

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”)




v1.5b [2017/04/01]

この文書はソースコード説明書です。一般の文書作成者向けの解説については、ユーザーマニュアル `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	33
4	フォントサイズ	33
5	レイアウト	39
5.1	ページレイアウト	40
6	改ページ (日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 開発コミュニティ版のみ)	47
7	ページスタイル	48
8	文書のマークアップ	52
8.1	表題	52
8.2	章・節	57
8.3	リスト環境	68
8.4	パラメータの設定	76
8.5	フロート	77
8.6	キャプション	78
9	フォントコマンド	80

10	相互参照	82
10.1	目次の類	82
10.2	参考文献	87
10.3	索引	89
10.4	脚注	90
11	段落の頭へのグルー挿入禁止	93
12	いろいろなロゴ	96
13	amsmath との衝突の回避	97
14	初期設定	97
付録 A	和文ドライバの仕様 	101
付録 B	和文ドライバ : minimal 	102
B.1	補助マクロ	102
B.2	(u)pTeX 用の設定	104
B.3	pdfTeX 用の処理	108
B.4	X _Y TeX 用の処理	109
B.5	後処理 (エンジン共通)	109
付録 C	和文ドライバ : standard 	111
C.1	共通処理 (1)	112
C.2	pTeX 用設定	116
C.3	pdfTeX 用設定 : CJK + bxcjkatype	119
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	121
C.5	LuaTeX 用設定 : LuaTeX-ja	123
C.6	共通処理 (2)	126
付録 D	和文ドライバ : modern 	126
D.1	フォント設定	127
D.2	fixltx2e 読込	127
D.3	和文カテゴリコード	127
D.4	完了	127
付録 E	和文ドライバ : pandoc 	128
E.1	dupload システム	128
E.2	lang 変数	129
E.3	geometry 変数	130
E.4	CJKmainfont 変数	130
E.5	fixltx2e パッケージ	130

E.6	cmap パッケージ	130
E.7	microtype パッケージ	130
E.8	完了	131
付録 F	補助パッケージ一覧 	131
付録 G	補助パッケージ : bxjscompat 	131
G.1	準備	131
G.2	X _g TeX 部分	132
G.3	LuaTeX 部分	132
G.4	完了	134
付録 H	補助パッケージ : bxjscjkat 	134
H.1	準備	134
H.2	和文カテゴリコードの設定	135
H.3	ギリシャ・キリル文字の扱い	135
H.4	初期設定	142
H.5	完了	142

1 はじめに

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

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

本ドキュメントクラスは奥村晴彦氏および日本語 TeX 開発コミュニティによる「p_lA_TE_X2_ε 新ドキュメントクラス」に改変を加えたものである。本ドキュメントクラスに関する説明は全てこの形式の枠の中に記す。枠の外にあるものは原版著者による原版に対する解説である。

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

[2009-02-22] 田中琢爾氏による up_lA_TE_X 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` と表すことにしています。 `openany` で偶数ページからでも始まるようになります。

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

```
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。森本さんにしただって訂正しました。

```

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` : \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=<dimen>` : 基底フォントサイズを直接指定する。(xxpt オプションの代用なので、既定値は 10 pt である。)
- `jbase=<dimen>` : 基底フォントサイズを “和文規準で” 直接指定する。
- `scale=<real>` : 和文フォントのスケールを表すマクロ `\jsScale` の値を設定する。もちろんこの値を何らかの方法で和文処理モジュールに渡さないと意味を成さない。既定値は 0.924715 (= 13 Q/10 pt)。
- `noscale` : `scale=1` と等価。
- `mag=<int>` : `\mag` 値の直接設定。既定は `base` から算出する。
- `paper={<dimen:width>}{<dimen:height>}` : 用紙サイズ設定。用紙サイズオプションの代用で、既定値は a4paper 相当。
- `ja=<name>` : 使用する和文ドライバの指定。
- `jafont=<name>` : 和文フォントプリセットの指定。
- `japaram=<name>` : 和文フォントパラメタの指定。
- `magstyle=<name>` : “版面拡大” の実現方法の選択。
- `dvi=<name>` : DVI モードの時のみに参照されるドライバ指定。
- `geometry={class | user}` : `geometry` パッケージの読込を自動的に行うかユーザに任せるか。
- `fancyhdr=<bool>` : `fancyhdr` パッケージ用の調整を行うか。
- `layout=<name>` : レイアウト変種の指定。
- `textwidth-limit=<number>` : `bxjsbook` における、`\textwidth` の上限の全角単位での値。
- `paragraph-mark=<char>` : パラグラフのマーク。
- `whole-zw-lines=<bool>` : 行長を全角単位に丸めるか。
- `hyperref-enc=<bool>` : `hyperref` の文字コード指定補正を行うか。

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

※局所化の `\begingroup`～`\endgroup` について、以前は `\group`～`\egroup` を使っていたが、これだと数式モード中では空のサブ数式を生み出してしまうため修正した。

※元の長さが 128 pt 以上の場合でも動作するように修正した。

```

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

```

■複合設定オプション

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

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

```
292 \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。

```
293 \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。

```

294   \def\bxjs@composite@proc{%
295     \bxjs@oldfontcommandstrue
296     \setkeys{bxjs}{ja=pandoc}%
297     \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…。

```
298 \def\bxjs@driver@opt{dvipdfmx}%
299 \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。

```
300 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
301 %\let\bxjs@engine@opt\@undefined
```

エンジン明示指定のオプションの処理。

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

```
302 \DeclareOption{autodetect-engine}{%
303   \let\bxjs@engine@given=*}
304 \DeclareOption{latex}{%
305   \def\bxjs@engine@opt{latex}%
306   \let\bxjs@engine@given=n}
307 \DeclareOption{platex}{%
308   \def\bxjs@engine@opt{platex}%
309   \let\bxjs@engine@given=j}
310 \DeclareOption{uplatex}{%
311   \def\bxjs@engine@opt{uplatex}%
312   \let\bxjs@engine@given=u}
313 \DeclareOption{xelatex}{%
314   \def\bxjs@engine@opt{xelatex}%
315   \let\bxjs@engine@given=x}
316 \DeclareOption{pdflatex}{%
317   \def\bxjs@engine@opt{pdflatex}%
318   \let\bxjs@engine@given=p}
319 \DeclareOption{lualatex}{%
320   \def\bxjs@engine@opt{lualatex}%
321   \let\bxjs@engine@given=l}
322 \DeclareOption{platex-ng}{%
323   \def\bxjs@engine@opt{platex-ng}%
324   \let\bxjs@engine@given=g}
325 \DeclareOption{platex-ng*}{%
326   \def\bxjs@engine@opt{platex-ng*}%
327   \let\bxjs@platexng@nodrv=t%
328   \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。

```

329 %\let\bxjs@driver@given\@undefined
330 \let\bxjs@driver@@dvimode=0
331 \let\bxjs@driver@@dvipdfmx=1
332 \let\bxjs@driver@@pdfmode=2
333 \let\bxjs@driver@@xetex=3
334 \let\bxjs@driver@@dvips=4
335 \let\bxjs@driver@@none=5

```

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

```

336 %\let\bxjs@driver@opt\@undefined

337 \DeclareOption{dvips}{%
338   \def\bxjs@driver@opt{dvips}%
339   \let\bxjs@driver@given\bxjs@driver@@dvips}
340 \DeclareOption{dviout}{%
341   \def\bxjs@driver@opt{dviout}%
342   \let\bxjs@driver@given\bxjs@driver@@dvimode}
343 \DeclareOption{xdvi}{%
344   \def\bxjs@driver@opt{xdvi}%
345   \let\bxjs@driver@given\bxjs@driver@@dvimode}
346 \DeclareOption{dvipdfmx}{%
347   \def\bxjs@driver@opt{dvipdfmx}%
348   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
349 \DeclareOption{nodvidriver}{%
350   \def\bxjs@driver@opt{nodvidriver}%
351   \let\bxjs@driver@given\bxjs@driver@@none}
352 \DeclareOption{pdftex}{%
353   \def\bxjs@driver@opt{pdftex}%
354   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
355 \DeclareOption{luatex}{%
356   \def\bxjs@driver@opt{luatex}%
357   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
358 \DeclareOption{xetex}{%
359   \def\bxjs@driver@opt{xetex}%
360   \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。

```

361 \DeclareOption{dvipdfmx-if-dvi}{%
362   \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。

```

363 \newif\ifbxjs@usezw \bxjs@usezwtrue

```

`zw`、`noz` オプションの定義。

```

364 \DeclareOption{nozw}{%
365   \bxjs@usezwfalse}
366 \DeclareOption{zw}{%
367   \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。

```

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

    nojs、js オプションの定義。
369 \DeclareOption{nojs}{%
370   \bxjs@disguise@jsfalse}
371 \DeclareOption{js}{%
372   \bxjs@disguise@jstrue}

```

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

```

373 \newif\ifbxjs@precisetext

    noprecisetext / precisetext オプションの定義。
374 \DeclareOption{noprecisetext}{%
375   \bxjs@precisetextfalse}
376 \DeclareOption{precisetext}{%
377   \bxjs@precisetexttrue}

```

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

```

378 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

    nosimplejasetup / simplejasetup オプションの定義。
379 \DeclareOption{nosimplejasetup}{%
380   \bxjs@simplejasetupfalse}
381 \DeclareOption{simplejasetup}{%
382   \bxjs@simplejasetuptrue}

```

`\ifbxjs@bigcode` up_{TeX} 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。

```

383 \newif\ifbxjs@bigcode \bxjs@bigcodefalse

```

その上で、「T_EX 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、「T_EX のバージョンが 3.14159265（2014 年 1 月）以上であるか」を採用する。

```

384 \edef\bxjs@tmpa{\expandafter\noexpand\curname\endcurname}
385 \def\bxjs@tmpb#1 #2#3\@nil{%
386   \ifx1#2\bxjs@bigcodetrue \fi}
387 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil

```

`nobigcode` / `bigcode` オプションの定義。

```

388 \DeclareOption{nobigcode}{%

```

```

389 \bxjs@bigcodefalse}
390 \DeclareOption{bigcode}{%
391 \bxjs@bigcodetrue}

```

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

```

392 \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。

```

393 \DeclareOption{nooldfontcommands}{%
394 \bxjs@oldfontcommandsfalse}
395 \DeclareOption{oldfontcommands}{%
396 \bxjs@oldfontcommandstrue}

```

■keyval 型のオプション

```

397 \def\bxjs@setkey{%
398 \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
399 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
400 \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`。

```

401 \def\bxjs@safe@setkeys#1#2{%
402 \let\bxjs@KV@errx\KV@errx
403 \let\KV@errx\bxjs@safe@setkeys@a
404 \setkeys{#1}{#2}%
405 \let\KV@errx\bxjs@KV@errx}
406 \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。

```

407 \def\bxjs@set@keyval#1#2#3{%
408 \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
409 \ifx\bxjs@next\relax
410 \bxjs@error@keyval{#1}{#2}%
411 #3%
412 \else \bxjs@next
413 \fi}
414 \@onlypreamble\bxjs@error@keyval
415 \def\bxjs@error@keyval#1#2{%
416 \ClassError\bxjs@clsname
417 {Invalid value '#2' for option #1}\@ehc}

```

`\ifbxjs@scaleset` 和文スケール値が指定されたか。

```

418 \newif\ifbxjs@scaleset

```

`\jsScale` 〔実数値マクロ〕 和文スケール値。

```

419 \def\jsScale{0.924715}

    base オプションの処理。
420 \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。
421 \@onlypreamble\bxjs@do@opt@jbase
422 \let\bxjs@do@opt@jbase\relax
423 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
424 \def\bxjs@setjbasefontsize#1{%
425     \def\bxjs@do@opt@jbase{%
426         \bxjs@setbasefontlength\@tempdima{#1}%
427         \bxjs@invscale\@tempdima\jsScale
428         \bxjs@setbasefontsize{\@tempdima}}

    scale オプションの処理。
429 \define@key{bxjs}{scale}{%
430     \bxjs@scalesettrue
431     \edef\jsScale{#1}}

    noscale オプションの処理。
432 \DeclareOption{noscale}{%
433     \bxjs@scalesettrue
434     \def\jsScale{1}}

```

`\bxjs@param@mag` mag オプションの値。

```

435 \let\bxjs@param@mag\relax

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

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

```

`\bxjs@jadriver` 和文ドライバの名前。

```

438 \let\bxjs@jadriver\relax
439 %\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）。
440 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
441 \define@key{bxjs}{ja}[\relax]{%
442     \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}

```

`\jsJaFont` 和文フォント設定の名前。

```

443 \let\jsJaFont\@empty

```

jafont オプションの処理。

```

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

```

\jsJaParam 和文ドライバパラメタの文字列。

```

445 \let\jsJaParam\@empty

```

japaram オプションの処理。

```

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

```

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

```

447 \let\bxjs@magstyle@mag=m
448 \let\bxjs@magstyle@real=r
449 \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。

```

450 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
451 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
452 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal

```

\bxjs@magstyle@default は既定の値を表す。

```

453 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
454 \ifx l\jsEngine \ifnum\luatexversion>86
455   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
456 \fi\fi
457 \ifjsWithpTeXng
458   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
459 \fi
460 \let\bxjs@magstyle\bxjs@magstyle@default

```

magstyle オプションの処理。

```

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

```

\bxjs@geometry geometry オプションの値。

```

469 \let\bxjs@geometry@class=c
470 \let\bxjs@geometry@user=u
471 \let\bxjs@geometry\bxjs@geometry@class

```

geometry オプションの処理。

```

472 \define@key{bxjs}{geometry}{%
473   \expandafter\let\expandafter\bxjs@geometry\csname

```

```

474   bxjs@geometry@#1\endcsname
475   \ifx\bxjs@geometry\relax
476     \ClassError\bxjs@clsname
477     {Invalid value '#1' for option geometry}\@ehc
478     \let\bxjs@geometry\bxjs@geometry@class
479   \fi}

```

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

```

480 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

    fancyhdr オプションの処理。
481 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
482 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
483 \define@key{bxjs}{fancyhdr}{%
484   \bxjs@set@keyval{fancyhdr}{#1}{}}

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

485 \newif\ifbxjs@dvi@opt

    DVI モードのドライバとドライバ種別との対応。
486 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
487 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvips
488 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
489 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
490 \let\bxjs@dvidriver@@nodvidriver=\bxjs@driver@@none

    dvi オプションの処理。
491 \define@key{bxjs}{dvi}{%
492   \expandafter\let\expandafter\bxjs@tmpa\csname
493     bxjs@dvidriver@@#1\endcsname
494   \ifx\bxjs@tmpa\relax
495     \ClassError\bxjs@clsname
496     {Invalid value '#1' for option dvi}\@ehc
497   \else

    \bxjs@driver@given を未定義にしていることに注意。
498     \def\bxjs@driver@opt{#1}%
499     \let\bxjs@driver@given\@undefined
500     \bxjs@dvi@opttrue
501   \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。

```

502 \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。

```

503 \newif\ifbxjs@force@chapterabstract
504 %<book|report>\bxjs@force@chapterabstracttrue

```

layout オプションの処理。

```
505 \@namedef{bxjs@kv@layout@v1}{%
506 %<book>\bxjs@layout@buggyhmargintrue
507 %<report>\bxjs@force@chapterabstracttrue
508 }
509 \@namedef{bxjs@kv@layout@v2}{%
510 %<book>\bxjs@layout@buggyhmarginfalse
511 %<report>\bxjs@force@chapterabstractfalse
512 }
513 \define@key{bxjs}{layout}{%
514   \bxjs@set@keyval{layout}{#1}{}}
```

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

```
515 %\let\bxjs@textwidth@limit\undefined
516 \define@key{bxjs}{textwidth-limit}{%
517   \edef\bxjs@textwidth@limit{#1}}
```

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

```
518 %\let\bxjs@paragraph@mark\undefined
519 \define@key{bxjs}{paragraph-mark}{%
520   \edef\bxjs@paragraph@mark{#1}}
```

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

```
521 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
522 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
523 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
524 \define@key{bxjs}{whole-zw-lines}{\bxjs@set@keyval{wholezwlines}{#1}{}}
```

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

```
525 \newif\ifbxjs@xkanjiskip@cmd \bxjs@xkanjiskip@cmdtrue
526 \let\bxjs@kv@xkanjiskipcmd@true\bxjs@xkanjiskip@cmdtrue
527 \let\bxjs@kv@xkanjiskipcmd@false\bxjs@xkanjiskip@cmdfalse
528 \define@key{bxjs}{xkanjiskip-cmd}{\bxjs@set@keyval{xkanjiskipcmd}{#1}{}}
```

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

```
529 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue
530 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
531 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
532 \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。

```
533 \let\bxjs@ltx@removeelement\@removeelement
534 \def\@removeelement#1#2#3{%
535   \def\reserved@a{#2}%
536   \ifx\reserved@a\@empty \let#3\@empty
537   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
538   \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。

```
539 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
540 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
541 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
542 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
543 \ProcessOptions\relax
544 \bxjs@composite@proc
```

グローバルオプションのトークン列に { } が含まれていると、やはり後のパッケージの読み込みで不具合を起こすようである (`\ProcessOptions*` がエラーになる)。従って、このようなオプションは除外することにする。

```
545 \@onlypreamble\bxjs@purge@brace@elts
546 \def\bxjs@purge@brace@elts{%
547   \def\bxjs@tmpa{\@gobble}%
548   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
549   \let\@classoptionslist\bxjs@tmpa}
550 \@onlypreamble\bxjs@purge@be@a
551 \def\bxjs@purge@be@a#1,{%
552   \ifx\@nil#1\relax\else
553     \bxjs@purge@be@b#1{\@nil
554     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
555     \expandafter\bxjs@purge@be@a
556   \fi}
557 \@onlypreamble\bxjs@purge@be@b
558 \def\bxjs@purge@be@b#1#\bxjs@purge@be@c}
559 \@onlypreamble\bxjs@purge@be@c
560 \def\bxjs@purge@be@c#1\@nil{%
561   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
562 \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。

```
563 \@expandtwoargs\@removeelement
564 {papersize}\@classoptionslist\@classoptionslist
565 \@expandtwoargs\@removeelement
```

```

566 {10pt}\@classoptionslist\@classoptionslist
567 \@expandtwoargs\@removeelement
568 {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。

```

569 \ifx\bxjs@jadriver\relax\else
570 \let\bxjs@jadriver@given\bxjs@jadriver
571 \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。

```

572 \let\bxjs@tmpb\jsEngine
573 \ifx j\bxjs@tmpb\ifjsWithpTeXng
574 \let\bxjs@tmpb=g
575 \fi\fi
576 \ifx j\bxjs@tmpb\ifjsWithupTeX
577 \let\bxjs@tmpb=u
578 \fi\fi
579 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
580 \let\bxjs@tmpb=n
581 \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。）

```

582 \ifx *\bxjs@engine@given
583 \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。

```

584 \ifx j\bxjs@engine@given
585 \g@addto@macro\@classoptionslist{,latex}
586 \else\ifx u\bxjs@engine@given
587 \g@addto@macro\@classoptionslist{,uplatex}
588 \fi\fi
589 \fi
590 \ifx\bxjs@engine@given\@undefined\else
591 \ifx\bxjs@engine@given\bxjs@tmpb\else
592 \ClassError\bxjs@clsname
593 {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
594 \fi
595 \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。

```

596 \ifjsWithpTeXng
597 \g@addto@macro\@classoptionslist{,uplatex}
598 \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。

```

599 \@tempwattrue
600 \ifx \bxjs@driver@given\@undefined\else

```

```

601 \ifjsInPdfMode
602   \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
603     \@tempswafalse
604   \fi
605 \else\ifx x\jsEngine
606   \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
607     \@tempswafalse
608   \fi
609 \else
610   \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
611     \@tempswafalse
612   \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
613     \@tempswafalse
614   \fi\fi
615   \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
616     \@tempswafalse
617   \fi\fi
618 \fi\fi
619 \fi
620 \if@tempswa\else
621   \ClassError\bxjs@clsname
622   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
623 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

624 \ifjsInPdfMode \@tempswafalse
625 \else\ifx x\jsEngine \@tempswafalse
626 \else\ifjsWithpTeXng \@tempswafalse
627 \else \@tempwattrue
628 \fi\fi\fi
629 \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)。

```

630 \ifx\bxjs@driver@opt\undefined \ifx\bxjs@jadriver@given\undefined\else
631   \ClassWarningNoLine\bxjs@clsname
632   {No driver option is given}
633 \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。)

```

634 \ifbxjs@dvi@opt
635   \edef\bxjs@nxt{%
636     \let\noexpand\bxjs@driver@given
637     \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
638     \noexpand\g@addto@macro\noexpand\@classoptionslist
639     {,\bxjs@driver@opt}%
640   }\bxjs@nxt
641 \fi

```

642 \fi

エンジンが pT_EX-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。

643 \ifjsWithpTeXng

644 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx

645 \let\bxjs@platexng@nodrv\@undefined

646 \else\ifx t\bxjs@platexng@nodrv\else

647 \g@addto@macro\@classoptionslist{,dvipdfmx}

648 \fi\fi

649 \fi

ドライバが nodvidriver であった場合の処理。DVI ウェア依存の処理を全て無効化する。

650 \ifx\bxjs@driver@given\bxjs@driver@@none

651 \bxjs@papersizefalse

652 \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)pT_EX である場合は standard に変える。

653 \def\bxjs@@minimal{minimal}

654 \ifx\bxjs@jadriver\relax

655 \ifx j\jsEngine

656 \def\bxjs@jadriver{standard}

657 \else

658 \let\bxjs@jadriver\bxjs@@minimal

659 \fi

660 \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。

661 \ifx\bxjs@jadriver@given\@undefined\else

662 \ifx\bxjs@engine@given\@undefined

663 \ClassError\bxjs@clsname

664 {An engine option must be explicitly given}%

665 {When you use a Japanese-driver you must specify a correct\MessageBreak

666 engine option.\MessageBreak\@ehc}

667 \fi\fi

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

668 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else

669 \ifx\bxjs@magstyle\bxjs@magstyle@mag

670 \let\bxjs@magstyle\bxjs@magstyle@default

671 \ClassError\bxjs@clsname

672 {The engine does not support 'magstyle=usemag'}%

673 {LuaTeX v0.87 or later no longer supports the "mag" feature of TeX.\MessageBreak

674 The default value 'nomag*' is used instead.\MessageBreak \@ehc}

```
675 \fi
676 \fi
```

オプション処理時に遅延させていた `jbaser` の処理をここで実行する。

```
677 \bxjs@do@opt@jbaser
```

`\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。

```
678 \let\Cjascale\jsScale
```

後処理

```
679 \if@slide
680 \def\maybeblue{\ifundefined{ver@color.sty}{\color{blue}}
681 \fi
682 \if@landscape
683 \setlength\@tempdima {\paperheight}
684 \setlength\paperheight{\paperwidth}
685 \setlength\paperwidth {\@tempdima}
686 \fi
```

8bit 欧文 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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。）

```
687 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
688 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
689 \catcode\@tempcnta\active
690 \advance\@tempcnta\@ne
691 \repeat
692 \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。

```
693 \ifbxjs@disguise@js
694 %<book|report>\def\bxjs@js@clsname{jsbook}
695 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
696 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
697 \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。

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

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

```
699 \ifbxjs@oldfontcommands
700 \AtEndOfClass{\allowoldfontcommands}
701 \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。

```
702 %<slide>\def\n@baseline{13}%
703 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
704 %<!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。

```
705 \ifx\bxjs@magstyle\bxjs@magstyle@mag
706   \jsc@magtrue
707 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
708   \jsc@mag@xrealtrue
709 \fi\fi
```

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

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

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

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

```
710 \ifx\bxjs@param@mag\relax
711   \@tempdima=\bxjs@param@basefontsize
712   \advance\@tempdima.001pt \multiply\@tempdima25
713   \divide\@tempdima16384\relax \@tempcmta\@tempdima\relax
714   \edef\bxjs@param@mag{\the\@tempcmta}
715 \else
716 % mag 値が直接指定された場合
717   \let\c@bxjs@cmta\@tempcmta
718   \setcounter{bxjs@cmta}{\bxjs@param@mag}
719   \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
720 % 有効な mag 値の範囲は 1--32768
721   \edef\bxjs@param@mag{\the\@tempcmta}
```

```

722 \advance\@tempcnta100000
723 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
724 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
725 \edef\bxjs@param@basefontsize{\the\@tempdima}
726 \fi
727 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
728 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
729 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
730 \edef\jsc@magscale{\strip@pt\@tempdima}
731 \let\jsBaseFontSize\bxjs@param@basefontsize
732 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
733 %\typeout{\string\jsEngine: \meaning\jsEngine}
734 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
735 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
736 %\typeout{\string\jsc@magscale: \jsc@magscale}
737 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
738 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}

```

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

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

```

739 \ifjsc@mag
740 \let\jsc@mpt\p@
741 \newdimen\jsc@mmm \jsc@mmm=1mm
742 \catcode\?=9 % \p@? read as \p@
743 \else
744 \newdimen\jsc@mpt \jsc@mpt=\jsc@magscale \p@
745 \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
746 \catcode\?=11 \let\p@?\jsc@mpt
747 \fi
748 \chardef\bxjs@qmc=\catcode\?\relax

```

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

```

749 \newdimen\jsZw
750 \jsZw=10\jsc@mpt \jsZw=\jsScale\jsZw
751 \ifbxjs@usezw
752 \providecommand*\zw{\jsZw}
753 \fi

```

\zwspace 全角幅の水平空き。

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

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

```
755 \ifjsc@mag@xreal
756   \RequirePackage{type1cm}
757   \let\jsc@invscale\bxjs@invscale
```

ムニャムニャムニャ……。

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

[2016-11-16] latex.ltx (ltspace.dtx) で定義されている \smallskip の、単位 pt を

`\jsc@empt` 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 796 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 797 %\def\jsc@medskip{\vspace\jsc@medskipamount}
798 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount 799 \newskip\jsc@smallskipamount
\jsc@bigskipamount 800 \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
801 %\newskip\jsc@medskipamount
802 %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
803 %\newskip\jsc@bigskipamount
804 %\jsc@bigskipamoun =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt
```

■pagesize スペシャルの出力

削除。

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

`\@` 欧文といえば、 \LaTeX の `\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。

```
805 \chardef\bxjs@periodchar=`\ .
806 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcodes\bxjs@periodchar}
807 \def\@{\bxjs@SE{}}
```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

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

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

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

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

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

```

808 \def\bxjs@tmpa{\def\setfontsize##1##2##3}
809 \expandafter\bxjs@tmpa\expandafter{%
810   \setfontsize{#1}{#2}{#3}%
811 % 末尾にコードを追加
812   \expandafter\def\expandafter\size@update\expandafter{%
813     \size@update
814     \jsFontSizeChanged}%
815 }
```

`\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。

```

816 \newcommand*\jsFontSizeChanged{%
817   \jsZw=\f@size\p@
818   \jsZw=\jsScale \jsZw
819   \ifdim\parindent>\z@
820     \if@english \parindent=1em
821     \else       \parindent=1\jsZw
822   \fi
823   \fi\relax
824   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
825 \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。

```
826 \ifjsc@mag
827   \let\jsc@setfontsize\@setfontsize
828 \else
829   \def\jsc@setfontsize#1#2#3{%
830     \@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}
831 \fi
```

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

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

```
832 \emergencystretch 3\jsZw
```

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

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

`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対し、

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

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

別行立て数式に入るときに `\narrowbaselines` が呼ばれるが、このコードでは「数式中で `\normalsize` などのサイズ命令 (`\@currsiz` 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。

```
833 \newif\ifnarrowbaselines
834 \if@english
835   \narrowbaselinestrue
836 \fi
837 \def\narrowbaselines{%
838   \narrowbaselinestrue
839   \skip0=\abovedisplayskip
840   \skip2=\abovedisplayshortskip
841   \skip4=\belowdisplayskip
842   \skip6=\belowdisplayshortskip
```

```

843 % 一時的に警告を無効化する
844 \let\bxjs@ltx@nomath\@nomath
845 \let\@nomath\@gobble
846 \@currsize\selectfont
847 \let\@nomath\bxjs@ltx@nomath
848 \abovedisplayskip=\skip0
849 \abovedisplayshortskip=\skip2
850 \belowdisplayskip=\skip4
851 \belowdisplayshortskip=\skip6\relax}
852 \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。

```

853 \def\bxjs@if@narrowbaselines{%
854   \ifnarrowbaselines\expandafter\@firstoftwo
855   \else \expandafter\@secondoftwo
856   \fi
857 }

```

`\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。

```

858 \renewcommand{\normalsize}{%
859   \bxjs@if@narrowbaselines{%
860     \jsc@setfontsize\normalsize\@xpt\@xiipt
861   }{%else
862     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
863   }%

```

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

[2003-02-16] ちょっと変えました。

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

```

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

```

```

865 \abovedisplayskip \z@ \@plus3\p@?
866 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
867 \belowdisplayshortskip \belowdisplayskip

```

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

```
868 \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。

```
869 \normalsize
```

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

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

`\Cwd`

`\Cvs` まず `\Cwd` 等の変数を定義する。

```

\Cht 870 \ifx\Cht\@undefined \newdimen\Cht \fi
\Cdp 871 \ifx\Cdp\@undefined \newdimen\Cdp \fi
872 \ifx\Cwd\@undefined \newdimen\Cwd \fi
873 \ifx\Cvs\@undefined \newdimen\Cvs \fi
874 \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。

```

875 \setlength\Cht{0.88\jsZw}
876 \setlength\Cdp{0.12\jsZw}
877 \setlength\Cwd{1\jsZw}
878 \setlength\Cvs{\baselineskip}
879 \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。

```

880 \newcommand{\small}{%
881   \bxjs@if@narrowbaselines{%
882     %<!kiyou> \jsc@setfontsize\small\@ixpt{11}%
883     %<kiyou> \jsc@setfontsize\small{8.8888}{11}%
884   }{%else
885     %<!kiyou> \jsc@setfontsize\small\@ixpt{13}%
886     %<kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
887   }%
888   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
889   \abovedisplayshortskip \z@ \@plus3\p@?

```

```

890 \belowdisplayskip \abovedisplayskip
891 \belowdisplayshortskip \belowdisplayskip
892 \def\@listi{\leftmargin\leftmargini
893         \topsep \z@
894         \parsep \z@
895         \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。

```

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

```

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

`\LARGE` [2004-11-03] `\HUGE` を追加。

```

\huge 912 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
913 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 914 \if@twocolumn
915 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
916 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
917 \else
918 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
919 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
920 \fi
921 %<!kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xivpt{21}}
922 %<kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE{12.222}{21}}
923 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xvipt{25}}
924 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
925 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
926 \newcommand{\HUGE}{\jsc@setfontsize\HUGE{30}{40}}

```

別行立て数式の中では `\narrowbaselines` にします。和文の行送りのままでは、行列や

場合分けの行送り，連分数の高さなどが不釣り合いに大きくなるためです。

本文中の数式の中では `\narrowbaselines` にしていません。本文中ではなるべく行送りが変わるような大きいものを使わず，行列は `amsmath` の `smallmatrix` 環境を使うのがいいでしょう。

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

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

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

```
928 % \newcommand{\headfont}{\bfseries}
```

```
929 \newcommand{\headfont}{\sffamily}
```

```
930 % \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。

```
931 %<kiyou>\setlength\columnsep{2\Cwd}
```

```
932 %<kiyou>\setlength\columnsep{28truebp}
```

```
933 \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 934 \setlength\lineskip{1\p@?}
```

```
\normallineskip 935 \setlength\normallineskip{1\p@?}
```

```
\normallineskiplimit 936 \setlength\lineskiplimit{1\p@?}
```

```
937 \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，たとえば `\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。

```
938 \renewcommand{\baselinestretch}{}
```

`\parskip` `\parskip` は段落間の追加スペースです。元は 0pt plus 1pt になっていましたが、ここでは
`\parindent` ゼロにしました。`\parindent` は段落の先頭の字下げ幅です。

```
939 \setlength\parskip{0\p@}
940 \if@slide
941   \setlength\parindent{0\p@}
942 \else
943   \setlength\parindent{1\Cwd}
944 \fi
```

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

```
\@highpenalty 945 \@lowpenalty 51
946 \@medpenalty 151
947 \@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。

```
948 % \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。

```
949 % \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。

```
950 \ifjsc@mag
951 \mag=\bxjs@param@mag
952 \fi
953 \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。

```
954 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
955 \def\bxjs@read@b{%
956   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
957   \else \expandafter\bxjs@read@d \fi}
958 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
959 \def\bxjs@read@d#1\@nil{}
960 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```


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

```
961 \edef\bxjs@layout@paper{%
962   \ifjsc@mag truedimen,\fi
963   \if@landscape landscape,\fi
964   \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。

```
965 %<*article|report>
966 \def\bxjs@layout{%
967   headheight=\topskip,footskip=0.03367\paperheight,%
968   headsep=\footskip-\topskip,includeheadfoot,%
969   hscale=0.76,hmarginratio=1:1,%
970   vscale=0.83,vmarginratio=1:1,%
971 }
972 %</article|report>
973 %<*book>
974 \ifbxjs@layout@buggyhmargin      %---
975 % アレ
976 \def\bxjs@layout{%
977   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
978   hmargin=36\jsc@mmm,hmarginratio=1:1,%
979   vscale=0.83,vmarginratio=1:1,%
980 }
981 \else                                %---
982 % 非アレ
983 \def\bxjs@layout{%
984   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
985   hmargin=18\jsc@mmm,%
986   vscale=0.83,vmarginratio=1:1,%
987 }
988 \fi                                  %---
989 %</book>
990 %<*slide>
991 \def\bxjs@layout{%
992   noheadfoot,%
993   hscale=0.9,hmarginratio=1:1,%
994   vscale=0.944,vmarginratio=1:1,%
995 }
996 %</slide>
```

`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```
997 \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。

```
998 %<*book>
999 \newcommand\jsTextWidthLimit{40}
1000 \ifx\bxjs@textwidth@limit\@undefined\else
1001   \let\c@bxjs@canta\@tempcanta
1002   \setcounter{bxjs@canta}{\bxjs@textwidth@limit}
```

```

1003 \long\edef\jsTextWidthLimit{\the\@tempcnta}
1004 \fi
1005 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

1006 \def\bxjs@postproc@layout{%
1007 % ドライバ再設定
1008 \ifx\bxjs@geometry@driver\relax\else
1009 \let\Gm@driver\bxjs@geometry@driver
1010 \fi
1011 % textwidth 調整
1012 \ifbxjs@whole@zw@lines
1013 \@tempdimb=\textwidth
1014 \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
1015 \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
1016 \advance\@tempdimb-\textwidth
1017 \advance\oddsidemargin 0.5\@tempdimb
1018 \advance\evensidemargin 0.5\@tempdimb
1019 \fi
1020 \fullwidth=\textwidth
1021 %<*book>
1022 \ifdim\textwidth>\jsTextWidthLimit\Cwd
1023 \textwidth=\jsTextWidthLimit\Cwd
1024 \addtolength\evensidemargin{\fullwidth-\textwidth}
1025 \fi
1026 %</book>
1027 % textheight 調整
1028 \@tempdimb=\textheight
1029 \advance\textheight-\topskip
1030 \divide\textheight\baselineskip \multiply\textheight\baselineskip
1031 \advance\textheight\topskip
1032 \advance\@tempdimb-\textheight
1033 \advance\topmargin0.5\@tempdimb
1034 % headheight 調整
1035 \@tempdima=\topskip
1036 \advance\headheight\@tempdima
1037 \advance\topmargin-\@tempdima
1038 % marginpar 設定
1039 \setlength\marginparsep{\columnsep}
1040 \setlength\marginparpush{\baselineskip}
1041 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1042 -\textwidth-10\jsc@mmm-\marginparsep}
1043 \ifbxjs@whole@zw@lines
1044 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1045 \fi
1046 % 連動する変数
1047 \maxdepth=.5\topskip
1048 \stockwidth=\paperwidth
1049 \stockheight=\paperheight

```

1050 }

`\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。

```
1051 \edef\jsGeometryOptions{%
1052   \bxjs@layout@paper,\bxjs@layout}
```

■ geometry パッケージ読込

geometry オプションの値に応じて分岐する。

まずは `geometry=class` の場合。

```
1053 \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。

```
1054 \@onlypreamble\bxjs@revert
1055 \let\bxjs@revert\empty
1056 \edef\bxjs@tmpa{\the\ht\strutbox}
1057 \ht\strutbox=10\p@?
1058 \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。

```
1059 \ifbxjs@papersize
1060   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1061     \PassOptionsToPackage{dvipdfm}{geometry}
1062   \else\ifx\bxjs@driver@given\bxjs@driver@@dvimode
1063     \PassOptionsToPackage{dvipdfm}{geometry}
1064   \fi\fi
1065   \let\bxPapersizeSpecialDone=t
1066 \else
1067   \PassOptionsToPackage{driver=none}{geometry}
1068 \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。

```
1069 \edef\bxjs@nxt{%
1070   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1071 \AtBeginDocument{\bxjs@pre@geometry@hook}
1072 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1073 \bxjs@nxt \bxjs@revert
1074 \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。

```
1075 \let\bxjs@geometry@driver\Gm@driver
```

```
1076 \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。

```
1077 \@onlypreamble\bxjs@pre@geometry@hook
```

```
1078 \def\bxjs@pre@geometry@hook{%
```

```
1079   \@ifpackageloaded{geometry}{%
```

```
1080     \@ifpackagelater{geometry}{2010/02/12}{}{%else
```

```
1081       \PackageWarningNoLine\bxjs@clsname
```

```
1082       {The 'geometry' package installed\MessageBreak
```

```
1083       is too old (< v5.0)}%
```

```
1084       \if x\jsEngine \ifnum\mag=\@m\else
```

```
1085         \def\bxjs@Gm@driver{pdftex}
```

```
1086         \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
```

```
1087         \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
```

```
1088         \fi\fi
```

```
1089     }%
```

```
1090     \ifjsWithpTeXng
```

```
1091       \ifx\Gm@driver\@empty
```

```
1092         \def\Gm@driver{pdftex}%
```

```
1093       \fi
```

```
1094     \fi
```

```
1095   }{}}
```

`\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。

```
1096 \def\setpagelayout{\@ifstar
```

```
1097   {\bxjs@reset@layout}{\bxjs@modify@layout}}
```

```
1098 \def\bxjs@modify@layout#1{%
```

```
1099   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
```

```
1100   \bxjs@nxt\bxjs@postproc@layout}
```

```
1101 \def\bxjs@reset@layout#1{%
```

```
1102   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
```

```
1103   \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。

```
1104 \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。

```
1105 \AtBeginDocument{\bxjs@check@page@layout}
1106 \@onlypreamble\bxjs@check@page@layout
1107 \def\bxjs@check@page@layout{%
1108   \ifdim\textwidth=.5\maxdimen
1109     \ClassError\bxjs@clsname
1110       {Page layout is not properly set}%
1111       {\@ehd}
1112   \fi}
1113 \def\jsUseMinimalPageLayout{%
1114   \setlength{\textwidth}{6.5in}%
1115   \setlength{\textheight}{8in}}

\setpagelayout はとりあえず無効にしておく。

1116 \let\bxjs@geometry@driver\relax
1117 \def\setpagelayout{\@ifstar
1118   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1119 \def\bxjs@pagelayout@a#1{%
1120   \ClassError\bxjs@clsname
1121     {Command '\string\setpagelayout' is not supported,\MessageBreak
1122     because 'geometry' value is not 'class'}\@eha}

geometry=user の場合の処理はここで終わり。
1123 \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用ことにする。

```
1124 \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。

```
1125 \setlength{\skip\footins}{16\p@? \@plus 5\p@? \@minus 2\p@?}
```

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

`\c@topnumber` `topnumber` カウンタは本文ページ上部のフロートの最大数です。
[2003-08-23] ちょっと増やしました。

```
1126 \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。

```
1127 \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。

```
1128 \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。

```
1129 \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。

```
1130 \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。

```
1131 \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。

```
1132 \renewcommand{\floatpagefraction}{.8}
```

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

```
1133 \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。

```
1134 \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。

```
1135 \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。

```
1136 \setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1137 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1138 \setlength\intextsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
```

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

```
\dbltextfloatsep 1139 \setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}
1140 \setlength\dbltextfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
```

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

```
\@fpbot 1141 \setlength\@fptop{0\p@? \@plus 1fil}
1142 \setlength\@fpsep{8\p@? \@plus 2fil}
1143 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpptop` 段抜きフロートについての値です。

```
\@dblfpsep 1144 \setlength\@dblfpptop{0\p@? \@plus 1fil}
1145 \setlength\@dblfpsep{8\p@? \@plus 2fil}
\@dblfpbot 1146 \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 pL_AT_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。

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

```
1147 %\def\pltx@cleartorightpage{\clearpage\if@twoside
1148 % \ifodd\c@page
1149 % \iftdir
1150 % \hbox{}\thispagestyle{empty}\newpage
1151 % \if@twocolumn\hbox{}\newpage\fi
1152 % \fi
1153 % \else
1154 % \ifydir
1155 % \hbox{}\thispagestyle{empty}\newpage
1156 % \if@twocolumn\hbox{}\newpage\fi
1157 % \fi
1158 % \fi\fi}
1159 %\def\pltx@cleartoleftpage{\clearpage\if@twoside
```

```

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

```

1181 \let\pltx@cleartorightpage\pltx@cleartooddpage
1182 \let\pltx@cleartoleftpage\pltx@cleartoevenpage

```

`\cleardoublepage` [2017-02-24] コミュニティ版 p \LaTeX 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。

```

1183 %<*book|report>
1184 \if@openleft
1185 \let\cleardoublepage\pltx@cleartoleftpage
1186 \else\if@openright
1187 \let\cleardoublepage\pltx@cleartorightpage
1188 \fi\fi
1189 %</book|report>

```

7 ページスタイル

ページスタイルとして、 \LaTeX 2 ϵ (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 \LaTeX 2 ϵ 本体で定義されています。

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

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

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

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

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

```
1190 % \def\ps@empty{%
1191 %   \let\mkboth\gobbletwo
1192 %   \let\@oddhead\@empty
1193 %   \let\@oddfoot\@empty
1194 %   \let\@evenhead\@empty
1195 %   \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。

```
1196 \def\ps@plainfoot{%
1197   \let\mkboth\gobbletwo
1198   \let\@oddhead\@empty
1199   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1200   \let\@evenhead\@empty
1201   \let\@evenfoot\@oddfoot}
1202 \def\ps@plainhead{%
1203   \let\mkboth\gobbletwo
1204   \let\@oddfoot\@empty
1205   \let\@evenfoot\@empty
1206   \def\@evenhead{%
1207     \ifmparswitch \hss \fi
1208     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1209     \ifmparswitch\else \hss \fi}%
1210   \def\@oddhead{%
1211     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1212 %<book>\let\ps@plain\ps@plainhead
```

```

1213 %<!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。

1214 %<*article|slide>
1215 \if@twoside
1216   \def\ps@headings{%
1217     \let\@oddfoot\@empty
1218     \let\@evenfoot\@empty
1219     \def\@evenhead{\if@mparswitch \hss \fi
1220       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1221     \if@mparswitch\else \hss \fi}%
1222     \def\@oddhead{%
1223       \underline{%
1224         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1225     \let\@mkboth\markboth
1226     \def\sectionmark##1{\markboth{%
1227       \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1228       ##1}{}}%
1229     \def\subsectionmark##1{\markright{%
1230       \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
1231       ##1}}%
1232   }
1233 \else % if not twoside
1234   \def\ps@headings{%
1235     \let\@oddfoot\@empty
1236     \def\@oddhead{%
1237       \underline{%
1238         \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1239     \let\@mkboth\markboth
1240     \def\sectionmark##1{\markright{%
1241       \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1242       ##1}}%
1243   \fi
1244 %</article|slide>

```

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

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

```

1245 %<*book|report>
1246 \def\bxjs@maybe@autoxspacing{%
1247   \ifx\autoxspacing\undefined\else \autoxspacing \fi}
1248 \newif\if@omit@number
1249 \def\ps@headings{%

```

```

1250 \let\@oddfoot\@empty
1251 \let\@evenfoot\@empty
1252 \def\@evenhead{%
1253   \if@mparswitch \hss \fi
1254   \underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1255     \textbf{\thepage}\hfil\leftmark}}}%
1256   \if@mparswitch\else \hss \fi}%
1257 \def\@oddhead{\underline{\hbox to \fullwidth{\bxjs@maybe@autoxspacing
1258   {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1259 \let\@mkboth\markboth
1260 \def\chaptermark##1{\markboth{%
1261   \ifnum \c@secnumdepth >\m@ne
1262     \if@mainmatter
1263       \if@omit@number\else
1264         \@chapapp\thechapter\@chappos\hskip1\jsZw
1265       \fi
1266     \fi
1267   \fi
1268   ##1}{}}%
1269 \def\sectionmark##1{\markright{%
1270   \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1271   ##1}}}%
1272 %</book|report>

    最後は学会誌の場合です。
1273 %<*jspf>
1274 \def\ps@headings{%
1275   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1276   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1277   \def\@oddhead{\normalfont\hfil \@title \hfil}
1278   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1279 %</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。

```

1280 \def\ps@myheadings{%
1281   \let\@oddfoot\@empty\let\@evenfoot\@empty
1282   \def\@evenhead{%
1283     \if@mparswitch \hss \fi%
1284     \hbox to \fullwidth{\thepage\hfil\leftmark}}%
1285     \if@mparswitch\else \hss \fi}%
1286   \def\@oddhead{%
1287     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1288   \let\@mkboth\@gobbletwo
1289 %<book|report> \let\chaptermark\@gobble
1290 \let\sectionmark\@gobble
1291 %<!book&!report> \let\subsectionmark\@gobble
1292 }

```

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 1293 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1294 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1295 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1296 % \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。

```
1297 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1298 %\let\bxjs@subtitle\undefined

      \title にフックを入れる。

1299 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1300 \AtBeginDocument{\bxjs@decl@subtitle}
1301 \def\bxjs@decl@subtitle{%
1302   \global\let\bxjs@decl@subtitle\relax
1303   \ifx\subtitle\undefined
1304     \global\let\subtitle\jsSubtitle
1305   \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。

```
1306 \def\bxjs@annihilate@subtitle{%
1307   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1308   \global\let\jsSubtitle\relax}
```

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

```
\eauthor 1309 %<*jspf>
\keywords 1310 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
          1311 \newcommand*{\eauthor}[1]{\gdef\@eauthor{#1}}
          1312 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
          1313 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
          1314 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
          1315 %</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。

```

1316 \def\plainifnotempty{%
1317   \ifx \@oddhead \@empty
1318     \ifx \@oddfoot \@empty
1319       \else
1320         \thispagestyle{plainfoot}%
1321       \fi
1322     \else
1323       \thispagestyle{plainhead}%
1324     \fi}

```

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

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

```

1325 %<article|book|report|slide>
1326 \if@titlepage
1327   \newcommand{\maketitle}{%
1328     \begin{titlepage}%
1329       \let\footnotesize\small
1330       \let\footnoterule\relax
1331       \let\footnote\thanks
1332       \null\vfil
1333       \if@slide
1334         {\footnotesize \@date}%
1335         \begin{center}
1336           \mbox{} \[\!1\jsZw]
1337           \large
1338           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1339           \jsc@smallskip
1340           \@title
1341           \ifx\bxjs@subtitle\@undefined\else
1342             \par\vskip\z@
1343             {\small \bxjs@subtitle\par}
1344           \fi
1345           \jsc@smallskip
1346           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1347           \vfill
1348           {\small \@author}%
1349         \end{center}
1350       \else
1351         \vskip 60\p@?

```

```

1352 \begin{center}%
1353 {\LARGE \@title \par}%
1354 \ifx\bxjs@subtitle\@undefined\else
1355 \vskip5\p@?
1356 {\normalsize \bxjs@subtitle\par}
1357 \fi
1358 \vskip 3em%
1359 {\large
1360 \lineskip .75em
1361 \begin{tabular}[t]{c}%
1362 \author
1363 \end{tabular}\par}%
1364 \vskip 1.5em
1365 {\large \@date \par}%
1366 \end{center}%
1367 \fi
1368 \par
1369 \@thanks\vfil\null
1370 \end{titlepage}%
1371 \setcounter{footnote}{0}%
1372 \global\let\thanks\relax
1373 \global\let\maketitle\relax
1374 \global\let\@thanks\@empty
1375 \global\let\@author\@empty
1376 \global\let\@date\@empty
1377 \global\let\@title\@empty
1378 \global\let\title\relax
1379 \global\let\author\relax
1380 \global\let\date\relax
1381 \global\let\and\relax
1382 \bxjs@annihilate@subtitle
1383 }%
1384 \else
1385 \newcommand{\maketitle}{\par
1386 \begin{group
1387 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1388 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1389 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1390 \parindent 1\jsZw\noindent
1391 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1392 \if@twocolumn
1393 \ifnum \col@number=\@ne
1394 \@maketitle
1395 \else
1396 \twocolumn[\@maketitle]%
1397 \fi
1398 \else
1399 \newpage
1400 \global\@topnum\z@ % Prevents figures from going at top of page.

```

```

1401     \@maketitle
1402     \fi
1403     \plainifnotempty
1404     \@thanks
1405     \endgroup
1406     \setcounter{footnote}{0}%
1407     \global\let\thanks\relax
1408     \global\let\maketitle\relax
1409     \global\let\@thanks\@empty
1410     \global\let\@author\@empty
1411     \global\let\@date\@empty
1412     \global\let\@title\@empty
1413     \global\let\title\relax
1414     \global\let\author\relax
1415     \global\let\date\relax
1416     \global\let\and\relax
1417     \bxjs@annihilate@subtitle
1418 }

```

\@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。

```

1419 \def\@maketitle{%
1420     \newpage\null
1421     \vskip 2em
1422     \begin{center}%
1423         \let\footnote\thanks
1424         {\LARGE \@title \par}%
1425         \ifx\bxjs@subtitle\@undefined\else
1426             \vskip3\p@?
1427             {\normalsize \bxjs@subtitle\par}
1428         \fi
1429         \vskip 1.5em
1430         {\large
1431             \lineskip .5em
1432             \begin{tabular}[t]{c}%
1433                 \@author
1434             \end{tabular}\par}%
1435         \vskip 1em
1436         {\large \@date}%
1437     \end{center}%
1438     \par\vskip 1.5em
1439 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1440 }
1441 \fi
1442 %</article|book|report|slide>
1443 %<*jspf>
1444 \newcommand{\maketitle}{\par
1445     \begingroup
1446     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1447     \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%

```

```

1448 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1449 \parindent 1\jsZw\noindent
1450 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1451 \twocolumn[\@maketitle]%
1452 \plainifnotempty
1453 \@thanks
1454 \endgroup
1455 \setcounter{footnote}{0}%
1456 \global\let\thanks\relax
1457 \global\let\maketitle\relax
1458 \global\let\@thanks\@empty
1459 \global\let\@author\@empty
1460 \global\let\@date\@empty
1461 % \global\let\@title\@empty % \@title は柱に使う
1462 \global\let\title\relax
1463 \global\let\author\relax
1464 \global\let\date\relax
1465 \global\let\and\relax
1466 \ifx\authors@mail\@undefined\else{%
1467 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1468 \footnotetext[0]{\itshape\authors@mail}%
1469 }\fi
1470 \global\let\authors@mail\@undefined}
1471 \def\@maketitle{%
1472 \newpage\null
1473 \vskip 6em % used to be 2em
1474 \begin{center}
1475 \let\footnote\thanks
1476 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1477 \lineskip .5em
1478 \ifx\@author\@undefined\else
1479 \vskip 1em
1480 \begin{tabular}[t]{c}%
1481 \@author
1482 \end{tabular}\par
1483 \fi
1484 \ifx\@etitle\@undefined\else
1485 \vskip 1em
1486 {\large \@etitle \par}%
1487 \fi
1488 \ifx\@eauthor\@undefined\else
1489 \vskip 1em
1490 \begin{tabular}[t]{c}%
1491 \@eauthor
1492 \end{tabular}\par
1493 \fi
1494 \vskip 1em
1495 \@date
1496 \end{center}

```



```

1497 \vskip 1.5em
1498 \centerline{\box\@abstractbox}
1499 \ifx\@keywords\@undefined\else
1500     \vskip 1.5em
1501     \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1502 \fi
1503 \vskip 1.5em}
1504 %</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anta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。

```

1505 \def\@startsection#1#2#3#4#5#6{%
1506     \if@noskipsec \leavevmode \fi
1507     \par
1508 % 見出し上の空きを \@tempskipa にセットする
1509     \@tempskipa #4\relax
1510 % \@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
1511     \if@english \@afterindentfalse \else \@afterindenttrue \fi
1512 % 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
1513     \ifdim \@tempskipa <\z@

```

```

1514 \@tempskipa -\@tempskipa \@afterindentfalse
1515 \fi
1516 \if@nobreak
1517 % \everypar{\everyparhook}% これは間違い
1518 \everypar{}%
1519 \else
1520 \addpenalty\@secpenalty
1521 % 次の行は削除
1522 % \addvspace\@tempskipa
1523 % 次の \noindent まで追加
1524 \ifdim \@tempskipa >\z@
1525 \if@slide\else
1526 \null
1527 \vspace*{-\baselineskip}%
1528 \fi
1529 \vskip\@tempskipa
1530 \fi
1531 \fi
1532 \noindent
1533 % 追加終わり
1534 \@ifstar
1535 {\@ssect{#3}{#4}{#5}{#6}}%
1536 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

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

1537 \def\@sect#1#2#3#4#5#6[#7]#8{%
1538 \ifnum #2>\c@secnumdepth
1539 \let\@svsec\@empty
1540 \else
1541 \refstepcounter{#1}%
1542 \protected@edef\@svsec{\@secntformat{#1}\relax}%
1543 \fi
1544 % 見出し後の空きを \@tempskipa にセット
1545 \@tempskipa #5\relax
1546 % 条件判断の順序を入れ替えました
1547 \ifdim \@tempskipa<\z@
1548 \def\@svsechd{%
1549 #6{\hskip #3\relax
1550 \@svsec #8}%
1551 \csname #1mark\endcsname{#7}}%
1552 \addcontentsline{toc}{#1}{%
1553 \ifnum #2>\c@secnumdepth \else
1554 \protect\numberline{\csname the#1\endcsname}%
1555 \fi
1556 #7}}% 目次にフルネームを載せるなら #8
1557 \else
1558 \begingroup
1559 \interlinepenalty \@M % 下から移動

```

```

1560      #6{%
1561          \@hangfrom{\hskip #3\relax\@svsec}%
1562 %      \interlinepenalty \@M % 上に移動
1563      #8\@par}%
1564  \endgroup
1565  \csname #1mark\endcsname{#7}%
1566  \addcontentsline{toc}{#1}{%
1567      \ifnum #2>\c@secnumdepth \else
1568          \protect\numberline{\csname the#1\endcsname}%
1569      \fi
1570      #7}% 目次にフルネームを載せるならここは #8
1571  \fi
1572  \@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。

```

1573 \def\@xsect#1{%
1574 % 見出しの後ろの空きを \@tempskipa にセット
1575     \@tempskipa #1\relax
1576 % 条件判断の順序を変えました
1577     \ifdim \@tempskipa<\z@
1578         \@nbreakfalse
1579         \global\@noskipsectrue
1580         \everypar{%
1581             \if@noskipsec
1582                 \global\@noskipsecfalse
1583                 {\setbox\z@\lastbox}%
1584                 \clubpenalty\@M
1585                 \begingroup \@svsechd \endgroup
1586                 \unskip
1587                 \@tempskipa #1\relax
1588                 \hskip -\@tempskipa
1589                 \bxjs<ltj<inhibitglue
1590             \else
1591                 \clubpenalty \@clubpenalty
1592                 \everypar{\everyparhook}%
1593                 \fi\everyparhook}%
1594     \else
1595         \par \nbreak
1596         \vskip \@tempskipa
1597         \@afterheading
1598     \fi
1599     \if@slide
1600         {\vskip\if@twocolumn-5\jsc@mpt\else-6\jsc@mpt\fi
1601         \maybeblue\hrule height0\jsc@mpt depth1\jsc@mpt
1602         \vskip\if@twocolumn 4\jsc@mpt\else 7\jsc@mpt\fi\relax}%

```

```

1603 \fi
1604 \par % 2000-12-18
1605 \ignorespaces}
1606 \def\@ssect#1#2#3#4#5{%
1607 \@tempskipa #3\relax
1608 \ifdim \@tempskipa<\z@
1609 \def\@svsechd{#4{\hskip #1\relax #5}}%
1610 \else
1611 \begingroup
1612 #4{%
1613 \@hangfrom{\hskip #1}%
1614 \interlinepenalty \@M #5\@@par}%
1615 \endgroup
1616 \fi
1617 \@xsect{#3}}

```

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

```

1618 \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 1619 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1620 % \newcommand*\sectionmark[1]{}
1621 % \newcommand*\subsectionmark[1]{}
\paragraphmark 1622 % \newcommand*\subsubsectionmark[1]{}
\subparagraphmark 1623 % \newcommand*\paragraphmark[1]{}
1624 % \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。

```

1625 %<!book&!report>\setcounter{secnumdepth}{3}
1626 %<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 1627 \newcounter{part}
1628 %<book|report>\newcounter{chapter}
\c@subsubsection 1629 %<book|report>\newcounter{section}[chapter]
\c@paragraph 1630 %<!book&!report>\newcounter{section}
\c@subparagraph 1631 \newcounter{subsection}[section]
1632 \newcounter{subsubsection}[subsection]
1633 \newcounter{paragraph}[subsubsection]
1634 \newcounter{subparagraph}[paragraph]

```

<code>\thepart</code>	カウンタの値を出力する命令 <code>\the</code> 何々 を定義します。
<code>\thechapter</code>	カウンタを出力するコマンドには次のものがあります。
<code>\thesection</code>	<code>\arabic{COUNTER}</code> 1, 2, 3, ...
<code>\thesubsection</code>	<code>\roman{COUNTER}</code> i, ii, iii, ...
<code>\thesubsubsection</code>	<code>\Roman{COUNTER}</code> I, II, III, ...
<code>\theparagraph</code>	<code>\alph{COUNTER}</code> a, b, c, ...
<code>\thesubparagraph</code>	<code>\Alph{COUNTER}</code> A, B, C, ...
	<code>\kansuji{COUNTER}</code> 一, 二, 三, ...

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

1635 \renewcommand{\thepart}{\@Roman\c@part}
1636 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1637 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1638 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1639 %<*book|report>
1640 \renewcommand{\thechapter}{\@arabic\c@chapter}
1641 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1642 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1643 %</book|report>
1644 \renewcommand{\thesubsubsection}{\%
1645   \thesubsection.\@arabic\c@subsubsection}
1646 \renewcommand{\theparagraph}{\%
1647   \thesubsubsection.\@arabic\c@paragraph}
1648 \renewcommand{\thesubparagraph}{\%
1649   \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。

```

1650 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1651 %<book|report>\newcommand{\@chappos}{\postchaptername}

```

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

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

[2017-03-05] `\frontmatter` と `\mainmatter` の 2 つの命令は, 改丁または改ページした後で `\pagenumbering{...}` でノンブルを 1 にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが, これではノンブルをリセットする際に偶奇逆転が起こる場合がありました。openany かどうかに関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, L^AT_EX 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 pL^AT_EX の標準クラス 2017/03/05 も参照)。

```

1652 %<*book|report>
1653 \newcommand\frontmatter{%

```

```

1654 \pltx@cleartooddpage
1655 \@mainmatterfalse
1656 \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。

```

1657 \newcommand\mainmatter{%
1658 \pltx@cleartooddpage
1659 \@mainmattertrue
1660 \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。

```

1661 \newcommand\backmatter{%
1662 \if@openleft
1663 \cleardoublepage
1664 \else\if@openright
1665 \cleardoublepage
1666 \else
1667 \clearpage
1668 \fi\fi
1669 \@mainmatterfalse}
1670 %</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。

```

1671 %<*&book&!report>
1672 \newcommand\part{%
1673 \if@noskipsec \leavevmode \fi
1674 \par
1675 \addvspace{4ex}%
1676 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1677 \secdef\@part\@spart}
1678 %</!book&!report>

```

`book` および `report` クラスの場合は、少し複雑です。

```

1679 %<*&book|report>

```

```

1680 \newcommand\part{%
1681   \if@openleft
1682     \cleardoublepage
1683   \else\if@openright
1684     \cleardoublepage
1685   \else
1686     \clearpage
1687   \fi\fi
1688   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1689   \if@twocolumn
1690     \onecolumn
1691     \@restonecoltrue
1692   \else
1693     \@restonecolfalse
1694   \fi
1695   \null\vfil
1696   \secdef\@part\@spart}
1697 %</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。

```

1698 %<!*book&!report>
1699 \def\@part[#1]#2{%
1700   \ifnum \c@secnumdepth >\m@ne
1701     \refstepcounter{part}%
1702     \addcontentsline{toc}{part}{%
1703       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1704   \else
1705     \addcontentsline{toc}{part}{#1}%
1706   \fi
1707   \markboth{}{}%
1708   {\parindent\z@
1709     \raggedright
1710     \interlinepenalty \@M
1711     \normalfont
1712     \ifnum \c@secnumdepth >\m@ne
1713       \Large\headfont\prepartname\thepart\postpartname
1714       \par\nobreak
1715     \fi
1716     \huge \headfont #2%
1717     \markboth{}{}\par}%
1718   \nobreak
1719   \vskip 3ex
1720   \@afterheading}
1721 %</!*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。

```

1722 %<*book|report>

```

```

1723 \def\@part[#1]#2{%
1724   \ifnum \c@secnumdepth >-2\relax
1725     \refstepcounter{part}%
1726     \addcontentsline{toc}{part}{%
1727       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1728   \else
1729     \addcontentsline{toc}{part}{#1}%
1730   \fi
1731   \markboth{}{}%
1732   {\centering
1733     \interlinepenalty \@M
1734     \normalfont
1735     \ifnum \c@secnumdepth >-2\relax
1736       \huge\headfont \prepartname\thepart\postpartname
1737       \par\vskip20\p@?
1738     \fi
1739     \Huge \headfont #2\par}%
1740   \@endpart}
1741 %</book|report>

```

`\@spart` 番号を付けない部です。

```

1742 %<!*book&!report>
1743 \def\@spart#1{%
1744   \parindent \z@ \raggedright
1745   \interlinepenalty \@M
1746   \normalfont
1747   \huge \headfont #1\par}%
1748   \nobreak
1749   \vskip 3ex
1750   \@afterheading}
1751 %</!*book&!report>
1752 %<*book|report>
1753 \def\@spart#1{%
1754   \centering
1755   \interlinepenalty \@M
1756   \normalfont
1757   \Huge \headfont #1\par}%
1758   \@endpart}
1759 %</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 \LaTeX では `classes.dtx` v1.4b (2000/05/19) で修正されています。

```

1760 %<*book|report>
1761 \def\@endpart{\vfil\newpage
1762   \if@twoside

```



```

1763 \if@openleft %% added (2017/02/24)
1764 \null\thispagestyle{empty}\newpage
1765 \else\if@openright %% added (2016/12/13)
1766 \null\thispagestyle{empty}\newpage
1767 \fi\fi %% added (2016/12/13, 2017/02/24)
1768 \fi
1769 \if@restonecol
1770 \twocolumn
1771 \fi}
1772 %</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。

```

1773 %<*book|report>
1774 \newcommand{\chapter}{%
1775 \if@openleft\cleardoublepage\else
1776 \if@openright\cleardoublepage\else\clearpage\fi\fi
1777 \plainifnotempty % 元: \thispagestyle{plain}
1778 \global\@topnum\z@
1779 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1780 \secdef
1781 {\@omit@numberfalse\@chapter}%
1782 {\@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。

```

1783 \def\@chapter[#1]#2{%
1784 \ifnum \c@secnumdepth >\m@ne
1785 \if@mainmatter
1786 \refstepcounter{chapter}%
1787 \typeout{\@chapapp\thechapter\@chappos}%
1788 \addcontentsline{toc}{chapter}%
1789 {\protect\numberline
1790 % {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
1791 {\@chapapp\thechapter\@chappos}%
1792 #1}%
1793 \else\addcontentsline{toc}{chapter}{#1}\fi
1794 \else
1795 \addcontentsline{toc}{chapter}{#1}%
1796 \fi
1797 \chaptermark{#1}%
1798 \addtocontents{lof}{\protect\addvspace{10\jsc@empt}}%
1799 \addtocontents{lot}{\protect\addvspace{10\jsc@empt}}%
1800 \if@twocolumn
1801 \@topnewpage[\@makechapterhead{#2}]%
1802 \else

```

```

1803 \makechapterhead{#2}%
1804 \afterheading
1805 \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。

```

1806 \def\makechapterhead#1{%
1807 \vspace*{2\Cvs}% 欧文は 50pt
1808 {\parindent \z@ \raggedright \normalfont
1809 \ifnum \c@secnumdepth >\m@ne
1810 \if@mainmatter
1811 \huge\headfont \@chapapp\thechapter\@chappos
1812 \par\nobreak
1813 \vskip \Cvs % 欧文は 20pt
1814 \fi
1815 \fi
1816 \interlinepenalty\@M
1817 \Huge \headfont #1\par\nobreak
1818 \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。

```

1819 \def\@schapter#1{%
1820 \chaptermark{#1}%
1821 \if@twocolumn
1822 \@topnewpage[\@makeschapterhead{#1}]%
1823 \else
1824 \@makeschapterhead{#1}\@afterheading
1825 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

1826 \def\@makeschapterhead#1{%
1827 \vspace*{2\Cvs}% 欧文は 50pt
1828 {\parindent \z@ \raggedright
1829 \normalfont
1830 \interlinepenalty\@M
1831 \Huge \headfont #1\par\nobreak
1832 \vskip 3\Cvs}} % 欧文は 40pt
1833 %</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。

```

1834 \if@twocolumn
1835 \newcommand{\section}{%
1836 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
1837 \@startsection{section}{1}{\z@}%
1838 %<!kiyou> {0.6\Cvs}{0.4\Cvs}%

```

```

1839 %<kiyou>    {\Cvs}{0.5\Cvs}%
1840 %    {\normalfont\large\headfont\@secapp}}
1841    {\normalfont\large\headfont\raggedright}}
1842 \else
1843    \newcommand{\section}{%
1844        \if@slide\clearpage\fi
1845        \@startsection{section}{1}{\z@}%
1846        {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1847        {.5\Cvs \@plus.3\Cdp}% 後アキ
1848 %    {\normalfont\Large\headfont\@secapp}}
1849    {\normalfont\Large\headfont\raggedright}}
1850 \fi

```

`\subsection` 同上です。

```

1851 \if@twocolumn
1852    \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1853        {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1854        {\normalfont\normalsize\headfont}}
1855 \else
1856    \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1857        {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1858        {.5\Cvs \@plus.3\Cdp}% 後アキ
1859        {\normalfont\large\headfont}}
1860 \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)。

```

1861 \if@twocolumn
1862    \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1863        {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1864        {\normalfont\normalsize\headfont}}
1865 \else
1866    \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1867        {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1868        {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
1869        {\normalfont\normalsize\headfont}}
1870 \fi

```

`\paragraph` 見出しの後ろで改行されません。

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

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

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

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

段落のマーク (■) が必ず和文フォントで出力されるようにする。

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

```
1871 \newcommand\jsParagraphMark{\jsJaChar{■}}
1872 \ifx\bxjs@paragraph@mark\@empty
1873   \let\jsParagraphMark\@empty
1874 \else\ifx\bxjs@paragraph@mark\@undefined\else
1875   \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
1876 \fi\fi
1877 \let\jsJaChar\@empty
1878 \if@twocolumn
1879   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1880     {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1881     {\normalfont\normalsize\headfont}}
1882 %!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1883 \else
1884   \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1885     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1886     {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1887     {\normalfont\normalsize\headfont}}
1888 %!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1889 \fi
```

`\subparagraph` 見出しの後ろで改行されません。

```
1890 \if@twocolumn
1891   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1892     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1893     {\normalfont\normalsize\headfont}}
1894 \else
1895   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1896     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1897     {\normalfont\normalsize\headfont}}
1898 \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。

```
1899 \if@slide
```

```

1900 \setlength\leftmargini{1\jsZw}
1901 \else
1902 \if@twocolumn
1903 \setlength\leftmargini{2\jsZw}
1904 \else
1905 \setlength\leftmargini{3\jsZw}
1906 \fi
1907 \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 1908 \if@slide
\leftmarginv 1909 \setlength\leftmarginii {1\jsZw}
1910 \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 1911 \setlength\leftmarginiv {1\jsZw}
1912 \setlength\leftmarginv {1\jsZw}
1913 \setlength\leftmarginvi {1\jsZw}
1914 \else
1915 \setlength\leftmarginii {2\jsZw}
1916 \setlength\leftmarginiii{2\jsZw}
1917 \setlength\leftmarginiv {2\jsZw}
1918 \setlength\leftmarginv {1\jsZw}
1919 \setlength\leftmarginvi {1\jsZw}
1920 \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。

```

1921 \setlength \labelsep {0.5\jsZw} % .5em
1922 \setlength \labelwidth{\leftmargini}
1923 \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。

```

1924 \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 1925 \@beginparpenalty -\@lowpenalty
\@itempenalty 1926 \@endparpenalty -\@lowpenalty
1927 \@itempenalty -\@lowpenalty

```

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

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

```
1928 \def\@listi{\leftmargin\leftmarginii
1929   \parsep \z@
1930   \topsep 0.5\baselineskip
1931   \itemsep \z@ \relax}
1932 \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）。

```
1933 \@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 1934 \def\@listii{\leftmargin\leftmarginii
\@listiv 1935   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1936   \topsep \z@
\@listv 1937   \parsep \z@
\@listvi 1938   \itemsep\parsep}
1939 \def\@listiii{\leftmargin\leftmarginiii
1940   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1941   \topsep \z@
1942   \parsep \z@
1943   \itemsep\parsep}
1944 \def\@listiv {\leftmargin\leftmarginiv
1945   \labelwidth\leftmarginiv
1946   \advance\labelwidth-\labelsep}
1947 \def\@listv {\leftmargin\leftmarginv
1948   \labelwidth\leftmarginv
1949   \advance\labelwidth-\labelsep}
1950 \def\@listvi {\leftmargin\leftmarginvi
1951   \labelwidth\leftmarginvi
1952   \advance\labelwidth-\labelsep}
```

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

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

```
1953 \renewcommand{\theenumi}{\@arabic\c@enumi}
1954 \renewcommand{\theenumii}{\@alph\c@enumii}
1955 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1956 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

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

```
\labelenumiv
```

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

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

```
1957 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1958 \newcommand*{\jsInJaParen}[1]{%
1959   \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1960 \newcommand{\labelenumi}{\theenumi.}
1961 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1962 \newcommand{\labelenumiii}{\theenumiii.}
1963 \newcommand{\labelenumiv}{\theenumiv.}
```

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

```
\p@enumiv 1964 \renewcommand{\p@enumii}{\theenumi}
1965 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1966 \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 1967 \newcommand\labelitemi{\textbullet}
\labelitemiii 1968 \newcommand\labelitemii{\normalfont\bfseries \textendash}
\labelitemiv 1969 \newcommand\labelitemiii{\textasteriskcentered}
1970 \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。

```
1971 \newenvironment{description}{%
1972   \list{}{%
1973     \labelwidth=\leftmargin
1974     \labelsep=1\jsZw
1975     \advance \labelwidth by -\labelsep
1976     \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。

```
1977 \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。

```
1978 %<*book|report>
1979 \newenvironment{chapterabstract}{%
1980   \begin{list}{}{%
1981     \listparindent=1\jsZw
1982     \itemindent=\listparindent
1983     \rightmargin=\z@
1984     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
1985 %</book|report>
```

“普通の” abstract 環境の定義。

```
1986 %<*article|report|slide>
1987 \newbox\@abstractbox
1988 \if@titlepage
1989   \newenvironment{abstract}{%
1990     \titlepage
1991     \null\vfil
1992     \@beginparpenalty\@lowpenalty
1993     \begin{center}%
1994       \headfont \abstractname
1995       \@endparpenalty\@M
1996     \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。

```
1997   \par}%
1998 {\par\vfil\null\endtitlepage}
1999 \else
2000   \newenvironment{abstract}{%
2001     \if@twocolumn
2002       \ifx\maketitle\relax
2003         \section*{\abstractname}%
2004       \else
2005         \global\setbox\@abstractbox\hbox\bgroup
2006         \begin{minipage}[b]{\textwidth}
2007           \small\parindent1\jsZw
2008           \begin{center}%
```



```

2009         {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2010     \end{center}%
2011     \list{ }{%
2012         \listparindent\parindent
2013         \itemindent \listparindent
2014         \rightmargin \leftmargin}%
2015     \item\relax
2016 \fi
2017 \else
2018     \small
2019     \begin{center}%
2020         {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
2021     \end{center}%
2022     \list{ }{%
2023         \listparindent\parindent
2024         \itemindent \listparindent
2025         \rightmargin \leftmargin}%
2026     \item\relax
2027 \fi}{\if@twocolumn
2028     \ifx\maketitle\relax
2029     \else
2030         \endlist\end{minipage}\egroup
2031     \fi
2032 \else
2033     \endlist
2034 \fi}
2035 \fi
2036 %</article|report|slide>
2037 %<*jspf>
2038 \newbox\@abstractbox
2039 \newenvironment{abstract}{%
2040     \global\setbox\@abstractbox\hbox\bgroup
2041     \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
2042     \small
2043     \if@english \parindent6mm \else \parindent1\jsZw \fi}%
2044 {\end{minipage}\egroup}
2045 %</jspf>

```

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

```

2046 %<*book|report>
2047 \ifbxjs@force@chapterabstract
2048     \let\abstract\chapterabstract
2049     \let\endabstract\endchapterabstract
2050 \fi
2051 %</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。

```
2052 %<*jspf>
2053 %\newbox\@keywordsbox
2054 %\newenvironment{keywords}{%
2055 % \global\setbox\@keywordsbox\hbox\bgroup
2056 % \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
2057 % \small\parindent0\jsZw}%
2058 % {\end{minipage}\egroup}
2059 %</jspf>
```

■verse 環境

verse 詩のための `verse` 環境です。

```
2060 \newenvironment{verse}{%
2061 \let \=\@centercr
2062 \list{}{%
2063 \itemsep \z@
2064 \itemindent -2\jsZw % 元: -1.5em
2065 \listparindent\itemindent
2066 \rightmargin \z@
2067 \advance\leftmargin 2\jsZw}% 元: 1.5em
2068 \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。

```
2069 \newenvironment{quotation}{%
2070 \list{}{%
2071 \listparindent\parindent
2072 \itemindent\listparindent
2073 \rightmargin \z@}%
2074 \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。

```
2075 \newenvironment{quote}%
2076 {\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。

```
2077 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
2078   \item[\hskip \labelsep{\headfont #1\ #2}]}
2079 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
2080   \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 p_{La}T_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 p_{La}T_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。

```
2081 \newenvironment{titlepage}{%
2082 %<book> \pltx@cleartooddpage %% 2017-02-24
2083   \if@twocolumn
2084     \@restonecoltrue\onecolumn
2085   \else
2086     \@restonecolfalse\newpage
2087   \fi
2088   \thispagestyle{empty}%
2089   \ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-24
2090 }%
2091 {\if@restonecol\twocolumn \else \newpage \fi
2092   \if@twoside\else
2093     \setcounter{page}\@ne
2094   \fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2095 %<!*book&!report>
2096 \newcommand{\appendix}{\par
2097   \setcounter{section}{0}%
2098   \setcounter{subsection}{0}%
2099   \gdef\presectionname{\appendixname}%
2100   \gdef\postsectionname{}}
2101 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2102 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2103 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
2104 %</!*book&!report>
2105 %<*book|report>
2106 \newcommand{\appendix}{\par

```

```

2107 \setcounter{chapter}{0}%
2108 \setcounter{section}{0}%
2109 \gdef\@chapapp{\appendixname}%
2110 \gdef\@chappos{}%
2111 \gdef\thechapter{\@Alph\c@chapter}}
2112 %</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。

```
2113 \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。

```
2114 \setlength\tabcolsep{6\p@?}
```

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

```
2115 \setlength\arrayrulewidth{.4\p@}
```

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

```
2116 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \ ' コマンドで入るアキです。

```
2117 \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。

```
2118 \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。

```
2119 \setlength\fboxsep{3\p@?}
```

```
2120 \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。

```
2121 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```

2122 %<*book|report>
2123 \@addtoreset{equation}{chapter}
2124 \renewcommand\theequation
2125   {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
2126 %</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。

```
2127 % \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。

```
2128 % \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。

```
2129 % \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。

```

2130 %<*!book&!report>
2131 \newcounter{figure}
2132 \renewcommand \thefigure {\@arabic\c@figure}
2133 %</!book&!report>
2134 %<*book|report>
2135 \newcounter{figure}[chapter]
2136 \renewcommand \thefigure
2137   {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2138 %</book|report>

```

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

`\ext@figure`

`\fnum@figure`

```

2139 \def\fps@figure{tbp}
2140 \def\ftype@figure{1}
2141 \def\ext@figure{lof}
2142 \def\fnun@figure{\figurename\nobreak\thefigure}

```

figure * 形式は段抜きのフロートです。

```

figure* 2143 \newenvironment{figure}%
2144         {\@float{figure}}%
2145         {\end@float}
2146 \newenvironment{figure*}%
2147         {\@dblfloat{figure}}%
2148         {\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。

```

2149 %<!*book&!report>
2150 \newcounter{table}
2151 \renewcommand\thetable{\@arabic\c@table}
2152 %</*!book&!report>
2153 %<*book|report>
2154 \newcounter{table}[chapter]
2155 \renewcommand \thetable
2156     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
2157 %</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、ここでは外しました。

```

\ext@table 2158 \def\fps@table{tbp}
2159 \def\ftype@table{2}
\fnun@table 2160 \def\ext@table{lot}
2161 \def\fnun@table{\tablename\nobreak\thetable}

```

table * は段抜きのフロートです。

```

table* 2162 \newenvironment{table}%
2163         {\@float{table}}%
2164         {\end@float}
2165 \newenvironment{table*}%
2166         {\@dblfloat{table}}%
2167         {\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 になっていたもので、キャプションを表の上につけた場合にキャプションと表がくっついてしまうのを直しました。

```
2168 \newlength\abovecaptionskip
2169 \newlength\belowcaptionskip
2170 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2171 \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。

```
2172 %<!*jspf>
2173 % \long\def\@makecaption#1#2{\small
2174 %   \advance\leftskip1cm
2175 %   \advance\rightskip1cm
2176 %   \vskip\abovecaptionskip
2177 %   \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2178 %   \ifdim \wd\@tempboxa >\hsize
2179 %     #1\hskip1\jsZw\relax #2\par
2180 %   \else
2181 %     \global \@minipagefalse
2182 %     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2183 %   \fi
2184 %   \vskip\belowcaptionskip}}
2185 \long\def\@makecaption#1#2{\small
2186   \advance\leftskip .0628\linewidth
2187   \advance\rightskip .0628\linewidth
2188   \vskip\abovecaptionskip
2189   \sbox\@tempboxa{#1\zwspace#2}%
2190   \ifdim \wd\@tempboxa <\hsize \centering \fi
2191   #1\zwspace#2\par
2192   \vskip\belowcaptionskip}}
2193 %</!*jspf>
2194 %<!*jspf>
2195 \long\def\@makecaption#1#2{%
2196   \vskip\abovecaptionskip
2197   \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2198   \ifdim \wd\@tempboxa >\hsize
2199     {\small\sffamily
2200       \list{#1}{%
2201         \renewcommand{\makelabel}[1]{##1\hfil}
2202         \itemsep \z@
2203         \itemindent \z@
2204         \labelsep \z@
2205         \labelwidth 11mm
2206         \listparindent\z@
2207         \leftmargin 11mm}\item\relax #2\endlist}
2208   \else
```

```

2209 \global \@minipagefalse
2210 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2211 \fi
2212 \vskip\belowcaptionskip}
2213 %</jspf>

```

9 フォントコマンド

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

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

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

```

\if@jsc@warnoldfontcmd
\if@jsc@warnoldfontcmdexception

```

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

```

2214 \newif\if@jsc@warnoldfontcmd
2215 \@jsc@warnoldfontcmdtrue
2216 \newif\if@jsc@warnoldfontcmdexception
2217 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand

2218 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2219 \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2220 \DeclareOldFontCommand{#1}{%
2221 \bxjs@oldfontcmd{#1}#2%
2222 }{%
2223 \bxjs@oldfontcmd{#1}#3%
2224 }%
2225 }
2226 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2227 \ClassInfo\bxjs@clsname
2228 {Old font command '\string#1' is used!!\MessageBreak
2229 The first occurrence is}%
2230 }

```

`\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。


```

2231 \newcommand*{\allowoldfontcommands}{%
2232   \@jsc@warnoldfontcmdexceptiontrue}
2233 \newcommand*{\disallowoldfontcommands}{%
2234   \@jsc@warnoldfontcmdexceptionfalse}

```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2235 \let\bxjs@oldfontcmd@list\@empty
2236 \def\bxjs@oldfontcmd#1{%
2237   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2238 \def\bxjs@oldfontcmd@a#1#2{%
2239   \if@jsc@warnoldfontcmdexception\else
2240     \global\@jsc@warnoldfontcmdfalse
2241     \ifx#1\relax
2242       \global\let#1=t%
2243       \jsc@warnoldfontcmd{#2}%
2244     \fi
2245   \fi}
2246 \def\bxjs@warnoldfontcmd@final{%
2247 % \par
2248 \global\let\bxjs@warnoldfontcmd@final\@empty
2249 \let\@tempa\@empty
2250 \def\do##1{%
2251   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2252     \edef\@tempa{\@tempa \space\string##1}\fi}
2253 \bxjs@oldfontcmd@list
2254 \ifx\@tempa\@empty\else
2255   \ClassWarningNoLine\bxjs@clsname
2256   {Some old font commands were used in text\MessageBreak
2257     (see the log file for detail)}%
2258   \ClassInfo\bxjs@clsname
2259   {Some old font commands were used in text:\MessageBreak
2260     \space\@tempa\MessageBreak
2261     You should note, that since 1994 LaTeX2e provides a\MessageBreak
2262     new font selection scheme called NFSS2 with several\MessageBreak
2263     new, combinable font commands. The
2264     class provides\MessageBreak
2265     the old font commands
2266     only for compatibility%
2267     \@gobble}%
2268 \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。

```

2269 \def\bxjs@warnoldfontcmd@kick@final{%
2270   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2271 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```
\gt 2272 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}  
\rm 2273 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}  
2274 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}  
\sf 2275 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}  
\tt 2276 \jsc@DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
```

`\bf` ボールドシリーズにします。通常のミディウムシリーズに戻るコマンドは `\mdseries` です。

```
2277 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

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

```
2278 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}  
2279 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}  
2280 \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 2281 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}  
2282 \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。

```
2283 \newcommand\@pnumwidth{1.55em}
2284 \newcommand\@tocrmarg{2.55em}
2285 \newcommand\@dotsep{4.5}
2286 %<!book&!report>\setcounter{tocdepth}{2}
2287 %<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)

```
2288 \newdimen\jsc@tocl@width
2289 \newcommand{\tableofcontents}{%
2290 %<*book|report>
2291 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2292 \settowidth\@tempdima{\headfont\appendixname}%
2293 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2294 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2295 \if@twocolumn
2296 \@restonecoltrue\onecolumn
2297 \else
2298 \@restonecolfalse
2299 \fi
2300 \chapter*{\contentsname}%
2301 \@mkboth{\contentsname}{}%
2302 %</book|report>
2303 %<*!book&!report>
2304 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2305 \settowidth\@tempdima{\headfont\appendixname}%
2306 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2307 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2308 \section*{\contentsname}%
```

```

2309 \mkboth{\contentsname}{\contentsname}%
2310 %</!book&!report>
2311 \@starttoc{toc}%
2312 %<book|report> \if@restonecol\twocolumn\fi
2313 }

```

\l@part 部の目次です。

```

2314 \newcommand*{\l@part}[2]{%
2315 \ifnum \c@tocdepth >-2\relax
2316 %<!book&!report> \addpenalty\@secpenalty
2317 %<book|report> \addpenalty{-\@highpenalty}%
2318 \addvspace{2.25em \@plus\p@}%
2319 \begingroup
2320 \parindent \z@
2321 % \@pnumwidth should be \@tocrmarg
2322 % \rightskip \@pnumwidth
2323 \rightskip \@tocrmarg
2324 \parfillskip -\rightskip
2325 {\leavevmode
2326 \large \headfont
2327 \setlength\@lnumwidth{4\jsZw}%
2328 #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2329 \nobreak
2330 %<book|report> \global\@nobreaktrue
2331 %<book|report> \everypar{\global\@nobreakfalse\everypar{}}%
2332 \endgroup
2333 \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)

```

2334 %<*book|report>
2335 \newcommand*{\l@chapter}[2]{%
2336 \ifnum \c@tocdepth >\m@ne
2337 \addpenalty{-\@highpenalty}%
2338 \addvspace{1.0em \@plus\p@}
2339 % \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2340 \begingroup
2341 \parindent \z@
2342 % \rightskip \@pnumwidth
2343 \rightskip \@tocrmarg
2344 \parfillskip -\rightskip
2345 \leavevmode\headfont
2346 % % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2347 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2348 \advance\leftskip\@lnumwidth \hskip-\leftskip
2349 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2350 \penalty\@highpenalty
2351 \endgroup

```

```

2352 \fi}
2353 %</book|report>

```

\l@section 節の目次です。

```

2354 %<!*book&!report>
2355 \newcommand*{\l@section}[2]{%
2356 \ifnum \c@tocdepth >\z@
2357 \addpenalty{\@secpenalty}%
2358 \addvspace{1.0em \@plus\p@?}%
2359 \begingroup
2360 \parindent\z@
2361 % \rightskip\@pnumwidth
2362 \rightskip\@tocrmarg
2363 \parfillskip-\rightskip
2364 \leavevmode\headfont
2365 % % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2366 \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2367 \advance\leftskip\@lnumwidth \hskip-\leftskip
2368 #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2369 \endgroup
2370 \fi}
2371 %</!*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。

```

2372 %<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 2373 %<!*book&!report>
2374 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2375 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2376 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2377 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2378 %
2379 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2380 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2381 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2382 %
2383 \newcommand*{\l@subsection}{%
2384 \tempdima\jsc@tocl@width \advance\tempdima -1\jsZw
2385 \@dottedtocline{2}{\tempdima}{3\jsZw}}
2386 \newcommand*{\l@subsubsection}{%
2387 \tempdima\jsc@tocl@width \advance\tempdima 0\jsZw
2388 \@dottedtocline{3}{\tempdima}{4\jsZw}}
2389 \newcommand*{\l@paragraph}{%
2390 \tempdima\jsc@tocl@width \advance\tempdima 1\jsZw

```

```

2391 \dottedtocline{4}{\@tempdima}{5\jsZw}}
2392 \newcommand*\l@subparagraph}{%
2393 \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2394 \dottedtocline{5}{\@tempdima}{6\jsZw}}
2395 %</!book&!report>
2396 %<*book|report>
2397 % \newcommand*\l@section}{\@dottedtocline{2}{3.8em}{3.2em}}
2398 % \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
2399 % \newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
2400 % \newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}
2401 \newcommand*\l@section}{%
2402 \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2403 \dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2404 \newcommand*\l@subsubsection}{%
2405 \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2406 \dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2407 \newcommand*\l@subsubsection}{%
2408 \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2409 \dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2410 \newcommand*\l@paragraph}{%
2411 \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2412 \dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2413 \newcommand*\l@subparagraph}{%
2414 \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2415 \dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2416 %</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。

```

2417 \newdimen\@lnumwidth
2418 \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。

```

2419 \def\dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2420 \vskip \z@ \@plus.2\p@?
2421 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2422 \parindent #2\relax\@afterindenttrue
2423 \interlinepenalty\@M
2424 \leavevmode
2425 \@lnumwidth #3\relax
2426 \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2427 {#4}\nobreak
2428 \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2429 mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
2430 \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```
2431 \newcommand{\listoffigures}{%
2432 %<*book|report>
2433 \if@twocolumn\@restonecoltrue\onecolumn
2434 \else\@restonecolfalse\fi
2435 \chapter*{\listfigurename}%
2436 \@mkboth{\listfigurename}{}%
2437 %</book|report>
2438 %<!*book&!report>
2439 \section*{\listfigurename}%
2440 \@mkboth{\listfigurename}{\listfigurename}%
2441 %</!book&!report>
2442 \@starttoc{lof}%
2443 %<book|report> \if@restonecol\twocolumn\fi
2444 }
```

`\l@figure` 図目次の項目を出力します。

```
2445 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}
```

`\listoftables` 表目次を出力します。

```
2446 \newcommand{\listoftables}{%
2447 %<*book|report>
2448 \if@twocolumn\@restonecoltrue\onecolumn
2449 \else\@restonecolfalse\fi
2450 \chapter*{\listtablename}%
2451 \@mkboth{\listtablename}{}%
2452 %</book|report>
2453 %<!*book&!report>
2454 \section*{\listtablename}%
2455 \@mkboth{\listtablename}{\listtablename}%
2456 %</!book&!report>
2457 \@starttoc{lot}%
2458 %<book|report> \if@restonecol\twocolumn\fi
2459 }
```

`\l@table` 表目次は図目次と同じです。

```
2460 \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。

```
2461 \newdimen\bibindent
2462 \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。

```

2463 \newenvironment{thebibliography}[1]{%
2464   \jsc@warnoldfontcmdexceptiontrue
2465   \global\let\presectionname\relax
2466   \global\let\postsectionname\relax
2467   %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2468   %<*kiyou>
2469   \vspace{1.5\baselineskip}
2470   \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2471   \vspace{0.5\baselineskip}
2472 %</kiyou>
2473 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2474 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2475   \list{\@biblabel{\@arabic{c@enumiv}}}%
2476     {\settowidth\labelwidth{\@biblabel{#1}}%
2477      \leftmargin\labelwidth
2478      \advance\leftmargin\labelsep
2479      \@openbib@code
2480      \usecounter{enumiv}%
2481      \let\p@enumiv\@empty
2482      \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
2483 %<kiyou> \small
2484   \sloppy
2485   \clubpenalty4000
2486   \@clubpenalty\clubpenalty
2487   \widowpenalty4000%
2488   \sfcode`.\@m}
2489 {\def\@noitemerr
2490  {\@latex@warning{Empty `thebibliography' environment}}}%
2491 \endlist}

```

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

```

2492 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

```

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

```

2493 \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。

```

2494 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

`\cite` 文献の番号を出力する部分は `ltbibl.dtx` で定義されていますが、コンマとカッコを和文

`\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必

`\@citex`

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

```
2495 % \def\@citex[#1]#2{\leavevmode
2496 %   \let\@citea\@empty
2497 %   \@cite{\@for\@citeb:=#2\do
2498 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2499 %     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2500 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2501 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2502 %       \G@refundefinedtrue
2503 %       \@latex@warning
2504 %         {Citation `'\@citeb' on page \thepage \space undefined}}%
2505 %       {\@citeofmt{\csname b@\@citeb\endcsname}}}{#1}}
2506 % \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。

```
2507 % \DeclareRobustCommand\cite{\unskip
2508 %   \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}]
2509 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2510 %   , \jsInhibitGlue\ #2\fi}})}$}
```

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くにマージンがずれる現象を直しました (Thanks: 藤村さん)。

```
2511 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2512   \if@twocolumn
2513     \onecolumn\@restonecolfalse
2514   \else
2515     \clearpage\@restonecoltrue
2516   \fi
2517   \columnseprule.4pt \columnsep 2\jsZw
2518   \ifx\multicols\@undefined
2519 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2520 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2521 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2522 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2523   \else
2524     \ifdim\textwidth<\fullwidth
2525       \setlength{\evensidemargin}{\oddsidemargin}
2526       \setlength{\textwidth}{\fullwidth}
2527       \setlength{\linewidth}{\fullwidth}
2528 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
2529 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2530 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2531 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
```

```

2532     \else
2533 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}]%
2534 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2535 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2536 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2537     \fi
2538 \fi
2539 %<book|report>      \@mkboth{\indexname}{}%
2540 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2541 \plainifnotempty % \thispagestyle{plain}
2542 \parindent\z@
2543 \parskip\z@ \@plus .3\p@?\relax
2544 \let\item\@idxitem
2545 \raggedright
2546 \footnotesize\narrowbaselines
2547 }{
2548 \ifx\multicols\@undefined
2549 \if@restonecol\onecolumn\fi
2550 \else
2551 \end{multicols}
2552 \fi
2553 \clearpage
2554 }

```

`\@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 2555 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2556 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2557 \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。

```

2558 \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。

```

2559 \newcommand\seename{\if@english see\else →\fi}
2560 \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。

```

2561 \ifx\pltx@foot@penalty\undefined
2562   \let\footnotes@ve=\footnote
2563   \def\footnote{\jsInhibitGlue\footnotes@ve}
2564   \let\footnotemarks@ve=\footnotemark
2565   \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}
2566 \fi

```

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

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

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

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

pTeX 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

```

2567 \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。

```

2568 % \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。

```

2569 \renewcommand{\footnoterule}{%
2570   \kern-2.6\p@? \kern-.4\p@
2571   \hrule width .4\columnwidth
2572   \kern 2.6\p@?}

```

\c@footnote 脚注番号は章ごとにリセットされます。

```

2573 %<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, *TeX and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9)

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

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

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

```
2574 \long\def\@footnotetext{%
2575   \insert\footins\bgroup
2576     \normalfont\footnotesize
2577     \interlinepenalty\interfootnotelinepenalty
2578     \splittopskip\footnotesep
2579     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
2580     \hsize\columnwidth \@parboxrestore
2581     \protected@edef\@currentlabel{%
2582       \csname p@footnote\endcsname\@thefnmark
2583     }%
2584     \color@begingroup
2585       \@makefnmark
2586       \rule{z@footnotesep}{\ignorespaces}%
2587       \futurelet\next\fo@t}
2588 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@@
2589                                     \else \let\next\fo@fi \next}
2590 \def\fo@@{\bgroup\aftergroup\@foot\let\next}
2591 \def\fo@t#1{#1\@foot}
2592 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2593   \ifx\pltx@foot@penalty\@undefined\else
2594     \ifhmode\null\fi
2595     \ifnum\pltx@foot@penalty=z@\else
2596       \penalty\pltx@foot@penalty
2597       \pltx@foot@penalty\z@
2598     \fi
2599   \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。

```
2600 \newcommand\@makefnmark[1]{%
2601   \advance\leftskip 3\jsZw
2602   \parindent 1\jsZw
2603   \noindent
2604   \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。

```
2605 % \def\@xfootnotenext[#1]{%
2606 %   \begingroup
```

```

2607 %      \ifnum#1>\z@
2608 %      \csname c@\@mpfn\endcsname #1\relax
2609 %      \unrestored@protected@xdef\@thefnmark{\thempfn}%
2610 %      \else
2611 %      \unrestored@protected@xdef\@thefnmark{}%
2612 %      \fi
2613 % \endgroup
2614 % \@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。

```
2615 \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。

```

2616 \def\everyparhook{\jsInhibitGlueAtParTop}
2617 \AtBeginDocument{\everypar{\everyparhook}}

```

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

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

[2017-02-13] `\jsc@tempa` は実はテンポラリーではなく「この処理専用のユニーク制御綴」である必要があります。間違っ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。)

```

2618 \def\@inhibitglue{%
2619   \futurelet\@let@token\@@inhibitglue}
2620 \begingroup
2621 \let\GDEF=\gdef
2622 \let\CATCODE=\catcode
2623 \let\ENDGROUP=\endgroup
2624 \CATCODE`k=12
2625 \CATCODE`a=12

```

```

2626 \CATCODE`n=12
2627 \CATCODE`j=12
2628 \CATCODE`i=12
2629 \CATCODE`c=12
2630 \CATCODE`h=12
2631 \CATCODE`r=12
2632 \CATCODE`t=12
2633 \CATCODE`e=12
2634 \GDEF\KANJI@CHARACTER{kanji character }
2635 \ENDGROUP
2636 \def\@@inhibitglue{%
2637   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
2638   \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
2639   \def\jsc@ig@temp{#1}%
2640   \ifx\jsc@ig@temp\@empty
2641     \ifnum\the\inhibitxspcode`#2=2\relax
2642       \inhibitglue
2643     \fi
2644   \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。

```

2645 \def\@doendpe{%
2646   \@endpetrue
2647   \def\par{%
2648     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2649   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

`\item` 命令の直後です。

```

2650 \def\@item[#1]{%
2651   \if@noperitem
2652     \@donoperitem
2653   \else
2654     \if@inlabel
2655       \indent \par
2656     \fi
2657     \ifhmode
2658       \unskip\unskip \par
2659     \fi
2660     \if@newlist
2661       \if@nobreak
2662         \@nbitem
2663       \else
2664         \addpenalty\@beginparpenalty

```

```

2665         \addvspace\@topsep
2666         \addvspace{-\parskip}%
2667     \fi
2668 \else
2669     \addpenalty\@itempenalty
2670     \addvspace\itemsep
2671 \fi
2672 \global\@inlabeltrue
2673 \fi
2674 \everypar{%
2675     \@minipagefalse
2676     \global\@newlistfalse
2677     \if@inlabel
2678         \global\@inlabelfalse
2679         {\setbox\z@\lastbox
2680         \ifvoid\z@
2681             \kern-\itemindent
2682         \fi}%
2683     \box\@labels
2684     \penalty\z@
2685 \fi
2686 \if@nobreak
2687     \@nobreakfalse
2688     \clubpenalty \@M
2689 \else
2690     \clubpenalty \@clubpenalty
2691     \everypar{\everyparhook}%
2692 \fi
2693 \bxjs\ltj\inhibitglue
2694 \everyparhook}%
2695 \if@noitemarg
2696     \@noitemargfalse
2697     \if@nmbrlist
2698         \refstepcounter\@listctr
2699     \fi
2700 \fi
2701 \sbox\@tempboxa{\makelabel{#1}}%
2702 \global\setbox\@labels\hbox{%
2703     \unhbox\@labels
2704     \hskip \itemindent
2705     \hskip -\labelwidth
2706     \hskip -\labelsep
2707     \ifdim \wd\@tempboxa >\labelwidth
2708         \box\@tempboxa
2709     \else
2710         \hbox to\labelwidth {\unhbox\@tempboxa}%
2711     \fi
2712     \hskip \labelsep}%
2713 \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。

```
2714 \def\@afterheading{%
2715   \@nobreaktrue
2716   \everypar{%
2717     \if@nobreak
2718       \@nobreakfalse
2719       \clubpenalty \@M
2720       \if@afterindent \else
2721         {\setbox\z@\lastbox}%
2722         \fi
2723       \else
2724         \clubpenalty \@clubpenalty
2725         \everypar{\everyparhook}%
2726       \fi\everyparhook}}
```

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

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

```
2727 \def\@gnewline #1{%
2728   \ifvmode
2729     \@nolnerr
2730   \else
2731     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
2732     \jsInhibitGlue \ignorespaces
2733   \fi}
```

12 いろいろなロゴ

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

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

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

※`\小`、`\上小` の制御綴は定義しない。

```
2734 \if@jslogo
2735   \IfFileExists{jslogo.sty}{%
2736     \RequirePackage{jslogo}%
2737   }{%
2738     \ClassWarningNoLine{bxjs}{clsname
```



```

2739     {The package 'jslogo' is not installed.\MessageBreak
2740     It is included in the recent release of\MessageBreak
2741     the 'jsclasses' bundle}
2742   }
2743 \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 2744 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2745 \newcommand{\postpartname}{\if@english\else 部\fi}
2746 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2747 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2748 \newcommand{\presectionname}{}% 第
\postsectionname 2749 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 2750 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2751 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
2752 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 2753 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2754 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
2755 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 2756 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
2757 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname
\abstractname 2758 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
2759 \newcommand{\appendixname}{\if@english \else 付録\fi}
2760 %<!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 `\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`

```
2761 \@tempwafalse
2762 \if p\jsEngine \@tempwattrue \fi
2763 \if n\jsEngine \@tempwattrue \fi
2764 \if@tempswa \expandafter\@firstoftwo
2765 \else \expandafter\@secondoftwo
2766 \fi
2767 {%
2768 % 欧文 8bitTeX の場合
2769 \newif\ifjsSeireki \jsSeirekitrue
2770 \def\bxjs@decl@Seireki@cmds{%
2771   \def\西暦{\jsSeirekitrue}%
2772   \def\和暦{\jsSeirekifalse}}
2773 \def\Seireki{\jsSeirekitrue}
2774 \def\Wareki{\jsSeirekifalse}
2775 \def\bxjs@if@use@seireki{%
2776   \ifjsSeireki \expandafter\@firstoftwo
2777   \else \expandafter\@secondoftwo \fi}
2778 }{%
2779 \newif\if 西暦 \西暦 true
2780 \def\bxjs@decl@Seireki@cmds{%
2781   \def\西暦{\西暦 true}%
2782   \def\和暦{\西暦 false}}
2783 \def\Seireki{\西暦 true}
2784 \def\Wareki{\西暦 false}
2785 \def\bxjs@if@use@seireki{%
2786   \if 西暦 \expandafter\@firstoftwo
2787   \else \expandafter\@secondoftwo \fi}
2788 }
2789 \bxjs@decl@Seireki@cmds
2790 % \bxjs@unxp
2791 \let\bxjs@unxp\@firstofone
2792 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2793 % \bxjs@iai
2794 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2795   \def\bxjs@iai{\noexpand~}
2796 \else \def\bxjs@iai{}
2797 \fi
2798 % \heisei
2799 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2800 % \today
2801 \edef\bxjs@today{%
```

```

2802 \if@english
2803   \ifcase\month\or
2804     January\or February\or March\or April\or May\or June\or
2805     July\or August\or September\or October\or November\or December\fi
2806     \space\number\day, \number\year
2807 \else
2808   \ifnum\heisei>\@ne
2809     \expandafter\noexpand\expandafter\bxjs@if@use@seireki
2810   \else \expandafter\@firstoftwo
2811   \fi {%
2812     \number\year\bxjs@iai\bxjs@unxp{年}%
2813     \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2814     \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2815   }{%
2816     \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2817     \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2818     \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2819   }%
2820 \fi}
2821 \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。

```

2822 \AtBeginDocument{%
2823   \ifx\bbl@jpn@Seirekitrue\@undefined\else
2824     \bxjs@decl@Seireki@cmds
2825     \g@addto@macro\datejapanese{%
2826       \let\today\bxjs@today}%
2827   \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)

```

2828 \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。

```

2829 %<slide>\pagestyle{empty}%
2830 %<article|report>\pagestyle{plain}%
2831 %<book>\pagestyle{headings}%
2832 \pagenumbering{arabic}
2833 \if@twocolumn
2834   \twocolumn
2835   \sloppy
2836   \flushbottom
2837 \else
2838   \onecolumn

```

```
2839 \raggedbottom
2840 \fi
2841 %<*slide>
2842 \renewcommand\familydefault{\sfdefault}
2843 \raggedright
2844 %</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。

```
2845 \catcode`\?=12
2846 \ifx\bxjs@jadriver\relax\else
2847 \input{bxjsja-\bxjs@jadriver.def}
2848 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
2849 \bxjs@restore@jltrcc
2850 %</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。

```
2851 %<*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

```
2852 %<*minimal>
```

```
2853 %% 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.
```

```
2854 \def\DeclareJaTextFontCommand#1#2{%
```

```
2855   \DeclareRobustCommand#1[1]{%
```

```
2856     \relax
```

```
2857     \ifmmode \expandafter\nfss@text \fi
```

```
2858     {#2##1}}%
```

```
2859 }
```

```
\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。
```

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

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

```
2862     \relax
```

```
2863     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
2864     \nfss@text{\fontfamily\familydefault
```

```
2865         \fontseries{m}\fontshape{n}\selectfont\relax
```

```
2866         #2##1}}%
```

```

2867 }%
2868 }

```

`\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。

```

2869 \long\def\bxjs@@CSsfdefault{\sfdefault}%
2870 \@onlypreamble\bxjs@if@sf@default
2871 \def\bxjs@if@sf@default#1{%
2872   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
2873   \AtBeginDocument{%
2874     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
2875 }

```

`\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。

```

2876 \def\jsLetHeadChar#1#2{%
2877   \begingroup
2878     \escapechar=`\ %
2879     \let\bxjs@tmpa={% brace-match-hack
2880       \bxjs@let@hchar@exp#2}%
2881   \endgroup
2882   \let#1\bxjs@g@tmpa}
2883 \def\bxjs@let@hchar@exp{%
2884   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2885 \def\bxjs@let@hchar@exp@a{%
2886   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
2887     \bxjs@let@hchar@out\let\relax
2888   }\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi{% 空白
2889     \bxjs@let@hchar@out\let\space%
2890   }\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
2891     \bxjs@let@hchar@out\let\@backslashchar
2892   }\bxjs@let@hchar@exp@b}}
2893 \def\bxjs@let@hchar@exp@b#1{%
2894   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2895 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2896   %\message{<#1#2>}%
2897   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴
2898     \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2899       \bxjs@let@hchar@out\let\relax
2900     }{%else
2901       \expandafter\bxjs@let@hchar@exp
2902     }%
2903   }{%else
2904     \bxjs@let@hchar@chr#1%

```

```

2905  }}
2906 \def\bxjs@let@hchar@chr#1{%
2907   \bxjs@let@hchar@out\def{{#1}}}
2908 \def\bxjs@let@hchar@out#1#2{%
2909   \global#1\bxjs@g@tmpa#2\relax
2910   \toks@{\bgroup}% skip to right brace

UTF-8 のバイト列を扱うコード。

2911 \chardef\bxjs@let@hchar@csta=128
2912 \chardef\bxjs@let@hchar@cstb=192
2913 \chardef\bxjs@let@hchar@cstc=224
2914 \chardef\bxjs@let@hchar@cstd=240
2915 \chardef\bxjs@let@hchar@cste=248
2916 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2917 \def\bxjs@let@hchar@chr@ue#1{%
2918   \@tempcnta=#1\relax
2919   \%message{\the\@tempcnta}%
2920   \bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2921     \bxjs@let@hchar@chr@ue@a#1%
2922   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2923     \bxjs@let@hchar@out\let\relax
2924   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2925     \bxjs@let@hchar@chr@ue@b
2926   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2927     \bxjs@let@hchar@chr@ue@c
2928   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2929     \bxjs@let@hchar@chr@ue@d
2930   }{%else
2931     \bxjs@let@hchar@out\let\relax
2932   }}}}
2933 \def\bxjs@let@hchar@chr@ue@a#1{%
2934   \bxjs@let@hchar@out\def{{#1}}}
2935 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2936   \bxjs@let@hchar@out\def{{#1#2}}}
2937 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2938   \bxjs@let@hchar@out\def{{#1#2#3}}}
2939 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2940   \bxjs@let@hchar@out\def{{#1#2#3#4}}}

```

B.2 (u)pTeX 用の設定

```
2941 \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。

```

2942 \def\bxjs@let@hchar@chr@pp#1{%
2943   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}

```



```

2944 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2945 %\message{(#1)}%
2946 \bxjs@cond@if#1t\fi{%
2947 \bxjs@let@hchar@chr@ue#3%
2948 }{%else
2949 \bxjs@let@hchar@out\def{#3}%
2950 }}
2951 \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。

```

2952 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
2953 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
2954 \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。

```

2955 \@onlypreamble\bxjs@declarefontshape
2956 \ifjsWithupTeX
2957 \def\bxjs@declarefontshape{%
2958 \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
2959 \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
2960 \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
2961 \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
2962 }
2963 \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。

```

2964 \else
2965 \def\bxjs@declarefontshape{%
2966 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
2967 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
2968 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
2969 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
2970 }
2971 \def\bxjs@sizereference{jis}
2972 \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。

```

2973 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%

```

```

2974 \def\bxjs@y{#5}}
2975 \ifjsWithpTeXng \def\bxjs@y{10}%
2976 \else
2977 \expandafter\expandafter\expandafter\bxjs@tmpa
2978 \expandafter\string\the\jfont\relax
2979 \fi
2980 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
2981             \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
2982 {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
2983 \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。

```

2984 \begingroup
2985 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2986 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2987 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2988 % 幅が丁度 10pt なら補正は不要
2989 \ifdim\wd\z@=10pt
2990 \global\let\bxjs@scale\jsScale
2991 \else
2992 %  $(10*s)/(10*f)$  として計算、\bxjs@invscale は BXJS で定義
2993 \edef\bxjs@tmpa{\strip@pt\wd\z@}
2994 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
2995 \bxjs@invscale\@tempdima\bxjs@tmpa
2996 \xdef\bxjs@scale{\strip@pt\@tempdima}
2997 \fi
2998 \endgroup
2999 %\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。

```

3000 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

3001 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3002 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
3003 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
3004 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
3005 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
3006 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
3007 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
3008 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
3009 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}

```

```

3010 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
3011 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
3012 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
3013 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
3014 \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。

```

3015 \DeclareRobustCommand\rmfamily
3016 { \not@math@alphabet\rmfamily\mathrm
3017   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3018 \DeclareRobustCommand\sffamily
3019 { \not@math@alphabet\sffamily\mathsf
3020   \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
3021 \DeclareRobustCommand\ttfamily
3022 { \not@math@alphabet\ttfamily\mathtt
3023   \romanfamily\ttddefault\kanjifamily\gtdefault\selectfont}
3024 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3025 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
3026 \bxjs@if@sf@default{%
3027   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

3028 \selectfont

```

■パラメタの設定

```

3029 \prebreakpenalty\jis"2147=10000
3030 \postbreakpenalty\jis"2148=10000
3031 \prebreakpenalty\jis"2149=10000
3032 \inhibitxspcode`!=1
3033 \inhibitxspcode`〒=2
3034 \xspcode`+=3
3035 \xspcode`\%=3

```

"80～"FF の範囲の \spcode を 3 に変更。

```

3036 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
3037   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

3038 \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。

```

3039 \begingroup
3040 \catcode`\!=0

```

\bxjs@ptex@dir 現在の組方向： t=縦、y=横、?=その他。

```

3041 \gdef\bxjs@ptex@dir{%
3042   !iftdir t%
3043   !else!ifydir y%
3044   !else ?%
3045   !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。

```

3046 % 古い \@makefnmark の定義
3047 \long\def\bxjs@tmpa{\hbox{%
3048   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
3049   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
3050 \ifx\@makefnmark\bxjs@tmpa
3051 \long\gdef\@makefnmark{%
3052   !ifydir \hbox{ }\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{ }%
3053   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
3054 \fi
3055 \endgroup

```

B.3 pdf_TE_X 用の処理

```

3056 \else\ifx p\jsEngine
3057 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
3058 \@onlypreamble\bxjs@cjk@loaded
3059 \def\bxjs@cjk@loaded{%
3060   \def\@footnotemark{%
3061     \leavevmode
3062     \ifhmode
3063       \edef\x@sf{\the\spacefactor}%
3064       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
3065         \unkern\unkern
3066         \ifdim\lastskip>\z@ \unskip \fi
3067       \fi\fi
3068       \nobreak
3069     \fi
3070     \@makefnmark
3071     \ifhmode \spacefactor\x@sf \fi
3072     \relax}%
3073 \let\bxjs@cjk@loaded\relax
3074 }
3075 \AtBeginDocument{%
3076   \@ifpackageloaded{CJK}{%
3077     \bxjs@cjk@loaded
3078   }{}%
3079 }

```

B.4 XeTeX 用の処理

```
3080 \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。

```
3081 \def\bxjs@let@hchar@chr#1{%
3082   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3083   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3084     \bxjs@let@hchar@chr@xe
3085   }\bxjs@let@hchar@out\def{{#1}}}}
3086 \def\bxjs@let@hchar@chr@xe#1{%
3087   \lccode`0=`#1\relax
3088   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3089 \ifx\XeTeXgenerateactualtext\@undefined\else
3090   \def\bxjs@do@precisetext{%
3091     \XeTeXgenerateactualtext=\@ne}
3092 \fi
```

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

```
3093 \@onlypreamble\bxjs@do@simplejasetup
3094 \def\bxjs@do@simplejasetup{%
3095   \ifnum\XeTeXinterchartokenstate>\z@
3096   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3097     \jsSimpleJaSetup
3098     \ClassInfo\bxjs@clsname
3099     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3100   \fi\fi}
```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```
3101 \newcommand*{\jsSimpleJaSetup}{%
3102   \XeTeXlinebreaklocale "ja"\relax
3103   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3104   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

```
3105 \fi\fi\fi
```

`simplejasetup` オプションの処理。

```
3106 \ifx\bxjs@do@simplejasetup\@undefined\else
3107   \AtBeginDocument{%
3108     \ifbxjs@simplejasetup
3109       \bxjs@do@simplejasetup
3110     \fi}
3111 \fi
```

`precisetext` オプションの処理。

```

3112 \ifbxjs@precisetext
3113   \ifx\bxjs@do@precisetext\@undefined
3114     \ClassWarning\bxjs@clsname
3115       {The current engine does not support the\MessageBreak
3116         'precisetext' option\@gobble}
3117   \else
3118     \bxjs@do@precisetext
3119   \fi
3120 \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。

```

3121 \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。

```

3122 \@onlypreamble\bxjs@adjust@fancyhdr
3123 \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。

```

3124 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3125 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3126 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3127 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3128 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3129 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3130 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3131 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3132 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3133 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3134 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3135 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3136 \def\bxjs@tmpa{\rm\thepage\strut}%
3137 \def\bxjs@tmpb{\thepage\strut}%
3138 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3139 \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。

```

3140 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3141   \setlength{\@tempdima}{\fullwidth-\textwidth}%
3142   \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3143     }\bxjs@tmpa

```

```

3144 \fi\fi
3145 \PackageInfo{bxjs@clsname
3146   {Patch to fancyhdr is applied\@gobble}}

```

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

```

3147 \def\bxjs@pagestyle@hook{%
3148   \@ifpackageloaded{fancyhdr}{%
3149     \bxjs@adjust@fancyhdr
3150     \global\let\bxjs@adjust@fancyhdr\relax
3151   }{}}

```

`\pagestyle` にフックを入れ込む。

```

3152 \let\bxjs@org@pagestyle\pagestyle
3153 \def\pagestyle{%
3154   \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。

```

3155 \AtBeginDocument{%
3156   \bxjs@pagestyle@hook
3157   \global\let\bxjs@pagestyle@hook\relax}

```

終わり。

```

3158 \fi

```

以上で終わり。

```

3159 %</minimal>

```

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

コマンドで使う“ファミリー”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。

```
3160 %<*standard>
3161 %% このファイルは日本語文字を含みます
3162 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3163 \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。

```
3164 \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。

```
\jH3165 \@tempdima=0.25mm
3166 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3167 \let\jH\jQ
```

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

```
\trueH3168 \ifjsc@mag
3169 \@tempdimb=\jsBaseFontSize\relax
3170 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3171 \@tempdima=2.5mm
3172 \bxjs@invscale\@tempdima\bxjs@tmpa
3173 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3174 \@tempdima=10pt
3175 \bxjs@invscale\@tempdima\bxjs@tmpa
3176 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3177 \else \let\trueQ\jQ \let\bxjs@truept\p@
3178 \fi
3179 \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。

```
3180 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3181 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3182 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3183 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3184 \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。

```
3185 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3186 \newcommand*\setkanjiskip[1]{%
```

```
3187   \edef\bxjs@kanjiskip{#1}%
```

```
3188   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3189 \newcommand*\getkanjiskip{%
```

```
3190   \bxjs@kanjiskip}
```

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

```
3191 \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 3192 \bxjs@robust@def\bxjs@enable@kanjiskip{%
```

```
3193   \bxjs@kanjiskip@enabledtrue
```

```
3194   \bxjs@reset@kanjiskip}
```

```
3195 \bxjs@robust@def\bxjs@disable@kanjiskip{%
```

```
3196   \bxjs@kanjiskip@enabledfalse
```

```
3197   \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。

```
3198 \bxjs@robust@def\bxjs@reset@kanjiskip{%
```

```
3199   \ifbxjs@kanjiskip@enabled
```

```
3200     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
```

```
3201     \else \@tempskipa\z@
```

```
3202     \fi
```

```
3203     \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 3204 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3205 \newcommand*\setxkanjiskip[1]{%
```

```
3206   \edef\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3207   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3208 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3209   \bxjs@xkanjiskip}
```

```
3210 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3211 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3212   \bxjs@xkanjiskip@enabledtrue
```

```
3213   \bxjs@reset@xkanjiskip}
```

```
3214 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3215   \bxjs@xkanjiskip@enabledfalse
```

```
3216   \bxjs@reset@xkanjiskip}
```

```
3217 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```

3218 \ifbxjs@xkanjiskip@enabled
3219 \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3220 \else \@tempskipa\z@
3221 \fi
3222 \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。

```

3223 \g@addto@macro\jsResetDimen{%
3224 \bxjs@reset@kanjiskip
3225 \bxjs@reset@xkanjiskip}
3226 \let\bxjs@apply@kanjiskip\relax
3227 \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。

```

3228 \@onlypreamble\bxjs@adjust@jafont
3229 \def\bxjs@adjust@jafont#1{%
3230 \ifx\jsJaFont\bxjs@@auto
3231 \bxjs@get@kanjiEmbed
3232 \ifx\bxjs@kanjiEmbed\relax
3233 \let\bxjs@tmpa\@empty
3234 \else
3235 \let\bxjs@tmpa\bxjs@kanjiEmbed
3236 \fi
3237 \else
3238 \let\bxjs@tmpa\jsJaFont
3239 \fi
3240 \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3241 \ClassWarningNoLine\bxjs@clsname
3242 {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3243 not available on the current situation}%
3244 \let\bxjs@tmpa\@empty
3245 \fi\fi
3246 }
3247 \def\bxjs@@auto{auto}
3248 \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。

```

3249 \let\bxjs@kanjiEmbed\relax

```

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

```

3250 \@onlypreamble\bxjs@get@kanjiEmbed
3251 \def\bxjs@get@kanjiEmbed{%

```

```

3252 \begingroup\setbox\z@=\hbox{%
3253   \global\let\bxjs@g@tmpa\relax
3254   \endlinechar\m@ne
3255   \let\do\@makeother\dospecials
3256   \catcode32=10 \catcode12=10 %form-feed
3257   \let\bxjs@tmpa\@empty
3258   \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3259   \ifeof\@inputcheck\else
3260     \read\@inputcheck to\bxjs@tmpa
3261     \closein\@inputcheck
3262   \fi
3263   \ifx\bxjs@tmpa\@empty\else
3264     \openin\@inputcheck="\bxjs@tmpa"\relax
3265     \@tempswatrue
3266     \loop\if@tempswa
3267       \read\@inputcheck to\bxjs@tmpa
3268       \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3269       \ifx\bxjs@tmpa\relax\else
3270         \global\let\bxjs@g@tmpa\bxjs@tmpa
3271         \@tempswafalse
3272       \fi
3273       \ifeof\@inputcheck \@tempswafalse \fi
3274     \repeat
3275   \fi
3276 } \endgroup
3277 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3278 }
3279 \@onlypreamble\bxjs@get@ke@a
3280 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3281   \ifx$#1$\def\bxjs@tmpa{#2}%
3282   \else \let\bxjs@tmpa\relax
3283   \fi}

```

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

```

3284 \newcommand*\jachar[1]{%
3285   \begingroup
3286     \jsLetHeadChar\bxjs@tmpa{#1}%
3287     \ifx\bxjs@tmpa\relax
3288       \ClassWarningNoLine\bxjs@clsname
3289         {Illegal argument given to \string\jachar}%
3290     \else
3291       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3292     \fi
3293   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3294 \let\jsJaChar\jachar

```

下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
3295 \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。

```
3296 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3297 \@onlypreamble\bxjs@fix@hyperref@unicode
3298 \def\bxjs@fix@hyperref@unicode#1{%
3299   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3300   \@namedef{KV@Hyp@bxjs/hook}##1{%
3301     \KV@Hyp@unicode{##1}%
3302     \def\KV@Hyp@unicode####1{%
3303       \expandafter\ifx\csname if##1\expandafter\endcsname
3304       \csname if####1\endcsname\else
3305       \ClassWarningNoLine{bxjs@clsname
3306       {Blcoked hyperref option 'unicode=####1'}}%
3307     \fi
3308   }%
3309 }%
3310 }
```

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

```
3311 \@onlypreamble\bxjs@urgent@special
3312 \def\bxjs@urgent@special#1{%
3313   \AtBeginDvi{\special{#1}}%
3314   \AtBeginDocument{%
3315     \@ifpackageloaded{atbegshi}{%
3316       \begingroup
3317       \toks\z@{\special{#1}}%
3318       \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3319       \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3320     \endgroup
3321   }{}%
3322 }%
3323 }
```

C.2 pTeX 用設定

```
3324 \if j\jsEngine
```

■共通命令の実装

```
3325 \def\bxjs@apply@kanjiskip{%
3326   \kanjiskip\@tempskipa}
3327 \def\bxjs@apply@xkanjiskip{%
3328   \xkanjiskip\@tempskipa}
```

`\jaJaChar` のサブマクロ。

```
3329 \def\bxjs@jachar#1{%
3330   \bxjs@jachar@a#1...\@nil}
3331 \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。

```
3332   \ifx.#2#1%
```

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

```
3333   \else\ifx.#3%
3334     \@tempcnta`#1 \multiply\@tempcnta64
3335     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3336     \bxjs@jachar@b
3337   \else\ifx.#4%
3338     \@tempcnta`#1 \multiply\@tempcnta64
3339     \advance\@tempcnta`#2 \multiply\@tempcnta64
3340     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3341     \bxjs@jachar@b
3342   \else
3343     \@tempcnta`#1 \multiply\@tempcnta64
3344     \advance\@tempcnta`#2 \multiply\@tempcnta64
3345     \advance\@tempcnta`#3 \multiply\@tempcnta64
3346     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3347     \bxjs@jachar@b
3348   \fi\fi\fi}
```

符号値が `\@tempcnta` の和文文字を出力する処理。

```
3349 \ifjsWithupTeX
3350   \def\bxjs@jachar@b{\kchar\@tempcnta}
3351 \else
3352   \def\bxjs@jachar@b{%
3353     \ifx\bxUInt\@undefined\else
3354       \bxUInt{\@tempcnta}%
3355     \fi}
3356 \fi
```

和欧文間空白の命令 `\>` の実装。

```
3357 \ifbxjs@xkanjiskip@cmd
3358   \def\bxjs@put@xkanjiskip{%
3359     \relax\ifmmode \mskip\medmuskip
3360     \else \hskip\xkanjiskip
3361     \fi}
3362   \ifjsWithTeX \protected\def\>{\bxjs@put@xkanjiskip}
3363   \else \def\>{\protect\bxjs@put@xkanjiskip}
3364   \fi
3365 \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。

```
3366 \let\bxjs@tmpa\jsJaFont
3367 \ifx\bxjs@tmpa\bxjs@@auto
3368   \let\bxjs@tmpa\@empty
3369 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3370   \def\bxjs@tmpa{noembed}
3371 \fi\fi
3372 \ifx\jsJaFont\@empty\else
3373   \edef\bxjs@nxt{%
3374     \noexpand\RequirePackage[\jsJaFont]
3375       {pxchfon}[2010/05/12]}% v0.5
3376   \bxjs@nxt
3377 \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……。)

```
3378 \begingroup
3379   \global\let\@gtempa\relax
3380   \catcode`\|=0 \catcode`\|=12
3381   |def|bxjs@check#1|@nil{%
3382     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3383   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3384     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3385   |catcode`\|=0 \catcode`\|=12
3386   \def\bxjs@check@b#1keyval#2\@nnil{%
3387     \ifx$#2$\else
3388       \xdef\@gtempa{%
3389         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3390     \fi}
3391 \@firstofone{%
3392   \catcode10=12 \endlinechar\m@ne
3393   \let\do\@makeother \dospecials \catcode32=10
3394   \openin\@inputcheck=otf.sty\relax
3395   \@tempwatrue
3396   \loop\if@tempwa
3397     \ifeof\@inputcheck \@tempwafalse \fi
3398     \if@tempwa
3399       \read\@inputcheck to\bxjs@line
3400       \expandafter\bxjs@check\bxjs@line\@nil
3401     \fi
3402   \repeat
3403   \closein\@inputcheck
3404 \endgroup}
3405 \@gtempa
```

■hyperref 対策 unicode にはいけない。

```
3406 \ifbxjs@hyperref@enc
3407   \bxjs@fix@hyperref@unicode{false}
3408 \fi
```

tounicode special 命令を出力する。

```
3409 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3410   \else\ifjsWithpTeXng T\else F\fi\fi T%
3411   \ifnum\jis"2121="A1A1 %euc
3412     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3413   \else\ifnum\jis"2121="8140 %sjis
3414     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3415   \else\ifnum\jis"2121="3000 %uptex
3416     \ifbxjs@bigcode
3417       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3418       \PassOptionsToPackage{bigcode}{pxjahyper}
3419     \else
3420       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3421     \fi
3422   \fi\fi\fi
3423   \let\bxToUnicodeSpecialDone=t
3424 \fi
```

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

```
3425 \ifx f\bxjs@enablejfam\else
3426   \@enablejfamtrue
3427 \fi
```

実際に和文用の数式ファミリの設定を行う。

```
3428 \if@enablejfam
3429   \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3430   \DeclareSymbolFontAlphabet{\mathmc}{\mincho}
3431   \SetSymbolFont{mincho}{\bold}{\jsc@JYn}{gt}{m}{n}
3432   \jfam\symmincho
3433   \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3434   \AtBeginDocument{%
3435     \ifx\reDeclareMathAlphabet\undefined\else
3436       \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3437       \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3438       \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3439     \fi}
3440 \fi
```

C.3 pdfTeX 用設定 : CJK + bxcjkatype

```
3441 \else\if p\jsEngine
```

■bxcjkjatype パッケージの読込 \jsJaFont が指定されている場合は、その値を bxcjkjatype 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 bxcjkjatype の側で行われる。

```
3442 \bxjs@adjust@jafont{f}
3443 \edef\bxjs@nxt{%
3444   \noexpand\RequirePackage[%
3445     \ifx\bxjs@tmpa\@empty\else \bxjs@tmpa,\fi
3446     whole,autotilde]{bxcjkjatype}[2013/10/15]}% v0.2c
3447 \bxjs@nxt
3448 \bxjs@cjk@loaded
```

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

※取りあえず固定はしない。

```
3449 \ifbxjs@hyperref@enc
3450   \PassOptionsToPackage{unicode}{hyperref}
3451 \fi
```

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

※ bxcjkjatype を whole 付きで使っていることが前提。

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

```
3452 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3453 \begingroup
3454   \CJK@input{UTF8.bdg}
3455 \endgroup
3456 \g@addto@macro\pdfstringdefPreHook{%
3457   \@nameuse{CJK@UTF8Binding}%
3458 }
3459 \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。

```
3460 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3461 \g@addto@macro\pdfstringdefPreHook{%
3462   \ifx~\bxjs@@CJKtilde
3463     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3464     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3465     \let~\@empty
3466   \fi
3467 }
3468 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3469 \def\bxjs@@tildecmd{~}
3470 \def\bxjs@LetUnexpandableSpace#1{%
3471   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
3472     \bxjs@org@LetUnexpandableSpace#1%
3473   \fi}
```


3474 \fi

■共通命令の実装

```
3475 \newskip\jsKanjiSkip
3476 \newskip\jsXKanjiSkip
3477 \ifx\CJKecglue\@undefined
3478   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3479 \fi
3480 \let\autospacing\bxjs@enable@kanjiskip
3481 \let\noautospacing\bxjs@disable@kanjiskip
3482 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3483 \def\bxjs@apply@kanjiskip{%
3484   \jsKanjiSkip\@tempskipa
3485   \let\CJKglue\bxjs@CJKglue}
3486 \let\autoxspacing\bxjs@enable@xkanjiskip
3487 \let\noautoxspacing\bxjs@disable@xkanjiskip
3488 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3489 \def\bxjs@apply@xkanjiskip{%
3490   \jsXKanjiSkip\@tempskipa
3491   \let\CJKecglue\bxjs@CJKecglue}
```

\jachar のサブマクロの実装。

```
3492 \def\bxjs@jachar#1{%
3493   \CJKforced{#1}}
```

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

```
3494 \ifbxjs@xkanjiskip@cmd
3495   \protected\def\>{%
3496     \relax\ifmmode \mskip\medmuskip \else \CJKecglue\ignorespaces \fi}
3497 \fi
```

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

```
3498 \ifx t\bxjs@enablejfam
3499   \ClassWarningNoLine\bxjs@clsname
3500   {You cannot use 'enablejfam=true', since the\MessageBreak
3501     CJK package does not support Japanese math}
3502 \fi
```

C.4 Xe_{La}TeX 用設定：xeCJK + zxjatype

```
3503 \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。

```
3504 \RequirePackage{zxjatype}
3505 \PassOptionsToPackage{no-math}{fontspec}%!
3506 \PassOptionsToPackage{xetex}{graphicx}%!
3507 \PassOptionsToPackage{xetex}{graphics}%!
3508 \ifx\zxJaFamilyName\@undefined
```

```

3509 \ClassError\bxjs@clsname
3510 {xeCJK or zxjatype is too old}\@ehc
3511 \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。

```

3512 \bxjs@adjust@jafont{f}
3513 \ifx\bxjs@tmpa\@empty
3514 \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3515 \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3516 \else
3517 \edef\bxjs@nxt{%
3518 \noexpand\RequirePackage[\bxjs@tmpa]%
3519 {zxjafont}[2013/01/28]}% v0.2a
3520 \bxjs@nxt
3521 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

X_ƎTeX の場合は、xdvipdfmx が UTF-8 → UTF-16 の変換を行う機能を持っているため、本来は special 命令の文字列の文字コード変換は不要である。ところが、hyperref での方針としては、X_ƎTeX の場合にもパッケージ側で文字コード変換を行う方が望ましいと考えている。実際、unicode を無効にしていると警告が出て強制的に有効化される。一方で、過去 (r35125 まで) の xdvipdfmx では、文字列を UTF-16 に変換した状態で与えるのは不正と見なしていて警告が発生する。

これを踏まえて、ここでは、「X_ƎTeX のバージョンが 0.99992 以上の場合に unicode を既定で有効にする」ことにする。

※取りあえず固定はしない。

```

3522 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3523 \ifbxjs@hyperref@enc
3524 \PassOptionsToPackage{unicode}{hyperref}
3525 \fi
3526 \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。

```

3527 \let\jsInhibitGlueAtParTop\@empty

```

■共通命令の実装

```

3528 \newskip\jsKanjiSkip
3529 \newskip\jsXKanjiSkip
3530 \ifx\CJKecglue\@undefined

```

```

3531 \def\CJKtilde{\CJK@global\def~{\CJKe glue\ignorespaces}}
3532 \fi
3533 \let\autospacing\bxjs@enable@kanjiskip
3534 \let\noautospacing\bxjs@disable@kanjiskip
3535 \protected\def\bxjs@CJKe glue{\hskip\jsKanjiSkip}
3536 \def\bxjs@apply@kanjiskip{%
3537   \jsKanjiSkip\@tempskipa
3538   \xeCJKsetup{CJKe glue={\bxjs@CJKe glue}}}
3539 \let\autoxspacing\bxjs@enable@xkanjiskip
3540 \let\noautoxspacing\bxjs@disable@xkanjiskip
3541 \protected\def\bxjs@CJKe glue{\hskip\jsXKanjiSkip}
3542 \def\bxjs@apply@xkanjiskip{%
3543   \jsXKanjiSkip\@tempskipa
3544   \xeCJKsetup{CJKe glue={\bxjs@CJKe glue}}}

```

`\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。

```

3545 \ifx\mcfamily\undefined
3546   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3547   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3548 \fi

```

`\jachar` のサブマクロの実装。

```

3549 \def\bxjs@jachar#1{%
3550   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3551   #1}

```

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

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

```

3552 \ifbxjs@xkanjiskip@cmd
3553   \protected\def\>{%
3554     \relax\ifmmode \mskip\medmuskip \else \CJKe glue\ignorespaces \fi}
3555 \fi

```

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

```

3556 \ifx t\bxjs@enablejfam
3557   \@enablejfamtrue
3558 \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。

```

3559 \if@enablejfam
3560   \xeCJKsetup{CJKmath=true}
3561 \fi

```

C.5 Lua_TE_X 用設定 : Lua_TE_X-ja

```
3562 \else\if 1\jsEngine
```

■LuaTeX-j_a 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。

```
3563 \let\zw\@undefined
```

```
3564 \RequirePackage{luatexja}
```

```
3565 \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？

```
3566 \ExplSyntaxOn
```

```
3567 \fp_gset:Nn \g_ltj_fontspec_scale_fp { \jsScale }
```

```
3568 \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）。

```
3569 \bxjs@adjust@jafont{t}
```

```
3570 \ifx\bxjs@tmpa\bxjs@@noEmbed
```

```
3571 \def\bxjs@tmpa{noembed}
```

```
3572 \fi
```

```
3573 \ifx\bxjs@tmpa\@empty
```

```
3574 \defaultjfontfeatures{ Kerning=Off }
```

```
3575 \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
```

```
3576 \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
```

```
3577 \else
```

```
3578 \edef\bxjs@nxt{%
```

```
3579 \noexpand\RequirePackage[\bxjs@tmpa]
```

```
3580 {luatexja-preset}}%
```

```
3581 \bxjs@nxt
```

```
3582 \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。

```
3583 \DeclareRobustCommand\rmfamily
```

```
3584 {\not@math@alphabet\rmfamily\mathrm
```

```
3585 \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
```

```
3586 \DeclareRobustCommand\sffamily
```

```
3587 {\not@math@alphabet\sffamily\mathsf
```

```
3588 \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
```

```
3589 \DeclareRobustCommand\ttfamily
```

```
3590 {\not@math@alphabet\ttfamily\mathtt
```

```

3591 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3592 \AtBeginDocument{%
3593 \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3594 \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
3595 \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
3596 \bxjs@if@sf@default{%
3597 \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

3598 % 次の3つは既定値の通り
3599 \ltjsetparameter{prebreakpenalty={` ,10000}}
3600 \ltjsetparameter{postbreakpenalty={` “,10000}}
3601 \ltjsetparameter{prebreakpenalty={` ”,10000}}
3602 \ltjsetparameter{jaxspmde={` ! ,1}}
3603 \ltjsetparameter{jaxspmde={` 〒,2}}
3604 \ltjsetparameter{alxspmde={` +,3}}
3605 \ltjsetparameter{alxspmde={` %,3}}

```

■段落頭でのグルー挿入禁止

```

3606 \protected\def\@inhibitglue{%
3607 \directlua{%
3608 \luatexja.jfmglue.create_beginpar_node()}}
3609 \let\bxjs@ltj@inhibitglue\inhibitglue
3610 \let\@inhibitglue\undefined

```

■hyperref 対策 unicode にするべき。

```

3611 \ifbxjs@hyperref@enc
3612 \bxjs@fix@hyperref@unicode{true}
3613 \fi

```

■共通命令の実装

```

3614 \protected\def\autospacing{%
3615 \ltjsetparameter{autospacing=true}}
3616 \protected\def\noautospacing{%
3617 \ltjsetparameter{autospacing=false}}
3618 \protected\def\autoxspacing{%
3619 \ltjsetparameter{autoxspacing=true}}
3620 \protected\def\noautoxspacing{%
3621 \ltjsetparameter{autoxspacing=false}}
3622 \def\bxjs@apply@kanjiskip{%
3623 \ltjsetparameter{kanjiskip={\@tempskipa}}}}
3624 \def\bxjs@apply@xkanjiskip{%
3625 \ltjsetparameter{xkanjiskip={\@tempskipa}}}}

```

\jachar のサブマクロの実装。

```

3626 \def\bxjs@jachar#1{%
3627 \ltjjachar`#1\relax}

```

\> の実装。

```

3628 \ifbxjs@xkanjiskip@cmd
3629   \protected\def\>{%
3630     \relax\ifmmode \mskip\medmuskip
3631     \else \hskip\ltjgetparameter{xkanjiskip}\ignorespaces
3632     \fi}
3633 \fi

```

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

```

3634 \ifx f\bxjs@enablejfam
3635   \ClassWarningNoLine{bxjs@clsname}
3636   {You cannot use 'enablejfam=false', since the\MessageBreak
3637     LuaTeX-jd always provides Japanese math families}
3638 \fi

```

C.6 共通処理 (2)

```

3639 \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 3640 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3641 \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。

```

3642 \ifx\mathmc\undefined
3643   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
3644   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
3645 \fi

```

■和文・和欧文間空白の初期値

```

3646 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3647 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3648 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3649 \fi

```

以上で終わり。

```

3650 %</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。

```

3651 %<*modern>
3652 \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。

```
3653 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3654 \def\encodingdefault{T1}%
3655 \input{t1enc.def}%
3656 \fontencoding\encodingdefault\selectfont
3657 \fi
```

基本フォントを Latin Modern フォントファミリに変更する。

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

```
3658 \ifnum0\if x\jsEngine1\fi\if 1\jsEngine1\fi=\z@
3659 \renewcommand{\rmdefault}{lmr}
3660 \renewcommand{\sfdefault}{lms}
3661 \renewcommand{\ttdefault}{lmtt}
3662 \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。

```
3663 \DeclareFontShape{OMX}{cmex}{m}{n}{%
3664   <-7.5>cmex7<7.5-8.5>cmex8%
3665   <8.5-9.5>cmex9<9.5->cmex10}{}%
3666 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
    amsmath 読み込時に上書きされるのを防ぐ。
3667 \def\cmex@opt{10}
```

D.2 fixltx2e 読み込

※ `fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```
3668 \ifx\@IncludeInRelease\@undefined
3669 \RequirePackage{fixltx2e}
3670 \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。

```
3671 \RequirePackage{bxjsjkat}
```

D.4 完了

おしまい。

```
3672 %</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。

```
3673 %<*pandoc>
```

```
3674 \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。

```
3675 \@onlypreamble\bxjs@set@dupload@proc
```

```
3676 \def\bxjs@set@dupload@proc#1{%
```

```
3677   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
3678 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
3679 \def\bxjs@set@dupload@proc@a#1{%
```

```
3680   \@onlypreamble#1\def#1##1}
```

`\@if@options` `\@if@options` の再定義。

```
3681 \@onlypreamble\bxjs@org@if@options
```

```
3682 \let\bxjs@org@if@options\@if@options
```

```
3683 \newif\ifbxjs@dlp
```

```
3684 \def\@if@options#1#2#3{%
```

```
3685   \bxjs@dlpfalse
```

```
3686   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@current}%
```

```
3687   \ifx\bxjs@tmpa\bxjs@tmpb
```

```
3688     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
```

```
3689       \bxjs@dlptrue \fi
```

```
3690   \fi
```

```
3691   \ifbxjs@dlp \expandafter\bxjs@do@dupload@proc
```

```
3692   \else \expandafter\bxjs@org@if@options
```

```
3693   \fi {#1}{#2}{#3}}
```

```
3694 \AtBeginDocument{%
```

```
3695   \let\@if@options\bxjs@org@if@options}
```

```
3696 \@onlypreamble\bxjs@do@dupload@proc
```

```
3697 \def\bxjs@do@dupload@proc#1#2#3{%
```

```
3698   \csname bxjs@dlp/#2.#1\endcsname{#3}%
```

```
3699   \@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。


```

3700 \def\bxjs@mark@as@loaded#1{%
3701   \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3702 \def\bxjs@mal@a#1#2{%
3703   \ifx#1\relax
3704     \def#1{2001/01/01}%
3705     \ClassInfo\bxjs@clsname
3706       {File '#2' marked as loaded\@gobble}%
3707   \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。

```

3708 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0

```

Polyglossia について。

```

3709 \bxjs@mark@as@loaded{polyglossia.sty}
3710 \bxjs@set@dupload@proc{polyglossia.sty}{%
3711   \ClassWarning\bxjs@clsname
3712     {Loading of polyglossia is blocked}}
3713 \ifx\setmainlanguage\@undefined
3714 \newcommand*\setmainlanguage[2][]{%
3715 \newcommand*\setotherlanguage[2][]{%
3716   \ifcat_#2_\else
3717     \expandafter\let\csname #2\endcsname\@empty
3718     \expandafter\let\csname end#2\endcsname\@empty
3719     \expandafter\let\csname text#2\endcsname\@firstofone
3720   \fi}
3721 \newcommand*\setotherlanguages[2][]{%
3722   \@for\bxjs@tmpa:={#2}\do{%
3723     \setotherlangauge{\bxjs@tmpa}}}
3724 \fi
3725 \else

```

Babel について。

```

3726 \bxjs@mark@as@loaded{babel.sty}
3727 \bxjs@set@dupload@proc{babel.sty}{%
3728   \ClassWarning\bxjs@clsname
3729     {Loading of babel is blocked}}
3730 \let\foreignlanguage\@secondoftwo
3731 \let\otherlanguage\@gobble
3732 \let\endotherlanguage\@empty
3733 \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。

```
3734 \bxjs@set@dupload@proc{geometry.sty}{%
3735   \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。

```
3736 \if 1\jsEngine
3737   \bxjs@mark@as@loaded{xeCJK.sty}
3738   \providecommand*\setCJKmainfont{\setmainfont}
3739 \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。

```
3740 \ifx\@IncludeInRelease\@undefined\else
3741   \bxjs@mark@as@loaded{fixltx2e.sty}
3742 \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。)

```
3743 \if j\jsEngine
3744   \bxjs@mark@as@loaded{cmap.sty}
3745 \fi
```

E.7 microtype パッケージ

警告が多すぎなので消す。

```
3746 \if j\jsEngine \else
3747   \PassOptionsToPackage{verbose=silent}{microtype}
3748 \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。

```
3749 \if j\jsEngine
3750   \bxjs@mark@as@loaded{microtype.sty}
3751   \newcommand*\UseMicrotypeSet[2][]{\fi}
3752 \fi
```

E.8 完了

おしまい。

```
3753 %</pandoc>
```

和文ドライバ実装はここまで。

```
3754 %</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。

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

```
3755 %<*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

```
3756 %<*compat>
3757 \def\bxac@pkgname{bxjscompat}

\bxjx@engine エンジンの種別。

3758 \let\bxac@engine=n
3759 \def\bxac@do#1#2{%
3760   \edef\bxac@tmpa{\string#1}%
3761   \edef\bxac@tmpb{\meaning#1}%
3762   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
3763 \bxac@do\XeTeXversion{\let\bxac@engine=x}
3764 \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。

```
3765 \ifx\jsAtEndOfClass\undefined
3766   \let\bxac@delayed@if@bxjs\@firstofone
```

```

3767 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
3768 \fi

```

\ImposeOldLuaTeXBehavior ムニャムニャ。

```

\RevokeOldLuaTeXBehavior 3769 \newif\ifbxac@in@old@behavior
3770 \let\ImposeOldLuaTeXBehavior\relax
3771 \let\RevokeOldLuaTeXBehavior\relax

```

G.2 XeTeX 部分

```

3772 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。
3773 \@onlypreamble\bxac@adjust@charclass
3774 \bxac@delayed@if@bxjs{%
3775   \@ifpackageloaded{xeCJK}{-}{%else
3776     \ifx\XeTeX@alloc@intercharclass\undefined\else
3777       \ifnum\XeTeX@alloc@intercharclass=\z@
3778         \PackageInfo\bxac@pkgname
3779         {Setting up interchar class for CJK...\@gobble}%
3780         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
3781           \XeTeX@alloc@intercharclass=3
3782         }{%else
3783           \PackageWarning\bxac@pkgname
3784             {Cannot find file 'load-unicode-xetex-classes.tex'%
3785             \@gobble}%
3786         }%
3787       \fi\fi
3788       \ifnum\XeTeX@charclass"3041=\z@
3789         \PackageInfo\bxac@pkgname
3790         {Adjusting interchar class for CJK...\@gobble}%
3791         \@for\bxac@x:={%
3792           3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3793           3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3794           30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3795           31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3796           31FF%
3797         }\do{\XeTeX@charclass"\bxac@x=\@ne}%
3798       \fi
3799     }%
3800 }

```

以上。

```

3801 \fi

```

G.3 LuaTeX 部分

```

3802 \ifx l\bxac@engine

ムニャムニャ。
3803 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85

```

```

3804 \chardef\pdftexversion=200
3805 \def\pdftexrevision{0}
3806 \let\pdftexbanner\luatexbanner
3807 \fi\fi

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 3808 \begingroup\expandafter\expandafter\expandafter\endgroup
3809 \expandafter\ifx\cename outputmode\endcsname\relax\else
3810 \def\bxac@ob@list{%
3811   \do{\let}\pdfoutput{\outputmode}%
3812   \do{\let}\pdfpagewidth{\pagewidth}%
3813   \do{\let}\pdfpageheight{\pageheight}%
3814   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
3815   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
3816 \def\bxac@ob@do#1#2{\begingroup
3817   \expandafter\bxac@ob@do@a\cename bxac@\string#2\endcsname{#1}#2}
3818 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3819   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3820   \else \let#3#1\relax \let#1\@undefined
3821   \fi}
3822 \protected\def\ImposeOldLuaTeXBehavior{%
3823   \unless\ifbxac@in@old@behavior
3824     \bxac@in@old@behaviortrue
3825     \let\do\bxac@ob@do \bxac@ob@list
3826   \fi}
3827 \protected\def\RevokeOldLuaTeXBehavior{%
3828   \ifbxac@in@old@behavior
3829     \bxac@in@old@behaviorfalse
3830     \let\do\bxac@ob@do \bxac@ob@list
3831   \fi}
3832 \fi

```

漢字および完成形ハングルのカテゴリコードのムニヤムニヤ。

```

3833 \ifnum\luatexversion>64 \directlua{
3834   local function range(cs, ce, cc, ff)
3835     if ff or not tex.getcatcode(cs) == cc then
3836       local setcc = tex.setcatcode
3837       for c = cs, ce do setcc(c, cc) end
3838     end
3839   end
3840   range(0x3400, 0x4DB5, 11, false)
3841   range(0x4DB5, 0x4DBF, 11, true)
3842   range(0x4E00, 0x9FCC, 11, false)
3843   range(0x9FCD, 0x9FFF, 11, true)
3844   range(0xAC00, 0xD7A3, 11, false)
3845   range(0x20000, 0x2A6D6, 11, false)
3846   range(0x2A6D7, 0x2A6FF, 11, true)
3847   range(0x2A700, 0x2B734, 11, false)
3848   range(0x2B735, 0x2B73F, 11, true)

```

```

3849     range(0x2B740, 0x2B81D, 11, false)
3850     range(0x2B81E, 0x2B81F, 11, true)
3851     range(0x2B820, 0x2CEA1, 11, false)
3852     range(0x2CEA2, 0x2FFFD, 11, true)
3853 } \fi

```

以上。

```

3854 \fi

```

G.4 完了

おしまい。

```

3855 %</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

```

3856 %<*cjkcat>
3857 \def\bxjx@pkgname{bxjscjkat}
3858 \newcount\bxjx@canta
\bxjx@engine エンジンの種別。
3859 \let\bxjx@engine=n
3860 \def\bxjx@do#1#2{%
3861   \edef\bxjx@tmpa{\string#1}%
3862   \edef\bxjx@tmpb{\meaning#1}%
3863   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3864 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3865 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3866 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3867 \bxjx@do\pdftexversion{\let\bxjx@engine=p}
3868 \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。

```

3869 \def\bxjx@do#1#2{%
3870   \if#1\bxjx@engine
3871     \ifpackageloaded{#2}{\%else
3872       \PackageError\bxjx@pkgname
3873         {Package '#2' must be loaded}%
3874         {Package loading is aborted.\MessageBreak\@ehc}%
3875     \endinput}
3876   \fi}
3877 \bxjx@do{p}{\bxcjkatype}

```

```

3878 \bxjx@do{x}{xeCJK}
3879 \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。

```

3880 \ifx\TextOrMath\undefined
3881   \RequirePackage{fixltx2e}
3882 \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。

```

3883 \if u\bxjx@engine
3884 \@for\bxjx@x:={%
3885 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3886 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3887 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3888 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3889 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3890 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
3891 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3892 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,44D0,A500,A640,%
3893 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3894 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
3895 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3896 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3897 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3898 10480,10500,10530,10600,10800,10840,10860,10880,%
3899 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
3900 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3901 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3902 11280,112B0,11300,11480,11580,11600,11680,11700,%
3903 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3904 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3905 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3906 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
3907 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3908 }\do{\kcatcode"\bxjx@x=15 }
3909 \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。

```

3910 \def\bxjx@grkcyr@list{%
3911 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
3912 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
3913 \do{0393}{LGR}{\textGamma}{\Gamma}%     % GR. C. L. GAMMA
3914 \do{0394}{LGR}{\textDelta}{\Delta}%     % GR. C. L. DELTA
3915 \do{0395}{LGR}{\textEpsilon}{E}%        % GR. C. L. EPSILON
3916 \do{0396}{LGR}{\textZeta}{Z}%           % GR. C. L. ZETA
3917 \do{0397}{LGR}{\textEta}{H}%            % GR. C. L. ETA
3918 \do{0398}{LGR}{\textTheta}{\Theta}%     % GR. C. L. THETA
3919 \do{0399}{LGR}{\textIota}{I}%           % GR. C. L. IOTA
3920 \do{039A}{LGR}{\textKappa}{K}%          % GR. C. L. KAPPA
3921 \do{039B}{LGR}{\textLambda}{\Lambda}%    % GR. C. L. LAMDA
3922 \do{039C}{LGR}{\textMu}{M}%             % GR. C. L. MU
3923 \do{039D}{LGR}{\textNu}{N}%             % GR. C. L. NU
3924 \do{039E}{LGR}{\textXi}{\Xi}%           % GR. C. L. XI
3925 \do{039F}{LGR}{\textOmicron}{O}%        % GR. C. L. OMICRON
3926 \do{03A0}{LGR}{\textPi}{\Pi}%           % GR. C. L. PI
3927 \do{03A1}{LGR}{\textRho}{P}%            % GR. C. L. RHO
3928 \do{03A3}{LGR}{\textSigma}{\Sigma}%     % GR. C. L. SIGMA
3929 \do{03A4}{LGR}{\textTau}{T}%             % GR. C. L. TAU
3930 \do{03A5}{LGR}{\textUpsilon}{\Upsilon}%  % GR. C. L. UPSILON
3931 \do{03A6}{LGR}{\textPhi}{\Phi}%          % GR. C. L. PHI
3932 \do{03A7}{LGR}{\textChi}{X}%            % GR. C. L. CHI
3933 \do{03A8}{LGR}{\textPsi}{\Psi}%          % GR. C. L. PSI
3934 \do{03A9}{LGR}{\textOmega}{\Omega}%     % GR. C. L. OMEGA
3935 \do{03B1}{LGR}{\textalpha}{\alpha}%     % GR. S. L. ALPHA
3936 \do{03B2}{LGR}{\textbeta}{\beta}%       % GR. S. L. BETA
3937 \do{03B3}{LGR}{\textgamma}{\gamma}%     % GR. S. L. GAMMA
3938 \do{03B4}{LGR}{\textdelta}{\delta}%     % GR. S. L. DELTA
3939 \do{03B5}{LGR}{\textepsilon}{\epsilon}%  % GR. S. L. EPSILON
3940 \do{03B6}{LGR}{\textzeta}{\zeta}%        % GR. S. L. ZETA
3941 \do{03B7}{LGR}{\texteta}{\eta}%           % GR. S. L. ETA
3942 \do{03B8}{LGR}{\texttheta}{\theta}%     % GR. S. L. THETA
3943 \do{03B9}{LGR}{\textiota}{\iota}%        % GR. S. L. IOTA
3944 \do{03BA}{LGR}{\textkappa}{\kappa}%      % GR. S. L. KAPPA
3945 \do{03BB}{LGR}{\textlambda}{\lambda}%    % GR. S. L. LAMDA
3946 \do{03BC}{LGR}{\textmu}{\mu}%           % GR. S. L. MU
3947 \do{03BD}{LGR}{\textnu}{\nu}%           % GR. S. L. NU
3948 \do{03BE}{LGR}{\textxi}{\xi}%           % GR. S. L. XI
3949 \do{03BF}{LGR}{\textomicron}{o}%          % GR. S. L. OMICRON
3950 \do{03C0}{LGR}{\textpi}{\pi}%           % GR. S. L. PI
3951 \do{03C1}{LGR}{\textrho}{\rho}%          % GR. S. L. RHO
3952 \do{03C2}{LGR}{\textvarsigma}{\varsigma}% % GR. S. L. FINAL SIGMA
3953 \do{03C3}{LGR}{\textsigma}{\sigma}%     % GR. S. L. SIGMA
3954 \do{03C4}{LGR}{\texttau}{\tau}%          % GR. S. L. TAU
3955 \do{03C5}{LGR}{\textupsilon}{\upsilon}%  % GR. S. L. UPSILON
3956 \do{03C6}{LGR}{\textphi}{\phi}%          % GR. S. L. PHI
3957 \do{03C7}{LGR}{\textchi}{\chi}%          % GR. S. L. CHI
3958 \do{03C8}{LGR}{\textpsi}{\psi}%          % GR. S. L. PSI

```


3959 \do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
3960 \do{0401}{T2A}{\CYRYO}{}%	% CY. C. L. IO
3961 \do{0410}{T2A}{\CYRA}{}%	% CY. C. L. A
3962 \do{0411}{T2A}{\CYRB}{}%	% CY. C. L. BE
3963 \do{0412}{T2A}{\CYRV}{}%	% CY. C. L. VE
3964 \do{0413}{T2A}{\CYRG}{}%	% CY. C. L. GHE
3965 \do{0414}{T2A}{\CYRD}{}%	% CY. C. L. DE
3966 \do{0415}{T2A}{\CYRE}{}%	% CY. C. L. IE
3967 \do{0416}{T2A}{\CYRZH}{}%	% CY. C. L. ZHE
3968 \do{0417}{T2A}{\CYRZ}{}%	% CY. C. L. ZE
3969 \do{0418}{T2A}{\CYRI}{}%	% CY. C. L. I
3970 \do{0419}{T2A}{\CYRISHRT}{}%	% CY. C. L. SHORT I
3971 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
3972 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
3973 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
3974 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
3975 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
3976 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
3977 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
3978 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
3979 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
3980 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
3981 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
3982 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
3983 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
3984 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
3985 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
3986 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
3987 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
3988 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
3989 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
3990 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
3991 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
3992 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
3993 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
3994 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
3995 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
3996 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
3997 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
3998 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
3999 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
4000 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
4001 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
4002 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
4003 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
4004 \do{043B}{T2A}{\cyrl}{}%	% CY. S. L. EL
4005 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
4006 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
4007 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O

```

4008 \do{043F}{T2A}{\cyrp}{}%           % CY. S. L. PE
4009 \do{0440}{T2A}{\cyrr}{}%           % CY. S. L. ER
4010 \do{0441}{T2A}{\cyrs}{}%           % CY. S. L. ES
4011 \do{0442}{T2A}{\cyrt}{}%           % CY. S. L. TE
4012 \do{0443}{T2A}{\cyrU}{}%           % CY. S. L. U
4013 \do{0444}{T2A}{\cyrF}{}%           % CY. S. L. EF
4014 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
4015 \do{0446}{T2A}{\cyrC}{}%           % CY. S. L. TSE
4016 \do{0447}{T2A}{\cyrch}{}%          % CY. S. L. CHE
4017 \do{0448}{T2A}{\cyrsh}{}%          % CY. S. L. SHA
4018 \do{0449}{T2A}{\cyrshch}{}%        % CY. S. L. SHCHA
4019 \do{044A}{T2A}{\cyrhrdsn}{}%       % CY. S. L. HARD SIGN
4020 \do{044B}{T2A}{\cyrery}{}%         % CY. S. L. YERU
4021 \do{044C}{T2A}{\cyrsoftsn}{}%      % CY. S. L. SOFT SIGN
4022 \do{044D}{T2A}{\cyrerev}{}%        % CY. S. L. E
4023 \do{044E}{T2A}{\cyrYU}{}%          % CY. S. L. YU
4024 \do{044F}{T2A}{\cyrYA}{}%          % CY. S. L. YA
4025 \do{0451}{T2A}{\cyrYO}{}%          % CY. S. L. IO
4026 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
4027 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
4028 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
4029 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
4030 \do{00B4}{TS1}{\textasciicute}{}%  % ACUTE ACCENT
4031 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
4032 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
4033 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
4034 }

4035 \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。

```
4036 \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。

```

4037 \newcommand*\greekasCJK{%
4038   \bxjx@gcc@cjktrue}
4039 \newcommand*\nogreekasCJK{%
4040   \bxjx@gcc@cjkfalse}

```

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

```

4041 \def\bxjx@do#1\relax{%
4042   \def\bxjx@fake@grk##1##2{%
4043     \expandafter\bxjx@fake@grk@a\meaning##2#1\@nil{##1}{##2}}%
4044   \def\bxjx@fake@grk@a##1#1##2\@nil##3##4{%
4045     \ifx\##1\%
4046       \bxjx@canta##4\divide\bxjx@canta\@cclvi
4047       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
4048       \mathchar\bxjx@canta

```

```

4049 \else ##3\fi}
4050 }\expandafter\bxjx@do\string\mathchar\relax

```

■pdfLaTeX・upLaTeX の場合

```

4051 \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。

```

4052 \@ifpackageloaded{inputenc}{\fi}{%else
4053 \RequirePackage[utf8]{inputenc}}
4054 \def\bxjx@tmpa{utf8}
4055 \ifx\bxjx@tmpa\inputencdoingname
4056 \PackageWarningNoLine\bxjx@pkgname
4057 {Input encoding changed to utf8}%
4058 \inputencoding{utf8}%
4059 \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。

```

4060 \if u\bxjx@engine
4061 \kcatcode"0370=15
4062 \kcatcode"0400=15
4063 \kcatcode"0500=15
4064 \fi

```

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

```

4065 \def\do#1{%
4066 \@tempcnta="#1\relax
4067 \@tempcntb=\@tempcnta \divide\@tempcntb256
4068 \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
4069 \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
4070 \def\bxjx@do@a#1#2#3#4#5{%
4071 \ifx\#5\%
4072 \def\bxjx@tmpa{\@inmathwarn#4}%
4073 \else\ifcat A\noexpand#5%
4074 \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4075 {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4076 \else \def\bxjx@tmpa{#5}%
4077 \fi\fi
4078 \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
4079 \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
4080 \if u\bxjx@engine
4081 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
4082 \def\bxjx@do@b#1#2#3#4#5{%
4083 \kchardef#1=\@tempcnta
4084 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
4085 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
4086 \else\if p\bxjx@engine
4087 \def\bxjx@do@b#1#2#3#4#5{%
4088 \mathchardef#1=\@tempcnta
4089 \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{UTF{#2}}{#3}{#4}}%

```

```

4090 \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}
4091 \fi\fi
4092 \bxjx@grkcyr@list
4093 \let\bxjx@do@a\undefined
4094 \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。

```

4095 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
4096 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
4097 \@onlypreamble\bxjx@DeclareUnicodeCharacter
4098 \def\bxjx@DeclareUnicodeCharacter#1#2{%
4099   \count@=#1\relax \bxjx@cmta\count@ \divide\bxjx@cmta256
4100   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cmta\endcsname\relax
4101     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4102   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
4103     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
4104   \else
4105     \wlog{ \space\space skipped defining Unicode char U+#1}%
4106   \fi\fi}

```

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

```

4107 \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。

```

4108 \ifbxjx@gcc@CJK #1%

```

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

```

4109 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
4110 \else \UseTextSymbol{#2}{#3}%
4111 \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。

```

4112 \begingroup
4113 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
4114 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
4115   \noexpand\bxjx@swap@DUC@cmd
4116   \the\toks@
4117   \noexpand\bxjx@swap@DUC@cmd}}
4118 \endgroup\next
4119 \def\bxjx@swap@DUC@cmd{%
4120   \let\bxjx@tmpa\DeclareUnicodeCharacter
4121   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4122   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■Xe_{La}TeX・Lua_{La}TeX の場合

```
4123 \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。

```
4124 \def\do#1{%
4125   \bxjx@canta="#1\relax
4126   \begingroup
4127     \lccode`~=\bxjx@canta
4128     \lowercase{\endgroup
4129     \bxjx@do@a{~}}{#1}}
4130 \def\bxjx@do@a#1#2#3#4#5{%
4131   \ifx\\#5\\\let\bxjx@tmpa\relax
4132   \else\ifcat A\noexpand#5%
4133     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4134       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
4135   \else \def\bxjx@tmpa{#5}%
4136   \fi\fi
4137   \ifx\bxjx@tmpa\relax\else
4138     \mathcode\bxjx@canta"8000 \let#1\bxjx@tmpa
4139   \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。

```
4140 \mathchardef\bxjx@tmpa="119
4141 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4142 \let\bxjx@do@a\undefined
```

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

```
4143 \if 1\bxjx@engine
4144   \protected\def\greekasCJK{%
4145     \bxjx@gcc@cjktrue
4146     \ltjsetparameter{jacharrange={+2, +8}}
4147   \protected\def\nogreekasCJK{%
4148     \bxjx@gcc@cjkfalse
4149     \ltjsetparameter{jacharrange={-2, -8}}
4150   \fi
```

Xe_{La}TeX における \(\text{no}\)greekasCJK の定義。

```
4151 \if x\bxjx@engine
4152   \protected\def\greekasCJK{%
4153     \bxjx@gcc@cjktrue
4154     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4155     \bxjx@grkcyr@list}
4156   \protected\def\nogreekasCJK{%
4157     \bxjx@gcc@cjkfalse
4158     \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4159     \bxjx@grkcyr@list}
4160   \fi
```

以上。

4161 \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。

4162 \nogreekasCJK

H.5 完了

おしまい。

4163 %</cjkat>

補助パッケージ実装はここまで。

4164 %</anc>