

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”)




v1.5 [2017/03/11]

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

目次

1	はじめに	3
2	オプション	7
3	和文フォントの変更	32
4	フォントサイズ	33
5	レイアウト	38
5.1	ページレイアウト	39
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	46
7	ページスタイル	48
8	文書のマークアップ	51
8.1	表題	51
8.2	章・節	56
8.3	リスト環境	68
8.4	パラメータの設定	75
8.5	フロート	76
8.6	キャプション	78
9	フォントコマンド	79

10	相互参照	81
10.1	目次の類	81
10.2	参考文献	87
10.3	索引	88
10.4	脚注	90
11	段落の頭へのグルー挿入禁止	92
12	いろいろなロゴ	96
13	amsmath との衝突の回避	96
14	初期設定	96
付録 A	和文ドライバの仕様 罨	100
付録 B	和文ドライバ : minimal 罨	101
B.1	補助マクロ	101
B.2	(u)pTeX 用の設定	103
B.3	pdfTeX 用の処理	107
B.4	X _Y TeX 用の処理	108
B.5	後処理 (エンジン共通)	108
付録 C	和文ドライバ : standard 罨	110
C.1	共通処理 (1)	111
C.2	pTeX 用設定	115
C.3	pdfTeX 用設定 : CJK + bxcjkatype	118
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	120
C.5	LuaTeX 用設定 : LuaTeX-ja	122
C.6	共通処理 (2)	124
付録 D	和文ドライバ : modern 罨	125
D.1	フォント設定	125
D.2	fixltx2e 読込	126
D.3	和文カテゴリコード	126
D.4	完了	126
付録 E	和文ドライバ : pandoc 罨	126
E.1	dupload システム	126
E.2	lang 変数	127
E.3	geometry 変数	128
E.4	CJKmainfont 変数	128
E.5	fixltx2e パッケージ	128

E.6	cmap パッケージ	129
E.7	microtype パッケージ	129
E.8	完了	129
付録 F	補助パッケージ一覧 	129
付録 G	補助パッケージ : bxjscompat 	129
G.1	準備	130
G.2	X _Y TeX 部分	130
G.3	LuaTeX 部分	131
G.4	完了	132
付録 H	補助パッケージ : bxjscjkat 	132
H.1	準備	132
H.2	和文カテゴリコードの設定	133
H.3	ギリシャ・キリル文字の扱い	134
H.4	初期設定	140
H.5	完了	140

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

これは L^AT_EX 3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいて奥村が改変したものです。権利については両者のものに従います。奥村は何の権利も主張しません。

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

以下では実際のコードに即して説明します。

■ BXJS クラス特有の設定

```

1 %<*cls>
2 %% このファイルは日本語文字を含みます。

長さ値の指定で式を利用可能にするため calc を読み込む。
3 \RequirePackage{calc}

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

クラスの本体ではこの他に geometry パッケージが読み込まれる。
  互換性のための補助パッケージを読み込む。

5 \IfFileExists{bxjscompat.sty}{%
6   \let\jsAtEndOfClass\@gobble
7   \RequirePackage{bxjscompat}%
8 }{}

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

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

\jsEngine 〔暗黙文字トークン〕 エンジン (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\TeXversion{\jsWitheTeXtrue}

非サポートのエンジンの場合は強制終了させる。
※ NTT jTeX と Omega 系。
41 \let\bxjs@tmpa\relax
42 \ifx J\jsEngine \def\bxjs@tmpa{NTT-jTeX}\fi
43 \ifx O\jsEngine \def\bxjs@tmpa{Omega}\fi
44 \ifx\bxjs@tmpa\relax \expandafter\@gobble
45 \else
46 \ClassError\bxjs@clsname
47 {The engine in use (\bxjs@tmpa) is not supported}
48 {It's a fatal error. I'll quit right now.}
49 \expandafter\@firstofone
50 \fi{\endinput\@@end}

\bxjs@protected  $\epsilon$ -TeX 拡張が有効な場合にのみ \protected の効果をもつ。
51 \ifjsWitheTeX \let\bxjs@protected\protected
52 \else \let\bxjs@protected\@empty
53 \fi

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

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

\bxjs@cond \bxjs@cond\ifXXX……\fi{(真)}{(偽)}

```

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

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

`\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 を右ページ起こしにするかどうかです。横組の書籍では真が標準で，要
              するに片起こし，奇数ページ起こしになります。
98 %<book|report>\newif\if@openright

\if@openleft  [2017-02-24] \chapter，\part を左ページ起こしにするかどうかです。
99 %<book|report>\newif\if@openleft

\if@mainmatter  真なら本文，偽なら前付け・後付けです。偽なら \chapter で章番号が出ません。
100 %<book|report>\newif\if@mainmatter \@mainmattertrue

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

JS クラスと異なり、初期値は偽とする。

```
101 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

`b5var` (B5 変形, $182\text{ mm} \times 230\text{ mm}$)，`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。

```

102 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
103 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
104 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
105 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
106 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
107 \DeclareOption{b4paper}{\bxjs@setpaper{{257truemm}{364truemm}}}
108 \DeclareOption{b5paper}{\bxjs@setpaper{{182truemm}{257truemm}}}
109 \DeclareOption{b6paper}{\bxjs@setpaper{{128truemm}{182truemm}}}
110 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
111 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
112 \DeclareOption{b4j}{\bxjs@setpaper{{257truemm}{364truemm}}}
113 \DeclareOption{b5j}{\bxjs@setpaper{{182truemm}{257truemm}}}
114 \DeclareOption{a4var}{\bxjs@setpaper{{210truemm}{283truemm}}}
115 \DeclareOption{b5var}{\bxjs@setpaper{{182truemm}{230truemm}}}
116 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
117 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
118 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

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

```

■slide オプション slide を新設しました。

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

```

122 \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。

```

123 %<!slide>\@slidefalse
124 %<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 へパッチを当てるオプション nomag* を新設しました。

\@ptsize は 10pt, 11pt, 12pt が指定された時のみ従来と同じ値とし、それ以外は \jsUnusualPtSize (= -20) にする。

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

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

```
128 \def\bxjs@setbasefontsize#1{%
129   \bxjs@setbasefontlength\@tempdima{#1}%
130   \edef\bxjs@param@basefontsize{\the\@tempdima}%
131   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
132   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
133   \else\ifdim\@tempdima=12pt  \long\def\@ptsize{2}%
134   \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。

```
135 \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。

```
136   \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。

```
137   \ifx j\jsEngine \setlength#1{#2}%
138   \else
139     \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
140     \ifx\bxjs@tmpa\relax \setlength#1{#2}%
141     \else \@tempdimc0.25mm #1=\bxjs@tmpa\@tempdimc
142       \fi
143   \fi}
144 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
145   \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
146   \else \def\bxjs@tmpa{#1}%
147   \fi}
148 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
149   \ifx\@nnil#2\@nnil\else
150     \ClassWarningNoLine\bxjs@clsname
151     {You should not use 'true' lengths here}%
152   \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。

```
153 \newif\ifjsc@mag
154 \newif\ifjsc@mag@xreal
155 %\let\jsc@magscale\@undefined
156 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
157 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
158 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
159 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
160 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
161 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
162 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
163 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
164 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
165 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
166 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
167 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
168 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
169 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
170 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
171 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
172 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
173 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
174 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}
```

JS クラス互換の `magstyle` 設定オプション。

```
175 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
176 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
177 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}
```

■ トンボオプション

欧文 \LaTeX のカーネルではサポートされないため削除。

ただし JS クラスとの互換のため以下の変数を用意する。

※ JS クラスでは、2017-01-11 の改修で、「トンボオプション指定時のみ `\stockwidth/height` を定義する」という仕様に変更された。BXJS での対応は要検討。

```
178 \newdimen\stockwidth \newdimen\stockheight
```

■ 面付け

欧文 \LaTeX 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。

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

■二段組 `twocolumn` で二段組になります。

```
182 \DeclareOption{onecolumn}{\@twocolumnfalse}
183 \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。

```
184 \DeclareOption{titlepage}{\@titlepagetrue}
185 \DeclareOption{notitlepage}{\@titlepagefalse}
```

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

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

```
186 %<book|report>\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
187 %<book|report>\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
188 %<book|report>\DeclareOption{openany}{\@openrightfalse\@openleftfalse}
```

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

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

```
189 \def\eqnarray{%
190   \stepcounter{equation}%
191   \def\@currentlabel{\p@equation\theequation}%
192   \global\@eqnswtrue
193   \m@th
194   \global\@eqcnt\z@
195   \tabskip\@centering
196   \let\\\@eqnocr
197   $$$\everycr{}\halign to\displaywidth\bgroup
198     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
199     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
200     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
201     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
202     \tabskip\z@skip
203   \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。

```

204 \DeclareOption{leqno}{\input{leqno.clo}}
205 \DeclareOption{fleqn}{\input{fleqn.clo}}
206 % fleqn 用の eqnarray 環境の再定義
207 \def\eqnarray{%
208   \stepcounter{equation}%
209   \def\@currentlabel{\p@equation\theequation}%
210   \global\@eqnswtrue\m@th
211   \global\@eqcnt\z@
212   \tabskip\mathindent
213   \let\=\@eqnocr
214   \setlength\abovedisplayskip{\topsep}%
215   \ifvmode
216     \addtolength\abovedisplayskip{\partopsep}%
217   \fi
218   \addtolength\abovedisplayskip{\parskip}%
219   \setlength\belowdisplayskip{\abovedisplayskip}%
220   \setlength\belowdisplayshortskip{\abovedisplayskip}%
221   \setlength\abovedisplayshortskip{\abovedisplayskip}%
222   $$\everycr{}\halign to\linewidth% $$
223   \bgroup
224   \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
225   &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
226   &\global\@eqcnt\tw@
227   $\displaystyle{##}$\hfil \tabskip\@centering
228   &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
229   \tabskip\z@skip\cr
230 }

```

■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。

```

231 % \DeclareOption{openbib}{%
232 %   \AtEndOfPackage{%
233 %     \renewcommand\@openbib@code{%
234 %       \advance\leftmargin\bibindent
235 %       \itemindent -\bibindent
236 %       \listparindent \itemindent
237 %       \parsep \z@}%
238 %     \renewcommand\newblock{\par}}

```

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

スでは標準で和文フォントを数式用に登録しないことにしていたのですが、従来のドキュメントクラスの仕様に合わせることにしました。

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

```
239 %\let\bxjs@enablejfam\undefined
```

`fancyhdr` オプションの処理。

```
240 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
241 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
242 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
243 \define@key{bxjs}{enablejfam}{%
244   \bxjs@set@keyval{enablejfam}{#1}{}}
```

JS クラスとの互換のため `disablejfam` オプションを定義する。

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

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

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

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

※ JS クラスの `\ifdraft` は 2016/07/13 版で廃止された。よって `\ifdraft` は 2.0 版で廃止を予定する。

```
246 \let\ifjsDraft\iffalse
247 \@onlypreamble\bxjs@draft
248 \def\bxjs@draft#1{%
249   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
250 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
251 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
252 \AtBeginDocument{%
253   \expandafter\ifx\csname ifdraft\endcsname\relax
254     \expandafter\let\csname ifdraft\expandafter\endcsname
255       \csname ifjsDraft\endcsname
256   \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。

```
257 \newif\ifbxjs@papersize
258 \bxjs@papersizetrue
259 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
260 \DeclareOption{papersize}{\bxjs@papersizetrue}
```

■英語化 オプション english を新設しました。

```
261 \newif\if@english
262 \@englishfalse
263 \DeclareOption{english}{\@englishttrue}
```

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

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

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

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

■jslogo パッケージの読み込み 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。

```
264 \newif\if@jslogo \@jslogofalse
265 \DeclareOption{jslogo}{\@jslogotrue}
266 \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` : \TeX の “generateactualtext” を有効／無効にする。
- `simplejasetup / nosimplejasetup` : \TeX の “linebreaklocale” を有効／無効にする。
- `bigcode / nobigcode` : up\TeX 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=\langle dimen \rangle` : 基底フォントサイズを直接指定する。(xxpt オプションの代用なので、既定値は 10 pt である。)
- `jbase=\langle dimen \rangle` : 基底フォントサイズを “和文規準で” 直接指定する。
- `scale=\langle real \rangle` : 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=\langle int \rangle` : `\mag` 値の直接設定。既定は `base` から算出する。
- `paper=\{\langle dimen:width \rangle\}\{\langle dimen:height \rangle\}` : 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=\langle name \rangle` : 使用する和文ドライバの指定。
- `jafont=\langle name \rangle` : 和文フォントプリセットの指定。
- `japaram=\langle name \rangle` : 和文フォントパラメタの指定。
- `magstyle=\langle name \rangle` : “版面拡大” の実現方法の選択。
- `dvi=\langle name \rangle` : 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=\langle bool \rangle` : `fancyhdr` パッケージ用の調整を行うか。
- `layout=\langle name \rangle` : レイアウト変種の指定。
- `textwidth-limit=\langle number \rangle` : `bxjsbook` における、`\textwidth` の上限の全角単位での値。
- `paragraph-mark=\langle char \rangle` : パラグラフのマーク。
- `whole-zw-lines=\langle bool \rangle` : 行長を全角単位に丸めるか。
- `hyperref-enc=\langle bool \rangle` : `hyperref` の文字コード指定補正を行うか。

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

※局所化の `\begingroup\sim\endgroup` について、以前は `\group\sim\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。

267 `\mathchardef\bxjs@csta=259`

```

268 \def\bxjs@invscale#1#2{%
269   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
270   \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
271   \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
272   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
273   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
274   \@tempdimb\@tempcnta\@ne
275   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
276   \advance\@tempcnta\bxjs@csta \@tempdimc\@tempcnta\@ne
277   \@whiledim\@tempdimb<\@tempdimc\do{%
278     \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
279     \advance\@tempcntb\@ne \divide\@tempcntb\@tw@
280     \ifdim #2\@tempcntb>\@tempdima
281       \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
282       \else \@tempdimb=\@tempcntb\@ne \fi}%
283   \xdef\bxjs@gtmpa{\the\@tempdimb}%
284   \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。

```
285 \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。

```
286 \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。

```

287   \def\bxjs@composite@proc{%
288     \bxjs@oldfontcommandstrue
289     \setkeys{bxjs}{ja=pandoc}%
290     \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…。

```

291   \def\bxjs@driver@opt{dvipdfmx}%
292   \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。

```
293 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
294 %\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。

```
295 \DeclareOption{autodetect-engine}{%
296   \let\bxjs@engine@given=*}
297 \DeclareOption{latex}{%
298   \def\bxjs@engine@opt{latex}%
299   \let\bxjs@engine@given=n}
300 \DeclareOption{platex}{%
301   \def\bxjs@engine@opt{platex}%
302   \let\bxjs@engine@given=j}
303 \DeclareOption{uplatex}{%
304   \def\bxjs@engine@opt{uplatex}%
305   \let\bxjs@engine@given=u}
306 \DeclareOption{xelatex}{%
307   \def\bxjs@engine@opt{xelatex}%
308   \let\bxjs@engine@given=x}
309 \DeclareOption{pdflatex}{%
310   \def\bxjs@engine@opt{pdflatex}%
311   \let\bxjs@engine@given=p}
312 \DeclareOption{lualatex}{%
313   \def\bxjs@engine@opt{lualatex}%
314   \let\bxjs@engine@given=l}
315 \DeclareOption{platex-ng}{%
316   \def\bxjs@engine@opt{platex-ng}%
317   \let\bxjs@engine@given=g}
318 \DeclareOption{platex-ng*}{%
319   \def\bxjs@engine@opt{platex-ng*}%
320   \let\bxjs@platexng@nodrv=t%
321   \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。

```
322 %\let\bxjs@driver@given\@undefined
323 \let\bxjs@driver@@dvimode=0
324 \let\bxjs@driver@@dvipdfmx=1
325 \let\bxjs@driver@@pdfmode=2
326 \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。

```
327 %\let\bxjs@driver@opt\@undefined
```

```

328 \DeclareOption{dvips}{%
329   \def\bxjs@driver@opt{dvips}%
330   \let\bxjs@driver@given\bxjs@driver@@dvimode}
331 \DeclareOption{dviout}{%
332   \def\bxjs@driver@opt{dviout}%
333   \let\bxjs@driver@given\bxjs@driver@@dvimode}
334 \DeclareOption{xdvi}{%
335   \def\bxjs@driver@opt{xdvi}%
336   \let\bxjs@driver@given\bxjs@driver@@dvimode}
337 \DeclareOption{dvipdfmx}{%
338   \def\bxjs@driver@opt{dvipdfmx}%
339   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
340 \DeclareOption{pdftex}{%
341   \def\bxjs@driver@opt{pdftex}%
342   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
343 \DeclareOption{luatex}{%
344   \def\bxjs@driver@opt{luatex}%
345   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
346 \DeclareOption{xetex}{%
347   \def\bxjs@driver@opt{xetex}%
348   \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。

```

349 \DeclareOption{dvipdfmx-if-dvi}{%
350   \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。

```

351 \newif\ifbxjs@usezw \bxjs@usezwtrue

```

`zw`、`noz` オプションの定義。

```

352 \DeclareOption{noz}{%
353   \bxjs@usezwfalse}
354 \DeclareOption{zw}{%
355   \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。

```

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

```

`nojs`、`js` オプションの定義。

```

357 \DeclareOption{nojs}{%
358   \bxjs@disguise@jsfalse}
359 \DeclareOption{js}{%
360   \bxjs@disguise@jstrue}

```

```

\ifbxjs@precisetext XeLaTeX の “generateactualtext” を有効にするか。既定は偽。
361 \newif\ifbxjs@precisetext

    noprecisetext / precisetext オプションの定義。
362 \DeclareOption{noprecisetext}{%
363   \bxjs@precisetextfalse}
364 \DeclareOption{precisetext}{%
365   \bxjs@precisetexttrue}

\ifbxjs@simplejasetup XeLaTeX 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）。
366 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

    nosimplejasetup / simplejasetup オプションの定義。
367 \DeclareOption{nosimplejasetup}{%
368   \bxjs@simplejasetupfalse}
369 \DeclareOption{simplejasetup}{%
370   \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。
371 \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。
372 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
373 \def\bxjs@tmpb#1 #2#3\@nil{%
374   \ifx1#2\bxjs@bigcodetrue \fi}
375 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil

    nobigcode / bigcode オプションの定義。
376 \DeclareOption{nobigcode}{%
377   \bxjs@bigcodefalse}
378 \DeclareOption{bigcode}{%
379   \bxjs@bigcodetrue}

\ifbxjs@oldfontcommands \allowoldfontcommands を既定で有効にするか。
380 \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。
381 \DeclareOption{nooldfontcommands}{%
382   \bxjs@oldfontcommandsfalse}
383 \DeclareOption{oldfontcommands}{%
384   \bxjs@oldfontcommandstrue}

```

■keyval 型のオプション

```
385 \def\bxjs@setkey{%
386   \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
387 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
388 \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`。

```
389 \def\bxjs@safe@setkeys#1#2{%
390   \let\bxjs@KV@errx\KV@errx
391   \let\KV@errx\bxjs@safe@setkeys@a
392   \setkeys{#1}{#2}%
393   \let\KV@errx\bxjs@KV@errx}
394 \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。

```
395 \def\bxjs@set@keyval#1#2#3{%
396   \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
397   \ifx\bxjs@next\relax
398     \bxjs@error@keyval{#1}{#2}%
399     #3%
400   \else \bxjs@next
401   \fi}
402 \@onlypreamble\bxjs@error@keyval
403 \def\bxjs@error@keyval#1#2{%
404   \ClassError\bxjs@clsname
405   {Invalid value '#2' for option #1}\@ehc}
```

`\ifbxjs@scaleset` 和文スケール値が指定されたか。

```
406 \newif\ifbxjs@scaleset
```

`\jsScale` [実数値マクロ] 和文スケール値。

```
407 \def\jsScale{0.924715}
```

base オプションの処理。

```
408 \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。

```
409 \@onlypreamble\bxjs@do@opt@jbase
410 \let\bxjs@do@opt@jbase\relax
411 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
412 \def\bxjs@setjbasefontsize#1{%
413   \def\bxjs@do@opt@jbase{%
414     \bxjs@setbasefontlength\@tempdima{#1}%
415     \bxjs@invscale\@tempdima\jsScale
```

```

416 \bxjs@setbasefontsize{\@tempdima}}
    scale オプションの処理。
417 \define@key{bxjs}{scale}{%
418 \bxjs@scalesettrue
419 \edef\jsScale{#1}}
    noscale オプションの処理。
420 \DeclareOption{noscale}{%
421 \bxjs@scalesettrue
422 \def\jsScale{1}}

\bxjs@param@mag mag オプションの値。
423 \let\bxjs@param@mag\relax
    mag オプションの処理。
424 \define@key{bxjs}{mag}{\edef\bxjs@param@mag{#1}}
    paper オプションの処理。
425 \define@key{bxjs}{paper}{\edef\bxjs@param@paper{#1}}

\bxjs@jadriver 和文ドライバの名前。
426 \let\bxjs@jadriver\relax
427 %\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)。
428 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
429 \define@key{bxjs}{ja}{\relax}{%
430 \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}

\jsJaFont 和文フォント設定の名前。
431 \let\jsJaFont\@empty
    jafont オプションの処理。
432 \define@key{bxjs}{jafont}{\edef\jsJaFont{#1}}

\jsJaParam 和文ドライバパラメタの文字列。
433 \let\jsJaParam\@empty
    japaram オプションの処理。
434 \define@key{bxjs}{japaram}{\edef\jsJaParam{#1}}

\bxjs@magstyle magstyle 設定値。(古いイマイチな名前。)
435 \let\bxjs@magstyle@mag=m
436 \let\bxjs@magstyle@real=r
437 \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。

```
438 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
439 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
440 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal
```

`\bxjs@magstyle@default` は既定の値を表す。

```
441 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
442 \ifx l\jsEngine \ifnum\luatexversion>86
443   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
444 \fi\fi
445 \ifjsWithpTeXng
446   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
447 \fi
448 \let\bxjs@magstyle\bxjs@magstyle@default
```

`magstyle` オプションの処理。

```
449 \define@key{bxjs}{magstyle}{%
450   \expandafter\let\expandafter\bxjs@magstyle\csname
451     bxjs@magstyle@#1\endcsname
452   \ifx\bxjs@magstyle\relax
453     \ClassError\bxjs{clsname
454       {Invalid value '#1' for option magstyle}}\@ehc
455     \let\bxjs@magstyle\bxjs@magstyle@default
456   \fi}
```

`\bxjs@geometry` `geometry` オプションの値。

```
457 \let\bxjs@geometry@class=c
458 \let\bxjs@geometry@user=u
459 \let\bxjs@geometry\bxjs@geometry@class
```

`geometry` オプションの処理。

```
460 \define@key{bxjs}{geometry}{%
461   \expandafter\let\expandafter\bxjs@geometry\csname
462     bxjs@geometry@#1\endcsname
463   \ifx\bxjs@geometry\relax
464     \ClassError\bxjs{clsname
465       {Invalid value '#1' for option geometry}}\@ehc
466     \let\bxjs@geometry\bxjs@geometry@class
467   \fi}
```

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

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

`fancyhdr` オプションの処理。

```
469 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
470 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
471 \define@key{bxjs}{fancyhdr}{%
```

```

472 \bxjs@set@keyval{fancyhdr}{#1}{}

\ifbxjs@dvi@opt dvi オプションが指定されたか。
473 \newif\ifbxjs@dvi@opt

DVI モードのドライバとドライバ種別との対応。
474 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
475 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvimode
476 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
477 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode

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

\bxjs@driver@given を未定義にしていることに注意。
485 \def\bxjs@driver@opt{#1}%
486 \let\bxjs@driver@given\undefined
487 \bxjs@dvi@opttrue
488 \fi}

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

\ifbxjs@force@chapterabstract [スイッチ] abstract 環境を chapterabstract にするか。
※ bxjsbook では常に真。bxjsreport では layout が v1 の場合に真になる。
※ただし 1.5 版では移行措置のため bxjsreport の既定値を真にする。
490 \newif\ifbxjs@force@chapterabstract
491 %<book|report>\bxjs@force@chapterabstracttrue

layout オプションの処理。
492 \@namedef{bxjs@kv@layout@v1}{%
493 %<book>\bxjs@layout@buggyhmargintrue
494 %<report>\bxjs@force@chapterabstracttrue
495 }
496 \@namedef{bxjs@kv@layout@v2}{%
497 %<book>\bxjs@layout@buggyhmarginfalse
498 %<report>\bxjs@force@chapterabstractfalse
499 }
500 \define@key{bxjs}{layout}{%
501 \bxjs@set@keyval{layout}{#1}{}

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

```

```

502 %\let\bxjs@textwidth@limit\@undefined
503 \define@key{bxjs}{textwidth-limit}{%
504   \edef\bxjs@textwidth@limit{#1}}

```

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

```

505 %\let\bxjs@paragraph@mark\@undefined
506 \define@key{bxjs}{paragraph-mark}{%
507   \edef\bxjs@paragraph@mark{#1}}

```

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

```

508 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
509 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
510 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
511 \define@key{bxjs}{whole-zw-lines}{\bxjs@set@keyval{wholezwlines}{#1}{}}

```

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

```

512 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
513 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
514 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
515 \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。

```

516 \let\bxjs@ltx@removeelement\@removeelement
517 \def\@removeelement#1#2#3{%
518   \def\reserved@a{#2}%
519   \ifx\reserved@a\@empty \let#3\@empty
520   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
521   \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。

```

522 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
523 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
524 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
525 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
526 \ProcessOptions\relax
527 \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。

```

528 \@onlypreamble\bxjs@purge@brace@elts
529 \def\bxjs@purge@brace@elts{%
530   \def\bxjs@tmpa{\@gobble}%
531   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
532   \let\@classoptionslist\bxjs@tmpa}
533 \@onlypreamble\bxjs@purge@be@a
534 \def\bxjs@purge@be@a#1,{%
535   \ifx\@nil#1\relax\else
536     \bxjs@purge@be@b#1{\@nil
537     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
538     \expandafter\bxjs@purge@be@a
539   \fi}
540 \@onlypreamble\bxjs@purge@be@b
541 \def\bxjs@purge@be@b#1#2{\bxjs@purge@be@c}
542 \@onlypreamble\bxjs@purge@be@c
543 \def\bxjs@purge@be@c#1\@nil{%
544   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
545 \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。

```

546 \@expandtwoargs\@removeelement
547 {papersize}\@classoptionslist\@classoptionslist
548 \@expandtwoargs\@removeelement
549 {10pt}\@classoptionslist\@classoptionslist
550 \@expandtwoargs\@removeelement
551 {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。

```

552 \ifx\bxjs@jadriver\relax\else
553   \let\bxjs@jadriver@given\bxjs@jadriver
554 \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。

```

555 \let\bxjs@tmpb\jsEngine
556 \ifx j\bxjs@tmpb\ifjsWithpTeXng
557   \let\bxjs@tmpb=g
558 \fi\fi
559 \ifx j\bxjs@tmpb\ifjsWithupTeX
560   \let\bxjs@tmpb=u
561 \fi\fi
562 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
563   \let\bxjs@tmpb=n
564 \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。)

```
565 \ifx *\bxjs@engine@given
566   \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。

```
567 \ifx j\bxjs@engine@given
568   \g@addto@macro\@classoptionslist{,latex}
569 \else\ifx u\bxjs@engine@given
570   \g@addto@macro\@classoptionslist{,uplatex}
571 \fi\fi
572 \fi
573 \ifx\bxjs@engine@given\@undefined\else
574   \ifx\bxjs@engine@given\bxjs@tmpb\else
575     \ClassError\bxjs@clsname
576       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
577   \fi
578 \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。

```
579 \ifjsWithpTeXng
580   \g@addto@macro\@classoptionslist{,uplatex}
581 \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。

```
582 \@tempswatrue
583 \ifx \bxjs@driver@given\@undefined\else
584   \ifjsInPdfMode
585     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
586       \@tempswafalse
587     \fi
588   \else\ifx x\jsEngine
589     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
590       \@tempswafalse
591     \fi
592   \else
593     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
594       \@tempswafalse
595     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
596       \@tempswafalse
597     \fi\fi
598     \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
599       \@tempswafalse
600     \fi\fi
601   \fi\fi
602 \fi
603 \if@tempswa\else
604   \ClassError\bxjs@clsname
```

```

605   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
606 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

607 \ifjsInPdfMode \@tempswafalse
608 \else\ifx x\jsEngine \@tempswafalse
609 \else\ifjsWithpTeXng \@tempswafalse
610 \else \@tempwattrue
611 \fi\fi\fi
612 \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)。

```

613   \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
614     \ClassWarningNoLine\bxjs@clsname
615     {No driver option is given}
616   \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。)

```

617   \ifbxjs@dvi@opt
618     \edef\bxjs@nxt{%
619       \let\noexpand\bxjs@driver@given
620       \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
621       \noexpand\g@addto@macro\noexpand\@classoptionslist
622       {,\bxjs@driver@opt}%
623     }\bxjs@nxt
624   \fi
625 \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。

```

626 \ifjsWithpTeXng
627   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
628     \let\bxjs@platexng@nodrv\@undefined
629   \else\ifx t\bxjs@platexng@nodrv\else
630     \g@addto@macro\@classoptionslist{,dvipdfmx}
631   \fi\fi
632 \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。

```

633 \def\bxjs@@minimal{minimal}
634 \ifx\bxjs@jadriver\relax
635   \ifx j\jsEngine
636     \def\bxjs@jadriver{standard}
637   \else
638     \let\bxjs@jadriver\bxjs@@minimal

```

```
639 \fi
640 \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。

```
641 \ifx\bxjs@jadriver@given\@undefined\else
642   \ifx\bxjs@engine@given\@undefined
643     \ClassError\bxjs@clsname
644     {An engine option must be explicitly given}%
645     {When you use a Japanese-driver you must specify a correct\MessageBreak
646       engine option.\MessageBreak \@ehc}
647 \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。)

```
648 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
649   \ifx\bxjs@magstyle\bxjs@magstyle@mag
650     \let\bxjs@magstyle\bxjs@magstyle@default
651     \ClassError\bxjs@clsname
652     {The engine does not support 'magstyle=usemag'}%
653     {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak
654       The default value 'nomag*' is used instead.\MessageBreak \@ehc}
655   \fi
656 \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。

```
657 \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。

```
658 \let\Cjascale\jsScale
```

後処理

```
659 \if@slide
660   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
661 \fi
662 \if@landscape
663   \setlength\@tempdima {\paperheight}
664   \setlength\paperheight{\paperwidth}
665   \setlength\paperwidth {\@tempdima}
666 \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。)

```
667 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
668   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
669     \catcode\@tempcnta\active
```

```

670 \advance\@tempcnta\@ne
671 \repeat
672 \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。

```

673 \ifbxjs@disguise@js
674 %<book|report>\def\bxjs@js@clsname{jsbook}
675 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
676 \@namedef{ver@bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
677 \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。

```

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

```

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

```

679 \ifbxjs@oldfontcommands
680 \AtEndOfClass{\allowoldfontcommands}
681 \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。

```

682 %<slide>\def\n@baseline{13}%
683 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
684 %<!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。

```

685 \ifx\bxjs@magstyle\bxjs@magstyle@mag
686 \jsc@magtrue
687 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
688 \jsc@mag@xrealtrue
689 \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。

```

690 \ifx\bxjs@param@mag\relax
691   \@tempdima=\bxjs@param@basefontsize
692   \advance\@tempdima.001pt \multiply\@tempdima25
693   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
694   \edef\bxjs@param@mag{\the\@tempcnta}
695 \else
696 % mag 値が直接指定された場合
697   \let\c@bxjs@cnta\@tempcnta
698   \setcounter{bxjs@cnta}{\bxjs@param@mag}
699   \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
700 % 有効な mag 値の範囲は 1--32768
701   \edef\bxjs@param@mag{\the\@tempcnta}
702   \advance\@tempcnta100000
703   \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
704   \expandafter\bxjs@tmpa\the\@tempcnta\@nil
705   \edef\bxjs@param@basefontsize{\the\@tempdima}
706 \fi
707 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
708 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
709 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
710 \edef\jsc@magscale{\strip@pt\@tempdima}
711 \let\jsBaseFontSize\bxjs@param@basefontsize
712 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
713 %\typeout{\string\jsEngine: \meaning\jsEngine}
714 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
715 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
716 %\typeout{\string\jsc@magscale: \jsc@magscale}
717 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
718 %\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。

```

719 \ifjsc@mag
720   \let\jsc@mpt\p@
721   \newdimen\jsc@mmm \jsc@mmm=1mm
722   \catcode`\?=9 % \p@? read as \p@
723 \else
724   \newdimen\jsc@mpt \jsc@mpt=\jsc@magscale \p@
725   \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
726   \catcode`\?=11 \let\p@?\jsc@mpt
727 \fi
728 \chardef\bxjs@qmcc=\catcode`\?\relax

```

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

`noz` 非指定時は `\zw` を `\jsZw` と同義にする。

```

729 \newdimen\jsZw
730 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
731 \ifbxjs@usezw
732   \providecommand*\zw{\jsZw}
733 \fi

```

`\zwspace` 全角幅の水平空き。

```

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

```

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

```

735 \ifjsc@mag@xreal
736   \RequirePackage{type1cm}
737   \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

738 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
739 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
740 \let\jsc@get@external@font\get@external@font
741 \def\get@external@font{%
742   \jsc@preadjust@extract@font
743   \jsc@get@external@font}
744 \def\jsc@fstrunc#1{%
745   \edef\jsc@tmpa{\strip@pt#1}%
746   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
747 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
748   \if#5*\else
749     \edef\jsc@tmpa{#1%
750       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
751   \fi}
752 \def\jsc@preadjust@extract@font{%
753   \let\jsc@req@size\f@size
754   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
755   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@

```

```

756 \let\jsc@ref@size\jsc@tmpa
757 \let\f@size\jsc@ref@size}
758 \def\execute@size@function#1{%
759 \let\jsc@c@ref@size\f@size
760 \let\f@size\jsc@req@size
761 \csname s@fct@#1\endcsname}
762 \let\jsc@DeclareErrorFont\DeclareErrorFont
763 \def\DeclareErrorFont#1#2#3#4#5{%
764 \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
765 \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
766 \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
767 \def\gen@sfcnt{%
768 \edef\mandatory@arg{\mandatory@arg\jsc@c@ref@size}%
769 \empty@sfcnt}
770 \def\genb@sfcnt{%
771 \edef\mandatory@arg{%
772 \mandatory@arg\expandafter\genb@x\jsc@c@ref@size..\@{}}%
773 \empty@sfcnt}
774 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
775 \fi

```

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

```

\jsc@smallskip
\jsc@medskip 776 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 777 %\def\jsc@medskip{\vspace\jsc@medskipamount}
778 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

```

```

\jsc@smallskipamount
\jsc@medskipamount 779 \newskip\jsc@smallskipamount
\jsc@bigskipamount 780 \jsc@smallskipamount=3\jsc@mppt plus 1\jsc@mppt minus 1\jsc@mppt
781 %\newskip\jsc@medskipamount
782 %\jsc@medskipamount =6\jsc@mppt plus 2\jsc@mppt minus 2\jsc@mppt
783 %\newskip\jsc@bigskipamount
784 %\jsc@bigskipamoun =12\jsc@mppt plus 4\jsc@mppt minus 4\jsc@mppt

```

■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、L^AT_EX の `\def\@{\spacefactor\@m}` という定義（\@m は 1000）では `I watch TV\@.` と書くと V とピリオドのペアカーニングが効かなくなります。そこで、次のような定義に直し、`I watch TV.\@` と書くことにします。

```
785 \chardef\bxjs@periodchar=`\.  
786 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcode\bxjs@periodchar}  
787 \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>\@vpt</code>	6	<code>\@vipt</code>	7
<code>\@viipt</code>	8	<code>\@ixpt</code>	9	<code>\@xpt</code>	10
<code>\@xipt</code>	10.95	<code>\@xipt</code>	12	<code>\@xivpt</code>	14.4

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

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

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

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

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

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

```

788 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
789 \expandafter\bxjs@tmpa\expandafter{%
790   \set@fontsize{#1}{#2}{#3}%
791 % 末尾にコードを追加
792   \expandafter\def\expandafter\size@update\expandafter{%
793     \size@update
794     \jsFontSizeChanged}%
795 }

```

`\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。

```

796 \newcommand*\jsFontSizeChanged{%
797   \jsZw=\f@size\p@
798   \jsZw=\jsScale \jsZw
799   \ifdim\parindent>\z@
800     \if@english \parindent=1em
801     \else       \parindent=1\jsZw
802   \fi
803   \fi\relax
804   \jsResetDimen}

```

`\jsResetDimen` ユーザ定義用のフック。

```

805 \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。

```

806 \ifjsc@mag
807   \let\jsc@setfontsize\@setfontsize
808 \else
809   \def\jsc@setfontsize#1#2#3{%
810     \@setfontsize#1{#2\jsc@mpt}{#3\jsc@mpt}}
811 \fi

```

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

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

```

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

[2008-02-18] `english` オプションで最初の段落のインデントをしないようにしました。
TODO: Hasumi さん [qa:54539] のご指摘は考慮中です。

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

```
813 \newif\ifnarrowbaselines
814 \if@english
815   \narrowbaselinestrue
816 \fi
817 \def\narrowbaselines{%
818   \narrowbaselinestrue
819   \skip0=\abovedisplayskip
820   \skip2=\abovedisplayshortskip
821   \skip4=\belowdisplayskip
822   \skip6=\belowdisplayshortskip
823 % 一時的に警告を無効化する
824   \let\bxjs@ltx@nomath\@nomath
825   \let\@nomath\@gobble
826   \@currsize\selectfont
827   \let\@nomath\bxjs@ltx@nomath
828   \abovedisplayskip=\skip0
829   \abovedisplayshortskip=\skip2
830   \belowdisplayskip=\skip4
831   \belowdisplayshortskip=\skip6\relax}
832 \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。

```
833 \def\bxjs@ifnarrowbaselines{%
834   \ifnarrowbaselines\expandafter\@firstoftwo
835   \else \expandafter\@secondoftwo
836   \fi
837 }
```

`\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。

```
838 \renewcommand{\normalsize}{%
839   \bxjs@if@narrowbaselines{%
840     \jsc@setfontsize\normalsize\@xpt\@xipt
841   }{%else
842     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
843   }%
```

数式の上のアキ(\abovedisplayskip), 短い数式の上のアキ(\abovedisplayskipshortskip), 数式の下のアキ(\belowdisplayshortskip) の設定です。

[2003-02-16] ちょっと変えました。

[2009-08-26] T_EX Q & A 52569 から始まる議論について逡巡していましたが、結局、微調節してみることにしました。

```
844 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
845 \abovedisplayskipshortskip \z@ \@plus3\p@?
846 \belowdisplayshortskip 9\p@? \@plus3\p@? \@minus4\p@?
847 \belowdisplayshortskip \belowdisplayshortskip
```

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

```
848 \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。

```
849 \normalsize
```

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

\Cwd

\Cvs まず \Cwd 等の変数を定義する。

```
\Chs 850 \ifx\Cht\@undefined \newdimen\Cht \fi
851 \ifx\Cdp\@undefined \newdimen\Cdp \fi
852 \ifx\Cwd\@undefined \newdimen\Cwd \fi
853 \ifx\Cvs\@undefined \newdimen\Cvs \fi
854 \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。

```
855 \setlength\Cht{0.88\jsZw}
856 \setlength\Cdp{0.12\jsZw}
857 \setlength\Cwd{1\jsZw}
858 \setlength\Cvs{\baselineskip}
```

859 \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。

```
860 \newcommand{\small}{%
861   \bxjs@if@narrowbaselines{%
862     <!kiyou> \jsc@setfontsize\small\@ixpt{11}%
863     <kiyou> \jsc@setfontsize\small{8.8888}{11}%
864   }{%else
865     <!kiyou> \jsc@setfontsize\small\@ixpt{13}%
866     <kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
867   }%
868   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
869   \abovedisplayshortskip \z@ \@plus3\p@?
870   \belowdisplayskip \abovedisplayskip
871   \belowdisplayshortskip \belowdisplayskip
872   \def\@listi{\leftmargin\leftmargin
873             \topsep \z@
874             \parsep \z@
875             \itemsep \parsep}}
```

\footnotesize \footnotesize も同様です。 \topsep と \parsep は、元はそれぞれ 3 ± 1 , 2 ± 1 ポイントでしたが、ここではゼロ (\z@) にしました。

```
876 \newcommand{\footnotesize}{%
877   \bxjs@if@narrowbaselines{%
878     <!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
879     <kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
880   }{%else
881     <!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
882     <kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
883   }%
884   \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
885   \abovedisplayshortskip \z@ \@plus2\p@?
886   \belowdisplayskip \abovedisplayskip
887   \belowdisplayshortskip \belowdisplayskip
888   \def\@listi{\leftmargin\leftmargin
889             \topsep \z@
890             \parsep \z@
891             \itemsep \parsep}}
```

\scriptsize それ以外のサイズは、本文に使うことがないので、単にフォントサイズと行送りだけ変更します。特に注意すべきは \large で、これは二段組のときに節見出しのフォントとして使い、

\large

\Large

\LARGE

\huge

\Huge

\HUGE

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。

[2004-11-03] `\HUGE` を追加。

```
892 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
893 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
894 \if@twocolumn
895 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
896 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
897 \else
898 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
899 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
900 \fi
901 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
902 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
903 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
904 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
905 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
906 \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` 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。

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

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

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

```
908 % \newcommand{\headfont}{\bfseries}
909 \newcommand{\headfont}{\sffamily}
910 % \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。

```
911 %<!kiyou>\setlength\columnsep{2\Cwd}
```

```

912 %<kiyou>\setlength\columnsep{28truebp}
913 \setlength\columnseprule{0\p@}

```

■段落

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

```

\lineskiplimit 914 \setlength\lineskip{1\p@?}
\normallineskip 915 \setlength\normallineskip{1\p@?}
\normallineskiplimit 916 \setlength\lineskiplimit{1\p@?}
917 \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。

```
918 \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、ここではゼロにしました。`\parindent` は段落の先頭の字下げ幅です。

```

919 \setlength\parskip{0\p@}
920 \if@slide
921   \setlength\parindent{0\p@}
922 \else
923   \setlength\parindent{1\Cwd}
924 \fi

```

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

```

\@highpenalty 925 \@lowpenalty 51
926 \@medpenalty 151
927 \@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。

```
928 % \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。

```
929 % \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。

```
930 \ifjsc@mag
931 \mag=\bxjs@param@mag
932 \fi
933 \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。

```
934 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
935 \def\bxjs@read@b{%
936   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
937   \else \expandafter\bxjs@read@d \fi}
938 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
939 \def\bxjs@read@d#1\@nil{}
940 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
941 \edef\bxjs@layout@paper{%
942   \ifjsc@mag truedimen,\fi
943   \if@landscape landscape,\fi
944   \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。

```
945 %<*article|report>
946 \def\bxjs@layout{%
947   headheight=\topskip,footskip=0.03367\paperheight,%
948   headsep=\footskip-\topskip,includeheadfoot,%
949   hscale=0.76,hmarginratio=1:1,%
950   vscale=0.83,vmarginratio=1:1,%
951 }
952 %</article|report>
953 %<*book>
954 \ifbxjs@layout@buggyhmargin %---
955 % アレ
956 \def\bxjs@layout{%
957   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
958   hmargin=36\jsc@mmm,hmarginratio=1:1,%
959   vscale=0.83,vmarginratio=1:1,%
960 }
961 \else %---
962 % 非アレ
963 \def\bxjs@layout{%
964   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
965   hmargin=18\jsc@mmm,%
966   vscale=0.83,vmarginratio=1:1,%
```



```

967 }
968 \fi                                     %---
969 %</book>
970 %<*slide>
971 \def\bxjs@layout{%
972   noheadfoot,%
973   hscale=0.9,hmarginratio=1:1,%
974   vscale=0.944,vmarginratio=1:1,%
975 }
976 %</slide>

```

\fullwidth 〔寸法レジスタ〕 ヘッダ・フッタ領域の横幅。

```

977 \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。

```

978 %<*book>
979 \newcommand\jsTextWidthLimit{40}
980 \ifx\bxjs@textwidth@limit\@undefined\else
981   \let\c@bxjs@cmta\@tempcmta
982   \setcounter{bxjs@cmta}{\bxjs@textwidth@limit}
983   \long\edef\jsTextWidthLimit{\the\@tempcmta}
984 \fi
985 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

986 \def\bxjs@postproc@layout{%
987 % ドライバ再設定
988   \ifx\bxjs@geometry@driver\relax\else
989     \let\Gm@driver\bxjs@geometry@driver
990   \fi
991 % textwidth 調整
992   \ifbxjs@whole@zw@lines
993     \@tempdimb=\textwidth
994     \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
995     \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
996     \advance\@tempdimb-\textwidth
997     \advance\oddsidemargin 0.5\@tempdimb
998     \advance\evensidemargin 0.5\@tempdimb
999   \fi
1000   \fullwidth=\textwidth
1001 %<*book>
1002   \ifdim\textwidth>\jsTextWidthLimit\Cwd
1003     \textwidth=\jsTextWidthLimit\Cwd
1004     \addtolength\evensidemargin{\fullwidth-\textwidth}
1005   \fi
1006 %</book>
1007 % textheight 調整
1008   \@tempdimb=\textheight
1009   \advance\textheight-\topskip

```

```

1010 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1011 \advance\textheight\topskip
1012 \advance\@tempdimb-\textheight
1013 \advance\topmargin0.5\@tempdimb
1014 % headheight 調整
1015 \@tempdima=\topskip
1016 \advance\headheight\@tempdima
1017 \advance\topmargin-\@tempdima
1018 % marginpar 設定
1019 \setlength\marginparsep{\columnsep}
1020 \setlength\marginparpush{\baselineskip}
1021 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1022     -\textwidth-10\jsc@mmm-\marginparsep}
1023 \ifbxjs@whole@zw@lines
1024     \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1025 \fi
1026 % 連動する変数
1027 \maxdepth=.5\topskip
1028 \stockwidth=\paperwidth
1029 \stockheight=\paperheight
1030 }

```

`\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。

```

1031 \edef\jsGeometryOptions{%
1032     \bxjs@layout@paper,\bxjs@layout}

```

■ geometry パッケージ読込

geoemtry オプションの値に応じて分岐する。

まずは geometry=class の場合。

```

1033 \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。

```

1034 \@onlypreamble\bxjs@revert
1035 \let\bxjs@revert\@empty
1036 \edef\bxjs@tmpa{\the\ht\strutbox}
1037 \ht\strutbox=10\p@?
1038 \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。

```

1039 \ifbxjs@papersize
1040   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1041     \PassOptionsToPackage{dvipdfm}{geometry}
1042   \fi
1043   \let\bxPapersizeSpecialDone=t
1044 \else
1045   \PassOptionsToPackage{driver=none}{geometry}
1046 \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。

```

1047 \edef\bxjs@nxt{%
1048   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}
1049 \AtBeginDocument{\bxjs@pre@geometry@hook}
1050 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1051 \bxjs@nxt \bxjs@revert
1052 \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。

```

1053 \let\bxjs@geometry@driver\Gm@driver
1054 \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。

```

1055 \@onlypreamble\bxjs@pre@geometry@hook
1056 \def\bxjs@pre@geometry@hook{%
1057   \@ifpackageloaded{geometry}{%
1058     \@ifpackagelater{geometry}{2010/02/12}{%}{%else
1059       \PackageWarningNoLine\bxjs@clsname
1060       {The 'geometry' package installed\MessageBreak
1061        is too old (< v5.0)}}%
1062     \if x\jsEngine \ifnum\mag=\@m\else
1063       \def\bxjs@Gm@driver{pdftex}
1064       \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1065       \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1066     \fi\fi
1067   }%
1068   \ifjsWithpTeXng
1069     \ifx\Gm@driver\@empty
1070       \def\Gm@driver{pdftex}%
1071     \fi
1072   \fi
1073 }{}

```

\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。

```

1074 \def\setpagelayout{\@ifstar
1075   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1076 \def\bxjs@modify@layout#1{%
1077   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}}%
1078   \bxjs@nxt\bxjs@postproc@layout}
1079 \def\bxjs@reset@layout#1{%
1080   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}}%
1081   \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。

```

1082 \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。

```

1083 \AtBeginDocument{\bxjs@check@page@layout}
1084 \@onlypreamble\bxjs@check@page@layout
1085 \def\bxjs@check@page@layout{%
1086   \ifdim\textwidth=.5\maxdimen
1087     \ClassError\bxjs@clsname
1088       {Page layout is not properly set}%
1089       {\@ehd}
1090   \fi}
1091 \def\jsUseMinimalPageLayout{%
1092   \setlength{\textwidth}{6.5in}%
1093   \setlength{\textheight}{8in}}

```

`\setpagelayout` はとりあえず無効にしておく。

```

1094 \let\bxjs@geometry@driver\relax
1095 \def\setpagelayout{\@ifstar
1096   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1097 \def\bxjs@pagelayout@a#1{%
1098   \ClassError\bxjs@clsname
1099   {Command '\string\setpagelayout' is not supported,\MessageBreak
1100    because 'geometry' value is not 'class'}\@eha}

```

`geometry=user` の場合の処理はここで終わり。

```

1101 \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。

```
1102 \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。

```
1103 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

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

[2003-08-23] ちょっと増やしました。

```
1104 \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。

```
1105 \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。

```
1106 \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。

```
1107 \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。

```
1108 \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。

```
1109 \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。

```
1110 \renewcommand{\floatpagefraction}{.8}
```

`\cdbltopnumber` 二段組のとき本文ページ上部に出力できる段抜きフロートの最大数です。

[2003-08-23] ちょっと増やしました。

```
1111 \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。

```
1112 \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。

```
1113 \renewcommand{\dblfloatpagefraction}{.8}
```

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

```
1114 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1115 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1116 \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。

```
1117 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1118 \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。

```
1119 \setlength\@fptop{0\p@? \@plus 1fil}
1120 \setlength\@fpsep{8\p@? \@plus 2fil}
1121 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
1122 \setlength\@dblfpsep{0\p@? \@plus 1fil}
1123 \setlength\@dblfpsep{8\p@? \@plus 2fil}
1124 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

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

`\pltx@cleartoleftpage`

`\pltx@cleartooddpage`

`\pltx@cleartoevenpage`

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

となっています。

```
1125 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1126 % \ifodd\c@page
1127 % \iftdir
1128 % \hbox{}\thispagestyle{empty}\newpage
1129 % \if@twocolumn\hbox{}\newpage\fi
1130 % \fi
1131 % \else
1132 % \ifydir
1133 % \hbox{}\thispagestyle{empty}\newpage
1134 % \if@twocolumn\hbox{}\newpage\fi
1135 % \fi
1136 % \fi\fi}
1137 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
1138 % \ifodd\c@page
1139 % \ifydir
1140 % \hbox{}\thispagestyle{empty}\newpage
1141 % \if@twocolumn\hbox{}\newpage\fi
1142 % \fi
1143 % \else
1144 % \iftdir
1145 % \hbox{}\thispagestyle{empty}\newpage
1146 % \if@twocolumn\hbox{}\newpage\fi
1147 % \fi
1148 % \fi\fi}
1149 \def\pltx@cleartooddpage{\clearpage\if@twoside
1150 \ifodd\c@page\else
1151 \hbox{}\thispagestyle{empty}\newpage
1152 \if@twocolumn\hbox{}\newpage\fi
1153 \fi\fi}
1154 \def\pltx@cleartoevenpage{\clearpage\if@twoside
1155 \ifodd\c@page
1156 \hbox{}\thispagestyle{empty}\newpage
1157 \if@twocolumn\hbox{}\newpage\fi
1158 \fi\fi}
```

BXJS クラスでは `\iftdir` 等が使えないので、横組を仮定した定義を用いる。

```
1159 \let\pltx@cleartorightpage\pltx@cleartooddpage
1160 \let\pltx@cleartoleftpage\pltx@cleartoevenpage
```

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

```
1161 %<book|report>
1162 \if@openleft
1163 \let\cleardoublepage\pltx@cleartoleftpage
1164 \else\if@openright
```

```

1165 \let\cleardoublepage\ptx@cleartorightpage
1166 \fi\fi
1167 %</book|report>

```

7 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

```

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

```

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

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

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

```

1168 % \def\ps@empty{%
1169 %   \let\mkboth\gobbletwo
1170 %   \let\@oddhead\@empty
1171 %   \let\@oddfoot\@empty
1172 %   \let\@evenhead\@empty
1173 %   \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。

```

1174 \def\ps@plainfoot{%

```



```

1175 \let\@mkboth\@gobbletwo
1176 \let\@oddhead\@empty
1177 \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1178 \let\@evenhead\@empty
1179 \let\@evenfoot\@oddfoot}
1180 \def\ps@plainhead{%
1181 \let\@mkboth\@gobbletwo
1182 \let\@oddfoot\@empty
1183 \let\@evenfoot\@empty
1184 \def\@evenhead{%
1185 \if@mparswitch \hss \fi
1186 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1187 \if@mparswitch\else \hss \fi}%
1188 \def\@oddhead{%
1189 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1190 %<book>\let\ps@plain\ps@plainhead
1191 %<!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。

```

1192 %<*article|slide>
1193 \if@twoside
1194 \def\ps@headings{%
1195 \let\@oddfoot\@empty
1196 \let\@evenfoot\@empty
1197 \def\@evenhead{\if@mparswitch \hss \fi
1198 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1199 \if@mparswitch\else \hss \fi}%
1200 \def\@oddhead{%
1201 \underline{%
1202 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1203 \let\@mkboth\markboth
1204 \def\sectionmark##1{\markboth{%
1205 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
1206 ##1}{}}%
1207 \def\subsectionmark##1{\markright{%
1208 \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsw\fi
1209 ##1}}%
1210 }
1211 \else % if not twoside
1212 \def\ps@headings{%
1213 \let\@oddfoot\@empty
1214 \def\@oddhead{%
1215 \underline{%
1216 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1217 \let\@mkboth\markboth
1218 \def\sectionmark##1{\markright{%

```

```

1219 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1220 ##1}}}}
1221 \fi
1222 %</article|slide>

```

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

```

1223 %<*book|report>
1224 \newif\if@omit@number
1225 \def\ps@headings{%
1226 \let\@oddfoot\@empty
1227 \let\@evenfoot\@empty
1228 \def\@evenhead{%
1229 \if@mparswitch \hss \fi
1230 \underline{\hbox to \fullwidth{\autoxspacing
1231 \textbf{\thepage}\hfil\leftmark}}}%
1232 \if@mparswitch\else \hss \fi}%
1233 \def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
1234 {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1235 \let\@mkboth\markboth
1236 \def\chaptermark##1{\markboth{%
1237 \ifnum \c@secnumdepth >\m@ne
1238 \if@mainmatter
1239 \if@omit@number\else
1240 \@chapapp\thechapter\@chappos\hskip1\jsZw
1241 \fi
1242 \fi
1243 \fi
1244 ##1}{}}}%
1245 \def\sectionmark##1{\markright{%
1246 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1247 ##1}}}%
1248 %</book|report>

```

最後は学会誌の場合です。

```

1249 %<*jspf>
1250 \def\ps@headings{%
1251 \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1252 \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1253 \def\@oddhead{\normalfont\hfil \@title \hfil}
1254 \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1255 %</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。

```

1256 \def\ps@myheadings{%
1257 \let\@oddfoot\@empty\let\@evenfoot\@empty
1258 \def\@evenhead{%

```

```

1259 \if@mparswitch \hss \fi%
1260 \hbox to \fullwidth{\thepage\hfil\leftmark}%
1261 \if@mparswitch\else \hss \fi}%
1262 \def\@oddhead{%
1263 \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1264 \let\@mkboth\@gobbletwo
1265 %<book|report> \let\chaptermark\@gobble
1266 \let\sectionmark\@gobble
1267 %<!book&!report> \let\subsectionmark\@gobble
1268 }

```

8 文書のマークアップ

8.1 表題

`\title` これらは L^AT_EX 本体で次のように定義されています。ここではコメントアウトした形で示します。

```

\date 1269 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1270 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1271 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1272 % \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。

```

1273 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1274 %\let\bxjs@subtitle\undefined

\title にフックを入れる。

1275 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1276 \AtBeginDocument{\bxjs@decl@subtitle}
1277 \def\bxjs@decl@subtitle{%
1278 \global\let\bxjs@decl@subtitle\relax
1279 \ifx\subtitle\undefined
1280 \global\let\subtitle\jsSubtitle
1281 \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。

```

1282 \def\bxjs@annihilate@subtitle{%
1283 \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1284 \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 1285 %<*jspf>
\keywords 1286 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1287 \newcommand*{\author}[1]{\gdef\@author{#1}}
1288 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1289 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1290 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1291 %</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。

```

1292 \def\plainifnotempty{%
1293   \ifx \@oddhead \@empty
1294     \ifx \@oddfoot \@empty
1295       \else
1296         \thispagestyle{plainfoot}%
1297       \fi
1298     \else
1299       \thispagestyle{plainhead}%
1300     \fi}

```

`\maketitle` 表題を出力します。著者名を出力する部分は、欧文の標準クラスファイルでは `\large`、和文のものでは `\Large` になっていましたが、ここでは `\large` にしました。

[2016-11-16] 新設された `nomag` および `nomag*` オプションの場合をデフォルト (`usemag` 相当) に合わせるため、`\smallskip` を `\jsc@smallskip` に置き換えました。`\smallskip` のままでは `nomag(*)` の場合にスケールしくなり、レイアウトが変わってしまいます。

```

1301 %<*article|book|report|slide>
1302 \if@titlepage
1303   \newcommand{\maketitle}{%
1304     \begin{titlepage}%
1305       \let\footnotesize\small
1306       \let\footnoterule\relax
1307       \let\footnote\thanks
1308       \null\vfil
1309       \if@slide
1310         {\footnotesize \@date}%
1311       \begin{center}
1312         \mbox{} \ \ [1\jsZw]
1313         \large
1314         {\maybeblue\hrule height0\p? depth2\p?\relax}\par
1315         \jsc@smallskip
1316         \@title

```

```

1317         \ifx\bxjs@subtitle\@undefined\else
1318         \par\vskip\z@
1319         {\small \bxjs@subtitle\par}
1320     \fi
1321     \jsc@smallskip
1322     {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1323     \vfill
1324     {\small \@author}%
1325 \end{center}
1326 \else
1327 \vskip 60\p@?
1328 \begin{center}%
1329 {\LARGE \@title \par}%
1330 \ifx\bxjs@subtitle\@undefined\else
1331 \vskip5\p@?
1332 {\normalsize \bxjs@subtitle\par}
1333 \fi
1334 \vskip 3em%
1335 {\large
1336 \lineskip .75em
1337 \begin{tabular}[t]{c}%
1338 \author
1339 \end{tabular}\par}%
1340 \vskip 1.5em
1341 {\large \@date \par}%
1342 \end{center}%
1343 \fi
1344 \par
1345 \@thanks\vfil\null
1346 \end{titlepage}%
1347 \setcounter{footnote}{0}%
1348 \global\let\thanks\relax
1349 \global\let\maketitle\relax
1350 \global\let\@thanks\@empty
1351 \global\let\@author\@empty
1352 \global\let\@date\@empty
1353 \global\let\@title\@empty
1354 \global\let\title\relax
1355 \global\let\author\relax
1356 \global\let\date\relax
1357 \global\let\and\relax
1358 \bxjs@annihilate@subtitle
1359 }%
1360 \else
1361 \newcommand{\maketitle}{\par
1362 \begin{group
1363 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1364 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1365 \long\def\@makefntext##1{\advance\leftskip 3\jsZw

```

```

1366     \parindent 1\jsZw\noindent
1367     \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1368     \if@twocolumn
1369         \ifnum \col@number=\@ne
1370             \@maketitle
1371         \else
1372             \twocolumn[\@maketitle]%
1373         \fi
1374     \else
1375         \newpage
1376         \global\@topnum\z@ % Prevents figures from going at top of page.
1377         \@maketitle
1378     \fi
1379     \plainifnotempty
1380     \@thanks
1381 \endgroup
1382 \setcounter{footnote}{0}%
1383 \global\let\thanks\relax
1384 \global\let\maketitle\relax
1385 \global\let\@thanks\@empty
1386 \global\let\@author\@empty
1387 \global\let\@date\@empty
1388 \global\let\@title\@empty
1389 \global\let\title\relax
1390 \global\let\author\relax
1391 \global\let\date\relax
1392 \global\let\and\relax
1393 \bxjs@annihilate@subtitle
1394 }

```

`\@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。

```

1395 \def\@maketitle{%
1396     \newpage\null
1397     \vskip 2em
1398     \begin{center}%
1399         \let\footnote\thanks
1400         {\LARGE \@title \par}%
1401         \ifx\bxjs@subtitle\@undefined\else
1402             \vskip3\p@?
1403             {\normalsize \bxjs@subtitle\par}
1404         \fi
1405         \vskip 1.5em
1406         {\large
1407             \lineskip .5em
1408             \begin{tabular}[t]{c}%
1409                 \@author
1410             \end{tabular}\par}%
1411         \vskip 1em
1412         {\large \@date}%

```

```

1413 \end{center}%
1414 \par\vskip 1.5em
1415 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1416 }
1417 \fi
1418 %</article|book|report|slide>
1419 %<*jspf>
1420 \newcommand\maketitle{\par
1421 \begingroup
1422 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1423 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1424 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1425 \parindent 1\jsZw\noindent
1426 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1427 \twocolumn[\@maketitle]%
1428 \plainifnotempty
1429 \@thanks
1430 \endgroup
1431 \setcounter{footnote}{0}%
1432 \global\let\thanks\relax
1433 \global\let\maketitle\relax
1434 \global\let\@thanks\@empty
1435 \global\let\@author\@empty
1436 \global\let\@date\@empty
1437 % \global\let\@title\@empty % \@title は柱に使う
1438 \global\let\title\relax
1439 \global\let\author\relax
1440 \global\let\date\relax
1441 \global\let\and\relax
1442 \ifx\authors@mail\@undefined\else{%
1443 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1444 \footnotetext[0]{\itshape\authors@mail}%
1445 }\fi
1446 \global\let\authors@mail\@undefined}
1447 \def\@maketitle{%
1448 \newpage\null
1449 \vskip 6em % used to be 2em
1450 \begin{center}
1451 \let\footnote\thanks
1452 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1453 \lineskip .5em
1454 \ifx\@author\@undefined\else
1455 \vskip 1em
1456 \begin{tabular}[t]{c}%
1457 \@author
1458 \end{tabular}\par
1459 \fi
1460 \ifx\@etitle\@undefined\else
1461 \vskip 1em

```

```

1462     {\large \@etitle \par}%
1463 \fi
1464 \ifx\@eauthor\@undefined\else
1465     \vskip 1em
1466     \begin{tabular}[t]{c}%
1467         \@eauthor
1468     \end{tabular}\par
1469 \fi
1470 \vskip 1em
1471 \@date
1472 \end{center}
1473 \vskip 1.5em
1474 \centerline{\box\@abstractbox}
1475 \ifx\@keywords\@undefined\else
1476     \vskip 1.5em
1477     \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1478 \fi
1479 \vskip 1.5em}
1480 %</jspf>

```

8.2 章・節

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

```

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

```

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

次は `\@startsection` の定義です。情報処理学会論文誌スタイルファイル

(ipsjcommon.sty) を参考にさせていただきましたが、完全に行送りが \baselineskip の整数倍にならなくてもいいから前の行と重ならないようにしました。

```

1481 \def\@startsection#1#2#3#4#5#6{%
1482   \if@noskipsec \leavevmode \fi
1483   \par
1484 % 見出し上の空きを \@tempskipa にセットする
1485   \@tempskipa #4\relax
1486 % \@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
1487   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1488 % 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
1489   \ifdim \@tempskipa <\z@
1490     \@tempskipa -\@tempskipa \@afterindentfalse
1491   \fi
1492   \if@nobreak
1493 %   \everypar{\everyparhook}% これは間違い
1494     \everypar{}%
1495   \else
1496     \addpenalty\@secpenalty
1497 % 次の行は削除
1498 %   \addvspace\@tempskipa
1499 % 次の \noindent まで追加
1500     \ifdim \@tempskipa >\z@
1501       \if@slide\else
1502         \null
1503         \vspace*{-\baselineskip}%
1504       \fi
1505       \vskip\@tempskipa
1506     \fi
1507   \fi
1508   \noindent
1509 % 追加終わり
1510   \@ifstar
1511     {\@ssect{#3}{#4}{#5}{#6}}%
1512     {\@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。

```

1513 \def\@sect#1#2#3#4#5#6[#7]#8{%
1514   \ifnum #2>\c@secnumdepth
1515     \let\@svsec\@empty
1516   \else
1517     \refstepcounter{#1}%
1518     \protected@edef\@svsec{\@secntformat{#1}\relax}%
1519   \fi
1520 % 見出し後の空きを \@tempskipa にセット
1521   \@tempskipa #5\relax
1522 % 条件判断の順序を入れ替えました
1523   \ifdim \@tempskipa <\z@

```

```

1524 \def\@svsechd{%
1525   #6{\hskip #3\relax
1526   \@svsec #8}%
1527   \csname #1mark\endcsname{#7}%
1528   \addcontentsline{toc}{#1}{%
1529     \ifnum #2>\c@secnumdepth \else
1530       \protect\numberline{\csname the#1\endcsname}%
1531     \fi
1532     #7}}% 目次にフルネームを載せるなら #8
1533 \else
1534   \begingroup
1535     \interlinepenalty \@M % 下から移動
1536     #6{%
1537       \@hangfrom{\hskip #3\relax\@svsec}%
1538 %     \interlinepenalty \@M % 上に移動
1539     #8\@@par}%
1540   \endgroup
1541   \csname #1mark\endcsname{#7}%
1542   \addcontentsline{toc}{#1}{%
1543     \ifnum #2>\c@secnumdepth \else
1544       \protect\numberline{\csname the#1\endcsname}%
1545     \fi
1546     #7}}% 目次にフルネームを載せるならここは #8
1547   \fi
1548   \@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。

```

1549 \def\@xsect#1{%
1550 % 見出しの後ろの空きを \@tempskipa にセット
1551   \@tempskipa #1\relax
1552 % 条件判断の順序を変えました
1553   \ifdim \@tempskipa<\z@
1554     \@nbreakfalse
1555     \global\@noskipsectrue
1556     \everypar{%
1557       \if@noskipsec
1558         \global\@noskipsecfalse
1559         {\setbox\z@\lastbox}%
1560         \clubpenalty\@M
1561         \begingroup \@svsechd \endgroup
1562         \unskip
1563         \@tempskipa #1\relax
1564         \hskip -\@tempskipa
1565         \bxjs@ltj@inhibitglue
1566       \else

```

```

1567      \clubpenalty \@clubpenalty
1568      \everypar{\everyparhook}%
1569      \fi\everyparhook}%
1570  \else
1571      \par \nobreak
1572      \vskip \@tempskipa
1573      \@afterheading
1574  \fi
1575  \if@slide
1576      {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1577       \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1578       \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1579  \fi
1580  \par % 2000-12-18
1581  \ignorespaces}
1582 \def\@ssect#1#2#3#4#5{%
1583   \@tempskipa #3\relax
1584   \ifdim \@tempskipa<\z@
1585     \def\@svsechd{#4{\hskip #1\relax #5}}%
1586   \else
1587     \begingroup
1588     #4{%
1589       \@hangfrom{\hskip #1}%
1590       \interlinepenalty \@M #5\@@par}%
1591   \endgroup
1592  \fi
1593  \@xsect{#3}}

```

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

```

1594 \let\bxjs@ltj@inhibitglue\@empty

```

■柱関係の命令

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

```

\subsectionmark 1595 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1596 % \newcommand*\sectionmark[1]{}
1597 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1598 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1599 % \newcommand*\paragraphmark[1]{}
1600 % \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。

```

1601 %<!book&!report>\setcounter{secnumdepth}{3}

```

```
1602 %<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 1603 \newcounter{part}
```

```
\c@subsubsection 1604 %<book|report>\newcounter{chapter}
```

```
1605 %<book|report>\newcounter{section}[chapter]
```

```
\c@paragraph 1606 %<!book&!report>\newcounter{section}
```

```
\c@subparagraph 1607 \newcounter{subsection}[section]
```

```
1608 \newcounter{subsubsection}[subsection]
```

```
1609 \newcounter{paragraph}[subsubsection]
```

```
1610 \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。

`\thesection` `\arabic{COUNTER}` 1, 2, 3, ...

`\thesubsection` `\roman{COUNTER}` i, ii, iii, ...

`\thesubsubsection` `\Roman{COUNTER}` I, II, III, ...

`\theparagraph` `\alph{COUNTER}` a, b, c, ...

`\thesubparagraph` `\Alph{COUNTER}` A, B, C, ...

`\kansuji{COUNTER}` 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。

```
1611 \renewcommand{\thepart}{\@Roman\c@part}
```

```
1612 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
```

```
1613 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
```

```
1614 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
```

```
1615 %<*book|report>
```

```
1616 \renewcommand{\thechapter}{\@arabic\c@chapter}
```

```
1617 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
```

```
1618 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
```

```
1619 %</book|report>
```

```
1620 \renewcommand{\thesubsubsection}{%
```

```
1621    \thesubsection.\@arabic\c@subsubsection}
```

```
1622 \renewcommand{\theparagraph}{%
```

```
1623    \thesubsubsection.\@arabic\c@paragraph}
```

```
1624 \renewcommand{\thesubparagraph}{%
```

```
1625    \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。

```
1626 %<book|report>\newcommand{\@chapapp}{\prechaptername}
```

```
1627 %<book|report>\newcommand{\@chappos}{\postchaptername}
```

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

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

[2017-03-05] `\frontmatter` と `\mainmatter` の2つの命令は、改丁または改ページした後で `\pagenumbering{...}` でノンブルを1にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが、これではノンブルをリセットする際に偶奇逆転が起こる場合があります。 `openany` がどうかにかかわらず奇数ページまで繰るように修正することで、問題を解消しました。実は、 \LaTeX の標準クラスでは1998年に修正されていた問題です (コミュニティ版 \LaTeX の標準クラス 2017/03/05 も参照)。

```
1628 %<*book|report>
1629 \pltx@cleartooddpage
1630 \@mainmatterfalse
1631 \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。

```
1632 \newcommand\mainmatter{%
1633 \pltx@cleartooddpage
1634 \@mainmattertrue
1635 \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。

```
1636 \newcommand\backmatter{%
1637 \if@openleft
1638 \cleardoublepage
1639 \else\if@openright
1640 \cleardoublepage
1641 \else
1642 \clearpage
1643 \fi\fi
1644 \@mainmatterfalse}
1645 %</book|report>
```

■部

`\part` 新しい部を始めます。

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

`\secdef{星なし}{星あり}`

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

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

まず book と report のクラス以外です。

```
1646 %<*!book&!report>
1647 \newcommand\part{%
1648   \if@noskipsec \leavevmode \fi
1649   \par
1650   \addvspace{4ex}%
1651   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1652   \secdef\@part\@spart}
1653 %</!book&!report>
```

book および report クラスの場合は、少し複雑です。

```
1654 %<*book|report>
1655 \newcommand\part{%
1656   \if@openleft
1657     \cleardoublepage
1658   \else\if@openright
1659     \cleardoublepage
1660   \else
1661     \clearpage
1662   \fi\fi
1663   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1664   \if@twocolumn
1665     \onecolumn
1666     \@restonecoltrue
1667   \else
1668     \@restonecolfalse
1669   \fi
1670   \null\vfil
1671   \secdef\@part\@spart}
1672 %</book|report>
```

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

book および report クラス以外では secnumdepth が -1 より大きいとき部番号を付けます。

```
1673 %<*!book&!report>
1674 \def\@part[#1]#2{%
1675   \ifnum \c@secnumdepth >\m@ne
1676     \refstepcounter{part}%
1677     \addcontentsline{toc}{part}{%
1678       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1679   \else
1680     \addcontentsline{toc}{part}{#1}%
1681   \fi
1682   \markboth{}{}%
1683   {\parindent\z@
1684     \raggedright
1685     \interlinepenalty \@M
1686     \normalfont
1687     \ifnum \c@secnumdepth >\m@ne
```

```

1688      \Large\headfont\prepartname\thepart\postpartname
1689      \par\nobreak
1690      \fi
1691      \huge \headfont #2%
1692      \markboth{}{}\par}%
1693      \nobreak
1694      \vskip 3ex
1695      \@afterheading}
1696 %</!book&!report>

```

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

```

1697 %<*book|report>
1698 \def\@part[#1]#2{%
1699   \ifnum \c@secnumdepth >-2\relax
1700     \refstepcounter{part}%
1701     \addcontentsline{toc}{part}{%
1702       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1703   \else
1704     \addcontentsline{toc}{part}{#1}%
1705   \fi
1706   \markboth{}{}\%
1707   {\centering
1708     \interlinepenalty \@M
1709     \normalfont
1710     \ifnum \c@secnumdepth >-2\relax
1711       \huge\headfont \prepartname\thepart\postpartname
1712       \par\vskip20\p@?
1713     \fi
1714     \Huge \headfont #2\par}%
1715   \@endpart}
1716 %</book|report>

```

\@spart 番号を付けない部です。

```

1717 %<*!book&!report>
1718 \def\@spart#1{%
1719   \parindent \z@ \raggedright
1720   \interlinepenalty \@M
1721   \normalfont
1722   \huge \headfont #1\par}%
1723   \nobreak
1724   \vskip 3ex
1725   \@afterheading}
1726 %</!book&!report>
1727 %<*book|report>
1728 \def\@spart#1{%
1729   \centering
1730   \interlinepenalty \@M
1731   \normalfont
1732   \Huge \headfont #1\par}%

```

```

1733 \endpart}
1734 %</book|report>

```

`\endpart` `\@part` と `\@spart` の最後で実行されるマクロです。両面印刷のときは白ページを追加します。二段組のときには、二段組に戻します。

[2016-12-13] `openany` のときには白ページが追加されるのは変なので、その場合は追加しないようにしました。このバグは L^AT_EX では `classes.dtx` v1.4b (2000/05/19) で修正されています。

```

1735 %<*book|report>
1736 \def\endpart{\vfil\newpage
1737 \if@twoside
1738 \if@openleft %% added (2017/02/24)
1739 \null\thispagestyle{empty}\newpage
1740 \else\if@openright %% added (2016/12/13)
1741 \null\thispagestyle{empty}\newpage
1742 \fi\fi %% added (2016/12/13, 2017/02/24)
1743 \fi
1744 \if@restonecol
1745 \twocolumn
1746 \fi}
1747 %</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。

```

1748 %<*book|report>
1749 \newcommand{\chapter}{%
1750 \if@openleft\cleardoublepage\else
1751 \if@openright\cleardoublepage\else\clearpage\fi\fi
1752 \plainifnotempty % 元: \thispagestyle{plain}
1753 \global\@topnum\z@
1754 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1755 \secdef
1756 {\@omit@numberfalse\@chapter}%
1757 {\@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。

```

1758 \def\@chapter[#1]#2{%
1759 \ifnum \c@secnumdepth >\m@ne
1760 \if@mainmatter
1761 \refstepcounter{chapter}%
1762 \typeout{\@chapapp\thechapter\@chappos}%
1763 \addcontentsline{toc}{chapter}%
1764 {\protect\numberline
1765 % {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%

```



```

1766      {\@chapapp\thechapter\@chappos}%
1767      #1}%
1768      \else\addcontentsline{toc}{chapter}{#1}\fi
1769      \else
1770      \addcontentsline{toc}{chapter}{#1}%
1771      \fi
1772      \chaptermark{#1}%
1773      \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
1774      \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
1775      \if@twocolumn
1776      \@topnewpage[\@makechapterhead{#2}]%
1777      \else
1778      \@makechapterhead{#2}%
1779      \@afterheading
1780      \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。

```

1781 \def\@makechapterhead#1{%
1782   \vspace*{2\Cvs}% 欧文は 50pt
1783   {\parindent \z@ \raggedright \normalfont
1784     \ifnum \c@secnumdepth >\m@ne
1785       \if@mainmatter
1786         \huge\headfont \@chapapp\thechapter\@chappos
1787         \par\nobreak
1788         \vskip \Cvs % 欧文は 20pt
1789       \fi
1790     \fi
1791     \interlinepenalty\@M
1792     \Huge \headfont #1\par\nobreak
1793     \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。

```

1794 \def\@schapter#1{%
1795   \chaptermark{#1}%
1796   \if@twocolumn
1797     \@topnewpage[\@makeschapterhead{#1}]%
1798   \else
1799     \@makeschapterhead{#1}\@afterheading
1800   \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

1801 \def\@makeschapterhead#1{%
1802   \vspace*{2\Cvs}% 欧文は 50pt
1803   {\parindent \z@ \raggedright
1804     \normalfont
1805     \interlinepenalty\@M
1806     \Huge \headfont #1\par\nobreak
1807     \vskip 3\Cvs}} % 欧文は 40pt
1808 %</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。

```
1809 \if@twocolumn
1810   \newcommand{\section}{%
1811     %<jspf>\ifx\maketitle\relax\else\maketitle\fi
1812     \@startsection{section}{1}{\z@}%
1813     %<kiyou>    {0.6\Cvs}{0.4\Cvs}%
1814     %<kiyou>    {\Cvs}{0.5\Cvs}%
1815     %    {\normalfont\large\headfont\@secapp}}
1816     {\normalfont\large\headfont\raggedright}}
1817 \else
1818   \newcommand{\section}{%
1819     \if@slide\clearpage\fi
1820     \@startsection{section}{1}{\z@}%
1821     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1822     {.5\Cvs \@plus.3\Cdp}% 後アキ
1823     %    {\normalfont\Large\headfont\@secapp}}
1824     {\normalfont\Large\headfont\raggedright}}
1825 \fi
```

`\subsection` 同上です。

```
1826 \if@twocolumn
1827   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1828     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1829     {\normalfont\normalsize\headfont}}
1830 \else
1831   \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1832     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1833     {.5\Cvs \@plus.3\Cdp}% 後アキ
1834     {\normalfont\large\headfont}}
1835 \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)。

```
1836 \if@twocolumn
1837   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1838     {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1839     {\normalfont\normalsize\headfont}}
1840 \else
1841   \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1842     {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1843     {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
1844     {\normalfont\normalsize\headfont}}
1845 \fi
```

`\paragraph` 見出しの後ろで改行されません。

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

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

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

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

段落のマーク（■）が必ず和文フォントで出力されるようにする。

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

```
1846 \newcommand\jsParagraphMark{\jsJaChar{■}}
1847 \ifx\bxjs@paragraph@mark\@empty
1848   \let\jsParagraphMark\@empty
1849 \else\ifx\bxjs@paragraph@mark\undefined\else
1850   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
1851 \fi\fi
1852 \let\jsJaChar\@empty
1853 \if@twocolumn
1854   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1855     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1856     <jspf> {\normalfont\normalsize\headfont}}
1857     <!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1858 \else
1859   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1860     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1861     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1862     <jspf> {\normalfont\normalsize\headfont}}
1863     <!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1864 \fi
```

`\subparagraph` 見出しの後ろで改行されません。

```
1865 \if@twocolumn
1866   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1867     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1868     {\normalfont\normalsize\headfont}}
1869 \else
1870   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1871     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1872     {\normalfont\normalsize\headfont}}
1873 \fi
```

8.3 リスト環境

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

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

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```
1874 \if@slide
1875   \setlength\leftmargini{1\jsZw}
1876 \else
1877   \if@twocolumn
1878     \setlength\leftmargini{2\jsZw}
1879   \else
1880     \setlength\leftmargini{3\jsZw}
1881   \fi
1882 \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 1883 \if@slide
\leftmarginv 1884   \setlength\leftmarginii {1\jsZw}
1885   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 1886   \setlength\leftmarginiv {1\jsZw}
1887   \setlength\leftmarginv {1\jsZw}
1888   \setlength\leftmarginvi {1\jsZw}
1889 \else
1890   \setlength\leftmarginii {2\jsZw}
1891   \setlength\leftmarginiii{2\jsZw}
1892   \setlength\leftmarginiv {2\jsZw}
1893   \setlength\leftmarginv {1\jsZw}
1894   \setlength\leftmarginvi {1\jsZw}
1895 \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。

```
1896 \setlength \labelsep {0.5\jsZw} % .5em
1897 \setlength \labelwidth{\leftmargini}
1898 \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。

```
1899 \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`

`\@itempenalty`

```

1900 \@beginparpenalty -\@lowpenalty
1901 \@endparpenalty -\@lowpenalty
1902 \@itempenalty -\@lowpenalty

```

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

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

```

1903 \def\@listi{\leftmargin\leftmargini
1904   \parsep \z@
1905   \topsep 0.5\baselineskip
1906   \itemsep \z@ \relax}
1907 \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）。

```

1908 \@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 1909 \def\@listii{\leftmargin\leftmarginii
\@listiv 1910   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1911   \topsep \z@
\@listv 1912   \parsep \z@
\@listvi 1913   \itemsep\parsep}
1914 \def\@listiii{\leftmargin\leftmarginiii
1915   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1916   \topsep \z@
1917   \parsep \z@
1918   \itemsep\parsep}
1919 \def\@listiv {\leftmargin\leftmarginiv
1920   \labelwidth\leftmarginiv
1921   \advance\labelwidth-\labelsep}
1922 \def\@listv {\leftmargin\leftmarginv
1923   \labelwidth\leftmarginv
1924   \advance\labelwidth-\labelsep}
1925 \def\@listvi {\leftmargin\leftmarginvi
1926   \labelwidth\leftmarginvi
1927   \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

`\theenumiii`

`\theenumiv`

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。

```
1928 \renewcommand{\theenumi}{\@arabic\c@enumi}
1929 \renewcommand{\theenumii}{\@alph\c@enumii}
1930 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1931 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

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

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

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

```
1932 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1933 \newcommand*{\jsInJaParen}[1]{%
1934   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1935 \newcommand{\labelenumi}{\theenumi.}
1936 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1937 \newcommand{\labelenumiii}{\theenumiii.}
1938 \newcommand{\labelenumiv}{\theenumiv.}
```

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

```
\p@enumiv 1939 \renewcommand{\p@enumii}{\theenumi}
1940 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1941 \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 1942 \newcommand\labelitemi{\textbullet}
\labelitemiii 1943 \newcommand\labelitemii{\normalfont\bfseries \textendash}
1944 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 1945 \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。

```
1946 \newenvironment{description}{%
1947   \list{}{%
```

```

1948 \labelwidth=\leftmargin
1949 \labelsep=1\jsZw
1950 \advance \labelwidth by -\labelsep
1951 \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。

```

1952 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}

```

■概要

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

JSPF スタイルでは実際の出力は `\maketitle` で行われます。

`bxjsreport` クラスの `abstract` 環境は：

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

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

```

1953 %<*book|report>
1954 \newenvironment{chapterabstract}{%
1955 \begin{list}{%
1956 \listparindent=1\jsZw
1957 \itemindent=\listparindent
1958 \rightmargin=\z@
1959 \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
1960 %</book|report>

```

“普通の” `abstract` 環境の定義。

```

1961 %<*article|report|slide>
1962 \newbox\@abstractbox
1963 \if@titlepage
1964 \newenvironment{abstract}{%
1965 \titlepage
1966 \null\vfil
1967 \@beginparpenalty\@lowpenalty
1968 \begin{center}%
1969 \headfont \abstractname
1970 \@endparpenalty\@M
1971 \end{center}%

```

BXJS クラスでは、概要の最初の段落に段落下げが入るようにする。

```

1972   \par}%
1973   {\par\vfil\null\endtitlepage}
1974 \else
1975   \newenvironment{abstract}{%
1976     \if@twocolumn
1977       \ifx\maketitle\relax
1978         \section*{\abstractname}%
1979       \else
1980         \global\setbox\@abstractbox\hbox\bgroup
1981         \begin{minipage}[b]{\textwidth}
1982           \small\parindent1\jsZw
1983           \begin{center}%
1984             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1985           \end{center}%
1986           \list{}{%
1987             \listparindent\parindent
1988             \itemindent \listparindent
1989             \rightmargin \leftmargin}%
1990           \item\relax
1991         \fi
1992       \else
1993         \small
1994         \begin{center}%
1995           {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1996         \end{center}%
1997         \list{}{%
1998           \listparindent\parindent
1999           \itemindent \listparindent
2000           \rightmargin \leftmargin}%
2001         \item\relax
2002       \fi}{\if@twocolumn
2003         \ifx\maketitle\relax
2004         \else
2005           \endlist\end{minipage}\egroup
2006         \fi
2007       \else
2008         \endlist
2009       \fi}
2010 \fi
2011 %</article|report|slide>
2012 %<*jspf>
2013 \newbox\@abstractbox
2014 \newenvironment{abstract}{%
2015   \global\setbox\@abstractbox\hbox\bgroup
2016   \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
2017     \small
2018     \if@english \parindent6mm \else \parindent1\jsZw \fi}%

```



```

2019 {\end{minipage}\egroup}
2020 %</jspf>

```

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

```

2021 %<*book|report>
2022 \ifbxjs@force@chapterabstract
2023 \let\abstract\chapterabstract
2024 \let\endabstract\endchapterabstract
2025 \fi
2026 %</book|report>

```

■キーワード

`keywords` キーワードを準備する環境です。実際の出力は `\maketitle` で行われます。

```

2027 %<*jspf>
2028 %\newbox\@keywordsbox
2029 %\newenvironment{keywords}{%
2030 % \global\setbox\@keywordsbox\hbox\bgroup
2031 % \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
2032 % \small\parindent0\jsZw}%
2033 % {\end{minipage}\egroup}
2034 %</jspf>

```

■verse 環境

`verse` 詩のための `verse` 環境です。

```

2035 \newenvironment{verse}{%
2036 \let \=\@centercr
2037 \list{}{%
2038 \itemsep \z@
2039 \itemindent -2\jsZw % 元: -1.5em
2040 \listparindent\itemindent
2041 \rightmargin \z@
2042 \advance\leftmargin 2\jsZw}% 元: 1.5em
2043 \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。

```

2044 \newenvironment{quotation}{%
2045 \list{}{%
2046 \listparindent\parindent
2047 \itemindent\listparindent
2048 \rightmargin \z@}%
2049 \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。

```
2050 \newenvironment{quote}%
2051   {\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。

```
2052 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2053   \item[\hskip \labelsep{\headfont #1\ #2}]}
2054 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2055   \item[\hskip \labelsep{\headfont #1\ #2 (#3) ]}}
```

titlepage タイトルを独立のページに出力するのに使われます。

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

```
2056 \newenvironment{titlepage}{%
2057   %<book> \pltx@cleartooddpage %% 2017-02-24
2058   \if@twocolumn
2059     \@restonecoltrue\onecolumn
2060   \else
2061     \@restonecolfalse\newpage
2062   \fi
2063   \thispagestyle{empty}%
2064   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-24
2065 }%
2066 {\if@restonecol\twocolumn \else \newpage \fi
2067   \if@twoside\else
2068     \setcounter{page}\@ne
2069   \fi}
```

■付録

\appendix 本文と付録を分離するコマンドです。

```

2070 %<*!book&!report>
2071 \newcommand{\appendix}{\par
2072   \setcounter{section}{0}%
2073   \setcounter{subsection}{0}%
2074   \gdef\presectionname{\appendixname}%
2075   \gdef\postsectionname{}}%
2076 % \gdef\thesection{\@Alph{c@section}% [2003-03-02]
2077   \gdef\thesection{\presectionname\@Alph{c@section}\postsectionname}%
2078   \gdef\thesubsection{\@Alph{c@section}.\@arabic{c@subsection}}
2079 %</!book&!report>
2080 %<*book|report>
2081 \newcommand{\appendix}{\par
2082   \setcounter{chapter}{0}%
2083   \setcounter{section}{0}%
2084   \gdef\@chapapp{\appendixname}%
2085   \gdef\@chappos{}}%
2086   \gdef\thechapter{\@Alph{c@chapter}}
2087 %</book|report>

```

8.4 パラメータの設定

■array と tabular 環境

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

```
2088 \setlength\arraycolsep{5\p@?}
```

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

```
2089 \setlength\tabcolsep{6\p@?}
```

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

```
2090 \setlength\arrayrulewidth{.4\p@}
```

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

```
2091 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \' コマンドで入るアキです。

```
2092 \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。

```
2093 \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。

```
2094 \setlength\fboxsep{3\p@?}
2095 \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。

```
2096 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
2097 %<*book|report>
2098 \@addtoreset{equation}{chapter}
2099 \renewcommand\theequation
2100 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
2101 %</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。

```
2102 % \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。

```
2103 % \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。

```
2104 % \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip\@italiccorr)}}
```

8.5 フロート

タイプ `TYPE` のフロートオブジェクトを扱うには、次のマクロを定義します。

`\fps@TYPE` フロートを置く位置 (float placement specifier) です。

`\ftype@TYPE` フロートの番号です。2 の累乗 (1, 2, 4, ...) でなければなりません。

`\ext@TYPE` フロートの目次を出力するファイルの拡張子です。

`\fnum@TYPE` キャプション用の番号を生成するマクロです。

`\@makecaption`(*num*)(*text*) キャプションを出力するマクロです。*num* は `\fnum@...` の生成する番号、*text* はキャプションのテキストです。テキストは適当な幅の `\parbox` に入ります。

■figure 環境

`\c@figure` 図番号のカウンタです。

`\thefigure` 図番号を出力するコマンドです。

```
2105 %<!*book&!report>
2106 \newcounter{figure}
2107 \renewcommand \thefigure {\@arabic\c@figure}
2108 %</!*book&!report>
2109 %<*book|report>
2110 \newcounter{figure}[chapter]
2111 \renewcommand \thefigure
2112     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@figure}
2113 %</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 2114 \def\fps@figure{tbp}
2115 \def\ftype@figure{1}
\fnun@figure 2116 \def\ext@figure{lof}
2117 \def\fnun@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2118 \newenvironment{figure}%
2119     {\@float{figure}}%
2120     {\end@float}
2121 \newenvironment{figure*}%
2122     {\@dblfloat{figure}}%
2123     {\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。

```
2124 %<!*book&!report>
2125 \newcounter{table}
2126 \renewcommand\thetable{\@arabic\c@table}
2127 %</!*book&!report>
2128 %<*book|report>
2129 \newcounter{table}[chapter]
2130 \renewcommand \thetable
2131     {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2132 %</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 2133 \def\fps@table{tbp}
2134 \def\ftype@table{2}
\fnun@table 2135 \def\ext@table{lot}
2136 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` * は段抜きのフロートです。

`table*`

```

2137 \newenvironment{table}%
2138         {\@float{table}}%
2139         {\end@float}
2140 \newenvironment{table*}%
2141         {\@dblfloat{table}}%
2142         {\end@dblfloat}

```

8.6 キャプション

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

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

```

2143 \newlength\abovecaptionskip
2144 \newlength\belowcaptionskip
2145 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2146 \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。

```

2147 %<!*jspf>
2148 % \long\def\@makecaption#1#2{{\small
2149 %   \advance\leftskip1cm
2150 %   \advance\rightskip1cm
2151 %   \vskip\abovecaptionskip
2152 %   \sbox\@tempboxa{#1\hskip1\jszw\relax #2}%
2153 %   \ifdim \wd\@tempboxa >\hsize
2154 %     #1\hskip1\jszw\relax #2\par
2155 %   \else
2156 %     \global \minipagefalse
2157 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2158 %   \fi
2159 %   \vskip\belowcaptionskip}}
2160 \long\def\@makecaption#1#2{{\small
2161   \advance\leftskip .0628\linewidth
2162   \advance\rightskip .0628\linewidth
2163   \vskip\abovecaptionskip
2164   \sbox\@tempboxa{#1\zwspace#2}%
2165   \ifdim \wd\@tempboxa <\hsize \centering \fi
2166   #1\zwspace#2\par
2167   \vskip\belowcaptionskip}}
2168 %</!*jspf>
2169 %<!*jspf>
2170 \long\def\@makecaption#1#2{%

```

```

2171 \vskip\abovecaptionskip
2172 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2173 \ifdim \wd\@tempboxa >\hsize
2174     {\small\sffamily
2175         \list{#1}{%
2176             \renewcommand{\makelabel}[1]{##1\hfil}
2177             \itemsep \z@
2178             \itemindent \z@
2179             \labelsep \z@
2180             \labelwidth 11mm
2181             \listparindent\z@
2182             \leftmargin 11mm}\item\relax #2\endlist}
2183 \else
2184     \global \@minipagefalse
2185     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2186 \fi
2187 \vskip\belowcaptionskip}
2188 %</jspf>

```

9 フォントコマンド

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

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

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

```

\if@jsc@warnoldfontcmd


---


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


---


2189 \newif\if@jsc@warnoldfontcmd
2190 \@jsc@warnoldfontcmdtrue
2191 \newif\if@jsc@warnoldfontcmdexception
2192 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand
2193 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2194     \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2195     \DeclareOldFontCommand{#1}{%
2196         \bxjs@oldfontcmd{#1}#2%
2197     }{%

```

```

2198     \bxjs@oldfontcmd{#1}#3%
2199   }%
2200 }
2201 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2202   \ClassInfo\bxjs@clsname
2203   {Old font command '\string#1' is used!!\MessageBreak
2204     The first occurrence is}%
2205 }

```

`\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。

```

2206 \newcommand*{\allowoldfontcommands}{%
2207   \@jsc@warnoldfontcmdexceptiontrue}
2208 \newcommand*{\disallowoldfontcommands}{%
2209   \@jsc@warnoldfontcmdexceptionfalse}

```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2210 \let\bxjs@oldfontcmd@list\@empty
2211 \def\bxjs@oldfontcmd#1{%
2212   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2213 \def\bxjs@oldfontcmd@a#1#2{%
2214   \if@jsc@warnoldfontcmdexception\else
2215     \global\@jsc@warnoldfontcmdfalse
2216     \ifx#1\relax
2217       \global\let#1=t%
2218       \jsc@warnoldfontcmd{#2}%
2219     \fi
2220   \fi}
2221 \def\bxjs@warnoldfontcmd@final{%
2222 % \par
2223   \global\let\bxjs@warnoldfontcmd@final\@empty
2224   \let\@tempa\@empty
2225   \def\do##1{%
2226     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2227       \edef\@tempa{\@tempa \space\string##1}\fi}
2228   \bxjs@oldfontcmd@list
2229   \ifx\@tempa\@empty\else
2230     \ClassWarningNoLine\bxjs@clsname
2231       {Some old font commands were used in text\MessageBreak
2232         (see the log file for detail)}%
2233     \ClassInfo\bxjs@clsname
2234       {Some old font commands were used in text:\MessageBreak
2235         \space\@tempa\MessageBreak
2236         You should note, that since 1994 LaTeX2e provides a\MessageBreak
2237         new font selection scheme called NFSS2 with several\MessageBreak
2238         new, combinable font commands. The

```



```

2239     class provides\MessageBreak
2240     the old font commands
2241     only for compatibility%
2242     \@gobble}%
2243 \fi}

```

単純に `\AtEndDocument` のフックの中で `\bxjs@warnoldfontcmd@final` を実行した場合、最終ページのヘッダ・フッタの中にある二文字フォント命令はそれより後に実行されるため捕捉できない。これに対処するため、`\end{document}` 中に実行される `\clearpage` の処理の直後に `\bxjs...final` が呼ばれるようにする。

```

2244 \def\bxjs@warnoldfontcmd@kick@final{%
2245   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2246 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2247 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2248 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2249 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb 2250 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2251 \jsc@DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

```

`\bf` ボールドシリーズにします。通常のミディアムシリーズに戻すコマンドは `\mdseries` です。

```

2252 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

```

2253 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2254 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2255 \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 2256 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2257 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}

```

10 相互参照

10.1 目次の類

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

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

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

`\numberline{番号}{見出し}`

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

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

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

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

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

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

という書式です。

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

インデント 左側の字下げ量です。

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

`\@pnumwidth` ページ番号の入る箱の幅です。

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

`\@dotsep` 点の間隔です (単位 mu)。

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

```
2258 \newcommand\@pnumwidth{1.55em}
2259 \newcommand\@tocrmarg{2.55em}
2260 \newcommand\@dotsep{4.5}
2261 %<!book&!report>\setcounter{tocdepth}{2}
2262 %<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)

```
2263 \newdimen\jsc@tocl@width
2264 \newcommand{\tableofcontents}{%
2265 %<*book|report>
2266 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2267 \settowidth\@tempdima{\headfont\appendixname}%
2268 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2269 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
```

```

2270 \if@twocolumn
2271 \@restonecoltrue\onecolumn
2272 \else
2273 \@restonecolfalse
2274 \fi
2275 \chapter*{\contentsname}%
2276 \mkboth{\contentsname}{}%
2277 %</book|report>
2278 %<*!book&!report>
2279 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2280 \settowidth\@tempdima{\headfont\appendixname}%
2281 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2282 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2283 \section*{\contentsname}%
2284 \mkboth{\contentsname}{\contentsname}%
2285 %</!book&!report>
2286 \@starttoc{toc}%
2287 %<book|report> \if@restonecol\twocolumn\fi
2288 }

```

\l@part 部の目次です。

```

2289 \newcommand*{\l@part}[2]{%
2290 \ifnum \c@tocdepth >-2\relax
2291 %<!book&!report> \addpenalty\@secpenalty
2292 %<book|report> \addpenalty{-\@highpenalty}%
2293 \addvspace{2.25em \@plus\p@}%
2294 \begingroup
2295 \parindent \z@
2296 % \@pnumwidth should be \@tocrmarg
2297 % \rightskip \@pnumwidth
2298 \rightskip \@tocrmarg
2299 \parfillskip -\rightskip
2300 {\leavevmode
2301 \large \headfont
2302 \setlength\@lnumwidth{4\jsZw}%
2303 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2304 \nobreak
2305 %<book|report> \global\@nobreaktrue
2306 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2307 \endgroup
2308 \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)

```

2309 %<*book|report>
2310 \newcommand*{\l@chapter}[2]{%
2311 \ifnum \c@tocdepth >\m@ne
2312 \addpenalty{-\@highpenalty}%

```

```

2313 \addvspace{1.0em \@plus\p@?}
2314 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2315 \begingroup
2316 \parindent\z@
2317 % \rightskip\@pnumwidth
2318 \rightskip\@tocrmarg
2319 \parfillskip-\rightskip
2320 \leavevmode\headfont
2321 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2322 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2323 \advance\leftskip\@lnumwidth \hskip-\leftskip
2324 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2325 \penalty\@highpenalty
2326 \endgroup
2327 \fi}
2328 %</book|report>

```

\l@section 節の目次です。

```

2329 %<!*book&!report>
2330 \newcommand*{\l@section}[2]{%
2331 \ifnum \c@tocdepth >\z@
2332 \addpenalty{\@secpenalty}%
2333 \addvspace{1.0em \@plus\p@?}%
2334 \begingroup
2335 \parindent\z@
2336 % \rightskip\@pnumwidth
2337 \rightskip\@tocrmarg
2338 \parfillskip-\rightskip
2339 \leavevmode\headfont
2340 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2341 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2342 \advance\leftskip\@lnumwidth \hskip-\leftskip
2343 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2344 \endgroup
2345 \fi}
2346 %</!book&!report>

```

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

```

2347 %<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 2348 %<!*book&!report>
2349 % \newcommand*{\l@subsection}{\@dottedtocline{2}{1.5em}{2.3em}}
2350 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2351 % \newcommand*{\l@paragraph}{\@dottedtocline{4}{7.0em}{4.1em}}

```

```

2352 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2353 %
2354 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2355 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2356 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2357 %
2358 \newcommand*{\l@section}{%
2359     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2360     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2361 \newcommand*{\l@subsubsection}{%
2362     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2363     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2364 \newcommand*{\l@paragraph}{%
2365     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2366     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2367 \newcommand*{\l@subparagraph}{%
2368     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2369     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2370 %<!/book&!report>
2371 %<*book|report>
2372 % \newcommand*{\l@subsection} {\@dottedtocline{2}{3.8em}{3.2em}}
2373 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2374 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2375 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2376 \newcommand*{\l@section}{%
2377     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2378     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2379 \newcommand*{\l@subsubsection}{%
2380     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2381     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2382 \newcommand*{\l@subsubsection}{%
2383     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2384     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2385 \newcommand*{\l@paragraph}{%
2386     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2387     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2388 \newcommand*{\l@subparagraph}{%
2389     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2390     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2391 %</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。

```

2392 \newdimen\@lnumwidth
2393 \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。

```

2394 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2395   \vskip \z@ \@plus.2\p@?
2396   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2397    \parindent #2\relax\@afterindenttrue
2398    \interlinepenalty\@M
2399    \leavevmode
2400    \@lnumwidth #3\relax
2401    \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2402    {#4}\nobreak
2403    \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2404     mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2405     \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2406 \newcommand{\listoffigures}{%
2407 %<*book|report>
2408 \if@twocolumn\@restonecoltrue\onecolumn
2409 \else\@restonecolfalse\fi
2410 \chapter*{\listfigurename}%
2411 \@mkboth{\listfigurename}{}%
2412 %</book|report>
2413 %<*&!book&!report>
2414 \section*{\listfigurename}%
2415 \@mkboth{\listfigurename}{\listfigurename}%
2416 %</&!book&!report>
2417 \@starttoc{lof}%
2418 %<book|report> \if@restonecol\twocolumn\fi
2419 }

```

`\l@figure` 図目次の項目を出力します。

```

2420 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2421 \newcommand{\listoftables}{%
2422 %<*book|report>
2423 \if@twocolumn\@restonecoltrue\onecolumn
2424 \else\@restonecolfalse\fi
2425 \chapter*{\listtablename}%
2426 \@mkboth{\listtablename}{}%
2427 %</book|report>
2428 %<*&!book&!report>
2429 \section*{\listtablename}%
2430 \@mkboth{\listtablename}{\listtablename}%
2431 %</&!book&!report>

```

```

2432 \@starttoc{lot}%
2433 %<book|report> \if@restonecol\twocolumn\fi
2434 }

```

`\l@table` 表目次は図目次と同じです。

```

2435 \let\l@table\l@figure

```

10.2 参考文献

`\bibindent` オープンスタイルの参考文献で使うインデント幅です。元は 1.5em でした。

```

2436 \newdimen\bibindent
2437 \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。

```

2438 \newenvironment{thebibliography}[1]{%
2439   \@jsc@warnoldfontcmdexceptiontrue
2440   \global\let\presectionname\relax
2441   \global\let\postsectionname\relax
2442   %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2443   %<*kiyou>
2444   \vspace{1.5\baselineskip}
2445   \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2446   \vspace{0.5\baselineskip}
2447   %</kiyou>
2448   %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2449   %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2450   \list{\@biblabel{\@arabic\c@enumiv}}%
2451     {\settowidth\labelwidth{\@biblabel{#1}}%
2452      \leftmargin\labelwidth
2453      \advance\leftmargin\labelsep
2454      \@openbib@code
2455      \usecounter{enumiv}%
2456      \let\p@enumiv\empty
2457      \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2458   %<kiyou> \small
2459   \sloppy
2460   \clubpenalty4000
2461   \@clubpenalty\clubpenalty
2462   \widowpenalty4000%
2463   \sfcode`\.\@m}
2464   {\def\@noitemerr
2465     {\@latex@warning{Empty `thebibliography' environment}}}%
2466   \endlist}

```

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

```
2467 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}
```

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

```
2468 \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。

```
2469 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}
```

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

```
2470 % \def\@citex[#1]#2{\leavevmode
2471 %   \let\@citea\@empty
2472 %   \@cite{\@for\@citeb:=#2\do
2473 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2474 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2475 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2476 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2477 %       \G@refundefinedtrue
2478 %       \@latex@warning
2479 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2480 %     {\@cite@ofmt{\csname b@\@citeb\endcsname}}}{#1}}
2481 % \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。

```
2482 % \DeclareRobustCommand\cite{\unskip
2483 %   \@ifnextchar [{\@tempwattrue\@citex}{\@tempwafalse\@citex[]}]
2484 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2485 %   , \jsInhibitGlue\ #2\fi}}}$}
```

10.3 索引

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

```
2486 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2487   \if@twocolumn
2488     \onecolumn\@restonecolfalse
2489   \else
2490     \clearpage\@restonecoltrue
2491   \fi
```



```

2492 \columnseprule.4pt \columnsep 2\jsZw
2493 \ifx\multicols\@undefined
2494 %<book|report> \twocolumn[\@makeschapterhead{\indexname}%
2495 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2496 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2497 %<!book&!report> \twocolumn[\section*{\indexname}]]%
2498 \else
2499 \ifdim\textwidth<\fullwidth
2500 \setlength{\evensidemargin}{\oddsidemargin}
2501 \setlength{\textwidth}{\fullwidth}
2502 \setlength{\linewidth}{\fullwidth}
2503 %<book|report> \begin{multicols}{3}[\chapter*{\indexname}%
2504 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2505 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2506 %<!book&!report> \begin{multicols}{3}[\section*{\indexname}]]%
2507 \else
2508 %<book|report> \begin{multicols}{2}[\chapter*{\indexname}%
2509 %<book|report> \addcontentsline{toc}{chapter}{\indexname}]]%
2510 %<!book&!report> \def\presectionname{}\def\postsectionname{}%
2511 %<!book&!report> \begin{multicols}{2}[\section*{\indexname}]]%
2512 \fi
2513 \fi
2514 %<book|report> \mkboth{\indexname}{}%
2515 %<!book&!report> \mkboth{\indexname}{\indexname}%
2516 \plainifnotempty % \thispagestyle{plain}
2517 \parindent\z@
2518 \parskip\z@ \@plus .3\p@?\relax
2519 \let\item\@idxitem
2520 \raggedright
2521 \footnotesize\narrowbaselines
2522 }{
2523 \ifx\multicols\@undefined
2524 \if@restonecol\onecolumn\fi
2525 \else
2526 \end{multicols}
2527 \fi
2528 \clearpage
2529 }

```

`\@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 2530 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2531 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2532 \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。

```

2533 \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。

```
2534 \newcommand\seename{\if@english see\else →\fi}  
2535 \newcommand\alsoname{\if@english see also\else →\fi}
```

10.4 脚注

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

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

```
2536 \ifx\pltx@foot@penalty\@undefined  
2537 \let\footnotes@ve=\footnote  
2538 \def\footnote{\jsInhibitGlue\footnotes@ve}  
2539 \let\footnotemarks@ve=\footnotemark  
2540 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}  
2541 \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)。

```
2542 \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。

```
2543 % \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。

```
2544 \renewcommand{\footnoterule}{%
2545   \kern-2.6\p@? \kern-.4\p@
2546   \hrule width .4\columnwidth
2547   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2548 %<book|report>\@addtoreset{footnote}{chapter}
```

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

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

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

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

```
2549 \long\def\@footnotetext{%
2550   \insert\footins\bgroup
2551   \normalfont\footnotesize
2552   \interlinepenalty\interfootnotelinepenalty
2553   \splittopskip\footnotesep
2554   \splitmaxdepth \dp\strutbox \floatingpenalty \MM
2555   \hsize\columnwidth \@parboxrestore
2556   \protected@edef\@currentlabel{%
2557     \csname p@footnote\endcsname\@thefnmark
2558   }%
2559   \color@begingroup
2560   \@makefnmark{%
2561     \rule{\z@\footnotesep}{\ignorespaces}%
2562     \futurelet\next\fo@t}
2563   \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@@
2564                                     \else \let\next\fo@t\fi \next}
2565   \def\fo@@{\bgroup\aftergroup\@foot\let\next}
2566   \def\fo@t#1{#1\@foot}
2567   \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2568     \ifx\pltx@foot@penalty\undefined\else
2569       \ifhmode\null\fi
2570       \ifnum\pltx@foot@penalty=\z@\else
2571         \penalty\pltx@foot@penalty
2572         \pltx@foot@penalty\z@
2573       \fi
2574     \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。

```

2575 \newcommand\@makefntext[1]{%
2576   \advance\leftskip 3\jsZw
2577   \parindent 1\jsZw
2578   \noindent
2579   \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。

```

2580 % \def\@xfootnotenext[#1]{%
2581 %   \begingroup
2582 %     \ifnum#1>\z@
2583 %       \csname c@\@mpfn\endcsname #1\relax
2584 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2585 %     \else
2586 %       \unrestored@protected@xdef\@thefnmark{}%
2587 %     \fi
2588 %   \endgroup
2589 %   \@footnotetext}

```

ここまでのコードは JS クラスを踏襲する。

11 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

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

```

2590 \let\jsInhibitGlueAtParTop\@empty

```

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

```

2591 \def\everyparhook{\jsInhibitGlueAtParTop}
2592 \AtBeginDocument{\everypar{\everyparhook}}

```

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

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

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

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

```

2593 \def\@inhibitglue{%
2594   \futurelet\@let@token\@inhibitglue}
2595 \begingroup
2596 \let\GDEF=\gdef
2597 \let\CATCODE=\catcode
2598 \let\ENDGROUP=\endgroup
2599 \CATCODE`k=12
2600 \CATCODE`a=12
2601 \CATCODE`n=12
2602 \CATCODE`j=12
2603 \CATCODE`i=12
2604 \CATCODE`c=12
2605 \CATCODE`h=12
2606 \CATCODE`r=12
2607 \CATCODE`t=12
2608 \CATCODE`e=12
2609 \GDEF\KANJI@CHARACTER{kanji character }
2610 \ENDGROUP
2611 \def\@inhibitglue{%
2612   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@token
2613   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc@
2614   \def\jsc@ig@temp{#1}%
2615   \ifx\jsc@ig@temp\@empty
2616     \ifnum\the\inhibitxspcode`#2=2\relax
2617       \inhibitglue
2618     \fi
2619   \fi}

```

これだけではいけないようです。あちこちに `\everypar` を初期化するコマンドが隠されていました。

まず、環境の直後の段落です。

[2016-11-19] `ltlists.dtx` 2015/05/10 v1.0t の変更に従って `\clubpenalty` のリセットを追加しました。

```

2620 \def\@doendpe{%
2621   \@endpetrue
2622   \def\par{%
2623     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2624   \everypar{{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

`\item` 命令の直後です。

```

2625 \def\@item[#1]{%
2626   \if@noparitem
2627     \@donoparitem
2628   \else
2629     \if@inlabel
2630       \indent \par
2631     \fi
2632     \ifhmode
2633       \unskip\unskip \par
2634     \fi
2635     \if@newlist
2636       \if@nobreak
2637         \@nbitem
2638       \else
2639         \addpenalty\@beginparpenalty
2640         \addvspace\@topsep
2641         \addvspace{-\parskip}%
2642       \fi
2643     \else
2644       \addpenalty\@itempenalty
2645       \addvspace\itemsep
2646     \fi
2647     \global\@inlabeltrue
2648   \fi
2649   \everypar{%
2650     \@minipagefalse
2651     \global\@newlistfalse
2652     \if@inlabel
2653       \global\@inlabelfalse
2654       {\setbox\z@\lastbox
2655        \ifvoid\z@
2656          \kern-\itemindent
2657        \fi}%
2658     \box\@labels
2659     \penalty\z@
2660   \fi
2661   \if@nobreak
2662     \@nobreakfalse
2663     \clubpenalty \@M
2664   \else
2665     \clubpenalty \@clubpenalty
2666     \everypar{\everyparhook}%
2667   \fi
2668   \bxjs@ltj@inhibitglue
2669   \everyparhook}%
2670 \if@noitemarg
2671   \@noitemargfalse
2672   \if@nmbrlist
2673     \refstepcounter\@listctr

```

```

2674 \fi
2675 \fi
2676 \sbox\@tempboxa{\makelabel{#1}}%
2677 \global\setbox\@labels\hbox{%
2678 \unhbox\@labels
2679 \hskip \itemindent
2680 \hskip -\labelwidth
2681 \hskip -\labelsep
2682 \ifdim \wd\@tempboxa >\labelwidth
2683 \box\@tempboxa
2684 \else
2685 \hbox to\labelwidth {\unhbox\@tempboxa}%
2686 \fi
2687 \hskip \labelsep}%
2688 \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。

```

2689 \def\@afterheading{%
2690 \@nobraektrue
2691 \everypar{%
2692 \if@nobraek
2693 \@nobraekfalse
2694 \clubpenalty \@M
2695 \if@afterindent \else
2696 {\setbox\z@\lastbox}%
2697 \fi
2698 \else
2699 \clubpenalty \@clubpenalty
2700 \everypar{\everyparhook}%
2701 \fi\everyparhook}}

```

`\@gnewline` についてはちょっと複雑な心境です。もともとの $\text{p}\text{A}\text{T}\text{E}\text{X} 2_{\epsilon}$ 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。

```

2702 \def\@gnewline #1{%
2703 \ifvmode
2704 \@nolnerr
2705 \else
2706 \unskip \reserved@a {\reserved@f#1}\nobraek \hfil \break \null
2707 \jsInhibitGlue \ignorespaces
2708 \fi}

```

12 いろいろなロゴ

L^AT_EX 関連のロゴを作り直します。

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

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

※`\小`、`\上小` の制御綴は定義しない。

```
2709 \if@jslogo
2710   \IfFileExists{jslogo.sty}{%
2711     \RequirePackage{jslogo}%
2712   }{%
2713     \ClassWarningNoLine{bxjs@clsname}
2714       {The package 'jslogo' is not installed.\MessageBreak
2715         It is included in the recent release of\MessageBreak
2716         the 'jsclasses' bundle}
2717   }
2718 \fi
```

13 amsmath との衝突の回避

最近の `\LaTeX` では該当の問題は対処されているので削除。

14 初期設定

■いろいろな語

```
\prepartname
\postpartname 2719 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2720 \newcommand{\postpartname}{\if@english\else 部\fi}
2721 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2722 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2723 \newcommand{\presectionname}{}% 第
\postsectionname 2724 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 2725 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2726 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
```



```

2727 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname

\bibname 2728 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2729 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
2730 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 2731 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
2732 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname

\abstractname 2733 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
2734 \newcommand{\appendixname}{\if@english \else 付録\fi}
2735 %<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

環境変数 SOURCE_DATE_EPOCH_TEX_PRIMITIVES が設定されている場合は“今日”が過去の日付になる可能性があるが、その場合、和暦表記は平成 2 年（1990 年）以降でのみサポートする。

※ “新元号” への対応？

```

\today

2736 \@tempwafalse
2737 \if p\jsEngine \@tempwattrue \fi
2738 \if n\jsEngine \@tempwattrue \fi
2739 \if@tempswa \expandafter\@firstoftwo
2740 \else \expandafter\@secondoftwo
2741 \fi
2742 {%
2743 % 欧文 8bitTeX の場合
2744 \newif\ifjsSeireki \jsSeirekitrue
2745 \def\bxjs@decl@Seireki@cmds{%
2746 \def\西暦{\jsSeirekitrue}%
2747 \def\和暦{\jsSeirekifalse}}
2748 \def\Seireki{\jsSeirekitrue}
2749 \def\Wareki{\jsSeirekifalse}
2750 \def\bxjs@if@use@seireki{%
2751 \ifjsSeireki \expandafter\@firstoftwo
2752 \else \expandafter\@secondoftwo \fi}
2753 }{%
2754 \newif\if 西暦 \西暦 true
2755 \def\bxjs@decl@Seireki@cmds{%
2756 \def\西暦{\西暦 true}%

```

```

2757 \def\和暦{\西暦 false}}
2758 \def\Seireki{\西暦 true}
2759 \def\Wareki{\西暦 false}
2760 \def\bxjs@if@use@seireki{%
2761 \if 西暦 \expandafter\@firstoftwo
2762 \else \expandafter\@secondoftwo \fi}
2763 }
2764 \bxjs@decl@Seireki@cmds
2765 % \bxjs@unxp
2766 \let\bxjs@unxp\@firstofone
2767 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2768 % \bxjs@iai
2769 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2770 \def\bxjs@iai{\noexpand~}
2771 \else \def\bxjs@iai{}
2772 \fi
2773 % \heisei
2774 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2775 % \today
2776 \edef\bxjs@today{%
2777 \if@english
2778 \ifcase\month\or
2779 January\or February\or March\or April\or May\or June\or
2780 July\or August\or September\or October\or November\or December\fi
2781 \space\number\day, \number\year
2782 \else
2783 \ifnum\heisei>\@ne
2784 \expandafter\noexpand\expandafter\bxjs@if@use@seireki
2785 \else \expandafter\@firstoftwo
2786 \fi {%
2787 \number\year\bxjs@iai\bxjs@unxp{年}%
2788 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2789 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2790 }{%
2791 \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2792 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2793 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2794 }%
2795 \fi}
2796 \let\today\bxjs@today

```

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

```

2797 \AtBeginDocument{%
2798 \ifx\bb1@jpn@Seirekitrue\@undefined\else
2799 \bxjs@decl@Seireki@cmds
2800 \g@addto@macro\datejapanese{%

```

```

2801         \let\today\bxjs@today}%
2802     \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）

```

2803 \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。

```

2804 %<slide>\pagestyle{empty}%
2805 %<article|report>\pagestyle{plain}%
2806 %<book>\pagestyle{headings}%
2807 \pagenumbering{arabic}
2808 \if@twocolumn
2809     \twocolumn
2810     \sloppy
2811     \flushbottom
2812 \else
2813     \onecolumn
2814     \raggedbottom
2815 \fi
2816 %<*>slide>
2817 \renewcommand\familydefault{\sfdefault}
2818 \raggedright
2819 %</>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。

```

2820 \catcode\?=12
2821 \ifx\bxjs@jadriver\relax\else
2822 \input{bxjsja-\bxjs@jadriver.def}
2823 \fi

```

最後に日本語文字のカテゴリコードを元に戻す。

```

2824 \bxjs@restore@jltrcc
2825 %</>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。

```
2826 %<*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

```
2827 %<*minimal>
```

```
2828 %% 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.
```

```
2829 \def\DeclareJaTextFontCommand#1#2{%
```

```
2830   \DeclareRobustCommand#1[1]{%
```

```
2831     \relax
```

```
2832     \ifmmode \expandafter\nfss@text \fi
```

```
2833     {#2##1}}%
```

```
2834 }
```

```
\DeclareJaMathFontCommand 和文数式フォントが無効な場合に、それをエミュレートするもの。
```

```
2835 \def\DeclareJaMathFontCommand#1#2{%
```

```
2836   \DeclareRobustCommand#1[1]{%
```

```
2837     \relax
```

```
2838     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
2839     \nfss@text{\fontfamily\familydefault
```

```
2840       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
2841       #2##1}}%
```

```

2842 }%
2843 }

```

`\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。

```

2844 \long\def\bxjs@@CSsfdefault{\sfdefault}%
2845 \@onlypreamble\bxjs@if@sf@default
2846 \def\bxjs@if@sf@default#1{%
2847   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
2848   \AtBeginDocument{%
2849     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
2850 }

```

`\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。

```

2851 \def\jsLetHeadChar#1#2{%
2852   \begingroup
2853     \escapechar=`\ %
2854     \let\bxjs@tmpa={% brace-match-hack
2855       \bxjs@let@hchar@exp#2}%
2856   \endgroup
2857   \let#1\bxjs@g@tmpa}
2858 \def\bxjs@let@hchar@exp{%
2859   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2860 \def\bxjs@let@hchar@exp@a{%
2861   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
2862     \bxjs@let@hchar@out\let\relax
2863   }\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi{% 空白
2864     \bxjs@let@hchar@out\let\space%
2865   }\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
2866     \bxjs@let@hchar@out\let\@backslashchar
2867   }\bxjs@let@hchar@exp@b}}}%
2868 \def\bxjs@let@hchar@exp@b#1{%
2869   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2870 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2871   \%message{<#1#2>}%
2872   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴
2873     \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2874       \bxjs@let@hchar@out\let\relax
2875     }{%else
2876       \expandafter\bxjs@let@hchar@exp
2877     }%
2878   }{%else
2879     \bxjs@let@hchar@chr#1%

```

```

2880  }}
2881 \def\bxjs@let@hchar@chr#1{%
2882  \bxjs@let@hchar@out\def{{#1}}}
2883 \def\bxjs@let@hchar@out#1#2{%
2884  \global#1\bxjs@g@tmpa#2\relax
2885  \toks@{\bgroup}% skip to right brace

UTF-8 のバイト列を扱うコード。
2886 \chardef\bxjs@let@hchar@csta=128
2887 \chardef\bxjs@let@hchar@cstb=192
2888 \chardef\bxjs@let@hchar@cstc=224
2889 \chardef\bxjs@let@hchar@cstd=240
2890 \chardef\bxjs@let@hchar@cste=248
2891 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2892 \def\bxjs@let@hchar@chr@ue#1{%
2893  \@tempcnta=#1\relax
2894  %\message{\the\@tempcnta}%
2895  \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2896    \bxjs@let@hchar@chr@ue@a#1%
2897  }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2898    \bxjs@let@hchar@out\let\relax
2899  }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2900    \bxjs@let@hchar@chr@ue@b
2901  }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2902    \bxjs@let@hchar@chr@ue@c
2903  }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2904    \bxjs@let@hchar@chr@ue@d
2905  }\{%else
2906    \bxjs@let@hchar@out\let\relax
2907  }}}}
2908 \def\bxjs@let@hchar@chr@ue@a#1{%
2909  \bxjs@let@hchar@out\def{{#1}}}
2910 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2911  \bxjs@let@hchar@out\def{{#1#2}}}
2912 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2913  \bxjs@let@hchar@out\def{{#1#2#3}}}
2914 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2915  \bxjs@let@hchar@out\def{{#1#2#3#4}}}

```

B.2 (u)pTeX 用の設定

```
2916 \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。

```

2917 \def\bxjs@let@hchar@chr@pp#1{%
2918  \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}

```

```

2919 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2920 %\message{(#1)}%
2921 \bxjs@cond\if#1t\fi{%
2922 \bxjs@let@hchar@chr@ue#3%
2923 }{%else
2924 \bxjs@let@hchar@out\def{#3}%
2925 }}
2926 \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。

```

2927 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
2928 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
2929 \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。

```

2930 \@onlypreamble\bxjs@declarefontshape
2931 \ifjsWithupTeX
2932 \def\bxjs@declarefontshape{%
2933 \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
2934 \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
2935 \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
2936 \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
2937 }
2938 \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。

```

2939 \else
2940 \def\bxjs@declarefontshape{%
2941 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
2942 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
2943 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
2944 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
2945 }
2946 \def\bxjs@sizereference{jis}
2947 \fi

```

既に使用されている標準和文フォント定義がもしあれば取り消す。

```

2948 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%

```



```

2949 \def\bxjs@y{#5}}
2950 \ifjsWithpTeXng \def\bxjs@y{10}%
2951 \else
2952 \expandafter\expandafter\expandafter\bxjs@tmpa
2953 \expandafter\string\the\jfont\relax
2954 \fi
2955 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
2956             \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
2957 {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
2958  \expandafter\let\csname\bxjs@x/\bxjs@y\endcsname=\@undefined}

```

■和文フォントスケールの補正 実、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。

```

2959 \begingroup
2960 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2961 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2962 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2963 % 幅が丁度 10pt なら補正は不要
2964 \ifdim\wd\z@=10pt
2965   \global\let\bxjs@scale\jsScale
2966 \else
2967 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
2968 \edef\bxjs@tmpa{\strip@pt\wd\z@}
2969 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
2970 \bxjs@invscale\@tempdima\bxjs@tmpa
2971 \xdef\bxjs@scale{\strip@pt\@tempdima}
2972 \fi
2973 \endgroup
2974 %\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。

```

2975 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

2976 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
2977 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
2978 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
2979 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
2980 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
2981 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
2982 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
2983 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
2984 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}

```

```

2985 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
2986 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
2987 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
2988 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
2989 \DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{ }

```

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

```

2990 \DeclareRobustCommand\rmfamily
2991 { \not@math@alphabet\rmfamily\mathrm
2992   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
2993 \DeclareRobustCommand\sffamily
2994 { \not@math@alphabet\sffamily\mathsf
2995   \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
2996 \DeclareRobustCommand\ttfamily
2997 { \not@math@alphabet\ttfamily\mathtt
2998   \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
2999 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3000 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3001 \bxjs@if@sf@default{%
3002   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3003 \selectfont

```

■パラメタの設定

```

3004 \prebreakpenalty\jis"2147=10000
3005 \postbreakpenalty\jis"2148=10000
3006 \prebreakpenalty\jis"2149=10000
3007 \inhibitxspcode`!=1
3008 \inhibitxspcode`〒=2
3009 \xspcode`+=3
3010 \xspcode`\%=3

```

"80～"FF の範囲の \spcode を 3 に変更。

```

3011 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3012   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3013 \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。

```

3014 \begingroup
3015 \catcode`\!=0

```

\bxjs@ptex@dir 現在の組方向： t=縦、y=横、?=その他。

```

3016 \gdef\bxjs@ptex@dir{%
3017   !iftdir t%
3018   !else!ifydir y%
3019   !else ?%
3020   !fi!fi}

```

新版の p_TE_X で脚注番号の周囲の空きが過大になる現象への対処。

※現在の p_LA_TE_X カーネルでは対処が既に行われている。ここでは、\@makefnmark の定義が古いものであった場合に、新しいものに置き換える。

```

3021 % 古い \@makefnmark の定義
3022 \long\def\bxjs@tmpa{\hbox{%
3023   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3024   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3025 \ifx\@makefnmark\bxjs@tmpa
3026 \long\gdef\@makefnmark{%
3027   !ifydir \hbox{ }\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{ }%
3028   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3029 \fi
3030 \endgroup

```

B.3 pdf_TE_X 用の処理

```

3031 \else\ifx p\jsEngine
3032 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3033 \@onlypreamble\bxjs@cjk@loaded
3034 \def\bxjs@cjk@loaded{%
3035   \def\@footnotemark{%
3036     \leavevmode
3037     \ifhmode
3038       \edef\@x@sf{\the\spacefactor}%
3039       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3040         \unkern\unkern
3041         \ifdim\lastskip>\z@ \unskip \fi
3042       \fi\fi
3043       \nobreak
3044     \fi
3045     \@makefnmark
3046     \ifhmode \spacefactor\@x@sf \fi
3047     \relax}%
3048   \let\bxjs@cjk@loaded\relax
3049 }
3050 \AtBeginDocument{%
3051   \@ifpackageloaded{CJK}{%
3052     \bxjs@cjk@loaded
3053   }{}%
3054 }

```

B.4 XeTeX 用の処理

3055 \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。

```
3056 \def\bxjs@let@hchar@chr#1{%
3057   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3058   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3059     \bxjs@let@hchar@chr@xe
3060   }\bxjs@let@hchar@out\def{{#1}}}}
3061 \def\bxjs@let@hchar@chr@xe#1{%
3062   \lccode`0=`#1\relax
3063   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3064 \ifx\XeTeXgenerateactualtext\@undefined\else
3065   \def\bxjs@do@precisetext{%
3066     \XeTeXgenerateactualtext=\@ne}
3067 \fi
```

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

```
3068 \@onlypreamble\bxjs@do@simplejasetup
3069 \def\bxjs@do@simplejasetup{%
3070   \ifnum\XeTeXinterchartokenstate>\z@
3071   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3072     \jsSimpleJaSetup
3073     \ClassInfo\bxjs@clsname
3074     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3075   \fi\fi}
```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```
3076 \newcommand*{\jsSimpleJaSetup}{%
3077   \XeTeXlinebreaklocale "ja"\relax
3078   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3079   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

3080 \fi\fi\fi

simplejasetup オプションの処理。

```
3081 \ifx\bxjs@do@simplejasetup\@undefined\else
3082   \AtBeginDocument{%
3083     \ifbxjs@simplejasetup
3084       \bxjs@do@simplejasetup
3085     \fi}
3086 \fi
```

precisetext オプションの処理。

```

3087 \ifbxjs@precisetext
3088   \ifx\bxjs@do@precisetext\@undefined
3089     \ClassWarning\bxjs@clsname
3090     {The current engine does not support the\MessageBreak
3091     'precisetext' option\@gobble}
3092   \else
3093     \bxjs@do@precisetext
3094   \fi
3095 \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。

```

3096 \ifbxjs@fancyhdr

```

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

```

3097 \@onlypreamble\bxjs@adjust@fancyhdr
3098 \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。

```

3099 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3100 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3101 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3102 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3103 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3104 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3105 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3106 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3107 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3108 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3109 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3110 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3111 \def\bxjs@tmpa{\rm\thepage\strut}%
3112 \def\bxjs@tmpb{\thepage\strut}%
3113 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3114 \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。

```

3115 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3116   \setlength{\@tempdima}{\fullwidth-\textwidth}%
3117   \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3118   }\bxjs@tmpa

```

```

3119 \fi\fi
3120 \PackageInfo{bxjs@clsname
3121   {Patch to fancyhdr is applied\@gobble}}

```

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

```

3122 \def\bxjs@pagestyle@hook{%
3123   \@ifpackageloaded{fancyhdr}{%
3124     \bxjs@adjust@fancyhdr
3125     \global\let\bxjs@adjust@fancyhdr\relax
3126   }{}}

```

`\pagestyle` にフックを入れ込む。

```

3127 \let\bxjs@org@pagestyle\pagestyle
3128 \def\pagestyle{%
3129   \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。

```

3130 \AtBeginDocument{%
3131   \bxjs@pagestyle@hook
3132   \global\let\bxjs@pagestyle@hook\relax}

```

終わり。

```

3133 \fi

```

以上で終わり。

```

3134 %</minimal>

```

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

コマンドで使う“ファミリー”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ーを表す。

C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
3135 %<*standard>
3136 %% このファイルは日本語文字を含みます
3137 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3138 \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。

```
3139 \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。

```
\jH3140 \@tempdima=0.25mm
3141 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3142 \let\jH\jQ
```

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

```
\trueH3143 \ifjsc@mag
3144 \@tempdimb=\jsBaseFontSize\relax
3145 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3146 \@tempdima=2.5mm
3147 \bxjs@invscale\@tempdima\bxjs@tmpa
3148 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3149 \@tempdima=10pt
3150 \bxjs@invscale\@tempdima\bxjs@tmpa
3151 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3152 \else \let\trueQ\jQ \let\bxjs@truept\p@
3153 \fi
3154 \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。

```
3155 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3156 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3157 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3158 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3159 \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。

```
3160 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3161 \newcommand*\setkanjiskip[1]{%
```

```
3162   \edef\bxjs@kanjiskip{#1}%
```

```
3163   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3164 \newcommand*\getkanjiskip{%
```

```
3165   \bxjs@kanjiskip}
```

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

```
3166 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

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

```
\bxjs@disable@kanjiskip 3167 \bxjs@robust@def\bxjs@enable@kanjiskip{%
```

```
3168   \bxjs@kanjiskip@enabledtrue
```

```
3169   \bxjs@reset@kanjiskip}
```

```
3170 \bxjs@robust@def\bxjs@disable@kanjiskip{%
```

```
3171   \bxjs@kanjiskip@enabledfalse
```

```
3172   \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。

```
3173 \bxjs@robust@def\bxjs@reset@kanjiskip{%
```

```
3174   \ifbxjs@kanjiskip@enabled
```

```
3175     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
```

```
3176   \else \@tempskipa\z@
```

```
3177   \fi
```

```
3178   \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 3179 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3180 \newcommand*\setxkanjiskip[1]{%
```

```
3181   \edef\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3182   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3183 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3184   \bxjs@xkanjiskip}
```

```
3185 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3186 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3187   \bxjs@xkanjiskip@enabledtrue
```

```
3188   \bxjs@reset@xkanjiskip}
```

```
3189 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3190   \bxjs@xkanjiskip@enabledfalse
```

```
3191   \bxjs@reset@xkanjiskip}
```

```
3192 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```



```

3193 \ifbxjs@xkanjiskip@enabled
3194 \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3195 \else \@tempskipa\z@
3196 \fi
3197 \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。

```

3198 \g@addto@macro\jsResetDimen{%
3199 \bxjs@reset@kanjiskip
3200 \bxjs@reset@xkanjiskip}
3201 \let\bxjs@apply@kanjiskip\relax
3202 \let\bxjs@apply@xkanjiskip\relax

```

■和文フォント指定の扱い

\bxjs@adjust@jafont \jsJaFont に入っている和文フォント設定の値を“調整”して、その結果を \bxjs@tmpa に返す。#1 が f の場合は“非埋込 (noEmbed)”の設定が禁止される。この禁止の場合も含め、何か異常がある場合は \bxjs@tmpa は空になる。

```

3203 \@onlypreamble\bxjs@adjust@jafont
3204 \def\bxjs@adjust@jafont#1{%
3205 \ifx\jsJaFont\bxjs@@auto
3206 \bxjs@get@kanjiEmbed
3207 \ifx\bxjs@kanjiEmbed\relax
3208 \let\bxjs@tmpa\@empty
3209 \else
3210 \let\bxjs@tmpa\bxjs@kanjiEmbed
3211 \fi
3212 \else
3213 \let\bxjs@tmpa\jsJaFont
3214 \fi
3215 \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3216 \ClassWarningNoLine\bxjs@clsname
3217 {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3218 not available on the current situation}%
3219 \let\bxjs@tmpa\@empty
3220 \fi\fi
3221 }
3222 \def\bxjs@@auto{auto}
3223 \def\bxjs@@noEmbed{noEmbed}

```

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

```

3224 \let\bxjs@kanjiEmbed\relax

```

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

```

3225 \@onlypreamble\bxjs@get@kanjiEmbed
3226 \def\bxjs@get@kanjiEmbed{%

```

```

3227 \begingroup\setbox\z@=\hbox{%
3228   \global\let\bxjs@g@tmpa\relax
3229   \endlinechar\m@ne
3230   \let\do\@makeother\dospecials
3231   \catcode32=10 \catcode12=10 %form-feed
3232   \let\bxjs@tmpa\@empty
3233   \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3234   \ifeof\@inputcheck\else
3235     \read\@inputcheck to\bxjs@tmpa
3236     \closein\@inputcheck
3237   \fi
3238   \ifx\bxjs@tmpa\@empty\else
3239     \openin\@inputcheck="\bxjs@tmpa"\relax
3240     \@tempswatrue
3241     \loop\if@tempswa
3242       \read\@inputcheck to\bxjs@tmpa
3243       \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3244       \ifx\bxjs@tmpa\relax\else
3245         \global\let\bxjs@g@tmpa\bxjs@tmpa
3246         \@tempswafalse
3247       \fi
3248       \ifeof\@inputcheck \@tempswafalse \fi
3249     \repeat
3250   \fi
3251 }\endgroup
3252 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3253 }
3254 \@onlypreamble\bxjs@get@ke@a
3255 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3256   \ifx$#1$\def\bxjs@tmpa{#2}%
3257   \else \let\bxjs@tmpa\relax
3258   \fi}

```

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

```

3259 \newcommand*\jachar[1]{%
3260   \begingroup
3261     \jsLetHeadChar\bxjs@tmpa{#1}%
3262     \ifx\bxjs@tmpa\relax
3263       \ClassWarningNoLine\bxjs@clsname
3264         {Illegal argument given to \string\jachar}%
3265     \else
3266       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3267     \fi
3268   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3269 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
3270 \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。

```
3271 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3272 \@onlypreamble\bxjs@fix@hyperref@unicode
3273 \def\bxjs@fix@hyperref@unicode#1{%
3274   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3275   \@namedef{KV@Hyp@bxjs/hook}##1{%
3276     \KV@Hyp@unicode{##1}%
3277     \def\KV@Hyp@unicode####1{%
3278       \expandafter\ifx\csname if##1\expandafter\endcsname
3279       \csname if####1\endcsname\else
3280       \ClassWarningNoLine{bxjs@clsname
3281       {Blcoked hyperref option 'unicode=####1'}}%
3282     \fi
3283   }%
3284 }%
3285 }
```

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

```
3286 \@onlypreamble\bxjs@urgent@special
3287 \def\bxjs@urgent@special#1{%
3288   \AtBeginDvi{\special{#1}}%
3289   \AtBeginDocument{%
3290     \@ifpackageloaded{atbegshi}{%
3291       \begingroup
3292       \toks\z@{\special{#1}}%
3293       \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3294       \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3295     \endgroup
3296   }{}%
3297 }%
3298 }
```

C.2 pTeX 用設定

```
3299 \if j\jsEngine
```

■共通命令の実装

```
3300 \def\bxjs@apply@kanjiskip{%
3301   \kanjiskip\@tempskipa}
3302 \def\bxjs@apply@xkanjiskip{%
3303   \xkanjiskip\@tempskipa}
```

`\jaJaChar` のサブマクロ。

```
3304 \def\bxjs@jachar#1{%
3305   \bxjs@jachar@a#1...\@nil}
3306 \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。

```
3307   \ifx.#2#1%
```

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

```
3308   \else\ifx.#3%
3309     \@tempcnta`#1 \multiply\@tempcnta64
3310     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3311     \bxjs@jachar@b
3312   \else\ifx.#4%
3313     \@tempcnta`#1 \multiply\@tempcnta64
3314     \advance\@tempcnta`#2 \multiply\@tempcnta64
3315     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3316     \bxjs@jachar@b
3317   \else
3318     \@tempcnta`#1 \multiply\@tempcnta64
3319     \advance\@tempcnta`#2 \multiply\@tempcnta64
3320     \advance\@tempcnta`#3 \multiply\@tempcnta64
3321     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3322     \bxjs@jachar@b
3323   \fi\fi\fi}
```

符号値が `\@tempcnta` の和文文字を出力する処理。

```
3324 \ifjsWithupTeX
3325   \def\bxjs@jachar@b{\kchar\@tempcnta}
3326 \else
3327   \def\bxjs@jachar@b{%
3328     \ifx\bxUInt\@undefined\else
3329       \bxUInt{\@tempcnta}%
3330     \fi}
3331 \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。

```
3332 \let\bxjs@tmpa\jsJaFont
3333 \ifx\bxjs@tmpa\bxjs@@auto
3334   \let\bxjs@tmpa\@empty
3335 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3336   \def\bxjs@tmpa{noembed}
3337 \fi\fi
3338 \ifx\jsJaFont\@empty\else
3339   \edef\bxjs@nxt{%
3340     \noexpand\RequirePackage[\jsJaFont]
```

```

3341 {pxchfon}[2010/05/12]]% v0.5
3342 \bxjs@nxt
3343 \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……。)

```

3344 \begingroup
3345 \global\let\@gtempa\relax
3346 \catcode`\|=0 \catcode`\|=12
3347 |def|bxjs@check@a#1|@nil{%
3348 |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3349 |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3350 |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3351 |catcode`\|=0 \catcode`\|=12
3352 \def|bxjs@check@b#1keyval#2\@nnil{%
3353 \ifx$#2$\else
3354 \xdef\@gtempa{%
3355 \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3356 \fi}
3357 \@firstofone{%
3358 \catcode10=12 \endlinechar\m@ne
3359 \let\do\@makeother \dospecials \catcode32=10
3360 \openin\@inputcheck=otf.sty\relax
3361 \@tempswatrue
3362 \loop\if@tempswa
3363 \ifeof\@inputcheck \@tempswafalse \fi
3364 \if@tempswa
3365 \read\@inputcheck to\bxjs@line
3366 \expandafter\bxjs@check\bxjs@line\@nil
3367 \fi
3368 \repeat
3369 \closein\@inputcheck
3370 \endgroup}
3371 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3372 \ifbxjs@hyperref@enc
3373 \bxjs@fix@hyperref@unicode{false}
3374 \fi

```

tunicode special 命令を出力する。

```

3375 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3376 \else\ifjsWithpTeXng T\else F\fi\fi T%
3377 \ifnum\jis"2121="A1A1 %euc
3378 \bxjs@urgent@special{pdf:tunicode EUC-UCS2}
3379 \else\ifnum\jis"2121="8140 %sjis

```

```

3380 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3381 \else\ifnum\jis"2121="3000 %uptex
3382 \ifbxjs@bigcode
3383 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3384 \PassOptionsToPackage{bigcode}{pxjahyper}
3385 \else
3386 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3387 \fi
3388 \fi\fi\fi
3389 \let\bxToUnicodeSpecialDone=t
3390 \fi

```

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

```

3391 \ifx f\bxjs@enablejfam\else
3392 \@enablejfamtrue
3393 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

3394 \if@enablejfam
3395 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3396 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3397 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3398 \jfam\symmincho
3399 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3400 \AtBeginDocument{%
3401 \ifx\reDeclareMathAlphabet\@undefined\else
3402 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3403 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3404 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3405 \fi}
3406 \fi

```

C.3 pdf_TE_X 用設定 : CJK + bxcjkatype

```

3407 \else\if p\jsEngine

```

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

```

3408 \bxjs@adjust@jafont{f}
3409 \edef\bxjs@nxt{%
3410 \noexpand\RequirePackage[%
3411 \ifx\bxjs@tmpa\@empty\else \bxjs@tmpa,\fi
3412 whole,autotilde]{bxcjkatype}[2013/10/15]}% v0.2c
3413 \bxjs@nxt
3414 \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。

```
3415 \ifbxjs@hyperref@enc
3416   \PassOptionsToPackage{unicode}{hyperref}
3417 \fi
```

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

※ bxcjkatype を whole 付きで使っていることが前提。

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

```
3418 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3419 \begingroup
3420   \CJK@input{UTF8.bdg}
3421 \endgroup
3422 \g@addto@macro\pdfstringdefPreHook{%
3423   \@nameuse{CJK@UTF8Binding}%
3424 }
3425 \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。

```
3426 \ifx\bxcjkatypeHyperrefPatchDone\@undefined
3427 \g@addto@macro\pdfstringdefPreHook{%
3428   \ifx~\bxjs@@CJKtilde
3429     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3430     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3431     \let~\@empty
3432   \fi
3433 }
3434 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3435 \def\bxjs@@tildecmd{~}
3436 \def\bxjs@LetUnexpandableSpace#1{%
3437   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
3438     \bxjs@org@LetUnexpandableSpace#1%
3439   \fi}
3440 \fi
```

■共通命令の実装

```
3441 \newskip\jsKanjiSkip
3442 \newskip\jsXKanjiSkip
3443 \ifx\CJKecglue\@undefined
3444   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3445 \fi
3446 \let\autospacing\bxjs@enable@kanjiskip
3447 \let\noautospacing\bxjs@disable@kanjiskip
3448 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3449 \def\bxjs@apply@kanjiskip{%
```

```

3450 \jsKanjiSkip\@tempskipa
3451 \let\CJKglue\bxjs@CJKglue}
3452 \let\autoxspacing\bxjs@enable@xkanjiskip
3453 \let\noautoxspacing\bxjs@disable@xkanjiskip
3454 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3455 \def\bxjs@apply@xkanjiskip{%
3456 \jsXKanjiSkip\@tempskipa
3457 \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

3458 \def\bxjs@jachar#1{%
3459 \CJKforced{#1}}

```

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

```

3460 \ifx t\bxjs@enablejfam
3461 \ClassWarningNoLine\bxjs@clsname
3462 {You cannot use 'enablejfam=true', since the\MessageBreak
3463 CJK package does not support Japanese math}
3464 \fi

```

C.4 Xe_{La}TeX 用設定：xeCJK + zxjatype

```

3465 \else\if x\jsEngine

```

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

```

3466 \RequirePackage{zxjatype}
3467 \PassOptionsToPackage{no-math}{fontspec}%!
3468 \PassOptionsToPackage{xetex}{graphicx}%!
3469 \PassOptionsToPackage{xetex}{graphics}%!
3470 \ifx\zxJaFamilyName\@undefined
3471 \ClassError\bxjs@clsname
3472 {xeCJK or zxjatype is too old}\@ehc
3473 \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。

```

3474 \bxjs@adjust@jafont{f}
3475 \ifx\bxjs@tmpa\@empty
3476 \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3477 \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3478 \else
3479 \edef\bxjs@nxt{%
3480 \noexpand\RequirePackage[\bxjs@tmpa]{
3481 {zxjafont}[2013/01/28]}% v0.2a
3482 \bxjs@nxt
3483 \fi

```


■hyperref 対策 unicode オプションの指定に関する話。

X_ƒT_EX 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_ƒT_EX 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_ƒT_EX 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。

```
3484 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3485   \ifbxjs@hyperref@enc
3486     \PassOptionsToPackage{unicode}{hyperref}
3487   \fi
3488 \fi
```

■段落頭でのグルー挿入禁止 どうやら、zxjatype の \inhibitglue の実装が極めて杜撰なため、1.0 版での実装では全く期待通りの動作をしていないし、そもそも（少なくとも現状の）xeCJK では、段落頭での \inhibitglue は実行しないほうが JS クラスの出力に近いものが得られるらしい。

従って、\jsInhibitGlueAtParTop は結局何もしないことにする。

```
3489 \let\jsInhibitGlueAtParTop\@empty
```

■共通命令の実装

```
3490 \newskip\jsKanjiSkip
3491 \newskip\jsXKanjiSkip
3492 \ifx\CJKecglue\@undefined
3493   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3494 \fi
3495 \let\autospacing\bxjs@enable@kanjiskip
3496 \let\noautospacing\bxjs@disable@kanjiskip
3497 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3498 \def\bxjs@apply@kanjiskip{%
3499   \jsKanjiSkip\@tempskipa
3500   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3501 \let\autoxspacing\bxjs@enable@xkanjiskip
3502 \let\noautoxspacing\bxjs@disable@xkanjiskip
3503 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3504 \def\bxjs@apply@xkanjiskip{%
3505   \jsXKanjiSkip\@tempskipa
3506   \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。

```
3507 \ifx\mcfamily\@undefined
```

```

3508 \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3509 \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3510 \fi

    \jachar のサブマクロの実装。
3511 \def\bxjs@jachar#1{%
3512   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3513   #1}

```

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

```

3514 \ifx t\bxjs@enablejfam
3515   \@enablejfamtrue
3516 \fi

```

実際に和文用の数式ファミリーの設定を行う。

※ FIXME: 要検討。

```

3517 \if@enablejfam
3518   \xeCJKsetup{CJKmath=true}
3519 \fi

```

C.5 LuaTeX 用設定: LuaTeX-ja

```

3520 \else\if 1\jsEngine

```

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

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

※ 1.0b 版から「graphics パッケージに `pdftex` オプションを渡す」処理を行っていたが、1.4 版で廃止された。

```

3521 \let\zw\undefined
3522 \RequirePackage{luatexja}
3523 \RequirePackage{luatexja-fontspec}

```

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

```

3524 \ExplSyntaxOn
3525 \fp_gset:Nn \g_ltx_fontscalescale_fp { \jsScale }
3526 \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)。

```

3527 \bxjs@adjust@jafont{t}
3528 \ifx\bxjs@tmpa\bxjs@@noEmbed

```

```

3529 \def\bxjs@tmpa{noembed}
3530 \fi
3531 \ifx\bxjs@tmpa\@empty
3532 \defaultjfontfeatures{ Kerning=Off }
3533 \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3534 \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3535 \else
3536 \edef\bxjs@nxt{%
3537 \noexpand\RequirePackage[\bxjs@tmpa]
3538 {luatexja-preset}}%
3539 \bxjs@nxt
3540 \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。

```

3541 \DeclareRobustCommand\rmfamily
3542 {\not@math@alphabet\rmfamily\mathrm
3543 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3544 \DeclareRobustCommand\sffamily
3545 {\not@math@alphabet\sffamily\mathsf
3546 \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3547 \DeclareRobustCommand\ttfamily
3548 {\not@math@alphabet\ttfamily\mathtt
3549 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3550 \AtBeginDocument{%
3551 \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3552 \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
3553 \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
3554 \bxjs@if@sf@default{%
3555 \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

3556 % 次の3つは既定値の通り
3557 %\ltjsetparameter{prebreakpenalty={` ,10000}}
3558 %\ltjsetparameter{postbreakpenalty={` “,10000}}
3559 %\ltjsetparameter{prebreakpenalty={` ”,10000}}
3560 \ltjsetparameter{jaxspmde={` ! ,1}}
3561 \ltjsetparameter{jaxspmde={` 〒,2}}
3562 \ltjsetparameter{alxspmde={` +,3}}
3563 \ltjsetparameter{alxspmde={` %,3}}

```

■段落頭でのグルー挿入禁止

```

3564 \protected\def\@inhibitglue{%
3565 \directlua{%
3566 luatexja.jfmglue.create_beginpar_node()}}
3567 \let\bxjs@ltj@inhibitglue\@inhibitglue
3568 \let\@inhibitglue\undefined

```

■hyperref 対策 unicode にするべき。

```
3569 \ifbxjs@hyperref@enc
3570   \bxjs@fix@hyperref@unicode{true}
3571 \fi
```

■共通命令の実装

```
3572 \protected\def\autospaceing{%
3573   \ltjsetparameter{autospaceing=true}}
3574 \protected\def\noautospaceing{%
3575   \ltjsetparameter{autospaceing=false}}
3576 \protected\def\autoxspaceing{%
3577   \ltjsetparameter{autoxspaceing=true}}
3578 \protected\def\noautoxspaceing{%
3579   \ltjsetparameter{autoxspaceing=false}}
3580 \def\bxjs@apply@kanjiskip{%
3581   \ltjsetparameter{kanjiskip={\@tempskipa}}}}
3582 \def\bxjs@apply@xkanjiskip{%
3583   \ltjsetparameter{xkanjiskip={\@tempskipa}}}}
```

\jachar のサブマクロの実装。

```
3584 \def\bxjs@jachar#1{%
3585   \ltjjachar`#1\relax}
```

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

```
3586 \ifx f\bxjs@enablejfam
3587   \ClassWarningNoLine\bxjs@clsname
3588     {You cannot use 'enablejfam=false', since the\MessageBreak
3589     LuaTeX-japanese always provides Japanese math families}
3590 \fi
```

C.6 共通処理 (2)

```
3591 \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 3592 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3593 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
```

\mathmc この時点で未定義である場合に限り、\DeclareJaMathFontCommand を利用したフォール

\mathgt バックの定義を行う。

```
3594 \ifx\mathmc\@undefined
3595   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
3596   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
3597 \fi
```

■和文・和欧文間空白の初期値

```
3598 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}  
3599 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}  
3600 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}  
3601 \fi
```

以上で終わり。

```
3602 %</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。

```
3603 %<*modern>  
3604 \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。

```
3605 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@  
3606 \def\encodingdefault{T1}%  
3607 \input{t1enc.def}%  
3608 \fontencoding\encodingdefault\selectfont  
3609 \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。

```
3610 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@  
3611 \renewcommand{\rmdefault}{lmr}  
3612 \renewcommand{\sfdefault}{lmss}  
3613 \renewcommand{\ttdefault}{lmtt}  
3614 \fi
```

大型演算子用の数式フォントの設定。

※ `amsfonts` パッケージと同等にする。

```
3615 \DeclareFontShape{OMX}{cmex}{m}{n}{%  
3616   <-7.5>cmex7<7.5-8.5>cmex8%  
3617   <8.5-9.5>cmex9<9.5->cmex10}{}%  
3618 \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。

```
3619 \def\cmex@opt{10}
```

D.2 fixltx2e 読込

※ fixltx2e 廃止前の L^AT_EX カーネルの場合。

```
3620 \ifx\@IncludeInRelease\@undefined
3621 \RequirePackage{fixltx2e}
3622 \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。

```
3623 \RequirePackage{bxjscjkcat}
```

D.4 完了

おしまい。

```
3624 %</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。

```
3625 %<*pandoc>
3626 \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。

```
3627 \@onlypreamble\bxjs@set@dupload@proc
3628 \def\bxjs@set@dupload@proc#1{%
3629   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
3630 \@onlypreamble\bxjs@set@dupload@proc@a
3631 \def\bxjs@set@dupload@proc@a#1{%
3632   \@onlypreamble#1\def#1##1}
```

`\@ifoptions` `\@ifoptions` の再定義。

```
3633 \@onlypreamble\bxjs@org@ifoptions
3634 \let\bxjs@org@ifoptions\@ifoptions
```

```

3635 \newif\ifbxjs@dlp
3636 \def\@ifoptions#1#2#3{%
3637   \bxjs@dlpfalse
3638   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
3639   \ifx\bxjs@tmpa\bxjs@tmpb
3640     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
3641       \bxjs@dlptrue \fi
3642   \fi
3643   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
3644   \else \expandafter\bxjs@org@ifoptions
3645   \fi {#1}{#2}{#3}}
3646 \AtBeginDocument{%
3647   \let\@ifoptions\bxjs@org@ifoptions}
3648 \@onlypreamble\bxjs@do@dupload@proc
3649 \def\bxjs@do@dupload@proc#1#2#3{%
3650   \csname bxjs@dlp/#2.#1\endcsname{#3}%
3651   \@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。

```

3652 \def\bxjs@mark@as@loaded#1{%
3653   \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3654 \def\bxjs@mal@a#1#2{%
3655   \ifx#1\relax
3656     \def#1{2001/01/01}%
3657     \ClassInfo\bxjs@clsname
3658     {File '#2' marked as loaded\@gobble}%
3659   \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。

```

3660 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi>0

```

Polyglossia について。

```

3661 \bxjs@mark@as@loaded{polyglossia.sty}
3662 \bxjs@set@dupload@proc{polyglossia.sty}{%
3663   \ClassWarning\bxjs@clsname
3664   {Loading of polyglossia is blocked}}
3665 \ifx\setmainlanguage\@undefined
3666 \newcommand*\setmainlanguage[2][{}]{%
3667 \newcommand*\setotherlanguage[2][{}]{%
3668   \ifcat_#2_\else
3669     \expandafter\let\csname #2\endcsname\@empty
3670     \expandafter\let\csname end#2\endcsname\@empty

```

```

3671 \expandafter\let\csname text#2\endcsname\@firstofone
3672 \fi}
3673 \newcommand*\setotherlanguages[2][]{\%
3674 \@for\bxjs@tmpa:={#2}\do{\%
3675 \setotherlangauge{\bxjs@tmpa}}}
3676 \fi
3677 \else

```

Babel について。

```

3678 \bxjs@mark@as@loaded{babel.sty}
3679 \bxjs@set@dupload@proc{babel.sty}{\%
3680 \ClassWarning\bxjs@clsname
3681 {Loading of babel is blocked}}
3682 \let\foreignlanguage\@secondoftwo
3683 \let\otherlanguage\@gobble
3684 \let\endotherlanguage\@empty
3685 \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。

```

3686 \bxjs@set@dupload@proc{geometry.sty}{\%
3687 \setpagelayout*{#1}}

```

E.4 CJKmainfont 変数

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

```

3688 \if 1\jsEngine
3689 \bxjs@mark@as@loaded{xeCJK.sty}
3690 \providecommand*\setCJKmainfont{\setmainfont}
3691 \fi

```

E.5 fixltx2e パッケージ

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

L^AT_EX カーネルが新しい場合は fixltx2e を読込済にする。

```

3692 \ifx\@IncludeInRelease\@undefined\else
3693 \bxjs@mark@as@loaded{fixltx2e.sty}
3694 \fi

```


E.6 cmap パッケージ

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

```
3695 \if j\jsEngine
3696   \bxjs@mark@as@loaded{cmap.sty}
3697 \fi
```

E.7 microtype パッケージ

警告が多すぎなので消す。

```
3698 \if j\jsEngine \else
3699   \PassOptionsToPackage{verbose=silent}{microtype}
3700 \fi
```

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

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

```
3701 \if j\jsEngine
3702   \bxjs@mark@as@loaded{microtype.sty}
3703   \newcommand*\UseMicrotypeSet[2][]{\}
3704 \fi
```

E.8 完了

おしまい。

```
3705 %</pandoc>
```

和文ドライバ実装はここまで。

```
3706 %</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。

```
3707 %<*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

```
3708 %<*compat>
3709 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

3710 \let\bxac@engine=n
3711 \def\bxac@do#1#2{%
3712   \edef\bxac@tmpa{\string#1}%
3713   \edef\bxac@tmpb{\meaning#1}%
3714   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
3715 \bxac@do\XeTeXversion{\let\bxac@engine=x}
3716 \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。

3717 \ifx\jsAtEndOfClass\@undefined
3718   \let\bxac@delayed@if@bxjs\@firstofone
3719 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
3720 \fi

\ImposeOldLuaTeXBehavior ムニャムニャ。

\RevokeOldLuaTeXBehavior 3721 \newif\ifbxac@in@old@behavior
3722 \let\ImposeOldLuaTeXBehavior\relax
3723 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 XeTeX 部分

```
3724 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。

3725 \@onlypreamble\bxac@adjust@charclass
3726 \bxac@delayed@if@bxjs{%
3727   \@ifpackageloaded{xeCJK}{-}{%else
3728     \ifx\xe@alloc@intercharclass\@undefined\else
3729       \ifnum\xe@alloc@intercharclass=\z@
3730         \PackageInfo\bxac@pkgname
3731           {Setting up interchar class for CJK...\@gobble}%
3732         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
3733           \xe@alloc@intercharclass=3
3734         }{%else
3735           \PackageWarning\bxac@pkgname
3736             {Cannot find file 'load-unicode-xetex-classes.tex'%
3737             \@gobble}%
3738         }%
3739       \fi\fi
3740       \ifnum\XeTeXcharclass"3041=\z@
3741         \PackageInfo\bxac@pkgname
```

```

3742      {Adjusting interchar class for CJK...\@gobble}%
3743      \@for\bxac@x:={%
3744          3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3745          3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3746          30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3747          31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3748          31FF%
3749      }\do{\XeTeXcharclass"\bxac@x=\@ne}%
3750  \fi
3751  }%
3752 }

```

以上。

```
3753 \fi
```

G.3 LuaTeX 部分

```

3754 \ifx l\bxac@engine
      ムニヤムニヤ。
3755 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
3756   \chardef\pdftexversion=200
3757   \def\pdftexrevision{0}
3758   \let\pdftexbanner\luatexbanner
3759 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。
\RevokeOldLuaTeXBehavior
3760 \begingroup\expandafter\expandafter\expandafter\endgroup
3761 \expandafter\ifx\csname outputmode\endcsname\relax\else
3762 \def\bxac@ob@list{%
3763   \do{\let}\pdfoutput{\outputmode}%
3764   \do{\let}\pdfpagewidth{\pagewidth}%
3765   \do{\let}\pdfpageheight{\pageheight}%
3766   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
3767   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
3768 \def\bxac@ob@do#1#2{\begingroup
3769   \expandafter\bxac@ob@do@a\csname bxac@string#2\endcsname{#1}#2}
3770 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3771   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3772   \else \let#3#1\relax \let#1\@undefined
3773   \fi}
3774 \protected\def\ImposeOldLuaTeXBehavior{%
3775   \unless\ifbxac@in@old@behavior
3776     \bxac@in@old@behaviortrue
3777     \let\do\bxac@ob@do \bxac@ob@list
3778   \fi}
3779 \protected\def\RevokeOldLuaTeXBehavior{%
3780   \ifbxac@in@old@behavior
3781     \bxac@in@old@behaviorfalse
3782     \let\do\bxac@ob@do \bxac@ob@list
3783   \fi}

```

3784 \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。

```
3785 \ifnum\luatexversion>64 \directlua{
3786   local function range(cs, ce, cc, ff)
3787     if ff or not tex.getcatcode(cs) == cc then
3788       local setcc = tex.setcatcode
3789       for c = cs, ce do setcc(c, cc) end
3790     end
3791   end
3792   range(0x3400, 0x4DB5, 11, false)
3793   range(0x4DB5, 0x4DBF, 11, true)
3794   range(0x4E00, 0x9FCC, 11, false)
3795   range(0x9FCD, 0x9FFF, 11, true)
3796   range(0xAC00, 0xD7A3, 11, false)
3797   range(0x20000, 0x2A6D6, 11, false)
3798   range(0x2A6D7, 0x2A6FF, 11, true)
3799   range(0x2A700, 0x2B734, 11, false)
3800   range(0x2B735, 0x2B73F, 11, true)
3801   range(0x2B740, 0x2B81D, 11, false)
3802   range(0x2B81E, 0x2B81F, 11, true)
3803   range(0x2B820, 0x2CEA1, 11, false)
3804   range(0x2CEA2, 0x2FFFD, 11, true)
3805 } \fi
```

以上。

3806 \fi

G.4 完了

おしまい。

3807 %</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

```
3808 %<*cjkat>
3809 \def\bxjx@pkgname{bxjscjkat}
3810 \newcount\bxjx@canta
\bxjx@engine エンジンの種別。
3811 \let\bxjx@engine=n
3812 \def\bxjx@do#1#2{%
3813   \edef\bxjx@tmpa{\string#1}%
3814   \edef\bxjx@tmpb{\meaning#1}%
```

```

3815 \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3816 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3817 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3818 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3819 \bxjx@do\pdfTeXversion{\let\bxjx@engine=p}
3820 \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。

```

3821 \def\bxjx@do#1#2{%
3822   \if#1\bxjx@engine
3823     \@ifpackageloaded{#2}{\fi}%else
3824     \PackageError\bxjx@pkgname
3825       {Package '#2' must be loaded}%
3826       {Package loading is aborted.\MessageBreak\@ehc}%
3827   \endinput}
3828 \fi}
3829 \bxjx@do{p}{bxcjkatype}
3830 \bxjx@do{x}{xeCJK}
3831 \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。

```

3832 \ifx\TextOrMath\@undefined
3833   \RequirePackage{fixltx2e}
3834 \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。

```

3835 \if u\bxjx@engine
3836 \@for\bxjx@x:={%
3837 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3838 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3839 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3840 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3841 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3842 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
3843 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3844 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,44D0,A500,A640,%
3845 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3846 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
3847 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3848 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3849 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3850 10480,10500,10530,10600,10800,10840,10860,10880,%
3851 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%

```

```

3852 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3853 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3854 11280,112B0,11300,11480,11580,11600,11680,11700,%
3855 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3856 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3857 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3858 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
3859 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3860 }\do{\kcatcode"\bxjx@x=15 }
3861 \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。

```

3862 \def\bxjx@grkcyr@list{%
3863 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
3864 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
3865 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
3866 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
3867 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
3868 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
3869 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
3870 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
3871 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
3872 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
3873 \do{039B}{LGR}{\textLambda}{\Lambda}%   % GR. C. L. LAMDA
3874 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
3875 \do{039D}{LGR}{\textNu}{N}%             % GR. C. L. NU
3876 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
3877 \do{039F}{LGR}{\textOmicron}{O}%       % GR. C. L. OMICRON
3878 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
3879 \do{03A1}{LGR}{\textRho}{P}%            % GR. C. L. RHO
3880 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
3881 \do{03A4}{LGR}{\textTau}{T}%             % GR. C. L. TAU
3882 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}% % GR. C. L. UPSILON
3883 \do{03A6}{LGR}{\textPhi}{\Phi}%         % GR. C. L. PHI
3884 \do{03A7}{LGR}{\textChi}{X}%            % GR. C. L. CHI
3885 \do{03A8}{LGR}{\textPsi}{\Psi}%         % GR. C. L. PSI
3886 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA
3887 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
3888 \do{03B2}{LGR}{\textbeta}{\beta}%       % GR. S. L. BETA
3889 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA
3890 \do{03B4}{LGR}{\textdelta}{\delta}%     % GR. S. L. DELTA
3891 \do{03B5}{LGR}{\textepsilon}{\epsilon}% % GR. S. L. EPSILON
3892 \do{03B6}{LGR}{\textzeta}{\zeta}%        % GR. S. L. ZETA

```

3893 \do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
3894 \do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
3895 \do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
3896 \do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
3897 \do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
3898 \do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
3899 \do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
3900 \do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
3901 \do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
3902 \do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
3903 \do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
3904 \do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
3905 \do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
3906 \do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
3907 \do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
3908 \do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
3909 \do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
3910 \do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
3911 \do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
3912 \do{0401}{T2A}{\CYRYO}{\O}%	% CY. C. L. IO
3913 \do{0410}{T2A}{\CYRA}{\A}%	% CY. C. L. A
3914 \do{0411}{T2A}{\CYRB}{\B}%	% CY. C. L. BE
3915 \do{0412}{T2A}{\CYRV}{\V}%	% CY. C. L. VE
3916 \do{0413}{T2A}{\CYRG}{\G}%	% CY. C. L. GHE
3917 \do{0414}{T2A}{\CYRD}{\D}%	% CY. C. L. DE
3918 \do{0415}{T2A}{\CYRE}{\E}%	% CY. C. L. IE
3919 \do{0416}{T2A}{\CYRZH}{\ZH}%	% CY. C. L. ZHE
3920 \do{0417}{T2A}{\CYRZ}{\Z}%	% CY. C. L. ZE
3921 \do{0418}{T2A}{\CYRI}{\I}%	% CY. C. L. I
3922 \do{0419}{T2A}{\CYRISHRT}{\SHRT}%	% CY. C. L. SHORT I
3923 \do{041A}{T2A}{\CYRK}{\K}%	% CY. C. L. KA
3924 \do{041B}{T2A}{\CYRL}{\L}%	% CY. C. L. EL
3925 \do{041C}{T2A}{\CYRM}{\M}%	% CY. C. L. EM
3926 \do{041D}{T2A}{\CYRN}{\N}%	% CY. C. L. EN
3927 \do{041E}{T2A}{\CYRO}{\O}%	% CY. C. L. O
3928 \do{041F}{T2A}{\CYRP}{\P}%	% CY. C. L. PE
3929 \do{0420}{T2A}{\CYRR}{\R}%	% CY. C. L. ER
3930 \do{0421}{T2A}{\CYRS}{\S}%	% CY. C. L. ES
3931 \do{0422}{T2A}{\CYRT}{\T}%	% CY. C. L. TE
3932 \do{0423}{T2A}{\CYRU}{\U}%	% CY. C. L. U
3933 \do{0424}{T2A}{\CYRF}{\F}%	% CY. C. L. EF
3934 \do{0425}{T2A}{\CYRH}{\H}%	% CY. C. L. HA
3935 \do{0426}{T2A}{\CYRC}{\C}%	% CY. C. L. TSE
3936 \do{0427}{T2A}{\CYRCH}{\CH}%	% CY. C. L. CHE
3937 \do{0428}{T2A}{\CYRSH}{\SH}%	% CY. C. L. SHA
3938 \do{0429}{T2A}{\CYRSHCH}{\SHCH}%	% CY. C. L. SHCHA
3939 \do{042A}{T2A}{\CYRHRDSN}{\HRDSN}%	% CY. C. L. HARD SIGN
3940 \do{042B}{T2A}{\CYRERY}{\ERY}%	% CY. C. L. YERU
3941 \do{042C}{T2A}{\CYRSFTSN}{\SFTSN}%	% CY. C. L. SOFT SIGN

```

3942 \do{042D}{T2A}{\CYREREV}{}% % CY. C. L. E
3943 \do{042E}{T2A}{\CYRYU}{}% % CY. C. L. YU
3944 \do{042F}{T2A}{\CYRYA}{}% % CY. C. L. YA
3945 \do{0430}{T2A}{\cyra}{}% % CY. S. L. A
3946 \do{0431}{T2A}{\cyrb}{}% % CY. S. L. BE
3947 \do{0432}{T2A}{\cyrv}{}% % CY. S. L. VE
3948 \do{0433}{T2A}{\cyrg}{}% % CY. S. L. GHE
3949 \do{0434}{T2A}{\cyrd}{}% % CY. S. L. DE
3950 \do{0435}{T2A}{\cyre}{}% % CY. S. L. IE
3951 \do{0436}{T2A}{\cyrzh}{}% % CY. S. L. ZHE
3952 \do{0437}{T2A}{\cyrz}{}% % CY. S. L. ZE
3953 \do{0438}{T2A}{\cyri}{}% % CY. S. L. I
3954 \do{0439}{T2A}{\cyrishrt}{}% % CY. S. L. SHORT I
3955 \do{043A}{T2A}{\cyrk}{}% % CY. S. L. KA
3956 \do{043B}{T2A}{\cyrl}{}% % CY. S. L. EL
3957 \do{043C}{T2A}{\cyrm}{}% % CY. S. L. EM
3958 \do{043D}{T2A}{\cyrn}{}% % CY. S. L. EN
3959 \do{043E}{T2A}{\cyro}{}% % CY. S. L. O
3960 \do{043F}{T2A}{\cyrp}{}% % CY. S. L. PE
3961 \do{0440}{T2A}{\cyrr}{}% % CY. S. L. ER
3962 \do{0441}{T2A}{\cyrs}{}% % CY. S. L. ES
3963 \do{0442}{T2A}{\cyrt}{}% % CY. S. L. TE
3964 \do{0443}{T2A}{\cyrul}{}% % CY. S. L. U
3965 \do{0444}{T2A}{\cyrf}{}% % CY. S. L. EF
3966 \do{0445}{T2A}{\cyrh}{}% % CY. S. L. HA
3967 \do{0446}{T2A}{\cyrc}{}% % CY. S. L. TSE
3968 \do{0447}{T2A}{\cyrch}{}% % CY. S. L. CHE
3969 \do{0448}{T2A}{\cyrsh}{}% % CY. S. L. SHA
3970 \do{0449}{T2A}{\cyrshch}{}% % CY. S. L. SHCHA
3971 \do{044A}{T2A}{\cyrhrdsn}{}% % CY. S. L. HARD SIGN
3972 \do{044B}{T2A}{\cyrery}{}% % CY. S. L. YERU
3973 \do{044C}{T2A}{\cyrsftsn}{}% % CY. S. L. SOFT SIGN
3974 \do{044D}{T2A}{\cyrerev}{}% % CY. S. L. E
3975 \do{044E}{T2A}{\cyryu}{}% % CY. S. L. YU
3976 \do{044F}{T2A}{\cyrya}{}% % CY. S. L. YA
3977 \do{0451}{T2A}{\cyryo}{}% % CY. S. L. IO
3978 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
3979 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
3980 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
3981 \do{00B1}{TS1}{\textpm}{\pm}% % PLUS-MINUS SIGN
3982 \do{00B4}{TS1}{\textasciicircum}{}% % ACUTE ACCENT
3983 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
3984 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
3985 \do{00F7}{TS1}{\textdiv}{\div}% % DIVISION SIGN
3986 }

3987 \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。


```

3988 \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。

3989 \newcommand*\greekasCJK{%
3990   \bxjx@gcc@CJKtrue}
3991 \newcommand*\nogreekasCJK{%
3992   \bxjx@gcc@CJKfalse}

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

3993 \def\bxjx@do#1\relax{%
3994   \def\bxjx@fake@grk##1##2{%
3995     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
3996   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
3997     \ifx\##1\%
3998       \bxjx@cmta##4\divide\bxjx@cmta\@cclvi
3999       \multiply\bxjx@cmta\@cclvi \advance\bxjx@cmta`##3\relax
4000       \mathchar\bxjx@cmta
4001     \else ##3\fi}
4002 }\expandafter\bxjx@do\string\mathchar\relax

```

■pdfLaTeX・upLaTeX の場合

```
4003 \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。

```

4004 \@ifpackageloaded{inputenc}{}{%else
4005   \RequirePackage[utf8]{inputenc}}
4006 \def\bxjx@tmpa{utf8}
4007 \ifx\bxjx@tmpa\inputencdoingname
4008   \PackageWarningNoLine\bxjx@pkgname
4009     {Input encoding changed to utf8}%
4010   \inputencoding{utf8}%
4011 \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。

```

4012 \if u\bxjx@engine
4013 \kcatcode"0370=15
4014 \kcatcode"0400=15
4015 \kcatcode"0500=15
4016 \fi

```

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

```

4017 \def\do#1{%
4018   \@tempcnta="#1\relax
4019   \@tempcntb=\@tempcnta \divide\@tempcntb256
4020   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4021   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4022 \def\bxjx@do@a#1#2#3#4#5{%

```

```

4023 \ifx\#5\%
4024 \def\bxjx@tmpa{\@inmathwarn#4}%
4025 \else\ifcat A\noexpand#5%
4026 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4027 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4028 \else \def\bxjx@tmpa{#5}%
4029 \fi\fi
4030 \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4031 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
4032 \if u\bxjx@engine
4033 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
4034 \def\bxjx@do@b#1#2#3#4#5{%
4035 \kchardef#1=\@tempcnta
4036 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4037 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}}
4038 \else\if p\bxjx@engine
4039 \def\bxjx@do@b#1#2#3#4#5{%
4040 \mathchardef#1=\@tempcnta
4041 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{UTF{#2}}{#3}{#4}}%
4042 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}}
4043 \fi\fi
4044 \bxjx@grkcyr@list
4045 \let\bxjx@do@a\undefined
4046 \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。

```

4047 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4048 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4049 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4050 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4051 \count@=#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
4052 \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
4053 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4054 \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4055 \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4056 \else
4057 \wlog{ \space\space skipped defining Unicode char U+#1}%
4058 \fi\fi}

```

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

```

4059 \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。

```

4060 \ifbxjx@gcc@cj k #1%

```

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

```

4061 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%

```

```

4062 \else \UseTextSymbol{#2}{#3}%
4063 \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。

```

4064 \begingroup
4065 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4066 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4067   \noexpand\bxjx@swap@DUC@cmd
4068   \the\toks@
4069   \noexpand\bxjx@swap@DUC@cmd}}
4070 \endgroup\next
4071 \def\bxjx@swap@DUC@cmd{%
4072   \let\bxjx@tmpa\DeclareUnicodeCharacter
4073   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4074   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■Xe_{La}TeX・Lua_{La}TeX の場合

```

4075 \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。

```

4076 \def\do#1{%
4077   \bxjx@cmta="#1\relax
4078   \begingroup
4079     \lccode`~=\bxjx@cmta
4080     \lowercase{\endgroup
4081       \bxjx@do@a{~}}{#1}}
4082 \def\bxjx@do@a#1#2#3#4#5{%
4083   \ifx\\#5\\\let\bxjx@tmpa\relax
4084   \else\ifcat A\noexpand#5%
4085     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4086       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4087   \else \def\bxjx@tmpa{#5}%
4088   \fi\fi
4089   \ifx\bxjx@tmpa\relax\else
4090     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4091   \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。

```

4092 \mathchardef\bxjx@tmpa="119
4093 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4094 \let\bxjx@do@a\undefined

```

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

```

4095 \if 1\bxjx@engine
4096   \protected\def\greekasCJK{%
4097     \bxjx@gcc@cjkttrue

```

```

4098 \ltjsetparameter{jacharrange={+2, +8}}
4099 \protected\def\nogreekasCJK{%
4100 \bxjx@gcc@cjkgfalse
4101 \ltjsetparameter{jacharrange={-2, -8}}
4102 \fi

```

X_gTeX における \no)greekasCJK の定義。

```

4103 \if x\bxjx@engine
4104 \protected\def\greekasCJK{%
4105 \bxjx@gcc@cjkgtrue
4106 \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4107 \bxjx@grkcyr@list}
4108 \protected\def\nogreekasCJK{%
4109 \bxjx@gcc@cjkgfalse
4110 \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4111 \bxjx@grkcyr@list}
4112 \fi

```

以上。

```

4113 \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。

```

4114 \nogreekasCJK

```

H.5 完了

おしまい。

```

4115 %</cjkgcat>

```

補助パッケージ実装はここまで。

```

4116 %</anc>

```