\@listv {\leftmargin\leftmarginv
1855 \labelwidth\leftmarginv
1856 \advance\labelwidth-\labelsep}
1857 \def\@listvi {\leftmargin\leftmarginvi
1858 \labelwidth\leftmarginvi
1859 \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。

```

1860 \renewcommand{\theenumi}{\@arabic\c@enumi}
1861 \renewcommand{\theenumii}{\@alph\c@enumii}
1862 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1863 \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。

```

1864 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1865 \newcommand*\jsInJaParen}[1]{%
1866   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1867 \newcommand{\labelenumi}{\theenumi.}
1868 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1869 \newcommand{\labelenumiii}{\theenumiii.}
1870 \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 1871 \renewcommand{\p@enumii}{\theenumi}
1872 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1873 \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 1874 \newcommand\labelitemi{\textbullet}
1875 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiii 1876 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 1877 \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。

```

1878 \newenvironment{description}{%
1879   \list{}{%
1880     \labelwidth=\leftmargin
1881     \labelsep=1\jsZw
1882     \advance \labelwidth by -\labelsep
1883     \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。

```

1884 \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。

```

1885 %<*book|report>
1886 \newenvironment{abstract}{%
1887   \begin{list}{}{%
1888     \listparindent=1\jsZw
1889     \itemindent=\listparindent
1890     \rightmargin=\z@
1891     \leftmargin=5\jsZw}\item[]{}\end{list}\vspace{\baselineskip}}
1892 %</book|report>
1893 %<*article|slide>
1894 \newbox\@abstractbox
1895 \if@titlepage
1896   \newenvironment{abstract}{%
1897     \titlepage
1898     \null\vfil
1899     \@beginparpenalty\@lowpenalty
1900     \begin{center}%
1901       \headfont \abstractname
1902       \@endparpenalty\@M
1903     \end{center}}%
1904   {\par\vfil\null\endtitlepage}
1905 \else
1906   \newenvironment{abstract}{%
1907     \if@twocolumn
1908       \ifx\maketitle\relax
1909         \section*{\abstractname}%
1910       \else
1911         \global\setbox\@abstractbox\hbox\bgroup
1912         \begin{minipage}[b]{\textwidth}
1913           \small\parindent1\jsZw
1914           \begin{center}%
1915             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1916           \end{center}%
1917           \list{}{%
1918             \listparindent\parindent
1919             \itemindent \listparindent
1920             \rightmargin \leftmargin}%
1921           \item\relax
1922         \fi
1923       \else
1924         \small
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}{\if@twocolumn
1934      \ifx\maketitle\relax
1935      \else
1936      \endlist\end{minipage}\egroup
1937      \fi
1938      \else
1939      \endlist
1940      \fi}
1941 \fi
1942 %</article|slide>
1943 %<*jspf>
1944 \newbox\@abstractbox
1945 \newenvironment{abstract}{%
1946   \global\setbox\@abstractbox\hbox\bgroup
1947   \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
1948     \small
1949     \if@english \parindent6mm \else \parindent1\jsZw \fi}%
1950   {\end{minipage}\egroup}
1951 %</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。

```

1952 %<*jspf>
1953 %\newbox\@keywordsbox
1954 %\newenvironment{keywords}{%
1955 %   \global\setbox\@keywordsbox\hbox\bgroup
1956 %   \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
1957 %     \small\parindent0\jsZw}%
1958 %   {\end{minipage}\egroup}
1959 %</jspf>

```

■verse 環境

verse 詩のための `verse` 環境です。

```

1960 \newenvironment{verse}{%
1961   \let \\\=\@centercr
1962   \list{}{%
1963     \itemsep \z@
1964     \itemindent -2\jsZw % 元: -1.5em
1965     \listparindent\itemindent
1966     \rightmargin \z@
1967     \advance\leftmargin 2\jsZw}% 元: 1.5em
1968   \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。

```
1969 \newenvironment{quotation}{%
1970   \list{}{%
1971     \listparindent\parindent
1972     \itemindent\listparindent
1973     \rightmargin \z@}%
1974   \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。

```
1975 \newenvironment{quote}%
1976   {\list{}{\rightmargin\z@}\item\relax}{\endlist}
```

■定理など `ltthm.dtx` 参照。たとえば次のように定義します。

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

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

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

```
1977 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
1978   \item[\hskip \labelsep{\headfont #1\ #2}]}
1979 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
1980   \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。

```
1981 \newenvironment{titlepage}{%
1982   %<book|report> \cleardoublepage
1983   \if@twocolumn
1984     \@restonecoltrue\onecolumn
1985   \else
1986     \@restonecolfalse\newpage
1987   \fi
1988   \thispagestyle{empty}%
1989   \setcounter{page}\@ne
1990 }%
1991 {\if@restonecol\twocolumn \else \newpage \fi
1992   \if@twoside\else
1993     \setcounter{page}\@ne
1994   \fi}
```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
1995 %<*!book&!report>
1996 \newcommand{\appendix}{\par
1997   \setcounter{section}{0}%
1998   \setcounter{subsection}{0}%
1999   \gdef\presectionname{\appendixname}%
2000   \gdef\postsectionname{}}%
2001 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2002   \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2003   \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
2004 %</!book&!report>
2005 %<*book|report>
2006 \newcommand{\appendix}{\par
2007   \setcounter{chapter}{0}%
2008   \setcounter{section}{0}%
2009   \gdef\@chapapp{\appendixname}%
2010   \gdef\@chappos{}}%
2011   \gdef\thechapter{\@Alph\c@chapter}}
2012 %</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 倍の幅の空きが入ります。

```
2013 \setlength\arraycolsep{5\p@?}
```

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

```
2014 \setlength\tabcolsep{6\p@?}
```

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

```
2015 \setlength\arrayrulewidth{.4\p@}
```

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

```
2016 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2017 \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。

```
2018 \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。

```
2019 \setlength\fboxsep{3\p@?}
2020 \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。

```
2021 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2022 %<*book|report>
2023 \@addtoreset{equation}{chapter}
2024 \renewcommand\theequation
2025 {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}
2026 %</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。

```
2027 % \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。

```
2028 % \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。

```
2029 % \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。

```
2030 %<!*book&!report>
2031 \newcounter{figure}
2032 \renewcommand \thefigure {\@arabic\c@figure}
2033 %</!*book&!report>
2034 %<*book|report>
2035 \newcounter{figure}[chapter]
2036 \renewcommand \thefigure
2037     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2038 %</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 2039 \def\fps@figure{tbp}
2040 \def\ftype@figure{1}
\fnun@figure 2041 \def\ext@figure{lof}
2042 \def\fnun@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2043 \newenvironment{figure}%
2044         {\@float{figure}}%
2045         {\end@float}
2046 \newenvironment{figure*}%
2047         {\@dblfloat{figure}}%
2048         {\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。

```
2049 %<!*book&!report>
2050 \newcounter{table}
2051 \renewcommand\thetable{\@arabic\c@table}
2052 %</!*book&!report>
2053 %<*book|report>
2054 \newcounter{table}[chapter]
2055 \renewcommand \thetable
2056     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2057 %</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 2058 \def\fps@table{tbp}
2059 \def\ftype@table{2}
\fnun@table 2060 \def\ext@table{lot}
2061 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` * は段抜きのフロートです。

`table*`

```

2062 \newenvironment{table}%
2063         {\@float{table}}%
2064         {\end@float}
2065 \newenvironment{table*}%
2066         {\@dblfloat{table}}%
2067         {\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。

```

2068 \newlength\abovecaptionskip
2069 \newlength\belowcaptionskip
2070 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2071 \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。

```

2072 %<!*jspf>
2073 % \long\def\@makecaption#1#2{{\small
2074 %   \advance\leftskip1cm
2075 %   \advance\rightskip1cm
2076 %   \vskip\abovecaptionskip
2077 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2078 %   \ifdim \wd\@tempboxa >\hsize
2079 %     #1\hskip1\jsZw\relax #2\par
2080 %   \else
2081 %     \global \minipagefalse
2082 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2083 %   \fi
2084 %   \vskip\belowcaptionskip}}
2085 \long\def\@makecaption#1#2{{\small
2086   \advance\leftskip .0628\linewidth
2087   \advance\rightskip .0628\linewidth
2088   \vskip\abovecaptionskip
2089   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2090   \ifdim \wd\@tempboxa <\hsize \centering \fi
2091   #1\hskip1\jsZw\relax #2\par
2092   \vskip\belowcaptionskip}}
2093 %</!*jspf>
2094 %<!*jspf>
2095 \long\def\@makecaption#1#2{%

```

```

2096 \vskip\abovecaptionskip
2097 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2098 \ifdim \wd\@tempboxa >\hsize
2099   {\small\sffamily
2100     \list{#1}{%
2101       \renewcommand{\makelabel}[1]{##1\hfil}
2102       \itemsep \z@
2103       \itemindent \z@
2104       \labelsep \z@
2105       \labelwidth 11mm
2106       \listparindent\z@
2107       \leftmargin 11mm}\item\relax #2\endlist}
2108   \else
2109     \global \@minipagefalse
2110     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2111   \fi
2112 \vskip\belowcaptionskip}
2113 %</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 \if@jsc@warnoldfontcmdexception は \allow/disallowoldfontcommands の状態
を表す。


---


2114 \newif\if@jsc@warnoldfontcmd
2115 \@jsc@warnoldfontcmdtrue
2116 \newif\if@jsc@warnoldfontcmdexception
2117 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand
2118 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2119   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2120   \DeclareOldFontCommand{#1}{%
2121     \bxjs@oldfontcmd{#1}#2%
2122   }{%

```

```

2123     \bxjs@oldfontcmd{#1}#3%
2124 }%
2125 }
2126 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2127   \ClassInfo\bxjs@clsname
2128   {Old font command '\string#1' is used!!\MessageBreak
2129     The first occurrence is}%
2130 }

```

`\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。

```

2131 \newcommand*{\allowoldfontcommands}{%
2132   \@jsc@warnoldfontcmdexceptiontrue}
2133 \newcommand*{\disallowoldfontcommands}{%
2134   \@jsc@warnoldfontcmdexceptionfalse}

  ※ 1.x 版では Warning ではなく Info に留めておく。

2135 \let\bxjs@oldfontcmd@list\@empty
2136 \def\bxjs@oldfontcmd#1{%
2137   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2138 \def\bxjs@oldfontcmd@a#1#2{%
2139   \if@jsc@warnoldfontcmdexception\else
2140     \global\@jsc@warnoldfontcmdfalse
2141     \ifx#1\relax
2142       \global\let#1=t%
2143       \jsc@warnoldfontcmd{#2}%
2144     \fi
2145   \fi}
2146 \def\bxjs@warnoldfontcmd@final{%
2147   \par
2148   \let\@tempa\@empty
2149   \def\do##1{%
2150     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2151       \edef\@tempa{\@tempa \space\string##1}\fi}
2152   \bxjs@oldfontcmd@list
2153   \ifx\@tempa\@empty\else
2154     \ClassWarningNoLine\bxjs@clsname
2155     {Some old font commands were used in text\MessageBreak
2156       (see the log file for detail)}%
2157     \ClassInfo\bxjs@clsname
2158     {Some old font commands were used in text:\MessageBreak
2159       \space\@tempa\MessageBreak
2160       You should note, that since 1994 LaTeX2e provides a\MessageBreak
2161       new font selection scheme called NFSS2 with several\MessageBreak
2162       new, combinable font commands. The
2163       class provides\MessageBreak

```

```

2164 the old font commands
2165 only for compatibility%
2166 \@gobble}%
2167 \fi}
2168 \AtEndDocument{\bxjs@warnoldfontcmd@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2169 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2170 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2171 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sff 2172 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2173 \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。

```

2174 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2175 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2176 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2177 \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 2178 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2179 \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。

```
2180 \newcommand\@pnumwidth{1.55em}
2181 \newcommand\@tocrmarg{2.55em}
2182 \newcommand\@dotsep{4.5}
2183 %<!book&!report>\setcounter{tocdepth}{2}
2184 %<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)

```
2185 \newdimen\jsc@tocl@width
2186 \newcommand{\tableofcontents}{%
2187 %<*book|report>
2188 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2189 \settowidth\@tempdima{\headfont\appendixname}%
2190 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2191 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2192 \if@twocolumn
2193 \@restonecoltrue\onecolumn
2194 \else
2195 \@restonecolfalse
2196 \fi
2197 \chapter*{\contentsname}%
2198 \@mkboth{\contentsname}{}%
2199 %</book|report>
```

```

2200 %<*!book&!report>
2201 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2202 \settowidth\@tempdima{\headfont\appendixname}%
2203 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2204 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2205 \section*{\contentsname}%
2206 \@mkboth{\contentsname}{\contentsname}%
2207 %</!book&!report>
2208 \@starttoc{toc}%
2209 %<book|report> \if@restonecol\twocolumn\fi
2210 }

```

\l@part 部の目次です。

```

2211 \newcommand*{\l@part}[2]{%
2212 \ifnum \c@tocdepth >-2\relax
2213 %<!book&!report> \addpenalty\@secpenalty
2214 %<book|report> \addpenalty{-\@highpenalty}%
2215 \addvspace{2.25em \@plus\p@}%
2216 \begingroup
2217 \parindent \z@
2218 % \@pnumwidth should be \@tocrmarg
2219 % \rightskip \@pnumwidth
2220 \rightskip \@tocrmarg
2221 \parfillskip -\rightskip
2222 {\leavevmode
2223 \large \headfont
2224 \setlength\@lnumwidth{4\jsZw}%
2225 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2226 \nobreak
2227 %<book|report> \global\@nobreaktrue
2228 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2229 \endgroup
2230 \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)

```

2231 %<*book|report>
2232 \newcommand*{\l@chapter}[2]{%
2233 \ifnum \c@tocdepth >\m@ne
2234 \addpenalty{-\@highpenalty}%
2235 \addvspace{1.0em \@plus\p@}
2236 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2237 \begingroup
2238 \parindent\z@
2239 % \rightskip\@pnumwidth
2240 \rightskip\@tocrmarg
2241 \parfillskip-\rightskip
2242 \leavevmode\headfont

```

```

2243 %      % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2244 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2245 \advance\leftskip\@lnumwidth \hskip-\leftskip
2246 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2247 \penalty\@highpenalty
2248 \endgroup
2249 \fi}
2250 %</book|report>

```

\l@section 節の目次です。

```

2251 %<*&!book&!report>
2252 \newcommand*{\l@section}[2]{%
2253 \ifnum \c@tocdepth >\z@
2254 \addpenalty{\@secpenalty}%
2255 \addvspace{1.0em \@plus\p@}%
2256 \begingroup
2257 \parindent\z@
2258 % \rightskip\@pnumwidth
2259 \rightskip\@tocrmarg
2260 \parfillskip-\rightskip
2261 \leavevmode\headfont
2262 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2263 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2264 \advance\leftskip\@lnumwidth \hskip-\leftskip
2265 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2266 \endgroup
2267 \fi}
2268 %</!book&!report>

```

インデントと幅はそれぞれ 1.5em, 2.3em でしたが⁵, 1zw, 3.683zw に変えました。

```

2269 %<book|report> % \newcommand*{\l@section}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

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

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

\l@subsubsection しれません。

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

```

\l@subparagraph 2270 %<*&!book&!report>
2271 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2272 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2273 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2274 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2275 %
2276 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2277 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2278 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2279 %
2280 \newcommand*{\l@subsection}{%
2281 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw

```



```

2282 \dottedtocline{2}{\@tempdima}{3\jsZw}}
2283 \newcommand*\l@subsubsection}{%
2284 \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2285 \dottedtocline{3}{\@tempdima}{4\jsZw}}
2286 \newcommand*\l@paragraph}{%
2287 \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2288 \dottedtocline{4}{\@tempdima}{5\jsZw}}
2289 \newcommand*\l@subparagraph}{%
2290 \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2291 \dottedtocline{5}{\@tempdima}{6\jsZw}}
2292 %</!book&!report>
2293 %<*book|report>
2294 % \newcommand*\l@section}{\@dottedtocline{2}{3.8em}{3.2em}}
2295 % \newcommand*\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2296 % \newcommand*\l@paragraph}{\@dottedtocline{4}{10em}{5em}}
2297 % \newcommand*\l@subparagraph}{\@dottedtocline{5}{12em}{6em}}
2298 \newcommand*\l@section}{%
2299 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2300 \dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2301 \newcommand*\l@subsubsection}{%
2302 \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2303 \dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2304 \newcommand*\l@subsubsection}{%
2305 \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2306 \dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2307 \newcommand*\l@paragraph}{%
2308 \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2309 \dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2310 \newcommand*\l@subparagraph}{%
2311 \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2312 \dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2313 %</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。

```

2314 \newdimen\@lnumwidth
2315 \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。

```

2316 \def\dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2317 \vskip \z@ \@plus.2\p@?
2318 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2319 \parindent #2\relax\@afterindenttrue
2320 \interlinepenalty\@M
2321 \leavevmode

```

```

2322 \@lnumwidth #3\relax
2323 \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2324 {#4}\nobreak
2325 \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2326 mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2327 \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2328 \newcommand{\listoffigures}{%
2329 %<*book|report>
2330 \if@twocolumn\@restonecoltrue\onecolumn
2331 \else\@restonecolfalse\fi
2332 \chapter*{\listfigurename}%
2333 \@mkboth{\listfigurename}{}%
2334 %</book|report>
2335 %<*&!book&!report>
2336 \section*{\listfigurename}%
2337 \@mkboth{\listfigurename}{\listfigurename}%
2338 %</&!book&!report>
2339 \@starttoc{lof}%
2340 %<book|report> \if@restonecol\twocolumn\fi
2341 }

```

`\l@figure` 図目次の項目を出力します。

```

2342 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsw}{3.683\jsw}}

```

`\listoftables` 表目次を出力します。

```

2343 \newcommand{\listoftables}{%
2344 %<*book|report>
2345 \if@twocolumn\@restonecoltrue\onecolumn
2346 \else\@restonecolfalse\fi
2347 \chapter*{\listtablename}%
2348 \@mkboth{\listtablename}{}%
2349 %</book|report>
2350 %<*&!book&!report>
2351 \section*{\listtablename}%
2352 \@mkboth{\listtablename}{\listtablename}%
2353 %</&!book&!report>
2354 \@starttoc{lot}%
2355 %<book|report> \if@restonecol\twocolumn\fi
2356 }

```

`\l@table` 表目次は図目次と同じです。

```

2357 \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。

```
2358 \newdimen\bibindent
2359 \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。

```
2360 \newenvironment{thebibliography}[1]{%
2361   \@jsc@warnoldfontcmdexceptiontrue
2362   \global\let\presectionname\relax
2363   \global\let\postsectionname\relax
2364   %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2365   %<*kiyou>
2366   \vspace{1.5\baselineskip}
2367   \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2368   \vspace{0.5\baselineskip}
2369   %</kiyou>
2370   %<book|report> \chapter*{\bibname}\@mkboth{\bibname}{}%
2371   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2372   \list{\@biblabel{\@arabic\c@enumiv}}%
2373     {\settowidth\labelwidth{\@biblabel{#1}}%
2374     \leftmargin\labelwidth
2375     \advance\leftmargin\labelsep
2376     \@openbib@code
2377     \usecounter{enumiv}%
2378     \let\p@enumiv\@empty
2379     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2380   %<kiyou> \small
2381   \sloppy
2382   \clubpenalty4000
2383   \@clubpenalty\clubpenalty
2384   \widowpenalty4000%
2385   \sfcode`\.\@m}
2386   {\def\@noitemerr
2387     {\@latex@warning{Empty `thebibliography' environment}}}%
2388   \endlist}
```

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

```
2389 \newcommand{\newblock}{\hspace .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。

```
2390 \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。

```
2391 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

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

```
2392 % \def\@citex[#1]#2{%
2393 %   \let\@citea\@empty
2394 %   \@cite{\@for\@citeb:=#2\do
2395 %     {\@citea\def\@citea{, \jsInhibitGlue\penalty\@m }%
2396 %     \edef\@citeb{\expandafter\@firstofone\@citeb}%
2397 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2398 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2399 %       \G@refundefinedtrue
2400 %       \@latex@warning
2401 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2402 %     {\hbox{\csname b@\@citeb\endcsname}}}{#1}}
2403 % \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。

```
2404 % \DeclareRobustCommand\cite{\unskip
2405 %   \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}]
2406 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2407 %   , \jsInhibitGlue\ #2\fi}})}$}
```

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きにマージンがずれる現象を直しまし
た (Thanks: 藤村さん)。

```
2408 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2409   \if@twocolumn
2410     \onecolumn\@restonecolfalse
2411   \else
2412     \clearpage\@restonecoltrue
2413   \fi
2414   \columnseprule.4pt \columnsep 2\jsZw
2415   \ifx\multicols\@undefined
2416 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2417 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2418 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2419 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2420   \else
```

```

2421 \ifdim\textwidth<\fullwidth
2422 \setlength{\evensidemargin}{\oddsidemargin}
2423 \setlength{\textwidth}{\fullwidth}
2424 \setlength{\linewidth}{\fullwidth}
2425 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}]%
2426 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]%
2427 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2428 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]%
2429 \else
2430 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}]%
2431 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]%
2432 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2433 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]%
2434 \fi
2435 \fi
2436 %<book|report> \@mkboth{\indexname}{}%
2437 %<!book&!report> \@mkboth{\indexname}{\indexname}%
2438 \plainifnotempty % \thispagestyle{plain}
2439 \parindent\z@
2440 \parskip\z@ \@plus .3\p@?\relax
2441 \let\item\@idxitem
2442 \raggedright
2443 \footnotesize\narrowbaselines
2444 }{
2445 \ifx\multicols\@undefined
2446 \if@restonecol\onecolumn\fi
2447 \else
2448 \end{multicols}
2449 \fi
2450 \clearpage
2451 }

```

`\@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 2452 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2453 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2454 \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。

```

2455 \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。

```

2456 \newcommand\seename{\if@english see\else →\fi}
2457 \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。

```
2458 \ifx\pltx@foot@penalty\@undefined
2459   \let\footnotes@ve=\footnote
2460   \def\footnote{\jsInhibitGlue\footnotes@ve}
2461   \let\footnotemarks@ve=\footnotemark
2462   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2463 \fi
```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

pT_EX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```
2464 \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。

```
2465 % \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。

```

2466 \renewcommand{\footnoterule}{%
2467   \kern-2.6\p@? \kern-.4\p@
2468   \hrule width .4\columnwidth
2469   \kern 2.6\p@?}

```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```

2470 %<book|report>\@addtoreset{footnote}{chapter}

```

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

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

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

```

2471 \long\def\@footnotetext{%
2472   \insert\footins\bgroup
2473     \normalfont\footnotesize
2474     \interlinepenalty\interfootnotelinepenalty
2475     \splittopskip\footnotesep
2476     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2477     \hsize\columnwidth \@parboxrestore
2478     \protected@edef\@currentlabel{%
2479       \csname p@footnote\endcsname\@thefnmark
2480     }%
2481     \color@begingroup
2482     \@makefnmark
2483     \rule{z@\footnotesep\ignorespaces}%
2484     \futurelet\next\fo@t}
2485 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
2486           \else \let\next\fo@t\fi \next}
2487 \def\fo@t{\bgroup\aftergroup\@foot\let\next}
2488 \def\fo@t#1{#1\@foot}
2489 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2490   \ifhmode\null\fi
2491   \ifx\pltx@foot@penalty\undefined\else
2492     \ifnum\pltx@foot@penalty=z@\else
2493       \penalty\pltx@foot@penalty
2494       \pltx@foot@penalty\z@
2495     \fi
2496   \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。

```

2497 \newcommand\@makefnmark[1]{%
2498   \advance\leftskip 3\jsZw
2499   \parindent 1\jsZw
2500   \noindent
2501   \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。

```
2502 % \def\@xfootnotenext[#1]{%
2503 %   \begingroup
2504 %     \ifnum#1>\z@
2505 %       \csname c@\@mpfn\endcsname #1\relax
2506 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2507 %     \else
2508 %       \unrestored@protected@xdef\@thefnmark{}%
2509 %     \fi
2510 %   \endgroup
2511 %   \@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。

```
2512 \let\jsInhibitGlueAtParTop\@empty
```

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

```
2513 \def\everyparhook{\jsInhibitGlueAtParTop}
2514 \AtBeginDocument{\everypar{\everyparhook}}
```

`\@inhibitglue` JS クラスでの `\jsInhibitGlueAtParTop` の実装。

```
2515 \def\@inhibitglue{%
2516   \futurelet\@let@token\@@inhibitglue}
2517 \def\@@inhibitglue{%
2518   \ifx\@let@token「%
2519     \jsInhibitGlue
2520   \else
2521     \ifx\@let@token( %
2522       \jsInhibitGlue
2523     \else
2524       \ifx\@let@token『%
```



```

2525     \jsInhibitGlue
2526   \else
2527     \ifx\@let@token [%
2528       \jsInhibitGlue
2529     \fi
2530   \fi
2531 \fi
2532 \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。

```

2533 \def\@doendpe{%
2534   \@endpetrue
2535   \def\par{%
2536     \@restorepar\everypar{\everyparhook}\par\@endpefalse}%
2537   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

`\item` 命令の直後です。

```

2538 \def\@item[#1]{%
2539   \if@noperitem
2540     \@donoperitem
2541   \else
2542     \if@inlabel
2543       \indent \par
2544     \fi
2545     \ifhmode
2546       \unskip\unskip \par
2547     \fi
2548     \if@newlist
2549       \if@nobreak
2550         \@nbitem
2551       \else
2552         \addpenalty\@beginparpenalty
2553         \addvspace\@topsep
2554         \addvspace{-\parskip}%
2555       \fi
2556     \else
2557       \addpenalty\@itempenalty
2558       \addvspace\itemsep
2559     \fi
2560     \global\@inlabeltrue
2561   \fi
2562   \everypar{%
2563     \@minipagefalse
2564     \global\@newlistfalse
2565     \if@inlabel
2566       \global\@inlabelfalse

```

```

2567     {\setbox\z@\lastbox
2568     \ifvoid\z@
2569     \kern-\itemindent
2570     \fi}%
2571     \box\@labels
2572     \penalty\z@
2573 \fi
2574 \if@nobreak
2575     \@nobreakfalse
2576     \clubpenalty \@M
2577 \else
2578     \clubpenalty \@clubpenalty
2579     \everypar{\everyparhook}%
2580 \fi
2581 \bxjs@ltj@inhibitglue
2582 \everyparhook}%
2583 \if@noitemarg
2584     \@noitemargfalse
2585 \if@nmbrlist
2586     \refstepcounter\@listctr
2587 \fi
2588 \fi
2589 \sbox\@tempboxa{\makelabel{#1}}%
2590 \global\setbox\@labels\hbox{%
2591     \unhbox\@labels
2592     \hskip \itemindent
2593     \hskip -\labelwidth
2594     \hskip -\labelsep
2595     \ifdim \wd\@tempboxa >\labelwidth
2596         \box\@tempboxa
2597     \else
2598         \hbox to\labelwidth {\unhbox\@tempboxa}%
2599     \fi
2600     \hskip \labelsep}%
2601 \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。

```

2602 \def\@afterheading{%
2603     \@nobreaktrue
2604     \everypar{%
2605         \if@nobreak
2606             \@nobreakfalse
2607             \clubpenalty \@M
2608             \if@afterindent \else
2609                 {\setbox\z@\lastbox}%
2610             \fi
2611         \else
2612             \clubpenalty \@clubpenalty

```

```

2613 \everypar{\everyparhook}%
2614 \fi\everyparhook}}

```

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

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

```

2615 \def\@gnewline #1{%
2616   \ifvmode
2617     \@nolnerr
2618   \else
2619     \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
2620     \jsInhibitGlue \ignorespaces
2621   \fi}

```

11 いろいろなロゴ

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

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

BXJS クラスでも `jslogo` オプション指定の場合に `jslogo` パッケージを読み込むようにした。ただし JS クラスと異なり、既定では読み込まない。
 ※`\小`、`\上小` の制御綴は定義しない。

```

2622 \if@jslogo
2623   \IfFileExists{jslogo.sty}{%
2624     \RequirePackage{jslogo}%
2625   }{%
2626     \ClassWarningNoLine{bxjs@clsname}
2627     {The package 'jslogo' is not installed.\MessageBreak
2628      It is included in the recent release of\MessageBreak
2629      the 'jsclasses' bundle}
2630   }
2631 \fi

```

12 amsmath との衝突の回避

最近の $\text{L}\text{A}\text{T}\text{E}\text{X}$ 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 2632 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2633 \newcommand{\postpartname}{\if@english\else 部\fi}
                2634 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2635 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2636 \newcommand{\presectionname}{}% 第
\postsectionname 2637 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 2638 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2639 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
                2640 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 2641 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2642 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
           2643 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 2644 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
           2645 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname
\abstractname 2646 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
           2647 \newcommand{\appendixname}{\if@english \else 付録\fi}
           2648 %<!book&!report>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}
```

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

```
\today
2649 \@tempswafalse
2650 \if p\jsEngine \@tempswatrue \fi
2651 \if n\jsEngine \@tempswatrue \fi
2652 \if@tempswa \expandafter\@firstoftwo
2653 \else \expandafter\@secondoftwo
2654 \fi
2655 {%
2656 % 欧文 8bitTeX の場合
2657 \newif\ifjsSeireki \jsSeirekitrue
2658 \def\西暦{\jsSeirekitrue}
2659 \def\和暦{\jsSeirekifalse}
2660 \def\Seireki{\jsSeirekitrue}
```

```

2661 \def\Wareki{\jsSeirekifalse}
2662 \def\bxjs@if@use@seireki{%
2663   \ifjsSeireki \expandafter\@firstoftwo
2664   \else \expandafter\@secondoftwo \fi}
2665 }{%
2666 \newif\if 西暦 \西暦 true
2667 \def\西暦{\西暦 true}
2668 \def\和暦{\西暦 false}
2669 \def\Seireki{\西暦 true}
2670 \def\Wareki{\西暦 false}
2671 \def\bxjs@if@use@seireki{%
2672   \if 西暦 \expandafter\@firstoftwo
2673   \else \expandafter\@secondoftwo \fi}
2674 }
2675 % \bxjs@unxp
2676 \let\bxjs@unxp\@firstofone
2677 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2678 % \bxjs@iai
2679 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2680 \def\bxjs@iai{\noexpand~}
2681 \else \def\bxjs@iai{}
2682 \fi
2683 % \heisei
2684 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2685 % \today
2686 \edef\today{%
2687   \if@english
2688     \ifcase\month\or
2689       January\or February\or March\or April\or May\or June\or
2690       July\or August\or September\or October\or November\or December\fi
2691     \space\number\day, \number\year
2692   \else
2693     \noexpand\bxjs@if@use@seireki{%
2694       \number\year\bxjs@iai\bxjs@unxp{年}%
2695       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2696       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2697     }{%
2698       \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2699       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2700       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2701     }%
2702   \fi}

```

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

```
2703 \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。

```

2704 %<slide>\pagestyle{empty}%
2705 %<article|report>\pagestyle{plain}%
2706 %<book>\pagestyle{headings}%
2707 \pagenumbering{arabic}
2708 \if@twocolumn
2709     \twocolumn
2710     \sloppy
2711     \flushbottom
2712 \else
2713     \onecolumn
2714     \raggedbottom
2715 \fi
2716 %<*slide>
2717 \renewcommand\familydefault{\sfdefault}
2718 \raggedright
2719 %</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。

```

2720 \catcode\?=12
2721 \ifx\bxjs@jadriver\relax\else
2722 \input{bxjsja-\bxjs@jadriver.def}
2723 \fi

```

最後に日本語文字のカテゴリコードを元に戻す。

```

2724 \bxjs@restore@jltrcc
2725 %</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。

```
2726 %<*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

```
2727 %<*minimal>
```

```
2728 %% 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.
```

```
2729 \def\DeclareJaTextFontCommand#1#2{%
```

```
2730   \DeclareRobustCommand#1[1]{%
```

```
2731     \relax
```

```
2732     \ifmmode \expandafter\nfss@text \fi
```

```
2733     {#2##1}}%
```

```
2734 }
```

```
\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。
```

```
2735 \long\def\bxjs@@CSsfdefault{\sfdefault}%
```

```
2736 \@onlypreamble\bxjs@if@sf@default
```

```
2737 \def\bxjs@if@sf@default#1{%
```

```
2738   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
```

```
2739   \AtBeginDocument{%
```

```
2740     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
```

```
2741 }
```


`\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。

```
2742 \def\jsLetHeadChar#1#2{%
2743   \begingroup
2744   \escapechar=`\ %
2745   \let\bxjs@tmpa={% brace-match-hack
2746   \bxjs@let@hchar@exp#2}%
2747   \endgroup
2748   \let#1\bxjs@g@tmpa}
2749 \def\bxjs@let@hchar@exp{%
2750   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2751 \def\bxjs@let@hchar@exp@a{%
2752   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
2753     \bxjs@let@hchar@out\let\relax
2754   }\bxjs@cond@ifcat\noexpand\@let@token\sptoken\fi{% 空白
2755     \bxjs@let@hchar@out\let\space%
2756   }\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
2757     \bxjs@let@hchar@out\let\@backslashchar
2758   }\bxjs@let@hchar@exp@b}}
2759 \def\bxjs@let@hchar@exp@b#1{%
2760   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2761 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2762   %\message{<#1#2>}%
2763   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴
2764     \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2765       \bxjs@let@hchar@out\let\relax
2766     }{%else
2767       \expandafter\bxjs@let@hchar@exp
2768     }%
2769   }{%else
2770     \bxjs@let@hchar@chr#1%
2771   }}
2772 \def\bxjs@let@hchar@chr#1{%
2773   \bxjs@let@hchar@out\def{#{1}}
2774 \def\bxjs@let@hchar@out#1#2{%
2775   \global#1\bxjs@g@tmpa#2\relax
2776   \toks@\bgroup}% skip to right brace
```

UTF-8 のバイト列を扱うコード。

```
2777 \chardef\bxjs@let@hchar@csta=128
2778 \chardef\bxjs@let@hchar@cstb=192
2779 \chardef\bxjs@let@hchar@cstc=224
```

```

2780 \chardef\bxjs@let@hchar@cstd=240
2781 \chardef\bxjs@let@hchar@cste=248
2782 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2783 \def\bxjs@let@hchar@chr@ue#1{%
2784   \@tempcnta=#1\relax
2785   \%message{\the\@tempcnta}%
2786   \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2787     \bxjs@let@hchar@chr@ue@a#1%
2788   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2789     \bxjs@let@hchar@out\let\relax
2790   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2791     \bxjs@let@hchar@chr@ue@b
2792   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2793     \bxjs@let@hchar@chr@ue@c
2794   }\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2795     \bxjs@let@hchar@chr@ue@d
2796   }{%else
2797     \bxjs@let@hchar@out\let\relax
2798   }}}}
2799 \def\bxjs@let@hchar@chr@ue@a#1{%
2800   \bxjs@let@hchar@out\def{{#1}}}
2801 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2802   \bxjs@let@hchar@out\def{{#1#2}}}
2803 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2804   \bxjs@let@hchar@out\def{{#1#2#3}}}
2805 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2806   \bxjs@let@hchar@out\def{{#1#2#3#4}}}

```

B.2 (u)pTeX 用の設定

```
2807 \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。

```

2808 \def\bxjs@let@hchar@chr@pp#1{%
2809   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
2810 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2811   \%message{(#1)}%
2812   \bxjs@cond\if#1t\fi{%
2813     \bxjs@let@hchar@chr@ue#3%
2814   }{%else
2815     \bxjs@let@hchar@out\def{{#3}}}%
2816   }}
2817 \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。

```
2818 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
2819 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
2820 \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。

```
2821 \@onlypreamble\bxjs@declarefontshape
2822 \ifjsWithupTeX
2823 \def\bxjs@declarefontshape{%
2824 \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
2825 \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpgnt-h}{}%
2826 \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
2827 \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpgnt-v}{}%
2828 }
2829 \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。

```
2830 \else
2831 \def\bxjs@declarefontshape{%
2832 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
2833 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
2834 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
2835 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
2836 }
2837 \def\bxjs@sizereference{jis}
2838 \fi
```

既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。

```
2839 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
2840   \def\bxjs@y{#5}}
2841 \ifjsWithpTeXng \def\bxjs@y{10}%
2842 \else
2843 \expandafter\expandafter\expandafter\bxjs@tmpa
2844 \expandafter\string\the\jfont\relax
2845 \fi
2846 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
2847               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
2848   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
2849    \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。

```
2850 \begingroup
2851 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2852 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2853 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2854 % 幅が丁度 10pt なら補正は不要
2855 \ifdim\wd\z@=10pt
2856 \global\let\bxjs@scale\jsScale
2857 \else
2858 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
2859 \edef\bxjs@tmpa{\strip@pt\wd\z@}
2860 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
2861 \bxjs@invscale\@tempdima\bxjs@tmpa
2862 \xdef\bxjs@scale{\strip@pt\@tempdima}
2863 \fi
2864 \endgroup
2865 %\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。

```
2866 \bxjs@declarefontshape

    フォント代替の明示的定義。

2867 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
2868 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
2869 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
2870 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
2871 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
2872 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
2873 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
2874 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
2875 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
2876 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
2877 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
2878 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
2879 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
2880 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
```

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

```
2881 \DeclareRobustCommand\rmfamily
2882 {\not@math@alphabet\rmfamily\mathrm
2883 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
```

```

2884 \DeclareRobustCommand\sffamily
2885   {\not@math@alphabet\sffamily\mathsf
2886     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
2887 \DeclareRobustCommand\ttfamily
2888   {\not@math@alphabet\ttfamily\mathtt
2889     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
2890 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
2891 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
2892 \bxjs@if@sf@default{%
2893   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

2894 \selectfont

```

■パラメタの設定

```

2895 \prebreakpenalty\jis"2147=10000
2896 \postbreakpenalty\jis"2148=10000
2897 \prebreakpenalty\jis"2149=10000
2898 \inhibitxspcode`!=1
2899 \inhibitxspcode`〒=2
2900 \xspcode`+=3
2901 \xspcode`\%=3

```

"80～"FF の範囲の \spcode を 3 に変更。

```

2902 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
2903   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

\jsInhibitGlueAtParTop の定義。

```

2904 \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。

```

2905 \beginingroup
2906 \catcode`\!=0

```

\bxjs@ptex@dir 現在の組方向： t=縦、y=横、?=その他。

```

2907 \gdef\bxjs@ptex@dir{%
2908   !ifttdir t%
2909   !else!ifydir y%
2910   !else ?%
2911   !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。

```

2912 % 古い \@makefnmark の定義

```

```

2913 \long\def\bxjs@tmpa{\hbox{%
2914   !ifydir \@textsuperscript{\normalfont\@thefnmark}}%
2915   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
2916 \ifx\@makefnmark\bxjs@tmpa
2917 \long\gdef\@makefnmark{%
2918   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
2919   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
2920 \fi
2921 \endgroup

```

B.3 pdfTeX 用の処理

```

2922 \else\ifx p\jsEngine
2923 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
2924 \@onlypreamble\bxjs@cjk@loaded
2925 \def\bxjs@cjk@loaded{%
2926   \def\@footnotemark{%
2927     \leavevmode
2928     \ifhmode
2929       \edef\x@sf{\the\spacefactor}%
2930       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
2931         \unkern\unkern
2932         \ifdim\lastskip>\z@ \unskip \fi
2933       \fi\fi
2934       \nobreak
2935     \fi
2936     \@makefnmark
2937     \ifhmode \spacefactor\x@sf \fi
2938     \relax}%
2939 \let\bxjs@cjk@loaded\relax
2940 }
2941 \AtBeginDocument{%
2942   \@ifpackageloaded{CJK}{%
2943     \bxjs@cjk@loaded
2944   }{}%
2945 }

```

B.4 XeTeX 用の処理

```

2946 \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。

2947 \def\bxjs@let@hchar@chr#1{%
2948   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
2949   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
2950     \bxjs@let@hchar@chr@xe
2951   }\bxjs@let@hchar@out\def{{#1}}}}
2952 \def\bxjs@let@hchar@chr@xe#1{%

```

```

2953 \lccode`0=`#1\relax
2954 \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

2955 \ifx\XeTeXgenerateactualtext\@undefined\else
2956 \def\bxjs@do@precisetext{%
2957 \XeTeXgenerateactualtext=\@ne}
2958 \fi

```

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

```

2959 \@onlypreamble\bxjs@do@simplejasetup
2960 \def\bxjs@do@simplejasetup{%
2961 \ifnum\XeTeXinterchartokenstate>\z@
2962 \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
2963 \jsSimpleJaSetup
2964 \ClassInfo\bxjs@clsname
2965 {'\string\jsSimpleJaSetup' is applied\@gobble}%
2966 \fi\fi}

```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```

2967 \newcommand*{\jsSimpleJaSetup}{%
2968 \XeTeXlinebreaklocale "ja"\relax
2969 \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
2970 \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理 (エンジン共通)

```

2971 \fi\fi\fi

```

`simplejasetup` オプションの処理。

```

2972 \ifx\bxjs@do@simplejasetup\@undefined\else
2973 \AtBeginDocument{%
2974 \ifbxjs@simplejasetup
2975 \bxjs@do@simplejasetup
2976 \fi}
2977 \fi

```

`precisetext` オプションの処理。

```

2978 \ifbxjs@precisetext
2979 \ifx\bxjs@do@precisetext\@undefined
2980 \ClassWarning\bxjs@clsname
2981 {The current engine does not support the\MessageBreak
2982 'precisetext' option\@gobble}
2983 \else
2984 \bxjs@do@precisetext
2985 \fi
2986 \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。

2987 `\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。

2988 `\@onlypreamble\bxjs@adjust@fancyhdr`

2989 `\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。

2990 `\def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%`

2991 `\def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%`

2992 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

2993 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

2994 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

2995 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

2996 `\def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%`

2997 `\def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%`

2998 `\ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi`

2999 `\ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi`

3000 `\ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi`

3001 `\ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi`

3002 `\def\bxjs@tmpa{\rm\thepage\strut}%`

3003 `\def\bxjs@tmpb{\thepage\strut}%`

3004 `\ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi`

3005 `\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。

3006 `\ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth`

3007 `\setlength{\@tempdima}{\fullwidth-\textwidth}%`

3008 `\edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%`

3009 `}\bxjs@tmpa`

3010 `\fi\fi`

3011 `\PackageInfo\bxjs{clsname`

3012 `{Patch to fancyhdr is applied\@gobble}}`

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

3013 `\def\bxjs@pagestyle@hook{%`

3014 `\@ifpackageloaded{fancyhdr}{%`

3015 `\bxjs@adjust@fancyhdr`

3016 `\global\let\bxjs@adjust@fancyhdr\relax`

3017 `}{}}`

`\pagestyle` にフックを入れ込む。

3018 `\let\bxjs@org@pagestyle\pagestyle`


```

3019 \def\pagestyle{%
3020   \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。
3021 \AtBeginDocument{%
3022   \bxjs@pagestyle@hook
3023   \global\let\bxjs@pagestyle@hook\relax}

終わり。
3024 \fi

以上で終わり。
3025 %</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。

```

3026 %<*standard>
3027 %% このファイルは日本語文字を含みます
3028 \input{bxjsja-minimal.def}

simplejasetup は standard では無効になる。
3029 \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。

```
3030 \ifjsWithTeX
```

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 3031 \@tempdima=0.25mm
      3032 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
      3033 \let\jH\jQ
```

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

```
\trueH 3034 \ifjsc@mag
      3035 \@tempdimb=\jsBaseFontSize\relax
      3036 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
      3037 \@tempdima=2.5mm
      3038 \bxjs@invscale\@tempdima\bxjs@tmpa
      3039 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
      3040 \@tempdima=10pt
      3041 \bxjs@invscale\@tempdima\bxjs@tmpa
      3042 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
      3043 \else \let\trueQ\jQ \let\bxjs@truept\p@
      3044 \fi
      3045 \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。

```
3046 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3047 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3048 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3049 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3050 \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。

```
3051 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3052 \newcommand*\setkanjiskip[1]{%
3053 \edef\bxjs@kanjiskip{#1}%
3054 \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3055 \newcommand*\getkanjiskip{%
3056 \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。

```
3057 \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 3058 \bxjs@robust@def\bxjs@enable@kanjiskip{%
3059   \bxjs@kanjiskip@enabledtrue
3060   \bxjs@reset@kanjiskip}
3061 \bxjs@robust@def\bxjs@disable@kanjiskip{%
3062   \bxjs@kanjiskip@enabledfalse
3063   \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。

```
3064 \bxjs@robust@def\bxjs@reset@kanjiskip{%
3065   \ifbxjs@kanjiskip@enabled
3066     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
3067   \else \@tempskipa\z@
3068   \fi
3069   \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 3070 \def\bxjs@xkanjiskip{Opt}
\getxkanjiskip 3071 \newcommand*\setxkanjiskip[1]{%
```

```
3072   \edef\bxjs@xkanjiskip{#1}%
\ifbxjs@xkanjiskip@enabled 3073   \bxjs@reset@xkanjiskip}
\bxjs@enable@xkanjiskip 3074 \newcommand*\getxkanjiskip{%
\bxjs@disable@xkanjiskip 3075   \bxjs@xkanjiskip}
3076 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
\bxjs@reset@xkanjiskip 3077 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
3078   \bxjs@xkanjiskip@enabledtrue
3079   \bxjs@reset@xkanjiskip}
3080 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
3081   \bxjs@xkanjiskip@enabledfalse
3082   \bxjs@reset@xkanjiskip}
3083 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
3084   \ifbxjs@xkanjiskip@enabled
3085     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3086   \else \@tempskipa\z@
3087   \fi
3088   \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。

```
3089 \g@addto@macro\jsResetDimen{%
3090   \bxjs@reset@kanjiskip
3091   \bxjs@reset@xkanjiskip}
3092 \let\bxjs@apply@kanjiskip\relax
3093 \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…。

```
3094 \@onlypreamble\bxjs@adjust@jafont
3095 \def\bxjs@adjust@jafont#1{%
3096   \ifx\jsJaFont\bxjs@@auto
3097     \bxjs@get@kanjiEmbed
3098     \ifx\bxjs@kanjiEmbed\relax
3099       \let\bxjs@tmpa\@empty
3100     \else
3101       \let\bxjs@tmpa\bxjs@kanjiEmbed
3102     \fi
3103   \else
3104     \let\bxjs@tmpa\jsJaFont
3105   \fi
3106   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3107     \ClassWarningNoLine\bxjs@clsname
3108     {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3109      not available on the current situation}%
3110     \let\bxjs@tmpa\@empty
3111   \fi\fi
3112 }
3113 \def\bxjs@@auto{auto}
3114 \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。

```
3115 \let\bxjs@kanjiEmbed\relax
```

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

```
3116 \@onlypreamble\bxjs@get@kanjiEmbed
3117 \def\bxjs@get@kanjiEmbed{%
3118   \begingroup\setbox\z@=\hbox{%
3119     \global\let\bxjs@g@tmpa\relax
3120     \endlinechar\m@ne
3121     \let\do\@makeother\dospecials
3122     \catcode32=10 \catcode12=10 %form-feed
3123     \let\bxjs@tmpa\@empty
3124     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3125     \ifeof\@inputcheck\else
3126       \read\@inputcheck to\bxjs@tmpa
3127       \closein\@inputcheck
3128     \fi
3129     \ifx\bxjs@tmpa\@empty\else
3130       \openin\@inputcheck="\bxjs@tmpa"\relax
3131       \@tempwattrue
3132       \loop\if@tempwa
3133         \read\@inputcheck to\bxjs@tmpa
3134         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
```

```

3135         \ifx\bxjs@tmpa\relax\else
3136         \global\let\bxjs@g@tmpa\bxjs@tmpa
3137         \@tempwafalse
3138         \fi
3139         \ifeof\@inputcheck \@tempwafalse \fi
3140     \repeat
3141 \fi
3142 }\endgroup
3143 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3144 }
3145 \@onlypreamble\bxjs@get@ke@a
3146 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3147     \ifx$#1$\def\bxjs@tmpa{#2}%
3148     \else \let\bxjs@tmpa\relax
3149     \fi}

```

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

```

3150 \newcommand*\jachar[1]{%
3151     \begingroup
3152     \jsLetHeadChar\bxjs@tmpa{#1}%
3153     \ifx\bxjs@tmpa\relax
3154         \ClassWarningNoLine\bxjs@clsname
3155         {Illegal argument given to \string\jachar}%
3156     \else
3157         \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3158     \fi
3159 \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3160 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```

3161 \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。

```

3162 \PassOptionsToPackage{setpagesize=false}{hyperref}

```

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

```

3163 \@onlypreamble\bxjs@fix@hyperref@unicode
3164 \def\bxjs@fix@hyperref@unicode#1{%
3165     \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3166     \@namedef{KV@Hyp@bxjs/hook}##1{%
3167         \KV@Hyp@unicode{##1}%
3168         \def\KV@Hyp@unicode####1{%
3169             \expandafter\ifx\csname if##1\expandafter\endcsname
3170             \csname if####1\endcsname\else

```

```

3171      \ClassWarningNoLine\bxjs@clsname
3172      {Blcoked hyperref option 'unicode=###1'}%
3173      \fi
3174      }%
3175      }%
3176 }

```

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

```

3177 \onlypreamble\bxjs@urgent@special
3178 \def\bxjs@urgent@special#1{%
3179   \AtBeginDvi{\special{#1}}%
3180   \AtBeginDocument{%
3181     \ifpackageloaded{atbegshi}{%
3182       \begingroup
3183       \toks\z@\special{#1}}%
3184       \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3185       \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3186     \endgroup
3187   }{}%
3188   }%
3189 }

```

C.2 pTeX 用設定

```
3190 \if j\jsEngine
```

■ 共通命令の実装

```

3191 \def\bxjs@apply@kanjiskip{%
3192   \kanjiskip\@tempkipa}
3193 \def\bxjs@apply@xkanjiskip{%
3194   \xkanjiskip\@tempkipa}

```

`\jaJaChar` のサブマクロ。

```

3195 \def\bxjs@jachar#1{%
3196   \bxjs@jachar@a#1...\@nil}
3197 \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。

```
3198   \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。

```

3199   \else\ifx.#3%
3200     \@tempcnta`#1 \multiply\@tempcnta64
3201     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3202     \bxjs@jachar@b
3203   \else\ifx.#4%
3204     \@tempcnta`#1 \multiply\@tempcnta64
3205     \advance\@tempcnta`#2 \multiply\@tempcnta64

```

```

3206 \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3207 \bxjs@jachar@b
3208 \else
3209 \@tempcnta`#1 \multiply\@tempcnta64
3210 \advance\@tempcnta`#2 \multiply\@tempcnta64
3211 \advance\@tempcnta`#3 \multiply\@tempcnta64
3212 \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3213 \bxjs@jachar@b
3214 \fi\fi\fi}

```

符号値が \@tempcnta の和文文字を出力する処理。

```

3215 \ifjsWithupTeX
3216 \def\bxjs@jachar@b{\kchar\@tempcnta}
3217 \else
3218 \def\bxjs@jachar@b{%
3219 \ifx\bxUInt\@undefined\else
3220 \bxUInt{\@tempcnta}%
3221 \fi}
3222 \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。

```

3223 \let\bxjs@tmpa\jsJaFont
3224 \ifx\bxjs@tmpa\bxjs@@auto
3225 \let\bxjs@tmpa\@empty
3226 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3227 \def\bxjs@tmpa{noembed}
3228 \fi\fi
3229 \ifx\jsJaFont\@empty\else
3230 \edef\bxjs@nxt{%
3231 \noexpand\RequirePackage[\jsJaFont]
3232 {pxchfon}[2010/05/12]}% v0.5
3233 \bxjs@nxt
3234 \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……。)

```

3235 \begingroup
3236 \global\let\@gtempa\relax
3237 \catcode`\|=0 \catcode`\|=12
3238 |def|bxjs@check#1|@nil{%
3239 |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3240 |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3241 |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%

```

```

3242 |catcode`\|=0 \catcode`\|=12
3243 \def\bxjs@check@b#1keyval#2\@nnil{%
3244   \ifx$#2$\else
3245     \xdef\@gtempa{%
3246       \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3247     \fi}
3248 \@firstofone{%
3249   \catcode10=12 \endlinechar\m@ne
3250   \let\do\@makeother \dospecials \catcode32=10
3251   \openin\@inputcheck=otf.sty\relax
3252   \@tempswatrue
3253   \loop\if@tempswa
3254     \ifeof\@inputcheck \@tempswafalse \fi
3255     \if@tempswa
3256       \read\@inputcheck to\bxjs@line
3257       \expandafter\bxjs@check\bxjs@line\@nil
3258     \fi
3259   \repeat
3260   \closein\@inputcheck
3261 \endgroup}
3262 \@gtempa

```

■hyperref 対策 unicode にはいけない。

```

3263 \bxjs@fix@hyperref@unicode{false}

tounicode special 命令を出力する。

3264 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3265   \else\ifjsWithpTeXng T\else F\fi\fi T%
3266   \ifnum\jis"2121="A1A1 %euc
3267     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3268   \else\ifnum\jis"2121="8140 %sjis
3269     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3270   \else\ifnum\jis"2121="3000 %uptex
3271     \ifbxjs@bigcode
3272       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3273       \PassOptionsToPackage{bigcode}{pxjahyper}
3274     \else
3275       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3276     \fi
3277   \fi\fi\fi
3278   \let\bxToUnicodeSpecialDone=t
3279 \fi

```

■microtype 対策

```

3280 \@namedef{ver@microtype.sty}{2000/01/01}
3281 \newcommand*\UseMicrotypeSet[2][{}]{

```


C.3 pdfTeX 用設定：CJK + bxcjkatype

```
3282 \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。

```
3283 \bxjs@adjust@jafont{f}
3284 \edef\bxjs@nxt{%
3285   \noexpand\RequirePackage[%
3286     \ifx\bxjs@tmpa\@empty\else \bxjs@tmpa,\fi
3287     whole,autotilde]{bxcjkatype}[2013/10/15]}% v0.2c
3288 \bxjs@nxt
3289 \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。

```
3290 \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。

```
3291 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3292 \begingroup
3293   \CJK@input{UTF8.bdg}
3294 \endgroup
3295 \g@addto@macro\pdfstringdefPreHook{%
3296   \@nameuse{CJK@UTF8Binding}%
3297 }
3298 \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。

3299 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3300 \g@addto@macro\pdfstringdefPreHook{%
3301   \ifx~\bxjs@@CJKtilde
3302     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3303     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3304     \let~\@empty
3305   \fi
3306 }
3307 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3308 \def\bxjs@@tildecmd{~}
3309 \def\bxjs@LetUnexpandableSpace#1{%
3310   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
```

```

3311 \bxjs@org@LetUnexpandableSpace#1%
3312 \fi}
3313 \fi

```

■共通命令の実装

```

3314 \newskip\jsKanjiSkip
3315 \newskip\jsXKanjiSkip
3316 \ifx\CJKecglue\@undefined
3317 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3318 \fi
3319 \let\autospacing\bxjs@enable@kanjiskip
3320 \let\noautospacing\bxjs@disable@kanjiskip
3321 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3322 \def\bxjs@apply@kanjiskip{%
3323 \jsKanjiSkip\@tempskipa
3324 \let\CJKglue\bxjs@CJKglue}
3325 \let\autoxspacing\bxjs@enable@xkanjiskip
3326 \let\noautoxspacing\bxjs@disable@xkanjiskip
3327 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3328 \def\bxjs@apply@xkanjiskip{%
3329 \jsXKanjiSkip\@tempskipa
3330 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

3331 \def\bxjs@jachar#1{%
3332 \CJKforced{#1}}

```

C.4 Xe_{La}TeX 用設定：xeCJK + zxjatype

```

3333 \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。

```

3334 \RequirePackage{zxjatype}
3335 \PassOptionsToPackage{no-math}{fontspec}%!
3336 \PassOptionsToPackage{xetex}{graphicx}%!
3337 \PassOptionsToPackage{xetex}{graphics}%!
3338 \ifx\zxJaFamilyName\@undefined
3339 \ClassError\bxjs@clsname
3340 {xeCJK or zxjatype is too old}\@ehc
3341 \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。

```

3342 \bxjs@adjust@jafont{f}
3343 \ifx\bxjs@tmpa\@empty
3344 \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3345 \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3346 \else

```

```

3347 \edef\bxjs@nxt{%
3348   \noexpand\RequirePackage[\bxjs@tmpa]%
3349     {zxjafont}[2013/01/28]}% v0.2a
3350 \bxjs@nxt
3351 \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。

```

3352 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3353   \PassOptionsToPackage{unicode}{hyperref}
3354 \fi

```

■段落頭でのグルー挿入禁止

```

3355 \let\jsInhibitGlueAtParTop\@inhibitglue

```

■共通命令の実装

```

3356 \newskip\jsKanjiSkip
3357 \newskip\jsXKanjiSkip
3358 \ifx\CJKecglue\@undefined
3359   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3360 \fi
3361 \let\autospacing\bxjs@enable@kanjiskip
3362 \let\noautospacing\bxjs@disable@kanjiskip
3363 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3364 \def\bxjs@apply@kanjiskip{%
3365   \jsKanjiSkip\@tempskipa
3366   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3367 \let\autoxspacing\bxjs@enable@xkanjiskip
3368 \let\noautoxspacing\bxjs@disable@xkanjiskip
3369 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3370 \def\bxjs@apply@xkanjiskip{%
3371   \jsXKanjiSkip\@tempskipa
3372   \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。

```

3373 \ifx\mcfamily\@undefined
3374   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}

```

```

3375 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3376 \fi

\jachar のサブマクロの実装。
3377 \def\bxjs@jachar#1{%
3378 \xeCJKDeclareCharClass{CJK}{`#1}\relax
3379 #1}

```

C.5 LuaTeX 用設定：LuaTeX-ja

```
3380 \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。

```

3381 \let\zw\@undefined
3382 \RequirePackage{luatexja}
3383 \RequirePackage{luatexja-fontspec}
3384 \PassOptionsToPackage{pdftex}{graphicx}%!
3385 \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？

```

3386 \ExplSyntaxOn
3387 \fp_gset:Nn \g_ltj_fontspec_scale_fp { \jsScale }
3388 \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）。

```

3389 \bxjs@adjust@jafont{t}
3390 \ifx\bxjs@tmpa\bxjs@@noEmbed
3391 \def\bxjs@tmpa{noembed}
3392 \fi
3393 \ifx\bxjs@tmpa@empty
3394 \defaultjfontfeatures{ Kerning=Off }
3395 \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3396 \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3397 \else
3398 \edef\bxjs@nxt{%
3399 \noexpand\RequirePackage[\bxjs@tmpa]
3400 {luatexja-preset}}%
3401 \bxjs@nxt
3402 \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。

```
3403 \DeclareRobustCommand\rmfamily
3404   {\not@math@alphabet\rmfamily\mathrm
3405     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3406 \DeclareRobustCommand\sffamily
3407   {\not@math@alphabet\sffamily\mathsf
3408     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3409 \DeclareRobustCommand\ttfamily
3410   {\not@math@alphabet\ttfamily\mathtt
3411     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3412 \AtBeginDocument{%
3413   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3414   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}}%
3415 \bxjs@if@sf@default{%
3416   \renewcommand\kanjifamilydefault{\gtdefault}}
```

■和文パラメタの設定

```
3417 % 次の3つは既定値の通り
3418 %\ltjsetparameter{prebreakpenalty={` ,10000}}
3419 %\ltjsetparameter{postbreakpenalty={` “,10000}}
3420 %\ltjsetparameter{prebreakpenalty={` ”,10000}}
3421 \ltjsetparameter{jaxspmode={` ! ,1}}
3422 \ltjsetparameter{jaxspmode={` 〒,2}}
3423 \ltjsetparameter{alxspmode={` +,3}}
3424 \ltjsetparameter{alxspmode={` \%,3}}
```

■段落頭でのグルー挿入禁止

```
3425 \protected\def\@inhibitglue{%
3426   \directlua{%
3427     luatexja.jfmglue.create_beginpar_node()}}
3428 \let\bxjs@ltj@inhibitglue\@inhibitglue
3429 \let\@inhibitglue\undefined
```

■hyperref 対策 unicode にするべき。

```
3430 \bxjs@fix@hyperref@unicode{true}
```

■共通命令の実装

```
3431 \protected\def\autospacing{%
3432   \ltjsetparameter{autospacing=true}}
3433 \protected\def\noautospacing{%
3434   \ltjsetparameter{autospacing=false}}
3435 \protected\def\autoxspacing{%
3436   \ltjsetparameter{autoxspacing=true}}
3437 \protected\def\noautoxspacing{%
3438   \ltjsetparameter{autoxspacing=false}}
3439 \def\bxjs@apply@kanjiskip{%
3440   \ltjsetparameter{kanjiskip={\@tempskipa}}}
```

```

3441 \def\bxjs@apply@xkanjiskip{%
3442   \ltjsetparameter{xkanjiskip={\@tempskipa}}}}

```

\jachar のサブマクロの実装。

```

3443 \def\bxjs@jachar#1{%
3444   \ltjjachar`#1\relax}

```

C.6 共通処理 (2)

```

3445 \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 3446 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3447 \DeclareJaTextFontCommand{\textgt}{\gtfamily}

```

■和文・和欧文間空白の初期値

```

3448 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3449 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3450 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3451 \fi

```

以上で終わり。

```

3452 %</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。

```

3453 %<*modern>
3454 \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。

```

3455 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3456 \def\encodingdefault{T1}%
3457 \input{t1enc.def}%
3458 \fontencoding\encodingdefault\selectfont
3459 \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。

```

3460 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
3461 \renewcommand{\rmdefault}{\lmr}
3462 \renewcommand{\sfdefault}{\lms}
3463 \renewcommand{\ttdefault}{\lmtt}
3464 \fi

    大型演算子用の数式フォントの設定。
    ※ amsmath パッケージと同等にする。
3465 \DeclareFontShape{OMX}{cmex}{m}{n}{%
3466   <-7.5>cmex7<7.5-8.5>cmex8%
3467   <8.5-9.5>cmex9<9.5->cmex10}{}%
3468 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax

    amsmath 読込時に上書きされるのを防ぐ。
3469 \def\cmex@opt{10}

```

D.2 fixltx2e 読込

※ `fixltx2e` 廃止前の \LaTeX カーネルの場合。

```

3470 \ifx\@IncludeInRelease\@undefined
3471 \RequirePackage{fixltx2e}
3472 \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。

```

3473 \RequirePackage{bxjsjkat}

```

D.4 完了

おしまい。

```

3474 %</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。

```

3475 %<*pandoc>
3476 \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。

```
3477 \@onlypreamble\bxjs@set@dupload@proc
3478 \def\bxjs@set@dupload@proc#1{%
3479   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
3480 \@onlypreamble\bxjs@set@dupload@proc@a
3481 \def\bxjs@set@dupload@proc@a#1{%
3482   \@onlypreamble#1\def#1##1}
```

`\@if@ptions` `\@if@ptions` の再定義。

```
3483 \@onlypreamble\bxjs@org@if@ptions
3484 \let\bxjs@org@if@ptions\@if@ptions
3485 \newif\ifbxjs@dlp
3486 \def\@if@ptions#1#2#3{%
3487   \bxjs@dlpfalse
3488   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
3489   \ifx\bxjs@tmpa\bxjs@tmpb
3490     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
3491       \bxjs@dlptrue \fi
3492   \fi
3493   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
3494   \else \expandafter\bxjs@org@if@ptions
3495   \fi {#1}{#2}{#3}}
3496 \AtBeginDocument{%
3497   \let\@if@ptions\bxjs@org@if@ptions}
3498 \@onlypreamble\bxjs@do@dupload@proc
3499 \def\bxjs@do@dupload@proc#1#2#3{%
3500   \csname bxjs@dlp/#2.#1\endcsname{#3}%
3501   \@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。

```
3502 \def\bxjs@mark@as@loaded#1{%
3503   \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3504 \def\bxjs@mal@a#1#2{%
3505   \ifx#1\relax
3506     \def#1{2001/01/01}%
3507     \ClassInfo\bxjs@clsname
3508     {File '#2' marked as loaded\@gobble}%
3509   \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。

```
3510 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0
```

Polyglossia について。

```
3511 \bxjs@mark@as@loaded{polyglossia.sty}
3512 \bxjs@set@dupload@proc{polyglossia.sty}{%
3513   \ClassWarning\bxjs@clsname
3514     {Loading of polyglossia is blocked}}
3515 \ifx\setmainlanguage\@undefined
3516 \newcommand*\setmainlanguage[2][]{%
3517 \newcommand*\setotherlanguage[2][]{%
3518   \ifcat_#2_\else
3519     \expandafter\let\csname #2\endcsname\@empty
3520     \expandafter\let\csname end#2\endcsname\@empty
3521     \expandafter\let\csname text#2\endcsname\@firstofone
3522   \fi}
3523 \newcommand*\setotherlanguages[2][]{%
3524   \@for\bxjs@tmpa:={#2}\do{%
3525     \setotherlangauge{\bxjs@tmpa}}}
3526 \fi
3527 \else
```

Babel について。

```
3528 \bxjs@mark@as@loaded{babel.sty}
3529 \bxjs@set@dupload@proc{babel.sty}{%
3530   \ClassWarning\bxjs@clsname
3531     {Loading of babel is blocked}}
3532 \let\foreignlanguage\@secondoftwo
3533 \let\otherlanguage\@gobble
3534 \let\endotherlanguage\@empty
3535 \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。

```
3536 \bxjs@set@dupload@proc{geometry.sty}{%
3537   \setpagelayout*{#1}}
```

E.4 CJKmainfont 変数

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

```
3538 \if 1\jsEngine
3539   \bxjs@mark@as@loaded{xeCJK.sty}
3540   \providecommand*\setCJKmainfont{\setmainjfont}
3541 \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。

```
3542 \ifx\@IncludeInRelease\@undefined\else
3543   \bxjs@mark@as@loaded{fixltx2e.sty}
3544 \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。）

```
3545 \if j\jsEngine
3546   \bxjs@mark@as@loaded{cmap.sty}
3547 \fi
```

E.7 microtype パッケージ

警告が多すぎなので消す。

```
3548 \PassOptionsToPackage{verbose=silent}{microtype}
```

E.8 完了

おしまい。

```
3549 %</pandoc>
```

和文ドライバ実装はここまで。

```
3550 %</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。

```
3551 %<*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

```
3552 %<*compat>
3553 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

3554 \let\bxac@engine=n
3555 \def\bxac@do#1#2{%
3556   \edef\bxac@tmpa{\string#1}%
3557   \edef\bxac@tmpb{\meaning#1}%
3558   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
3559 \bxac@do\XeTeXversion{\let\bxac@engine=x}
3560 \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。

3561 \ifx\jsAtEndOfClass\@undefined
3562   \let\bxac@delayed@if@bxjs\@firstofone
3563 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
3564 \fi

\ImposeOldLuaTeXBehavior ムニャムニャ。

\RevokeOldLuaTeXBehavior 3565 \newif\ifbxac@in@old@behavior
3566 \let\ImposeOldLuaTeXBehavior\relax
3567 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 XeTeX 部分

```
3568 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。

3569 \@onlypreamble\bxac@adjust@charclass
3570 \bxac@delayed@if@bxjs{%
3571   \@ifpackageloaded{xeCJK}{-}{%else
3572     \ifx\xe@alloc@intercharclass\@undefined\else
3573       \ifnum\xe@alloc@intercharclass=\z@
3574         \PackageInfo\bxac@pkgname
3575           {Setting up interchar class for CJK...\@gobble}%
3576         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
3577           \xe@alloc@intercharclass=3
3578         }{%else
3579           \PackageWarning\bxac@pkgname
3580             {Cannot find file 'load-unicode-xetex-classes.tex'%
3581             \@gobble}%
3582         }%
3583       \fi\fi
3584     \ifnum\XeTeXcharclass"3041=\z@
3585       \PackageInfo\bxac@pkgname
```

```

3586      {Adjusting interchar class for CJK...\@gobble}%
3587      \@for\bxac@x:={%
3588          3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3589          3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3590          30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3591          31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3592          31FF%
3593      }\do{\XeTeXcharclass"\bxac@x=\@ne}%
3594  \fi
3595  }%
3596 }

```

以上。

```
3597 \fi
```

G.3 LuaTeX 部分

```

3598 \ifx l\bxac@engine
      ムニヤムニヤ。
3599 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
3600   \chardef\pdftexversion=200
3601   \def\pdftexrevision{0}
3602   \let\pdftexbanner\luatexbanner
3603 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior
3604 \begingroup\expandafter\expandafter\expandafter\endgroup
3605 \expandafter\ifx\csname outputmode\endcsname\relax\else
3606 \def\bxac@ob@list{%
3607   \do{\let}\pdfoutput{\outputmode}%
3608   \do{\let}\pdfpagewidth{\pagewidth}%
3609   \do{\let}\pdfpageheight{\pageheight}%
3610   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}%
3611   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}
3612 \def\bxac@ob@do#1#2{\begingroup
3613   \expandafter\bxac@ob@do@a\csname bxac@string#2\endcsname{#1}#2}
3614 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3615   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3616   \else \let#3#1\relax \let#1\@undefined
3617   \fi}
3618 \protected\def\ImposeOldLuaTeXBehavior{%
3619   \unless\ifbxac@in@old@behavior
3620     \bxac@in@old@behaviortrue
3621     \let\do\bxac@ob@do \bxac@ob@list
3622   \fi}
3623 \protected\def\RevokeOldLuaTeXBehavior{%
3624   \ifbxac@in@old@behavior
3625     \bxac@in@old@behaviorfalse
3626     \let\do\bxac@ob@do \bxac@ob@list
3627   \fi}

```

3628 \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。

```
3629 \ifnum\luatexversion>64 \directlua{
3630   local function range(cs, ce, cc, ff)
3631     if ff or not tex.getcatcode(cs) == cc then
3632       local setcc = tex.setcatcode
3633       for c = cs, ce do setcc(c, cc) end
3634     end
3635   end
3636   range(0x3400, 0x4DB5, 11, false)
3637   range(0x4DB5, 0x4DBF, 11, true)
3638   range(0x4E00, 0x9FCC, 11, false)
3639   range(0x9FCD, 0x9FFF, 11, true)
3640   range(0xAC00, 0xD7A3, 11, false)
3641   range(0x20000, 0x2A6D6, 11, false)
3642   range(0x2A6D7, 0x2A6FF, 11, true)
3643   range(0x2A700, 0x2B734, 11, false)
3644   range(0x2B735, 0x2B73F, 11, true)
3645   range(0x2B740, 0x2B81D, 11, false)
3646   range(0x2B81E, 0x2B81F, 11, true)
3647   range(0x2B820, 0x2CEA1, 11, false)
3648   range(0x2CEA2, 0x2FFFD, 11, true)
3649 } \fi
```

以上。

3650 \fi

G.4 完了

おしまい。

3651 %</compat>

付録 H 補助パッケージ : bxjscjkat

modern ドライバ用の和文カテゴリを適用する。

H.1 準備

```
3652 %<*cjkat>
3653 \def\bxjx@pkgname{bxjscjkat}
3654 \newcount\bxjx@canta

\bxjx@engine エンジンの種別。

3655 \let\bxjx@engine=n
3656 \def\bxjx@do#1#2{%
3657   \edef\bxjx@tmpa{\string#1}%
3658   \edef\bxjx@tmpb{\meaning#1}%
```

```

3659 \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3660 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3661 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3662 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3663 \bxjx@do\pdfTeXversion{\let\bxjx@engine=p}
3664 \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。

```

3665 \def\bxjx@do#1#2{%
3666   \if#1\bxjx@engine
3667     \@ifpackageloaded{#2}{\}%else
3668     \PackageError\bxjx@pkgname
3669       {Package '#2' must be loaded}%
3670       {Package loading is aborted.\MessageBreak\@ehc}%
3671     \endinput}
3672 \fi}
3673 \bxjx@do{p}{bxcjkatype}
3674 \bxjx@do{x}{xeCJK}
3675 \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。

```

3676 \ifx\TextOrMath\@undefined
3677   \RequirePackage{fixltx2e}
3678 \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。

```

3679 \if u\bxjx@engine
3680 \@for\bxjx@x:={%
3681 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3682 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3683 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3684 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3685 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3686 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
3687 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3688 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
3689 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3690 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
3691 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3692 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3693 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3694 10480,10500,10530,10600,10800,10840,10860,10880,%
3695 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%

```

```

3696 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3697 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3698 11280,112B0,11300,11480,11580,11600,11680,11700,%
3699 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3700 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3701 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3702 1F030,1FOA0,1F100,1F200,1F300,1F600,1F650,1F680,%
3703 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3704 } \do{\kcatcode"\bxjx@x=15 }
3705 \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。

```

3706 \def\bxjx@grkcyr@list{%
3707 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
3708 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
3709 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
3710 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
3711 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
3712 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
3713 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
3714 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
3715 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
3716 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
3717 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
3718 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
3719 \do{039D}{LGR}{\textNu}{N}%             % GR. C. L. NU
3720 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
3721 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
3722 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
3723 \do{03A1}{LGR}{\textRho}{P}%            % GR. C. L. RHO
3724 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
3725 \do{03A4}{LGR}{\textTau}{T}%            % GR. C. L. TAU
3726 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
3727 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
3728 \do{03A7}{LGR}{\textChi}{X}%            % GR. C. L. CHI
3729 \do{03A8}{LGR}{\textPsi}{\Psi}%         % GR. C. L. PSI
3730 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA
3731 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
3732 \do{03B2}{LGR}{\textbeta}{\beta}%       % GR. S. L. BETA
3733 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA
3734 \do{03B4}{LGR}{\textdelta}{\delta}%     % GR. S. L. DELTA
3735 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
3736 \do{03B6}{LGR}{\textzeta}{\zeta}%        % GR. S. L. ZETA

```

3737 \do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
3738 \do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
3739 \do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
3740 \do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
3741 \do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
3742 \do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
3743 \do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
3744 \do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
3745 \do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
3746 \do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
3747 \do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
3748 \do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
3749 \do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
3750 \do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
3751 \do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
3752 \do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
3753 \do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
3754 \do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
3755 \do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
3756 \do{0401}{T2A}{\CYRYO}{\O}%	% CY. C. L. IO
3757 \do{0410}{T2A}{\CYRA}{\A}%	% CY. C. L. A
3758 \do{0411}{T2A}{\CYRB}{\B}%	% CY. C. L. BE
3759 \do{0412}{T2A}{\CYRV}{\V}%	% CY. C. L. VE
3760 \do{0413}{T2A}{\CYRG}{\G}%	% CY. C. L. GHE
3761 \do{0414}{T2A}{\CYRD}{\D}%	% CY. C. L. DE
3762 \do{0415}{T2A}{\CYRE}{\E}%	% CY. C. L. IE
3763 \do{0416}{T2A}{\CYRZH}{\ZH}%	% CY. C. L. ZHE
3764 \do{0417}{T2A}{\CYRZ}{\Z}%	% CY. C. L. ZE
3765 \do{0418}{T2A}{\CYRI}{\I}%	% CY. C. L. I
3766 \do{0419}{T2A}{\CYRISHRT}{\SHRT}%	% CY. C. L. SHORT I
3767 \do{041A}{T2A}{\CYRK}{\K}%	% CY. C. L. KA
3768 \do{041B}{T2A}{\CYRL}{\L}%	% CY. C. L. EL
3769 \do{041C}{T2A}{\CYRM}{\M}%	% CY. C. L. EM
3770 \do{041D}{T2A}{\CYRN}{\N}%	% CY. C. L. EN
3771 \do{041E}{T2A}{\CYRO}{\O}%	% CY. C. L. O
3772 \do{041F}{T2A}{\CYRP}{\P}%	% CY. C. L. PE
3773 \do{0420}{T2A}{\CYRR}{\R}%	% CY. C. L. ER
3774 \do{0421}{T2A}{\CYRS}{\S}%	% CY. C. L. ES
3775 \do{0422}{T2A}{\CYRT}{\T}%	% CY. C. L. TE
3776 \do{0423}{T2A}{\CYRU}{\U}%	% CY. C. L. U
3777 \do{0424}{T2A}{\CYRF}{\F}%	% CY. C. L. EF
3778 \do{0425}{T2A}{\CYRH}{\H}%	% CY. C. L. HA
3779 \do{0426}{T2A}{\CYRC}{\C}%	% CY. C. L. TSE
3780 \do{0427}{T2A}{\CYRCH}{\CH}%	% CY. C. L. CHE
3781 \do{0428}{T2A}{\CYRSH}{\SH}%	% CY. C. L. SHA
3782 \do{0429}{T2A}{\CYRSHCH}{\SHCH}%	% CY. C. L. SHCHA
3783 \do{042A}{T2A}{\CYRHRDSN}{\HRDSN}%	% CY. C. L. HARD SIGN
3784 \do{042B}{T2A}{\CYRERY}{\ERY}%	% CY. C. L. YERU
3785 \do{042C}{T2A}{\CYRSFTSN}{\SFTSN}%	% CY. C. L. SOFT SIGN


```

3786 \do{042D}{T2A}{\CYREREV}{}% % CY. C. L. E
3787 \do{042E}{T2A}{\CYRYU}{}% % CY. C. L. YU
3788 \do{042F}{T2A}{\CYRYA}{}% % CY. C. L. YA
3789 \do{0430}{T2A}{\cyra}{}% % CY. S. L. A
3790 \do{0431}{T2A}{\cyrb}{}% % CY. S. L. BE
3791 \do{0432}{T2A}{\cyrv}{}% % CY. S. L. VE
3792 \do{0433}{T2A}{\cyrg}{}% % CY. S. L. GHE
3793 \do{0434}{T2A}{\cyrd}{}% % CY. S. L. DE
3794 \do{0435}{T2A}{\cyre}{}% % CY. S. L. IE
3795 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE
3796 \do{0437}{T2A}{\cyrz}{}% % CY. S. L. ZE
3797 \do{0438}{T2A}{\cyri}{}% % CY. S. L. I
3798 \do{0439}{T2A}{\cyrishrt}{}% % CY. S. L. SHORT I
3799 \do{043A}{T2A}{\cyrk}{}% % CY. S. L. KA
3800 \do{043B}{T2A}{\cyrl}{}% % CY. S. L. EL
3801 \do{043C}{T2A}{\cyrm}{}% % CY. S. L. EM
3802 \do{043D}{T2A}{\cyrn}{}% % CY. S. L. EN
3803 \do{043E}{T2A}{\cyro}{}% % CY. S. L. O
3804 \do{043F}{T2A}{\cyrp}{}% % CY. S. L. PE
3805 \do{0440}{T2A}{\cyrr}{}% % CY. S. L. ER
3806 \do{0441}{T2A}{\cyrs}{}% % CY. S. L. ES
3807 \do{0442}{T2A}{\cyrt}{}% % CY. S. L. TE
3808 \do{0443}{T2A}{\cyru}{}% % CY. S. L. U
3809 \do{0444}{T2A}{\cyrf}{}% % CY. S. L. EF
3810 \do{0445}{T2A}{\cyrh}{}% % CY. S. L. HA
3811 \do{0446}{T2A}{\cyrc}{}% % CY. S. L. TSE
3812 \do{0447}{T2A}{\cyrch}{}% % CY. S. L. CHE
3813 \do{0448}{T2A}{\cyrsh}{}% % CY. S. L. SHA
3814 \do{0449}{T2A}{\cyrshch}{}% % CY. S. L. SHCHA
3815 \do{044A}{T2A}{\cyrhrdsn}{}% % CY. S. L. HARD SIGN
3816 \do{044B}{T2A}{\cyrery}{}% % CY. S. L. YERU
3817 \do{044C}{T2A}{\cyrsftsn}{}% % CY. S. L. SOFT SIGN
3818 \do{044D}{T2A}{\cyrerev}{}% % CY. S. L. E
3819 \do{044E}{T2A}{\cyryu}{}% % CY. S. L. YU
3820 \do{044F}{T2A}{\cyrya}{}% % CY. S. L. YA
3821 \do{0451}{T2A}{\cyryo}{}% % CY. S. L. IO
3822 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
3823 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
3824 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
3825 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN
3826 \do{00B4}{TS1}{\textasciicircum}{}% % ACUTE ACCENT
3827 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
3828 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
3829 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN
3830 }

3831 \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。

```

3832 \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。

3833 \newcommand*\greekasCJK{%
3834   \bxjx@gcc@CJKtrue}
3835 \newcommand*\nogreekasCJK{%
3836   \bxjx@gcc@CJKfalse}

\bx@fake@grk  \bx@fake@grk{<出力文字>}{<基準文字>} :

3837 \def\bxjx@do#1\relax{%
3838   \def\bxjx@fake@grk##1##2{%
3839     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
3840   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
3841     \ifx\##1\%
3842       \bxjx@cmta##4\divide\bxjx@cmta\@cclvi
3843       \multiply\bxjx@cmta\@cclvi \advance\bxjx@cmta`##3\relax
3844       \mathchar\bxjx@cmta
3845     \else ##3\fi}
3846 } \expandafter\bxjx@do\string\mathchar\relax

```

■pdfLaTeX・upLaTeX の場合

```

3847 \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。

```

3848 \@ifpackageloaded{inputenc}{}{%else
3849   \RequirePackage[utf8]{inputenc}}
3850 \def\bxjx@tmpa{utf8}
3851 \ifx\bxjx@tmpa\inputencdoingname
3852   \PackageWarningNoLine\bxjx@pkgnam
3853     {Input encoding changed to utf8}%
3854   \inputencoding{utf8}%
3855 \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。

```

3856 \if u\bxjx@engine
3857 \kcatcode"0370=15
3858 \kcatcode"0400=15
3859 \kcatcode"0500=15
3860 \fi

```

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

```

3861 \def\do#1{%
3862   \@tempcnta="#1\relax
3863   \@tempcntb=\@tempcnta \divide\@tempcntb256
3864   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
3865   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
3866 \def\bxjx@do@a#1#2#3#4#5{%

```

```

3867 \ifx\#5\%
3868 \def\bxjx@tmpa{\@inmathwarn#4}%
3869 \else\ifcat A\noexpand#5%
3870 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3871 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
3872 \else \def\bxjx@tmpa{#5}%
3873 \fi\fi
3874 \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
3875 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
3876 \if u\bxjx@engine
3877 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
3878 \def\bxjx@do@b#1#2#3#4#5{%
3879 \kchardef#1=\@tempcnta
3880 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
3881 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}}
3882 \else\if p\bxjx@engine
3883 \def\bxjx@do@b#1#2#3#4#5{%
3884 \mathchardef#1=\@tempcnta
3885 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{UTF{#2}}{#3}{#4}}%
3886 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}}
3887 \fi\fi
3888 \bxjx@grkcyr@list
3889 \let\bxjx@do@a\undefined
3890 \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。

```

3891 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
3892 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
3893 \@onlypreamble\bxjx@DeclareUnicodeCharacter
3894 \def\bxjx@DeclareUnicodeCharacter#1#2{%
3895 \count@=#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
3896 \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
3897 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3898 \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
3899 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3900 \else
3901 \wlog{ \space\space skipped defining Unicode char U+#1}%
3902 \fi\fi}

```

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

```

3903 \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。

```

3904 \ifbxjx@gcc@CJK #1%

```

\nogreekasCJK の場合は、エンコーディングを固定して欧文用のコードを実行するが、そのエンコーディングが未定義の場合は（フォールバックとして）和文用コードを使う。

```

3905 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%

```

```

3906 \else \UseTextSymbol{#2}{#3}%
3907 \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。

```

3908 \begingroup
3909 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
3910 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
3911   \noexpand\bxjx@swap@DUC@cmd
3912   \the\toks@
3913   \noexpand\bxjx@swap@DUC@cmd}}
3914 \endgroup\next
3915 \def\bxjx@swap@DUC@cmd{%
3916   \let\bxjx@tmpa\DeclareUnicodeCharacter
3917   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
3918   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■ $\mathrm{X}\mathrm{Y}\mathrm{T}\mathrm{E}\mathrm{X}$ ・ $\mathrm{Lua}\mathrm{T}\mathrm{E}\mathrm{X}$ の場合

```

3919 \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。

```

3920 \def\do#1{%
3921   \bxjx@cmta="#1\relax
3922   \begingroup
3923     \lccode`~=\bxjx@cmta
3924     \lowercase{\endgroup
3925     \bxjx@do@a{~}}{#1}}
3926 \def\bxjx@do@a#1#2#3#4#5{%
3927   \ifx\#5\\\let\bxjx@tmpa\relax
3928   \else\ifcat A\noexpand#5%
3929     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3930       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
3931   \else \def\bxjx@tmpa{#5}%
3932   \fi\fi
3933   \ifx\bxjx@tmpa\relax\else
3934     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
3935   \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。

```

3936 \mathchardef\bxjx@tmpa="119
3937 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
3938 \let\bxjx@do@a\undefined

```

$\mathrm{Lua}\mathrm{T}\mathrm{E}\mathrm{X}$ における `\(no)greekasCJK` の定義。 `jacharrange` の設定を変更する。

```

3939 \if 1\bxjx@engine
3940   \protected\def\greekasCJK{%
3941     \bxjx@gcc@cjkttrue

```

```

3942 \ltjsetparameter{jacharrange={+2, +8}}
3943 \protected\def\nogreekasCJK{%
3944 \bxjx@gcc@cj>false
3945 \ltjsetparameter{jacharrange={-2, -8}}
3946 \fi

```

X_gTeX における \no)greekasCJK の定義。

```

3947 \if x\bxjx@engine
3948 \protected\def\greekasCJK{%
3949 \bxjx@gcc@cj>true
3950 \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
3951 \bxjx@grkcyr@list}
3952 \protected\def\nogreekasCJK{%
3953 \bxjx@gcc@cj>false
3954 \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
3955 \bxjx@grkcyr@list}
3956 \fi

```

以上。

```

3957 \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。

```

3958 \nogreekasCJK

```

H.5 完了

おしまい。

```

3959 %</cjkat>

```

補助パッケージ実装はここまで。

```

3960 %</anc>

```