

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”)

v1.2 [2016/08/01]

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

目次

1	はじめに	1
2	オプション	4
3	和文フォントの変更	26
4	フォントサイズ	26
5	レイアウト	32
5.1	ページレイアウト	33
6	ページスタイル	40
7	文書のマークアップ	43
7.1	表題	43
7.2	章・節	48
7.3	リスト環境	59
7.4	パラメータの設定	66
7.5	フロート	67
7.6	キャプション	68
8	フォントコマンド	70
9	相互参照	72

9.1	目次の類	72
9.2	参考文献	77
9.3	索引	79
9.4	脚注	80
10	段落の頭へのグルー挿入禁止	82
11	いろいろなロゴ	85
12	amsmath との衝突の回避	85
13	初期設定	85
付録 A	和文ドライバの仕様 ㊦	89
付録 B	和文ドライバ : minimal ㊦	90
B.1	補助マクロ	90
B.2	(u)pTeX 用の設定	92
B.3	pdfTeX 用の処理	96
B.4	X _Y TeX 用の処理	96
B.5	後処理 (エンジン共通)	97
付録 C	和文ドライバ : standard ㊦	98
C.1	共通処理 (1)	98
C.2	pTeX 用設定	103
C.3	pdfTeX 用設定 : CJK + bxcjkatype	105
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	107
C.5	LuaTeX 用設定 : LuaTeX-ja	108
C.6	共通処理 (2)	110
付録 D	和文ドライバ : modern ㊦	111
D.1	フォント設定	111
D.2	fixltx2e 読込	111
D.3	和文カテゴリコード	112
D.4	完了	112
付録 E	和文ドライバ : pandoc ㊦	112
E.1	dupload システム	112
E.2	lang 変数	113
E.3	geometry 変数	114
E.4	CJKmainfont 変数	114
E.5	fixltx2e パッケージ	114
E.6	cmap パッケージ	114

E.7	microtype パッケージ	115
E.8	完了	115
付録 F	補助パッケージ一覧 𐄂	115
付録 G	補助パッケージ : bxjscompat 𐄂	115
G.1	準備	115
G.2	X _Y TeX 部分	116
G.3	LuaTeX 部分	116
G.4	完了	118
付録 H	補助パッケージ : bxjscjcat 𐄂	118
H.1	準備	118
H.2	和文カテゴリコードの設定	119
H.3	ギリシャ・キリル文字の扱い	119
H.4	初期設定	126
H.5	完了	126

1 はじめに

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

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

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

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

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

以下では実際のコードに即して説明します。

■ BXJS クラス特有の設定 𐄂

```
1 %<*cls>
```

```

2 %% このファイルは日本語文字を含みます
長さ値の指定で式を利用可能にするため calc を読み込む。
3 \RequirePackage{calc}
クラスオプションで key-value 形式を使用するため keyval を読み込む。
4 \RequirePackage{keyval}
クラスの本体ではこの他に geometry パッケージが読み込まれる。
  互換性のための補助パッケージを読み込む。
5 \IfFileExists{bxjscompat.sty}{%
6   \let\jsAtEndOfClass\@gobble
7   \RequirePackage{bxjscompat}%
8 }{}

\jsEngine [暗黙文字トークン] エンジン (TEX の種類) の種別 : n = オリジナル TEX、j = pTEX また
は upTEX、J = NTT jTEX、x = XqTEX、p = pdfTEX (含 DVI モード)、l = LuaTEX。
9 \let\jsEngine=n
10 \def\bxjs@test@engine#1#2{%
11   \edef\bxjs@tmpa{\string#1}%
12   \edef\bxjs@tmpb{\meaning#1}%
13   \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
14 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
15 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
16 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
17 \bxjs@test@engine\pdftexversion{\let\jsEngine=p}
18 \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。
19 \newif\ifjsWithupTeX
20 \ifx\ucs\@undefined\else \ifnum\ucs"3000="3000
21   \jsWithupTeXtrue
22 \fi\fi
23 \let\if@jsc@uplatex\ifjsWithupTeX

\ifjsWitheTeX [スイッチ] エンジンが ε-TEX 拡張をもつか。
24 \newif\ifjsWitheTeX
25 \bxjs@test@engine\eTeXversion{\jsWitheTeXtrue}

\bxjs@protected ε-TEX 拡張が有効な場合にのみ \protected の効果をもつ。
26 \ifjsWitheTeX \let\bxjs@protected\protected
27 \else \let\bxjs@protected\@empty
28 \fi

\bxjs@robust@def 無引数の頑強な命令を定義する。
29 \ifjsWitheTeX
30   \def\bxjs@robust@def{\protected\def}
31 \else
32   \def\bxjs@robust@def{\DeclareRobustCommand*}
33 \fi

```

`\ifjsInPdfMode` [スイッチ] pdf \TeX / Lua \TeX が PDF モードで動作しているか。
 ※ Lua \TeX 0.8x 版でのプリミティブ名変更に対応。

```

34 \newif\ifjsInPdfMode
35 \@nameuse{ImposeOldLuaTeXBehavior}
36 \let\bxjs@tmpa\PackageWarningNoLine
37 \let\PackageWarningNoLine\PackageInfo % suppress warning
38 \RequirePackage{ifpdf}
39 \let\PackageWarningNoLine\bxjs@tmpa
40 \@nameuse{RevokeOldLuaTeXBehavior}
41 \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。

```

42 \@gobbletwo\if\if \def\bxjs@cond#1\fi{%
43   #1\expandafter\@firstoftwo
44   \else\expandafter\@secondoftwo
45   \fi}

```

`\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`。

```

46 \let\jsArticle=a
47 \let\jsBook=b
48 \let\jsReport=r
49 \let\jsSlide=s
50 %<article>\let\jsDocClass\jsArticle
51 %<article>\def\bxjs@clsname{bxjsarticle}
52 %<book>\let\jsDocClass\jsBook
53 %<book>\def\bxjs@clsname{bxjsbook}
54 %<report>\let\jsDocClass\jsReport
55 %<report>\def\bxjs@clsname{bxjsreport}
56 %<slide>\let\jsDocClass\jsSlide
57 %<slide>\def\bxjs@clsname{bxjsslide}

```

`\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。)

```

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

```

Lua \TeX 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 Lua $\mathcal{T}\mathcal{E}\mathcal{X}$ 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。

```

60 \@onlypreamble\bxjs@restore@jltrcc
61 \let\bxjs@restore@jltrcc\@empty
62 \if 1\jsEngine
63 \def\bxjs@change@jltrcc#1{%
64   \xdef\bxjs@restore@jltrcc{%

```

```

65 \bxjs@restore@jltrcc
66 \catcode`#1=\the\catcode`#1\relax}%
67 \catcode`#1=11\relax}
68 \@tfor\bxjs@x:=西暦\do
69 {\expandafter\bxjs@change@jltrcc\bxjs@x}
70 \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。

```

71 \bxjs@robust@def\jsInhibitGlue{%
72 \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。

```

73 \if@compatibility
74 \ClassError\bxjs@clsname
75 {Something went chaotic!\MessageBreak
76 I cannot go a single step further...}
77 {If the chant of \string\documentstyle\space was just a blunder
78 of yours,\MessageBreak then there'll still be hope....}
79 \expandafter\@firstofone
80 \else \expandafter\@gobble
81 \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。
82 \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。
83 \newif\if@titlepage

\if@openright \chapter, \part を奇数ページ起こしにするかどうかです。書籍では真が標準です。
84 %<book|report>\newif\if@openright

\if@mainmatter 真なら本文，偽なら前付け・後付けです。偽なら \chapter で章番号が出ません。
85 %<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。
86 \newif\if@enablejfam \@enablejfamtrue

```

以下で各オプションを宣言します。

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

B 判は JIS と ISO で定義が異なります。JIS では B0 判の面積が 1.5 m^2 ですが，ISO では B1 判の辺の長さが A0 判と A1 判の辺の長さの幾何平均です。したがって ISO の B0 判は $1000\text{ mm} \times 1414\text{ mm}$ です。このため， $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 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。

```

87 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
88 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
89 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
90 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
91 \DeclareOption{a6paper}{\bxjs@setpaper{a6paper}}
92 \DeclareOption{b4paper}{\bxjs@setpaper{{257true mm}{364true mm}}}
93 \DeclareOption{b5paper}{\bxjs@setpaper{{182true mm}{257true mm}}}
94 \DeclareOption{b6paper}{\bxjs@setpaper{{128true mm}{182true mm}}}
95 \DeclareOption{a4j}{\bxjs@setpaper{a4paper}}
96 \DeclareOption{a5j}{\bxjs@setpaper{a5paper}}
97 \DeclareOption{b4j}{\bxjs@setpaper{{257true mm}{364true mm}}}
98 \DeclareOption{b5j}{\bxjs@setpaper{{182true mm}{257true mm}}}
99 \DeclareOption{a4var}{\bxjs@setpaper{{210true mm}{283true mm}}}
100 \DeclareOption{b5var}{\bxjs@setpaper{{182true mm}{230true mm}}}
101 \DeclareOption{letterpaper}{\bxjs@setpaper{letterpaper}}
102 \DeclareOption{legalpaper}{\bxjs@setpaper{legalpaper}}
103 \DeclareOption{executivepaper}{\bxjs@setpaper{executivepaper}}

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

104 \newif\if@landscape
105 \@landscapefalse
106 \DeclareOption{landscape}{\@landscape true}

```

■slide オプション `slide` を新設しました。

```

107 \newif\if@slide
108 %<!slide>\@slidefalse
109 %<slide>\@slide true

```

■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。

`\@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。

```
110 \newcommand{\@ptsize}{0}
111 \def\bxjs@param@basefontsize{10pt}
112 \def\jsUnusualPtSize{-20}

113 \def\bxjs@setbasefontsize#1{%
114   \bxjs@setbasefontlength\@tempdima{#1}%
115   \edef\bxjs@param@basefontsize{\the\@tempdima}%
116   \ifdim\@tempdima=10pt      \long\def\@ptsize{0}%
117   \else\ifdim\@tempdima=10.95pt \long\def\@ptsize{1}%
118   \else\ifdim\@tempdima=12pt  \long\def\@ptsize{2}%
119   \else \long\edef\@ptsize{\jsUnusualPtSize}\fi\fi\fi}
```

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

`\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。

```
120 \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。
121   \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。
122   \ifx j\jsEngine \setlength#1{#2}%
123   \else
124     \bxjs@setbasefontlength@b#2\@nil Q\@nil\@nnil
125     \ifx\bxjs@tmpa\relax \setlength#1{#2}%
126     \else \@tempdimc0.25mm #1=\bxjs@tmpa\@tempdimc
127     \fi
128   \fi}
129 \def\bxjs@setbasefontlength@b#1Q\@nil#2\@nnil{%
130   \ifx\@nnil#2\@nnil \let\bxjs@tmpa\relax
```

```

131 \else \def\bxjs@tmpa{#1}%
132 \fi}
133 \def\bxjs@setbasefontlength@a#1true#2\@nnil{%
134 \ifx\@nnil#2\@nnil\else
135 \ClassWarningNoLine\bxjs@clsname
136 {You should not use 'true' lengths here}%
137 \fi}

```

\ifjsc@mag は「\mag を使うか」を表すスイッチ。

```

138 \newif\ifjsc@mag
139 %\let\jsc@magscale\@undefined
140 \DeclareOption{8pt}{\bxjs@setbasefontsize{8pt}}
141 \DeclareOption{9pt}{\bxjs@setbasefontsize{9pt}}
142 \DeclareOption{10pt}{\bxjs@setbasefontsize{10pt}}
143 \DeclareOption{11pt}{\bxjs@setbasefontsize{10.95pt}}
144 \DeclareOption{12pt}{\bxjs@setbasefontsize{12pt}}
145 \DeclareOption{14pt}{\bxjs@setbasefontsize{14.4pt}}
146 \DeclareOption{17pt}{\bxjs@setbasefontsize{17.28pt}}
147 \DeclareOption{20pt}{\bxjs@setbasefontsize{20pt}}
148 \DeclareOption{21pt}{\bxjs@setbasefontsize{20.74pt}}
149 \DeclareOption{25pt}{\bxjs@setbasefontsize{24.88pt}}
150 \DeclareOption{30pt}{\bxjs@setbasefontsize{29.86pt}}
151 \DeclareOption{36pt}{\bxjs@setbasefontsize{35.83pt}}
152 \DeclareOption{43pt}{\bxjs@setbasefontsize{43pt}}
153 \DeclareOption{12Q}{\bxjs@setjbasefontsize{3mm}}
154 \DeclareOption{14Q}{\bxjs@setjbasefontsize{3.5mm}}
155 \DeclareOption{10ptj}{\bxjs@setjbasefontsize{10pt}}
156 \DeclareOption{10.5ptj}{\bxjs@setjbasefontsize{10.5pt}}
157 \DeclareOption{11ptj}{\bxjs@setjbasefontsize{11pt}}
158 \DeclareOption{12ptj}{\bxjs@setjbasefontsize{12pt}}

```

JS クラス互換の magstyle 設定オプション。

```

159 \DeclareOption{usemag}{\let\bxjs@magstyle\bxjs@magstyle@usemag}
160 \DeclareOption{nomag}{\let\bxjs@magstyle\bxjs@magstyle@nomag}
161 \DeclareOption{nomag*}{\let\bxjs@magstyle\bxjs@magstyle@xreal}

```

■ トンボオプション

欧文 L^AT_EX のカーネルではサポートされないため削除。
ただし JS クラスとの互換のため以下の変数を用意する。

```

162 \newdimen\stockwidth \newdimen\stockheight

```

■ 面付け

欧文 L^AT_EX のカーネルではサポートされないため削除。

■両面, 片面オプション `twoside` で奇数ページ・偶数ページのレイアウトが変わります。

[2003-04-29] `vartwoside` でどちらのページも傍注が右側になります。

```
163 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
164 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
165 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
166 \DeclareOption{onecolumn}{\@twocolumnfalse}
167 \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。

```
168 \DeclareOption{titlepage}{\@titlepagetrue}
169 \DeclareOption{notitlepage}{\@titlepagefalse}
```

■右左起こし 書籍では章は通常は奇数ページ起こしになりますが, `openany` で偶数ページからでも始まるようになります。

```
170 %<book|report>\DeclareOption{openright}{\@openrighttrue}
171 %<book|report>\DeclareOption{openany}{\@openrightfalse}
```

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

`eqnarray` \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。

```
172 \def\eqnarray{%
173   \stepcounter{equation}%
174   \def\@currentlabel{\p@equation\theequation}%
175   \global\@eqnswtrue
176   \m@th
177   \global\@eqcnt\z@
178   \tabskip\@centering
179   \let\\\@eqnocr
180   $$\everycr{}\halign to\displaywidth\bgroup
181     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
182     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
183     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
184     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
185     \tabskip\z@skip
186     \cr}
```

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

```
187 \DeclareOption{leqno}{\input{leqno.clo}}
188 \DeclareOption{fleqn}{\input{fleqn.clo}}
189 % fleqn 用の eqnarray 環境の再定義
190 \def\eqnarray{%
```

```

191 \stepcounter{equation}%
192 \def\@currentlabel{\p@equation\theequation}%
193 \global\@eqnswtrue\m@th
194 \global\@eqcnt\z@
195 \tabskip\mathindent
196 \let\=\@eqnocr
197 \setlength\abovedisplayskip{\topsep}%
198 \ifvmode
199 \addtolength\abovedisplayskip{\partopsep}%
200 \fi
201 \addtolength\abovedisplayskip{\parskip}%
202 \setlength\belowdisplayskip{\abovedisplayskip}%
203 \setlength\belowdisplayshortskip{\abovedisplayskip}%
204 \setlength\abovedisplayshortskip{\abovedisplayskip}%
205 $$\everycr{}\halign to\linewidth% $$
206 \bgroup
207 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnsele
208 &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
209 &\global\@eqcnt\tw@
210 $\displaystyle{##}$\hfil \tabskip\@centering
211 &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss#\egroup
212 \tabskip\z@skip\cr
213 }}

```

■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。

```

214 % \DeclareOption{openbib}{%
215 % \AtEndOfPackage{%
216 % \renewcommand\@openbib@code{%
217 % \advance\leftmargin\bibindent
218 % \itemindent -\bibindent
219 % \listparindent \itemindent
220 % \parsep \z@}%
221 % \renewcommand\newblock{\par}}

```

■数式フォントとして和文フォントを登録しないオプション

ここは和文ドライバの管轄。

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

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。

※\ifdraft は 2.0 版で廃止予定。

```

222 \let\ifjsDraft\iffalse
223 \@onlypreamble\bxjs@draft
224 \def\bxjs@draft#1{%
225   \expandafter\let\expandafter\ifjsDraft\csname if#1\endcsname}
226 \DeclareOption{draft}{\bxjs@draft{true}\setlength\overfullrule{5pt}}
227 \DeclareOption{final}{\bxjs@draft{false}\setlength\overfullrule{0pt}}
228 \AtBeginDocument{%
229   \expandafter\ifx\csname ifdraft\endcsname\relax
230     \expandafter\let\csname ifdraft\endcsname\expandafter\endcsname
231     \csname ifjsDraft\endcsname
232   \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。

既定で有効であるが、nopapersize オプションで無効にできる。

```

233 \newif\ifbxjs@papersize
234 \bxjs@papersizetrue
235 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
236 \DeclareOption{papersize}{\bxjs@papersizetrue}

```

■英語化 オプション english を新設しました。

```

237 \newif\if@english
238 \@englishfalse
239 \DeclareOption{english}{\@englishttrue}

```

■jsreport 相当 オプション report を新設しました。

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

■BXJS 特有のオプションの一覧

-
-
- エンジンオプション：xelatex 等。
 - ドライバオプション：dvipdfmx 等。
 - 複合設定オプション：pandoc 等。
 - nopapersize：papersize（既定で有効）の否定。

- `zw / nozw` : `\jsZw` と等価な命令として `\zw` を定義する／しない。
- `js / nojs` : JS クラスを読込済として扱う／扱わない。
- `precisetext / noprecisetext` : \TeX の “generateactualtext” を有効／無効にする。
- `simplejasetup / nosimplejasetup` : \TeX の “linebreaklocale” を有効／無効にする。
- `bigcode / nobigcode` : up\TeX で CMap として UTF8-UCS2 の代わりに UTF8-UTF16 を使う／使わない。
- `oldfontcommands / nooldfontcommands` : 古い “二文字フォント命令” に対する警告を抑止する／しない。
- `base=<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=<name>` : `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。

`\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、これだと数詞モード中では空のサブ数式を生み出してしまうため修正した。

```

240 \mathchardef\bxjs@csta=259
241 \def\bxjs@invscale#1#2{%
242   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
243   \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
244   \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
245   \@tempcntb\p@ \divide\@tempcntb\@tempdimb
246   \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
247   \@tempdimb\@tempcnta\@ne
248   \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
249   \advance\@tempcnta\bxjs@csta \@tempdimc\@tempcnta\@ne

```

```

250 \whiledim\@tempdimb<\@tempdimc\do{%
251 \tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
252 \advance\@tempcntb\@ne \divide\@tempcntb\tw@
253 \ifdim #2\@tempcntb>\@tempdima
254 \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
255 \else \@tempdimb=\@tempcntb\@ne \fi}%
256 \xdef\bxjs@gtmpa{\the\@tempdimb}%
257 \endgroup #1=\bxjs@gtmpa\relax}

```

■ 複合設定オプション

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

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

```
258 \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。

```
259 \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。

```

260 \def\bxjs@composite@proc{%
261 \bxjs@oldfontcommandstrue
262 \setkeys{bxjs}{ja=pandoc}%
263 \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…。

```

264 \def\bxjs@driver@opt{dvipdfmx}%
265 \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。

```
266 %\let\bxjs@engine@given\undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
267 %\let\bxjs@engine@opt\undefined
```

エンジン明示指定のオプションの処理。

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

```

268 \DeclareOption{autodetect-engine}{%
269   \let\bxjs@engine@given=*}
270 \DeclareOption{latex}{%
271   \def\bxjs@engine@opt{latex}%
272   \let\bxjs@engine@given=n}
273 \DeclareOption{platex}{%
274   \def\bxjs@engine@opt{platex}%
275   \let\bxjs@engine@given=j}
276 \DeclareOption{uplatex}{%
277   \def\bxjs@engine@opt{uplatex}%
278   \let\bxjs@engine@given=u}
279 \DeclareOption{xelatex}{%
280   \def\bxjs@engine@opt{xelatex}%
281   \let\bxjs@engine@given=x}
282 \DeclareOption{pdflatex}{%
283   \def\bxjs@engine@opt{pdflatex}%
284   \let\bxjs@engine@given=p}
285 \DeclareOption{lualatex}{%
286   \def\bxjs@engine@opt{lualatex}%
287   \let\bxjs@engine@given=l}

```

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

```

288 %\let\bxjs@driver@given\@undefined
289 \let\bxjs@driver@@dvimode=0
290 \let\bxjs@driver@@dviPDFmx=1
291 \let\bxjs@driver@@pdfmode=2
292 \let\bxjs@driver@@xetex=3

```

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

```

293 %\let\bxjs@driver@opt\@undefined

294 \DeclareOption{dvips}{%
295   \def\bxjs@driver@opt{dvips}%
296   \let\bxjs@driver@given\bxjs@driver@@dvimode}
297 \DeclareOption{dviout}{%
298   \def\bxjs@driver@opt{dviout}%
299   \let\bxjs@driver@given\bxjs@driver@@dvimode}
300 \DeclareOption{xdvi}{%
301   \def\bxjs@driver@opt{xdvi}%
302   \let\bxjs@driver@given\bxjs@driver@@dvimode}
303 \DeclareOption{dviPDFmx}{%
304   \def\bxjs@driver@opt{dviPDFmx}%
305   \let\bxjs@driver@given\bxjs@driver@@dviPDFmx}
306 \DeclareOption{pdftex}{%
307   \def\bxjs@driver@opt{pdftex}%
308   \let\bxjs@driver@given\bxjs@driver@@pdfmode}
309 \DeclareOption{luatex}{%

```

```

310 \def\bxjs@driver@opt{luatex}%
311 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
312 \DeclareOption{xetex}{%
313 \def\bxjs@driver@opt{xetex}%
314 \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。

```

315 \DeclareOption{dvipdfmx-if-dvi}{%
316 \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。

```

317 \newif\ifbxjs@usezw \bxjs@usezwtrue

```

`zw`、`noz` オプションの定義。

```

318 \DeclareOption{noz}{%
319 \bxjs@usezwfalse}
320 \DeclareOption{zw}{%
321 \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。

```

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

```

`nojs`、`js` オプションの定義。

```

323 \DeclareOption{nojs}{%
324 \bxjs@disguise@jsfalse}
325 \DeclareOption{js}{%
326 \bxjs@disguise@jstrue}

```

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

```

327 \newif\ifbxjs@precisetext

```

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

```

328 \DeclareOption{noprecisetext}{%
329 \bxjs@precisetextfalse}
330 \DeclareOption{precisetext}{%
331 \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）。

```

332 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

```

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

```

333 \DeclareOption{nosimplejasetup}{%

```

```

334 \bxjs@simplejasetupfalse}
335 \DeclareOption{simplejasetup}{%
336 \bxjs@simplejasetuptrue}

```

`\ifbxjs@bigcode` upTeX で有効化する ToUnicode CMap として「UTF8-UCS2」の代わりに「UTF8-UTF16」を使うか。BMP 外の文字に対応できる「UTF8-UTF16」の方が望ましいのであるが、このファイルが利用可能かの確実な判定が困難であるため、オプションで指定することとする。

```

337 \newif\ifbxjs@bigcode \bxjs@bigcodefalse

```

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

```

338 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
339 \def\bxjs@tmpb#1 #2#3\@nil{%
340 \ifx1#2\bxjs@bigcodetrue \fi}
341 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil

```

`nobigcode` / `bigcode` オプションの定義。

```

342 \DeclareOption{nobigcode}{%
343 \bxjs@bigcodefalse}
344 \DeclareOption{bigcode}{%
345 \bxjs@bigcodetrue}

```

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

```

346 \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。

```

347 \DeclareOption{nooldfontcommands}{%
348 \bxjs@oldfontcommandsfalse}
349 \DeclareOption{oldfontcommands}{%
350 \bxjs@oldfontcommandstrue}

```

■ keyval 型のオプション

```

351 \def\bxjs@setkey{%
352 \expandafter\bxjs@setkey@a\expandafter{\CurrentOption}}
353 \def\bxjs@setkey@a{\bxjs@safe@setkeys{bxjs}}
354 \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`。

```

355 \def\bxjs@safe@setkeys#1#2{%
356 \let\bxjs@KV@errx\KV@errx
357 \let\KV@errx\bxjs@safe@setkeys@a
358 \setkeys{#1}{#2}%

```

```

359 \let\KV@errx\bxjs@KV@errx}
360 \def\bxjs@safe@setkeys@a#1{

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

\jsScale 和文スケール値。
362 \def\jsScale{0.924715}
363 \def\Cjascale{\jsScale}

base オプションの処理。
364 \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。
365 \@onlypreamble\bxjs@do@opt@jbase
366 \let\bxjs@do@opt@jbase\relax
367 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
368 \def\bxjs@setjbasefontsize#1{%
369 \def\bxjs@do@opt@jbase{%
370 \bxjs@setbasefontlength\@tempdima{#1}%
371 \bxjs@invscale\@tempdima\jsScale
372 \bxjs@setbasefontsize{\@tempdima}}}

scale オプションの処理。
373 \define@key{bxjs}{scale}{%
374 \bxjs@scalesettrue
375 \edef\jsScale{#1}}

noscale オプションの処理。
376 \DeclareOption{noscale}{%
377 \bxjs@scalesettrue
378 \def\jsScale{1}}

CJK オプションは廃止された。
379 \DeclareOption{CJK}{%
380 \ClassWarning\bxjs@clsname
381 {Option `CJK' was abolished}}

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

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

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

\bxjs@jadriver 和文ドライバの名前。
385 \let\bxjs@jadriver\relax
386 %\let\bxjs@jadriver@given\@undefined

```

ja オプション (jadriver オプション) の処理。

```

387 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
388 \define@key{bxjs}{ja}[\relax]{%
389   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}

```

\jsJaFont 和文フォント設定の名前。

```

390 \let\jsJaFont\@empty

```

jafont オプションの処理。

```

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

```

\jsJaParam 和文ドライバパラメタの文字列。

```

392 \let\jsJaParam\@empty

```

japaram オプションの処理。

```

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

```

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

```

394 \let\bxjs@magstyle@mag=m
395 \let\bxjs@magstyle@real=r
396 \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。

```

397 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
398 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
399 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal

```

\bxjs@magstyle@default は既定の値を表す。

```

400 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
401 \ifx l\jsEngine \ifnum\luatexversion>86
402   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
403 \fi\fi
404 \let\bxjs@magstyle\bxjs@magstyle@default

```

magstyle オプションの処理。

```

405 \define@key{bxjs}{magstyle}{%
406   \expandafter\let\expandafter\bxjs@magstyle\csname
407     bxjs@magstyle@#1\endcsname
408   \ifx\bxjs@magstyle\relax
409     \ClassError\bxjs@clsname
410     {Invalid magstyle value `#1'}\@ehc
411     \let\bxjs@magstyle\bxjs@magstyle@default
412   \fi}

```

\bxjs@geometry geometry オプションの値。

```

413 \let\bxjs@geometry@class=c
414 \let\bxjs@geometry@user=u
415 \let\bxjs@geometry\bxjs@geometry@class

```

geometry オプションの処理。

```
416 \define@key{bxjs}{geometry}{%
417   \expandafter\let\expandafter\bxjs@geometry\csname
418     bxjs@geometry@#1\endcsname
419   \ifx\bxjs@geometry\relax
420     \ClassError\bxjs@clsname
421     {Invalid geometry value `#1'}\@ehc
422     \let\bxjs@geometry\bxjs@geometry@class
423   \fi}
```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```
424 \newif\ifbxjs@dvi@opt
```

DVI モードのドライバとドライバ種別との対応。

```
425 \let\bxjs@dvidriver@@dvipdfmx=\bxjs@driver@@dvipdfmx
426 \let\bxjs@dvidriver@@dvips=\bxjs@driver@@dvimode
427 \let\bxjs@dvidriver@@dviout=\bxjs@driver@@dvimode
428 \let\bxjs@dvidriver@@xdvi=\bxjs@driver@@dvimode
```

dvi オプションの処理。

```
429 \define@key{bxjs}{dvi}{%
430   \expandafter\let\expandafter\bxjs@tmpa\csname
431     bxjs@dvidriver@@#1\endcsname
432   \ifx\bxjs@tmpa\relax
433     \ClassError\bxjs@clsname
434     {Invalid dvi value `#1'}\@ehc
435   \else
```

\bxjs@driver@given を未定義にしていることに注意。

```
436   \def\bxjs@driver@opt{#1}%
437   \let\bxjs@driver@given\@undefined
438   \bxjs@dvi@opttrue
439 \fi}
```

■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。

```
440 \let\bxjs@ltx@removeelement\@removeelement
441 \def\@removeelement#1#2#3{%
442   \def\reserved@a{#2}%
443   \ifx\reserved@a\@empty \let#3\@empty
444   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
445 \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。

```
446 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
447 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
448 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
449 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
450 \ProcessOptions\relax
451 \bxjs@composite@proc
```

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

```
452 \@onlypreamble\bxjs@purge@brace@elts
453 \def\bxjs@purge@brace@elts{%
454   \def\bxjs@tmpa{\@gobble}%
455   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
456   \let\@classoptionslist\bxjs@tmpa}
457 \@onlypreamble\bxjs@purge@be@a
458 \def\bxjs@purge@be@a#1{%
459   \ifx\@nil#1\relax\else
460     \bxjs@purge@be@b#1{}\@nil
461     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
462     \expandafter\bxjs@purge@be@a
463   \fi}
464 \@onlypreamble\bxjs@purge@be@b
465 \def\bxjs@purge@be@b#1#\bxjs@purge@be@c}
466 \@onlypreamble\bxjs@purge@be@c
467 \def\bxjs@purge@be@c#1\@nil{%
468   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
469 \bxjs@purge@brace@elts
```

papersize と 10pt の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。

```
470 \@expandtwoargs\@removeelement
471 {papersize}\@classoptionslist\@classoptionslist
472 \@expandtwoargs\@removeelement
473 {10pt}\@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。

```
474 \ifx\bxjs@jadriver\relax\else
475   \let\bxjs@jadriver@given\bxjs@jadriver
476 \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。

```

477 \let\bxjs@tmpb\jsEngine
478 \ifx j\bxjs@tmpb\ifjsWithupTeX
479   \let\bxjs@tmpb=u
480 \fi\fi
481 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
482   \let\bxjs@tmpb=n
483 \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。)

```

484 \ifx *\bxjs@engine@given
485   \let\bxjs@engine@given\bxjs@tmpb

```

エンジン指定が autotetec-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。

```

486 \ifx j\bxjs@engine@given
487   \g@addto@macro\@classoptionslist{,latex}
488 \else\ifx u\bxjs@engine@given
489   \g@addto@macro\@classoptionslist{,uplatex}
490 \fi\fi
491 \fi
492 \ifx\bxjs@engine@given\@undefined\else
493   \ifx\bxjs@engine@given\bxjs@tmpb\else
494     \ClassError\bxjs@clsname%
495       {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
496   \fi
497 \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。

```

498 \@tempswatrue
499 \ifx \bxjs@driver@given\@undefined\else
500   \ifjsInPdfMode
501     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
502       \@tempswafalse
503     \fi
504   \else\ifx x\jsEngine
505     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
506       \@tempswafalse
507     \fi
508   \else
509     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
510       \@tempswafalse
511     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
512       \@tempswafalse
513     \fi\fi
514   \fi\fi
515 \fi
516 \if@tempswa\else
517   \ClassError\bxjs@clsname%

```

```

518 {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
519 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

520 \ifjsInPdfMode \@tempswafalse
521 \else\ifx x\jsEngine \@tempswafalse
522 \else \@tempswatru
523 \fi\fi
524 \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)。

```

525 \ifx\bxjs@driver@opt\@undefined \ifx\bxjs@jadriver@given\@undefined\else
526 \ClassWarningNoLine\bxjs@clsname
527 {No driver option is given}
528 \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。)

```

529 \ifbxjs@dvi@opt
530 \edef\bxjs@nxt{%
531 \let\noexpand\bxjs@driver@given
532 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
533 \noexpand\g@addto@macro\noexpand\@classoptionslist
534 {,\bxjs@driver@opt}%
535 }\bxjs@nxt
536 \fi
537 \fi

```

\bxjs@jadriver の正規化。値が未指定の場合は minimal に変える。ただしエンジンが (u)pTeX である場合は standard に変える。

```

538 \def\bxjs@@minimal{minimal}
539 \ifx\bxjs@jadriver\relax
540 \ifx j\jsEngine
541 \def\bxjs@jadriver{standard}
542 \else
543 \let\bxjs@jadriver\bxjs@@minimal
544 \fi
545 \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。

```

546 \ifx\bxjs@jadriver@given\@undefined\else
547 \ifx\bxjs@engine@given\@undefined
548 \ClassError\bxjs@clsname%
549 {An engine option must be explicitly given}%
550 {When you employ a Japanese-driver you must specify a
551 correct\MessageBreak engine option.\MessageBreak\@ehc}
552 \fi\fi

```

新しい Lua_T_EX (0.87 版以降) では mag がアレなので、magstyle=mag 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 xreal であり、エラーの場合は既定値に置き換えられる。)

```

553 \ifx\bxjs@magstyle@default\bxjs@magstyle@mag\else
554   \ifx\bxjs@magstyle\bxjs@magstyle@mag
555     \let\bxjs@magstyle\bxjs@magstyle@default
556     \ClassError\bxjs@clsname%
557     {The engine does not support 'magstyle=mag'}%
558     {LuaTeX v0.87 or later no longer supports "mag" feature.\MessageBreak
559       The default value is used instead.\MessageBreak \@ehc}
560   \fi
561 \fi

```

オプション処理時に遅延させていた jbase の処理をここで実行する。

```

562 \bxjs@do@opt@jbase

```

後処理

```

563 \if@slide
564   \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
565 \fi
566 \if@landscape
567   \setlength\@tempdima {\paperheight}
568   \setlength\paperheight{\paperwidth}
569   \setlength\paperwidth {\@tempdima}
570 \fi

```

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

```

571 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
572   \@tempcnta="80 \loop \ifnum\@tempcnta<"100
573     \catcode\@tempcnta\active
574     \advance\@tempcnta\@ne
575   \repeat
576 \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。

```

577 \ifbxjs@disguise@js
578 %<book|report>\def\bxjs@js@clsname{jsbook}
579 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
580 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
581 \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。

```

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

```

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

```
583 \ifbxjs@oldfontcommands
584   \jsAtEndOfClass{\allowoldfontcommands}
585 \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。

```
586 %<slide>\def\n@baseline{13}%
587 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
588 %<!slide>\else \def\n@baseline{16}\fi
```

■拡大率の設定

\ifbxjs@nfss@patch NFSS パッチするか。

```
589 \newif\ifbxjs@nfss@patch
590 \ifx\bxjs@magstyle\bxjs@magstyle@mag
591   \jsc@magtrue
592 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
593   \bxjs@nfss@patchtrue
594 \fi\fi
```

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

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

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

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

```
595 \ifx\bxjs@param@mag\relax
596   \@tempdima=\bxjs@param@basefontsize
```

```

597 \advance\@tempdima.001pt \multiply\@tempdima25
598 \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
599 \edef\bxjs@param@mag{\the\@tempcnta}
600 \else
601 \let\c@bxjs@cmta\@tempcnta
602 \setcounter{bxjs@cmta}{\bxjs@param@mag}
603 \ifnum\@tempcnta<\z@ \@tempcnta=\z@ \fi
604 % 有効な mag 値の範囲は 1--32768
605 \edef\bxjs@param@mag{\the\@tempcnta}
606 \advance\@tempcnta100000
607 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
608 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
609 \edef\bxjs@param@basefontsize{\the\@tempdima}
610 \fi
611 \@tempcnta\bxjs@param@mag \advance\@tempcnta100000
612 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
613 \expandafter\bxjs@tmpa\the\@tempcnta\@nil
614 \edef\jsc@magscale{\strip@pt\@tempdima}
615 \let\jsBaseFontSize\bxjs@param@basefontsize
616 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
617 %\typeout{\string\jsEngine: \meaning\jsEngine}
618 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
619 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
620 %\typeout{\string\jsc@magscale: \jsc@magscale}
621 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
622 %\typeout{\string\ifbxjs@nfss@patch: \meaning\ifbxjs@nfss@patch}

```

[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。

```

623 \ifjsc@mag
624 \let\jsc@mpt\p@
625 \newdimen\jsc@mmm \jsc@mmm=1mm
626 \catcode`\?=9 % \p@? read as \p@
627 \else
628 \newdimen\jsc@mpt \jsc@mpt=\jsc@magscale \p@
629 \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
630 \catcode`\?=11 \let\p@?\jsc@mpt
631 \fi
632 \chardef\bxjs@qmcc=\catcode`\?\relax

```

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

noz 非指定時は \zw を \jsZw と同義にする。

```

633 \newdimen\jsZw

```

```

634 \jsZw=10\jsc@empt \jsZw=\jsScale\jsZw
635 \ifbxjs@usezw
636 \providecommand*{zw}{\jsZw}
637 \fi

```

■NFSS パッチ

ムニャムニャムニャ……。

```

638 \ifbxjs@nfss@patch
639 \RequirePackage{type1cm}
640 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
641 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
642 \let\bxjs@get@external@font\get@external@font
643 \def\get@external@font{%
644   \bxjs@preadjust@extract@font
645   \bxjs@get@external@font}
646 \def\bxjs@fstrunc#1{%
647   \edef\bxjs@tmpa{\strip@pt#1}%
648   \expandafter\bxjs@fstrunc@a\bxjs@tmpa.****\@nil}
649 \def\bxjs@fstrunc@a#1.#2#3#4#5#6\@nil{%
650   \if#5*else
651     \edef\bxjs@tmpa{#1%
652       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
653   \fi}
654 \def\bxjs@preadjust@extract@font{%
655   \let\bxjs@req@size\f@size
656   \dimen@f@size\p@ \bxjs@invscale\dimen@jsc@magscale
657   \advance\dimen@.005pt\relax \bxjs@fstrunc\dimen@
658   \let\bxjs@ref@size\bxjs@tmpa
659   \let\f@size\bxjs@ref@size}
660 \def\execute@size@function#1{%
661   \let\bxjs@cref@size\f@size
662   \let\f@size\bxjs@req@size
663   \csname s@fct@#1\endcsname}
664 \let\bxjs@DeclareErrorFont\DeclareErrorFont
665 \def\DeclareErrorFont#1#2#3#4#5{%
666   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
667   \edef\bxjs@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
668   \expandafter\bxjs@DeclareErrorFont\bxjs@tmpa}
669 \def\gen@sfcnt{%
670   \edef\mandatory@arg{\mandatory@arg\bxjs@cref@size}%
671   \empty@sfcnt}
672 \def\genb@sfcnt{%
673   \edef\mandatory@arg{%
674     \mandatory@arg\expandafter\genb@x\bxjs@cref@size..\@{}}%
675   \empty@sfcnt}
676 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}

```

■pagesize スペシャルの出力

削除。

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

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

```
678 \chardef\bxjs@periodchar=`\.  
679 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcode\bxjs@periodchar}  
680 \def\@{\bxjs@SE{}}
```

4 フォントサイズ

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

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

のようにして行います。これは

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

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

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

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

\kanjiskip は $\text{p\LaTeX 2}_\epsilon$ で $\text{0pt plus .4pt minus .5pt}$ に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追

い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

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

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

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

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

```
681 \def\bxjs@tmpa{\def\set@fontsize##1##2##3%
682 \expandafter\bxjs@tmpa\expandafter{%
683   \set@fontsize{#1}{#2}{#3}%
684 % 末尾にコードを追加
685   \expandafter\def\expandafter\size@update\expandafter{%
686     \size@update
687     \jsFontSizeChanged}%
688 }
```

`\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。

```
689 \newcommand*\jsFontSizeChanged{%
690   \jsZw=\f@size\p@
691   \jsZw=\jsScale \jsZw
692   \ifdim\parindent>\z@
693     \if@english \parindent=1em
694     \else       \parindent=1\jsZw
695     \fi
696   \fi\relax
697   \jsResetDimen}
```

`\jsResetDimen` ユーザ定義用のフック。

```
698 \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。

```
699 \ifjsc@mag
700   \let\jsc@setfontsize\@setfontsize
701 \else
702   \def\jsc@setfontsize#1#2#3{%
703     \@setfontsize#1{#2\jsc@mp}{#3\jsc@mp}}
704 \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。

705 `\emergencystretch 3\jsZw`

`\ifnarrowbaselines` 欧文用に行間を狭くする論理変数と、それを真・偽にするためのコマンドです。
`\narrowbaselines` [2003-06-30] 数式に入るところで `\narrowbaselines` を実行しているので
`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対して、
しっぽ愛好家さんが次の修正を教えてくださいました。
[2008-02-18] `english` オプションで最初の段落のインデントをしないようにしました。
TODO: Hasumi さん [qa:54539] のご指摘は考慮中です。

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

```
706 \newif\ifnarrowbaselines
707 \if@english
708   \narrowbaselinestrue
709 \fi
710 \def\narrowbaselines{%
711   \narrowbaselinestrue
712   \skip0=\abovedisplayskip
713   \skip2=\abovedisplayshortskip
714   \skip4=\belowdisplayskip
715   \skip6=\belowdisplayshortskip
716 % 一時的に警告を無効化する
717   \let\bxjs@ltx@nomath\@nomath
718   \let\@nomath\@gobble
719   \@currsize\selectfont
720   \let\@nomath\bxjs@ltx@nomath
721   \abovedisplayskip=\skip0
722   \abovedisplayshortskip=\skip2
723   \belowdisplayskip=\skip4
724   \belowdisplayshortskip=\skip6\relax}
725 \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。

```
726 \def\bxjs@if@narrowbaselines{%  
727   \ifnarrowbaselines\expandafter\@firstoftwo  
728   \else \expandafter\@secondoftwo  
729   \fi  
730 }
```

`\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。

```
731 \renewcommand{\normalsize}{%  
732   \bxjs@if@narrowbaselines{%  
733     \jsc@setfontsize\normalsize\@xpt\@xipt  
734   }{%else  
735     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%  
736   }%
```

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

[2003-02-16] ちょっと変えました。

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

```
737 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?  
738 \abovedisplayshortskip \z@ \@plus3\p@?  
739 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?  
740 \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。

```
741 \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。

```
742 \normalsize
```

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

`\Cwd`

`\Cvs` まず `\Cwd` 等の変数を定義する。

`\Chs`

```

743 \ifx\Cht\@undefined \newdimen\Cht \fi
744 \ifx\Cdp\@undefined \newdimen\Cdp \fi
745 \ifx\Cwd\@undefined \newdimen\Cwd \fi
746 \ifx\Cvs\@undefined \newdimen\Cvs \fi
747 \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。

```

748 \setlength\Cht{0.88\jsZw}
749 \setlength\Cdp{0.12\jsZw}
750 \setlength\Cwd{1\jsZw}
751 \setlength\Cvs{\baselineskip}
752 \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。

```

753 \newcommand{\small}{%
754   \bxjs@if@narrowbaselines{%
755     %<!kiyou> \jsc@setfontsize\small\@ixpt{11}%
756     %<kiyou> \jsc@setfontsize\small{8.8888}{11}%
757   }{%else
758     %<!kiyou> \jsc@setfontsize\small\@ixpt{13}%
759     %<kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
760   }%
761   \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
762   \abovedisplayshortskip \z@ \@plus3\p@?
763   \belowdisplayskip \abovedisplayskip
764   \belowdisplayshortskip \belowdisplayskip
765   \def\@listi{\leftmargin\leftmargin
766     \topsep \z@
767     \parsep \z@
768     \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。

```

769 \newcommand{\footnotesize}{%
770   \bxjs@if@narrowbaselines{%
771     %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
772     %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%

```

```

773 }{%else
774 %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
775 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
776 }%
777 \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
778 \abovedisplayshortskip \z@ \@plus2\p@?
779 \belowdisplayskip \abovedisplayskip
780 \belowdisplayshortskip \belowdisplayskip
781 \def\@listi{\leftmargin\leftmargini
782 \topsep \z@
783 \parsep \z@
784 \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 785 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
786 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 787 \if@twocolumn
788 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
789 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
790 \else
791 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
792 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
793 \fi
794 %<!kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xivpt{21}}
795 %<kiyou>\newcommand{\LARGE}{\jsc@setfontsize\LARGE{12.222}{21}}
796 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xviipt{25}}
797 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
798 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
799 \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。

```

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

```

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

見出し用のフォントは `\bfseries` 固定ではなく、`\headfont` という命令で定めることにします。これは太ゴシックが使えるときは `\sffamily \bfseries` でいいと思いますが、通常の中ゴシックでは単に `\sffamily` だけのほうがよさそうです。『 $\text{\LaTeX} 2_{\epsilon}$ 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。

```
801 % \newcommand{\headfont}{\bfseries}
802 \newcommand{\headfont}{\sffamily}
803 % \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。

```
804 %<kiyou>\setlength\columnsep{2\Cwd}
805 %<kiyou>\setlength\columnsep{28truebp}
806 \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 807 \setlength\lineskip{1\p@?}
\normallineskip 808 \setlength\normallineskip{1\p@?}
\normallineskiplimit 809 \setlength\lineskiplimit{1\p@?}
810 \setlength\normallineskiplimit{1\p@?}
```

`\baselinestretch` 実際の行送りが `\baselineskip` の何倍かを表すマクロです。たとえば

```
\renewcommand{\baselinestretch}{2}
```

とすると、行送りが通常の 2 倍になります。ただし、これを設定すると、たとえ `\baselineskip` が伸縮するように設定しても、行送りの伸縮ができなくなります。行送りの伸縮はしないのが一般的です。

```
811 \renewcommand{\baselinestretch}{}
```

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

```
812 \setlength\parskip{0\p@}
813 \if@slide
814   \setlength\parindent{0\p@}
815 \else
816   \setlength\parindent{1\Cwd}
817 \fi
```

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

```
\@highpenalty 818 \@lowpenalty 51
```

```
819 \@medpenalty 151
820 \@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。

```
821 % \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。

```
822 % \brokenpenalty 100
```

5.1 ページレイアウト

`geometry` パッケージに任せる。

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
823 \ifjsc@mag
824 \mag=\bxjs@param@mag
825 \fi
826 \setlength{\topskip}{10\p0?}
```

`\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。

```
827 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}
828 \def\bxjs@read@b{%
829   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
830   \else \expandafter\bxjs@read@d \fi}
831 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
832 \def\bxjs@read@d#1\@nil{}
833 \expandafter\bxjs@read@a\bxjs@param@paper\@nil
```

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

```
834 \edef\bxjs@layout@paper{%
835   \ifjsc@mag truedimen,\fi
836   \if@landscape landscape,\fi
837   \bxjs@param@paper}
```

`\bxjs@layout` は `geometry` のページレイアウトのオプション列であり、文書クラス毎に異なる。

```
838 %<*article|report>
```

```

839 \def\bxjs@layout{%
840   headheight=\topskip,footskip=0.03367\paperheight,%
841   headsep=\footskip-\topskip,includeheadfoot,%
842   hscale=0.76,hmarginratio=1:1,%
843   vscale=0.83,vmarginratio=1:1,%
844 }
845 %</article|report>
846 %<*book>
847 \def\bxjs@layout{%
848   headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
849   hmargin=36\jsc@mmm,hmarginratio=1:1,%
850   vscale=0.83,vmarginratio=1:1,%
851 }
852 %</book>
853 %<*slide>
854 \def\bxjs@layout{%
855   noheadfoot,%
856   hscale=0.9,hmarginratio=1:1,%
857   vscale=0.944,vmarginratio=1:1,%
858 }
859 %</slide>

```

geometry の後処理。

\jsTextWidthLimit は bxjsbook での \textwidth の上限で、全角単位の実数値マクロ。

```

860 \newdimen\fullwidth
861 %<book>\newcommand\jsTextWidthLimit{40}
862 \def\bxjs@postproc@layout{%
863 %   ドライバ再設定
864   \ifx\bxjs@geometry@driver\relax\else
865     \let\Gm@driver\bxjs@geometry@driver
866   \fi
867 %   textwidth 調整
868   \@tempdimb=\textwidth
869   \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
870   \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
871   \advance\@tempdimb-\textwidth
872   \advance\oddsidemargin 0.5\@tempdimb
873   \advance\evensidemargin 0.5\@tempdimb
874   \fullwidth=\textwidth
875 %<*book>
876   \ifdim\textwidth>\jsTextWidthLimit\Cwd
877     \textwidth=\jsTextWidthLimit\Cwd
878     \addtolength\evensidemargin{\fullwidth-\textwidth}
879   \fi
880 %</book>

```

```

881 % textheight 調整
882 \@tempdimb=\textheight
883 \advance\textheight-\topskip
884 \divide\textheight\baselineskip \multiply\textheight\baselineskip
885 \advance\textheight\topskip
886 \advance\@tempdimb-\textheight
887 \advance\topmargin0.5\@tempdimb
888 % headheight 調整
889 \@tempdima=\topskip
890 \advance\headheight\@tempdima
891 \advance\topmargin-\@tempdima
892 % marginpar 設定
893 \setlength\marginparsep{\columnsep}
894 \setlength\marginparpush{\baselineskip}
895 \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
896     -\textwidth-10\jsc@mmm-\marginparsep}
897 \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
898 % 連動する変数
899 \maxdepth=.5\topskip
900 \stockwidth=\paperwidth
901 \stockheight=\paperheight
902 }

```

`\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。

```

903 \edef\jsGeometryOptions{%
904     \bxjs@layout@paper,\bxjs@layout}

```

ここで `geoemtry` オプションの値に応じて分岐する。

まずは `geometry=class` の場合。

```

905 \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。

```

906 \@onlypreamble\bxjs@revert
907 \let\bxjs@revert\@empty
908 \edef\bxjs@tmpa{\the\ht\strutbox}
909 \ht\strutbox=10\p@?
910 \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。

```

911 \ifbxjs@papersize
912   \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
913     \PassOptionsToPackage{dvipdfm}{geometry}
914   \fi
915   \let\bxPapersizeSpecialDone=t
916 \else
917   \PassOptionsToPackage{driver=none}{geometry}
918 \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。

```

919 \edef\bxjs@nxt{%
920   \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
921 \AtBeginDocument{\bxjs@pre@geometry@hook}
922 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
923 \bxjs@nxt \bxjs@revert
924 \AtBeginDocument{\RevokeOldLuaTeXBehavior}

```

`\bxjs@geometry@driver` は `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。

```

925 \let\bxjs@geometry@driver\Gm@driver
926 \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。

```

927 \@onlypreamble\bxjs@pre@geometry@hook
928 \def\bxjs@pre@geometry@hook{%
929   \@ifpackageloaded{geometry}{\@ifpackagelater{geometry}{2010/02/12}{}{%
930     \PackageWarningNoLine\bxjs@clsname
931       {The 'geometry' package installed\MessageBreak
932         is too old (< v5.0)}}%
933   \if x\jsEngine \ifnum\mag=\@m\else
934     \def\bxjs@Gm@driver{pdftex}
935     \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi

```

```

936     \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
937     \fi\fi
938   }}{}}

```

- `\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。

```

939 \def\setpagelayout{\@ifstar
940   {\bxjs@reset@layout}{\bxjs@modify@layout}}
941 \def\bxjs@modify@layout#1{%
942   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
943   \bxjs@nxt\bxjs@postproc@layout}
944 \def\bxjs@reset@layout#1{%
945   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
946   \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。

```

947 \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`）になっている場合はエラーを出す。

※`\jsUseMinimalPageLayout` は動作テスト用。

```

948 \AtBeginDocument{\bxjs@check@page@layout}
949 \@onlypreamble\bxjs@check@page@layout
950 \def\bxjs@check@page@layout{%
951   \ifdim\textwidth=.5\maxdimen
952     \ClassError\bxjs@clsname
953       {Page layout is not properly set}
954     \@ehd
955   \fi}
956 \def\jsUseMinimalPageLayout{%
957   \setlength{\textwidth}{6.5in}%
958   \setlength{\textheight}{8in}}

```

`\setpagelayout` はとりあえず無効にしておく。

```

959 \let\bxjs@geometry@driver\relax
960 \def\setpagelayout{\@ifstar
961   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
962 \def\bxjs@pagelayout@a#1{%
963   \ClassError\bxjs@clsname
964   {Command \string\setpagelayout\space is not supported,\MessageBreak
965     because 'geometry' value is not 'class'}\@eha}

```

geometry=user の場合の処理はここで終わり。

```

966 \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。

```

967 \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。

```

968 \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。

969 `\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。

970 `\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。

971 `\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。

972 `\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。

973 `\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。

974 `\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。

975 `\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。

976 `\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。

977 `\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。

978 `\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。

979 `\setlength\floatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 980 `\setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}`
 981 `\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` 982 `\setlength\dblfloatsep {12\p@? \@plus 2\p@? \@minus 2\p@?}`
 983 `\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` 984 `\setlength\@fptop{0\p@? \@plus 1fil}`
985 `\setlength\@fpsep{8\p@? \@plus 2fil}`
986 `\setlength\@fpbot{0\p@? \@plus 1fil}`

`\@dblftop` 段抜きフロートについての値です。
`\@dblfpsep` 987 `\setlength\@dblftop{0\p@? \@plus 1fil}`
`\@dblfpbot` 988 `\setlength\@dblfpsep{8\p@? \@plus 2fil}`
989 `\setlength\@dblfpbot{0\p@? \@plus 1fil}`

6 ページスタイル

ページスタイルとして, L^AT_EX 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 L^AT_EX 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, フッタ) を出力する命令です。これらは `\fullwidth` 幅の `\hbox` の中で呼び出されます。
`\@evenfoot` `\ps@...` の中で定義しておきます。
`\@oddfoot` 柱の内容は, `\chapter` が呼び出す `\chaptermark{何々}`, `\section` が呼び出す `\sectionmark{何々}` で設定します。柱を扱う命令には次のものがあります。

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

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

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

`\ps@empty` `empty` ページスタイルの定義です。L^AT_EX 本体で定義されているものをコメントアウトした形で載せておきます。
990 % `\def\ps@empty{%`
991 % `\let\mkboth\gobbletwo`
992 % `\let\@oddhead\@empty`
993 % `\let\@oddfoot\@empty`

```

994 % \let\@evenhead\@empty
995 % \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。
996 \def\ps@plainfoot{%
997 \let\@mkboth\@gobbletwo
998 \let\@oddhead\@empty
999 \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1000 \let\@evenhead\@empty
1001 \let\@evenfoot\@oddfoot}
1002 \def\ps@plainhead{%
1003 \let\@mkboth\@gobbletwo
1004 \let\@oddfoot\@empty
1005 \let\@evenfoot\@empty
1006 \def\@evenhead{%
1007 \if@mparswitch \hss \fi
1008 \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1009 \if@mparswitch\else \hss \fi}%
1010 \def\@oddhead{%
1011 \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}%
1012 %<book>\let\ps@plain\ps@plainhead
1013 %<!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。
1014 %<*article|slide>
1015 \if@twoside
1016 \def\ps@headings{%
1017 \let\@oddfoot\@empty
1018 \let\@evenfoot\@empty
1019 \def\@evenhead{\if@mparswitch \hss \fi
1020 \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1021 \if@mparswitch\else \hss \fi}%
1022 \def\@oddhead{%
1023 \underline{%
1024 \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1025 \let\@mkboth\markboth
1026 \def\sectionmark##1{\markboth{%
1027 \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsw\fi
1028 ##1}{}}%
1029 \def\subsectionmark##1{\markright{%
1030 \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsw\fi
1031 ##1}}%
1032 }
1033 \else % if not twoside

```

```

1034 \def\ps@headings{%
1035     \let\@oddfoot\@empty
1036     \def\@oddhead{%
1037         \underline{%
1038             \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1039     \let\@mkboth\markboth
1040     \def\sectionmark##1{\markright{%
1041         \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1042         ##1}}}%
1043 \fi
1044 %</article|slide>

```

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

```

1045 %<*book|report>
1046 \newif\if@omit@number
1047 \def\ps@headings{%
1048     \let\@oddfoot\@empty
1049     \let\@evenfoot\@empty
1050     \def\@evenhead{%
1051         \if@mparswitch \hss \fi
1052         \underline{\hbox to \fullwidth{\autoxspacing
1053             \textbf{\thepage}\hfil\leftmark}}}%
1054         \if@mparswitch\else \hss \fi}%
1055     \def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
1056         {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1057     \let\@mkboth\markboth
1058     \def\chaptermark##1{\markboth{%
1059         \ifnum \c@secnumdepth >\m@ne
1060             \if@mainmatter
1061                 \if@omit@number\else
1062                     \@chapapp\thechapter\@chappos\hskip1\jsZw
1063                 \fi
1064             \fi
1065             \fi
1066             ##1}{}}}%
1067     \def\sectionmark##1{\markright{%
1068         \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1069         ##1}}}%
1070 %</book|report>

```

最後は学会誌の場合です。

```

1071 %<*jspf>
1072 \def\ps@headings{%
1073     \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1074     \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1075     \def\@oddhead{\normalfont\hfil \@title \hfil}
1076     \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}}
1077 %</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。

```
1078 \def\ps@myheadings{%
1079   \let\@oddfoot\@empty\let\@evenfoot\@empty
1080   \def\@evenhead{%
1081     \if@mparswitch \hss \fi%
1082     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1083     \if@mparswitch\else \hss \fi}%
1084   \def\@oddhead{%
1085     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1086   \let\@mkboth\@gobbletwo
1087 %<book|report> \let\chaptermark\@gobble
1088 \let\sectionmark\@gobble
1089 %<!book&!report> \let\subsectionmark\@gobble
1090 }
```

7 文書のマークアップ

7.1 表題

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

```
\author ます。
\date 1091 % \newcommand*{\title}[1]{\gdef\@title{#1}}
      1092 % \newcommand*{\author}[1]{\gdef\@author{#1}}
      1093 % \newcommand*{\date}[1]{\gdef\@date{#1}}
      1094 % \date{\today}
```

`\subtitle` 副題を設定する。

※プレアンブルにおいて `\newcommand*{\subtitle}{...}` が行われることへの対策として、`\subtitle` の定義を `\title` の実行まで遅延させることにする。もしどうしても主題より前に副題を設定したい場合は、`\jsSubtitle` 命令を直接用いればよい。

本体を `\jsSubtitle` として定義する。

```
1095 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1096 %\let\bxjs@subtitle\@undefined

\title にフックを入れる。

1097 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1098 \AtBeginDocument{\bxjs@decl@subtitle}
1099 \def\bxjs@decl@subtitle{%
1100   \global\let\bxjs@decl@subtitle\relax
1101   \ifx\subtitle\@undefined
1102     \global\let\subtitle\jsSubtitle
1103   \fi}
```

```

1104 \def\bxjs@annihilate@subtitle{%
1105   \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1106   \global\let\jsSubtitle\relax}

```

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

```

\author 1107 %<*jspf>
1108 \newcommand*\etitle[1]{\gdef\etitle{#1}}
\keywords 1109 \newcommand*\eauthor[1]{\gdef\eauthor{#1}}
1110 \newcommand*\keywords[1]{\gdef\keywords{#1}}
1111 \newcommand*\email[1]{\gdef\authors@mail{#1}}
1112 \newcommand*\AuthorsEmail[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1113 %</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。

```

1114 \def\plainifnotempty{%
1115   \ifx \@oddhead \@empty
1116     \ifx \@oddfoot \@empty
1117       \else
1118         \thispagestyle{plainfoot}%
1119       \fi
1120     \else
1121       \thispagestyle{plainhead}%
1122     \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。

```

1123 %<*article|book|report|slide>
1124 \if@titlepage
1125   \newcommand*\maketitle{%
1126     \begin{titlepage}%
1127       \let\footnotesize\small
1128       \let\footnoterule\relax
1129       \let\footnote\thanks
1130       \null\vfil
1131       \if@slide
1132         {\footnotesize \@date}%
1133       \begin{center}
1134         \mbox{} \ \ [1\jsZw]
1135         \large
1136         {\maybblue\hrule height0\p@? depth2\p@?\relax}\par
1137         \smallskip
1138         \@title

```

```

1139         \ifx\bxjs@subtitle\@undefined\else
1140             \par\vskip\z@
1141             {\small \bxjs@subtitle\par}
1142         \fi
1143         \smallskip
1144         {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1145         \vfill
1146         {\small \@author}%
1147     \end{center}
1148 \else
1149     \vskip 60\p@?
1150     \begin{center}%
1151         {\LARGE \@title \par}%
1152         \ifx\bxjs@subtitle\@undefined\else
1153             \vskip5\p@?
1154             {\normalsize \bxjs@subtitle\par}
1155         \fi
1156         \vskip 3em%
1157         {\large
1158             \lineskip .75em
1159             \begin{tabular}[t]{c}%
1160                 \@author
1161             \end{tabular}\par}%
1162         \vskip 1.5em
1163         {\large \@date \par}%
1164     \end{center}%
1165     \fi
1166     \par
1167     \@thanks\vfil\null
1168 \end{titlepage}%
1169 \setcounter{footnote}{0}%
1170 \global\let\thanks\relax
1171 \global\let\maketitle\relax
1172 \global\let\@thanks\@empty
1173 \global\let\@author\@empty
1174 \global\let\@date\@empty
1175 \global\let\@title\@empty
1176 \global\let\title\relax
1177 \global\let\author\relax
1178 \global\let\date\relax
1179 \global\let\and\relax
1180 \bxjs@annihilate@subtitle
1181 }%
1182 \else
1183     \newcommand{\maketitle}{\par
1184         \begingroup
1185             \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1186             \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1187             \long\def\@makefntext##1{\advance\leftskip 3\jsZw

```

```

1188     \parindent 1\jsZw\noindent
1189     \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1190 \if@twocolumn
1191     \ifnum \col@number=\@ne
1192         \@maketitle
1193     \else
1194         \twocolumn[\@maketitle]%
1195     \fi
1196 \else
1197     \newpage
1198     \global\@topnum\z@ % Prevents figures from going at top of page.
1199     \@maketitle
1200 \fi
1201 \plainifnotempty
1202 \@thanks
1203 \endgroup
1204 \setcounter{footnote}{0}%
1205 \global\let\thanks\relax
1206 \global\let\maketitle\relax
1207 \global\let\@thanks\@empty
1208 \global\let\@author\@empty
1209 \global\let\@date\@empty
1210 \global\let\@title\@empty
1211 \global\let\title\relax
1212 \global\let\author\relax
1213 \global\let\date\relax
1214 \global\let\and\relax
1215 \bxjs@annihilate@subtitle
1216 }

```

`\@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。

```

1217 \def\@maketitle{%
1218     \newpage\null
1219     \vskip 2em
1220     \begin{center}%
1221         \let\footnote\thanks
1222         {\LARGE \@title \par}%
1223         \ifx\bxjs@subtitle\@undefined\else
1224             \vskip3\p@?
1225             {\normalsize \bxjs@subtitle\par}
1226         \fi
1227         \vskip 1.5em
1228         {\large
1229             \lineskip .5em
1230             \begin{tabular}[t]{c}%
1231                 \@author
1232             \end{tabular}\par}%
1233         \vskip 1em
1234         {\large \@date}%

```

```

1235 \end{center}%
1236 \par\vskip 1.5em
1237 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1238 }
1239 \fi
1240 %</article|book|report|slide>
1241 %<*jspf>
1242 \newcommand{\maketitle}{\par
1243 \begin{group}
1244 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1245 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1246 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1247 \parindent 1\jsZw\noindent
1248 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1249 \twocolumn[\@maketitle]%
1250 \plainifnotempty
1251 \@thanks
1252 \end{group}
1253 \setcounter{footnote}{0}%
1254 \global\let\thanks\relax
1255 \global\let\maketitle\relax
1256 \global\let\@thanks\@empty
1257 \global\let\@author\@empty
1258 \global\let\@date\@empty
1259 % \global\let\@title\@empty % \@title は柱に使う
1260 \global\let\title\relax
1261 \global\let\author\relax
1262 \global\let\date\relax
1263 \global\let\and\relax
1264 \ifx\authors@mail\@undefined\else{%
1265 \def\@makefntext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1266 \footnotetext[0]{\itshape\authors@mail}%
1267 }\fi
1268 \global\let\authors@mail\@undefined}
1269 \def\@maketitle{%
1270 \newpage\null
1271 \vskip 6em % used to be 2em
1272 \begin{center}
1273 \let\footnote\thanks
1274 \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1275 \lineskip .5em
1276 \ifx\@author\@undefined\else
1277 \vskip 1em
1278 \begin{tabular}[t]{c}%
1279 \@author
1280 \end{tabular}\par
1281 \fi
1282 \ifx\@etitle\@undefined\else
1283 \vskip 1em

```

```

1284     {\large \@etitle \par}%
1285     \fi
1286     \ifx\@eauthor\@undefined\else
1287         \vskip 1em
1288         \begin{tabular}[t]{c}%
1289             \@eauthor
1290         \end{tabular}\par
1291     \fi
1292     \vskip 1em
1293     \@date
1294 \end{center}
1295 \vskip 1.5em
1296 \centerline{\box\@abstractbox}
1297 \ifx\@keywords\@undefined\else
1298     \vskip 1.5em
1299     \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1300 \fi
1301 \vskip 1.5em}
1302 %</jspf>

```

7.2 章・節

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

```

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

```

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

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

字下げ 見出しの字下げ量です。

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

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

スタイル 見出しの文字スタイルの設定です。

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

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

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

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

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

```

1303 \def\@startsection#1#2#3#4#5#6{%
1304   \if@noskipsec \leavevmode \fi
1305   \par
1306 % 見出し上の空きを \@tempskipa にセットする
1307   \@tempskipa #4\relax
1308 % \@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
1309   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1310 % 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
1311   \ifdim \@tempskipa <\z@
1312     \@tempskipa -\@tempskipa \@afterindentfalse
1313   \fi
1314   \if@nobreak
1315 %   \everypar{\everyparhook}% これは間違い
1316     \everypar{}%
1317   \else
1318     \addpenalty\@secpenalty
1319 % 次の行は削除
1320 %   \addvspace\@tempskipa
1321 % 次の \noindent まで追加
1322     \ifdim \@tempskipa >\z@
1323       \if@slide\else
1324         \null
1325         \vspace*{-\baselineskip}%
1326       \fi
1327       \vskip\@tempskipa
1328     \fi
1329   \fi
1330   \noindent
1331 % 追加終わり
1332   \@ifstar
1333     {\@ssect{#3}{#4}{#5}{#6}}%
1334     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

```

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

```

1335 \def\@sect#1#2#3#4#5#6[#7]#8{%
1336   \ifnum #2>\c@secnumdepth
1337     \let\@svsec\@empty
1338   \else
1339     \refstepcounter{#1}%
1340     \protected@edef\@svsec{\@seccntformat{#1}\relax}%
1341   \fi
1342 % 見出し後の空きを \@tempskipa にセット
1343   \@tempskipa #5\relax
1344 % 条件判断の順序を入れ替えました
1345   \ifdim \@tempskipa <\z@

```

```

1346 \def\@svsechd{%
1347   #6{\hskip #3\relax
1348   \@svsec #8}%
1349   \csname #1mark\endcsname{#7}%
1350   \addcontentsline{toc}{#1}{%
1351     \ifnum #2>\c@secnumdepth \else
1352       \protect\numberline{\csname the#1\endcsname}%
1353     \fi
1354     #7}}% 目次にフルネームを載せるなら #8
1355 \else
1356   \begingroup
1357   \interlinepenalty \@M % 下から移動
1358   #6{%
1359     \@hangfrom{\hskip #3\relax\@svsec}%
1360 %   \interlinepenalty \@M % 上に移動
1361     #8\@@par}%
1362   \endgroup
1363   \csname #1mark\endcsname{#7}%
1364   \addcontentsline{toc}{#1}{%
1365     \ifnum #2>\c@secnumdepth \else
1366       \protect\numberline{\csname the#1\endcsname}%
1367     \fi
1368     #7}}% 目次にフルネームを載せるならここは #8
1369   \fi
1370   \@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。

```

1371 \def\@xsect#1{%
1372 % 見出しの後ろの空きを \@tempkipa にセット
1373   \@tempkipa #1\relax
1374 % 条件判断の順序を変えました
1375   \ifdim \@tempkipa<\z@
1376     \@nbreakfalse
1377     \global\@noskipsectrue
1378     \everypar{%
1379       \if@noskipsec
1380         \global\@noskipsecfalse
1381         {\setbox\z@\lastbox}%
1382         \clubpenalty \@M
1383         \begingroup \@svsechd \endgroup
1384         \unskip
1385         \@tempkipa #1\relax
1386         \hskip -\@tempkipa
1387         \bxjs@ltj@inhibitglue
1388       \else
1389         \clubpenalty \@clubpenalty
1390         \everypar{\everyparhook}%
1391       \fi\everyparhook}%

```

```

1392 \else
1393   \par \nobreak
1394   \vskip \@tempskipa
1395   \@afterheading
1396 \fi
1397 \if@slide
1398   {\vskip-6\p@?\maybeblue\hrule height0\p@? depth1\p@?\vskip7\p@?\relax}%
1399 \fi
1400 \par % 2000-12-18
1401 \ignorespaces}
1402 \def\@ssect#1#2#3#4#5{%
1403   \@tempskipa #3\relax
1404   \ifdim \@tempskipa<\z@
1405     \def\@svsechd{#4{\hskip #1\relax #5}}%
1406   \else
1407     \begingroup
1408       #4{%
1409         \@hangfrom{\hskip #1}%
1410         \interlinepenalty \@M #5\@@par}%
1411     \endgroup
1412   \fi
1413   \@xsect{#3}}

```

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

```

1414 \let\bxjs@ltj@inhibitglue\@empty

```

■柱関係の命令

`\chaptermark` `\...mark` の形の命令を初期化します (第 6 節参照)。`\chaptermark` 以外は L^AT_EX 本体で定義済みです。

```

\subsectionmark 1415 \newcommand*\chaptermark[1]{}
\subsubsectionmark 1416 % \newcommand*\{sectionmark}[1]{}
1417 % \newcommand*\{subsectionmark}[1]{}
\paragraphmark 1418 % \newcommand*\{subsubsectionmark}[1]{}
\subparagraphmark 1419 % \newcommand*\{paragraphmark}[1]{}
1420 % \newcommand*\{subparagraphmark}[1]{}

```

■カウンタの定義

`\c@secnumdepth` `secnumdepth` は第何レベルの見出しまで番号を付けるかを定めるカウンタです。

```

1421 %<!book&!report>\setcounter{secnumdepth}{3}
1422 %<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 1423 \newcounter{part}
\c@subsubsection
\c@paragraph
\c@subparagraph

```

```

1424 %<book|report>\newcounter{chapter}
1425 %<book|report>\newcounter{section}[chapter]
1426 %<!book&!report>\newcounter{section}
1427 \newcounter{subsection}[section]
1428 \newcounter{subsubsection}[subsection]
1429 \newcounter{paragraph}[subsubsection]
1430 \newcounter{subparagraph}[paragraph]

```

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

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

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

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```

1431 \renewcommand{\thepart}{\@Roman\c@part}
1432 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1433 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1434 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
1435 %<*book|report>
1436 \renewcommand{\thechapter}{\@arabic\c@chapter}
1437 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1438 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1439 %</book|report>
1440 \renewcommand{\thesubsubsection}{\%
1441   \thesubsection.\@arabic\c@subsubsection}
1442 \renewcommand{\theparagraph}{\%
1443   \thesubsubsection.\@arabic\c@paragraph}
1444 \renewcommand{\thesubparagraph}{\%
1445   \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。

```

1446 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1447 %<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。

```

1448 %<*book|report>
1449 \newcommand\frontmatter{%

```

```

1450 \if@openright
1451   \cleardoublepage
1452 \else
1453   \clearpage
1454 \fi
1455 \@mainmatterfalse
1456 \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。

```

1457 \newcommand\mainmatter{%
1458 % \if@openright
1459   \cleardoublepage
1460 % \else
1461 %   \clearpage
1462 % \fi
1463 \@mainmattertrue
1464 \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。

```

1465 \newcommand\backmatter{%
1466   \if@openright
1467     \cleardoublepage
1468   \else
1469     \clearpage
1470   \fi
1471   \@mainmatterfalse}
1472 %</book|report>

```

■部

`\part` 新しい部を始めます。

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

```
\secdef{星なし}{星あり}
```

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

```

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

```

まず book クラス以外です。

```

1473 %<*&!book&!report>
1474 \newcommand\part{%
1475   \if@noskipsec \leavevmode \fi
1476   \par

```

```

1477 \addvspace{4ex}%
1478 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1479 \secdef\@part\@spart}
1480 %</!book&!report>

```

book スタイルの場合は、少し複雑です。

```

1481 %<*book|report>
1482 \newcommand\part{%
1483 \if@openright
1484 \cleardoublepage
1485 \else
1486 \clearpage
1487 \fi
1488 \thispagestyle{empty}% 欧文用標準スタイルでは plain
1489 \if@twocolumn
1490 \onecolumn
1491 \@restonecoltrue
1492 \else
1493 \@restonecolfalse
1494 \fi
1495 \null\vfil
1496 \secdef\@part\@spart}
1497 %</book|report>

```

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

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

```

1498 %<*!book&!report>
1499 \def\@part[#1]#2{%
1500 \ifnum \c@secnumdepth >\m@ne
1501 \refstepcounter{part}%
1502 \addcontentsline{toc}{part}{%
1503 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1504 \else
1505 \addcontentsline{toc}{part}{#1}%
1506 \fi
1507 \markboth{}{}%
1508 {\parindent\z@
1509 \raggedright
1510 \interlinepenalty \@M
1511 \normalfont
1512 \ifnum \c@secnumdepth >\m@ne
1513 \Large\headfont\prepartname\thepart\postpartname
1514 \par\nobreak
1515 \fi
1516 \huge \headfont #2%
1517 \markboth{}{}\par}%
1518 \nobreak
1519 \vskip 3ex
1520 \@afterheading}

```

1521 %</*!book&!report>

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

1522 %<*book|report>

1523 \def\@part[#1]#2{%

1524 \ifnum \c@secnumdepth >-2\relax

1525 \refstepcounter{part}%

1526 \addcontentsline{toc}{part}{%

1527 \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%

1528 \else

1529 \addcontentsline{toc}{part}{#1}%

1530 \fi

1531 \markboth{}{}%

1532 {\centering

1533 \interlinepenalty \@M

1534 \normalfont

1535 \ifnum \c@secnumdepth >-2\relax

1536 \huge\headfont \prepartname\thepart\postpartname

1537 \par\vskip20\p@?

1538 \fi

1539 \Huge \headfont #2\par}%

1540 \@endpart}

1541 %</book|report>

\@spart 番号を付けない部です。

1542 %</*!book&!report>

1543 \def\@spart#1{%

1544 \parindent \z@ \raggedright

1545 \interlinepenalty \@M

1546 \normalfont

1547 \huge \headfont #1\par}%

1548 \nobreak

1549 \vskip 3ex

1550 \@afterheading}

1551 %</*!book&!report>

1552 %<*book|report>

1553 \def\@spart#1{%

1554 \centering

1555 \interlinepenalty \@M

1556 \normalfont

1557 \Huge \headfont #1\par}%

1558 \@endpart}

1559 %</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。

1560 %<*book|report>

1561 \def\@endpart{\vfil\newpage

1562 \if@twoside

```

1563 \null
1564 \thispagestyle{empty}%
1565 \newpage
1566 \fi
1567 \if@restonecol
1568 \twocolumn
1569 \fi}
1570 %</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。

```

1571 %<*book|report>
1572 \newcommand{\chapter}{%
1573 \if@openright\cleardoublepage\else\clearpage\fi
1574 \plainifnotempty % 元: \thispagestyle{plain}
1575 \global\@topnum\z@
1576 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1577 \secdef
1578 {\@omit@numberfalse\@chapter}%
1579 {\@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。

```

1580 \def\@chapter[#1]#2{%
1581 \ifnum \c@secnumdepth >\m@ne
1582 \if@mainmatter
1583 \refstepcounter{chapter}%
1584 \typeout{\@chapapp\thechapter\@chappos}%
1585 \addcontentsline{toc}{chapter}%
1586 {\protect\numberline
1587 % {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
1588 {\@chapapp\thechapter\@chappos}%
1589 #1}%
1590 \else\addcontentsline{toc}{chapter}{#1}\fi
1591 \else
1592 \addcontentsline{toc}{chapter}{#1}%
1593 \fi
1594 \chaptermark{#1}%
1595 \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
1596 \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
1597 \if@twocolumn
1598 \@topnewpage[\@makechapterhead{#2}]%
1599 \else
1600 \@makechapterhead{#2}%
1601 \@afterheading
1602 \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。

```
1603 \def\@makechapterhead#1{%
1604   \vspace*{2\Cvs}% 欧文は 50pt
1605   {\parindent \z@ \raggedright \normalfont
1606     \ifnum \c@secnumdepth >\m@ne
1607       \if@mainmatter
1608         \huge\headfont \@chapapp\thechapter\@chappos
1609         \par\nobreak
1610         \vskip \Cvs % 欧文は 20pt
1611       \fi
1612     \fi
1613     \interlinepenalty\@M
1614     \Huge \headfont #1\par\nobreak
1615     \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。

```
1616 \def\@schapter#1{%
1617   \chaptermark{#1}%
1618   \if@twocolumn
1619     \@topnewpage[\@makeschapterhead{#1}]%
1620   \else
1621     \@makeschapterhead{#1}\@afterheading
1622   \fi}
```

`\@makeschapterhead` 番号なしの章見出しです。

```
1623 \def\@makeschapterhead#1{%
1624   \vspace*{2\Cvs}% 欧文は 50pt
1625   {\parindent \z@ \raggedright
1626     \normalfont
1627     \interlinepenalty\@M
1628     \Huge \headfont #1\par\nobreak
1629     \vskip 3\Cvs}} % 欧文は 40pt
1630 %</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。

```
1631 \if@twocolumn
1632   \newcommand{\section}{%
1633     %<jspf>\ifx\maketitle\relax\else\maketitle\fi
1634     \@startsection{section}{1}{\z@}%
1635     %<kiyou> {0.6\Cvs}{0.4\Cvs}%
1636     %<kiyou> {\Cvs}{0.5\Cvs}%
1637     % {\normalfont\large\headfont\@secapp}}
1638     {\normalfont\large\headfont\raggedright}}
1639 \else
```

```

1640 \newcommand{\section}{%
1641 \if@slide\clearpage\fi
1642 \@startsection{section}{1}{\z@}%
1643 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1644 {.5\Cvs \@plus.3\Cdp}% 後アキ
1645 % {\normalfont\Large\headfont\@secapp}%
1646 {\normalfont\Large\headfont\raggedright}}
1647 \fi

```

`\subsection` 同上です。

```

1648 \if@twocolumn
1649 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1650 {\z@}{\z@}%
1651 {\normalfont\normalsize\headfont}}
1652 \else
1653 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1654 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1655 {.5\Cvs \@plus.3\Cdp}% 後アキ
1656 {\normalfont\large\headfont}}
1657 \fi

```

`\subsubsection`

```

1658 \if@twocolumn
1659 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1660 {\z@}{\z@}%
1661 {\normalfont\normalsize\headfont}}
1662 \else
1663 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1664 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1665 {\z@}%
1666 {\normalfont\normalsize\headfont}}
1667 \fi

```

`\paragraph` 見出しの後ろで改行されません。

段落のマーク (■) が必ず和文フォントで出力されるようにする。さらに、このマークを
変更可能にするため `\jsParagraphMark` というマクロに切り出した。

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

```

1668 \newcommand\jsParagraphMark{\jsJaChar{■}}
1669 \let\jsJaChar\empty
1670 \if@twocolumn
1671 \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1672 {\z@}{-1\jsZw}% 改行せず 1\jsZw のアキ
1673 %<jspf> {\normalfont\normalsize\headfont}}
1674 %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1675 \else

```

```

1676 \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1677     {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1678     {-1\jsZw}% 改行せず 1\jsZw のアキ
1679 %<jspf>     {\normalfont\normalsize\headfont}}
1680 %<!jspf>     {\normalfont\normalsize\headfont\jsParagraphMark}}
1681 \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

1682 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1683     {\z@}{-1\jsZw}%
1684     {\normalfont\normalsize\headfont}}

```

7.3 リスト環境

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

`\leftmargini` 二段組であるかないかに応じてそれぞれ 2em, 2.5em ですが、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。

```

1685 \if@slide
1686 \setlength\leftmargini{1\jsZw}
1687 \else
1688 \if@twocolumn
1689 \setlength\leftmargini{2\jsZw}
1690 \else
1691 \setlength\leftmargini{3\jsZw}
1692 \fi
1693 \fi

```

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

```

\leftmarginiv 1694 \if@slide
\leftmarginiv 1695 \setlength\leftmarginii {1\jsZw}
\leftmarginiv 1696 \setlength\leftmarginiii {1\jsZw}
\leftmarginiv 1697 \setlength\leftmarginiv {1\jsZw}
1698 \setlength\leftmarginiv {1\jsZw}
1699 \setlength\leftmarginvi {1\jsZw}
1700 \else
1701 \setlength\leftmarginii {2\jsZw}
1702 \setlength\leftmarginiii {2\jsZw}
1703 \setlength\leftmarginiv {2\jsZw}
1704 \setlength\leftmarginiv {1\jsZw}
1705 \setlength\leftmarginvi {1\jsZw}
1706 \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。

```
1707 \setlength \labelsep {0.5\jsZw} % .5em
1708 \setlength \labelwidth{\leftmargini}
1709 \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。

```
1710 \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 1711 \@beginparpenalty -\@lowpenalty
\@itempenalty 1712 \@endparpenalty -\@lowpenalty
1713 \@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。

```
1714 \def\@listi{\leftmargin\leftmargini
1715 \parsep \z@
1716 \topsep 0.5\baselineskip
1717 \itemsep \z@ \relax}
1718 \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）。

```
1719 \@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 1720 \def\@listii{\leftmargin\leftmarginii
\@listiv 1721 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1722 \topsep \z@
\@listv 1723 \parsep \z@
\@listvi 1724 \itemsep\parsep}
1725 \def\@listiii{\leftmargin\leftmarginiii
1726 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1727 \topsep \z@
1728 \parsep \z@
1729 \itemsep\parsep}
1730 \def\@listiv {\leftmargin\leftmarginiv
1731 \labelwidth\leftmarginiv
1732 \advance\labelwidth-\labelsep}
1733 \def\@listv {\leftmargin\leftmarginv
```

```

1734 \labelwidth\leftmarginv
1735 \advance\labelwidth-\labelsep}
1736 \def\@listvi {\leftmargin\leftmarginvi
1737 \labelwidth\leftmarginvi
1738 \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。

```

1739 \renewcommand{\theenumi}{\@arabic\c@enumi}
1740 \renewcommand{\theenumii}{\@alph\c@enumii}
1741 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1742 \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。

```

1743 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1744 \newcommand*{\jsInJaParen}[1]{%
1745 \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1746 \newcommand{\labelenumi}{\theenumi.}
1747 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1748 \newcommand{\labelenumiii}{\theenumiii.}
1749 \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 1750 \renewcommand{\p@enumii}{\theenumi}
1751 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1752 \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`
`\labelitemiii`
`\labelitemiv`

```

1753 \newcommand\labelitemi{\textbullet}
1754 \newcommand\labelitemii{\normalfont\bfseries \textendash}
1755 \newcommand\labelitemiii{\textasteriskcentered}
1756 \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。

```

1757 \newenvironment{description}{%
1758   \list{}{%
1759     \labelwidth=\leftmargin
1760     \labelsep=1\jsZw
1761     \advance \labelwidth by -\labelsep
1762     \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。

```

1763 \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。

```

1764 %<*book|report>
1765 \newenvironment{abstract}{%
1766   \begin{list}{}{%
1767     \listparindent=1\jsZw
1768     \itemindent=\listparindent
1769     \rightmargin=\z@
1770     \leftmargin=5\jsZw\item[]}{\end{list}\vspace{\baselineskip}}
1771 %</book|report>
1772 %<*article|slide>
1773 \newbox\@abstractbox
1774 \if@titlepage
1775   \newenvironment{abstract}{%
1776     \titlepage
1777     \null\vfil
1778     \@beginparpenalty\@lowpenalty
1779     \begin{center}%
1780       \headfont \abstractname
1781       \@endparpenalty\@M
1782     \end{center}}%
1783   {\par\vfil\null\endtitlepage}

```

```

1784 \else
1785   \newenvironment{abstract}{%
1786     \if@twocolumn
1787       \ifx\maketitle\relax
1788         \section*{\abstractname}%
1789       \else
1790         \global\setbox\@abstractbox\hbox\bgroup
1791         \begin{minipage}[b]{\textwidth}
1792           \small\parindent1\jsZw
1793           \begin{center}%
1794             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1795           \end{center}%
1796           \list{}{%
1797             \listparindent\parindent
1798             \itemindent \listparindent
1799             \rightmargin \leftmargin}%
1800           \item\relax
1801         \fi
1802       \else
1803         \small
1804         \begin{center}%
1805           {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1806         \end{center}%
1807         \list{}{%
1808           \listparindent\parindent
1809           \itemindent \listparindent
1810           \rightmargin \leftmargin}%
1811         \item\relax
1812       \fi}{\if@twocolumn
1813         \ifx\maketitle\relax
1814         \else
1815           \endlist\end{minipage}\egroup
1816         \fi
1817       \else
1818         \endlist
1819       \fi}
1820 \fi
1821 %</article|slide>
1822 %<*jspf>
1823 \newbox\@abstractbox
1824 \newenvironment{abstract}{%
1825   \global\setbox\@abstractbox\hbox\bgroup
1826   \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
1827     \small
1828     \if@english \parindent6mm \else \parindent1\jsZw \fi}%
1829   {\end{minipage}\egroup}
1830 %</jspf>

```

■キーワード

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

```
1831 %<*jspf>
1832 %\newbox\@keywordsbox
1833 %\newenvironment{keywords}{%
1834 % \global\setbox\@keywordsbox\hbox\bgroup
1835 % \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
1836 % \small\parindent0\jsZw}%
1837 % {\end{minipage}\egroup}
1838 %</jspf>
```

■verse 環境

verse 詩のための `verse` 環境です。

```
1839 \newenvironment{verse}{%
1840 \let \\\=\@centercr
1841 \list{}{%
1842 \itemsep \z@
1843 \itemindent -2\jsZw % 元: -1.5em
1844 \listparindent\itemindent
1845 \rightmargin \z@
1846 \advance\leftmargin 2\jsZw}% 元: 1.5em
1847 \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。

```
1848 \newenvironment{quotation}{%
1849 \list{}{%
1850 \listparindent\parindent
1851 \itemindent\listparindent
1852 \rightmargin \z@}%
1853 \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。

```
1854 \newenvironment{quote}%
1855 {\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。

```
1856 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
1857   \item[\hskip \labelsep{\headfont #1\ #2}]}
1858 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
1859   \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。

```
1860 \newenvironment{titlepage}{%
1861   %<book|report> \cleardoublepage
1862   \if@twocolumn
1863     \@restonecoltrue\onecolumn
1864   \else
1865     \@restonecolfalse\newpage
1866   \fi
1867   \thispagestyle{empty}%
1868   \setcounter{page}\@ne
1869 }%
1870 {\if@restonecol\twocolumn \else \newpage \fi
1871   \if@twoside\else
1872     \setcounter{page}\@ne
1873   \fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
1874 %<!*book&!report>
1875 \newcommand{\appendix}{\par
1876   \setcounter{section}{0}%
1877   \setcounter{subsection}{0}%
1878   \gdef\presectionname{\appendixname}%
1879   \gdef\postsectionname{}}%
1880 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
1881 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
1882 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
1883 %</!*book&!report>
1884 %<*book|report>
1885 \newcommand{\appendix}{\par
1886   \setcounter{chapter}{0}%
1887   \setcounter{section}{0}%
1888   \gdef\@chapapp{\appendixname}%
1889   \gdef\@chappos{}}%
1890 \gdef\thechapter{\@Alph\c@chapter}}
1891 %</book|report>

```

7.4 パラメータの設定

■array と tabular 環境

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

```
1892 \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。

```
1893 \setlength\tabcolsep{6\p@?}
```

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

```
1894 \setlength\arrayrulewidth{.4\p@}
```

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

```
1895 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` `\'` コマンドで入るアキです。

```
1896 \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。

```
1897 \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。

```
1898 \setlength\fboxsep{3\p@?}
```

```
1899 \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。

```
1900 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```
1901 %<*book|report>
```

```
1902 \@addtoreset{equation}{chapter}
```

```
1903 \renewcommand\theequation
```

```
1904 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

```
1905 %</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。

```
1906 % \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。

```
1907 % \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。

```
1908 % \def\tagform@#1{\maketag@@@{ (\ignorespaces#1\unskip\@italiccorr) }}
```

7.5 フロート

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

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

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

`\ext@TYPE` フロートの目次を出力するファイルの拡張子です。

`\fnum@TYPE` キャプション用の番号を生成するマクロです。

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

■figure 環境

`\c@figure` 図番号のカウンタです。

`\thefigure` 図番号を出力するコマンドです。

```
1909 %<!*book&!report>
1910 \newcounter{figure}
1911 \renewcommand \thefigure {\@arabic\c@figure}
1912 %</!*book&!report>
1913 %<!*book|report>
1914 \newcounter{figure}[chapter]
1915 \renewcommand \thefigure
1916     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
1917 %</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 1918 \def\fps@figure{tbp}
1919 \def\ftype@figure{1}
\fnum@figure 1920 \def\ext@figure{lof}
1921 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 1922 \newenvironment{figure}%
1923     {\@float{figure}}%
1924     {\end@float}
```

```

1925 \newenvironment{figure*}%
1926         {\@dblfloat{figure}}%
1927         {\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。

```

1928 %<!*book&!report>
1929 \newcounter{table}
1930 \renewcommand\thetable{\@arabic\c@table}
1931 %<\/!book&!report>
1932 %<*book|report>
1933 \newcounter{table}[chapter]
1934 \renewcommand \thetable
1935         {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
1936 %<\/book|report>

```

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

```

\ext@table 1937 \def\fps@table{tbp}
1938 \def\ftype@table{2}
\fnun@table 1939 \def\ext@table{lot}
1940 \def\fnun@table{\tablename\nobreak\thetable}

```

`table` * は段抜きのフロートです。

```

table* 1941 \newenvironment{table}%
1942         {\@float{table}}%
1943         {\end@float}
1944 \newenvironment{table*}%
1945         {\@dblfloat{table}}%
1946         {\end@dblfloat}

```

7.6 キャプション

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

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

```

1947 \newlength\abovecaptionskip
1948 \newlength\belowcaptionskip
1949 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
1950 \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。

```
1951 %<*!jspf>
1952 % \long\def\@makecaption#1#2{{\small
1953 % \advance\leftskip1cm
1954 % \advance\rightskip1cm
1955 % \vskip\abovecaptionskip
1956 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
1957 % \ifdim \wd\@tempboxa >\hsize
1958 % #1\hskip1\jsZw\relax #2\par
1959 % \else
1960 % \global \@minipagefalse
1961 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1962 % \fi
1963 % \vskip\belowcaptionskip}}
1964 \long\def\@makecaption#1#2{{\small
1965 \advance\leftskip .0628\linewidth
1966 \advance\rightskip .0628\linewidth
1967 \vskip\abovecaptionskip
1968 \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
1969 \ifdim \wd\@tempboxa <\hsize \centering \fi
1970 #1\hskip1\jsZw\relax #2\par
1971 \vskip\belowcaptionskip}}
1972 %</!jspf>
1973 %<*jspf>
1974 \long\def\@makecaption#1#2{%
1975 \vskip\abovecaptionskip
1976 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
1977 \ifdim \wd\@tempboxa >\hsize
1978 {\small\sffamily
1979 \list{#1}{%
1980 \renewcommand{\makelabel}[1]{##1\hfil}
1981 \itemsep \z@
1982 \itemindent \z@
1983 \labelsep \z@
1984 \labelwidth 11mm
1985 \listparindent\z@
1986 \leftmargin 11mm}\item\relax #2\endlist}
1987 \else
1988 \global \@minipagefalse
1989 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1990 \fi
1991 \vskip\belowcaptionskip}
1992 %</jspf>
```

8 フォントコマンド

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

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

[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。

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

```

1993 \newif\if@jsc@warnoldfontcmd
1994 \@jsc@warnoldfontcmdtrue
1995 \newif\if@jsc@warnoldfontcmdexception
1996 \@jsc@warnoldfontcmdexceptionfalse

1
1997 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
1998   \g@addto@macro\bxjs@oldfontcmd\list{\do#1}%
1999   \DeclareOldFontCommand{#1}{%
2000     \bxjs@oldfontcmd{#1}#2%
2001   }{%
2002     \bxjs@oldfontcmd{#1}#3%
2003   }%
2004 }
2005 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2006   \ClassInfo\bxjs@clsname
2007   {Old font command '\string#1' is used!!\MessageBreak
2008     First occurrence is}%
2009 }

```

`\allowoldfontcommands` “二文字フォント命令”の使用を許可する（警告しない）。

`\disallowoldfontcommands` “三文字フォント命令”の使用に対して警告を出す。

```
2010 \newcommand*{\allowoldfontcommands}{%
2011   \@jsc@warnoldfontcmdexceptiontrue}%
2012 \newcommand*{\disallowoldfontcommands}{%
2013   \@jsc@warnoldfontcmdexceptionfalse}
```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2014 \let\bxjs@oldfontcmd@list\@empty
2015 \def\bxjs@oldfontcmd#1{%
2016   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2017 \def\bxjs@oldfontcmd@a#1#2{%
2018   \if@jsc@warnoldfontcmdexception\else
2019     \global\@jsc@warnoldfontcmdfalse
2020     \ifx#1\relax
2021       \global\let#1=t%
2022       \jsc@warnoldfontcmd{#2}%
2023     \fi
2024   \fi}
2025 \def\bxjs@warnoldfontcmd@final{%
2026   \par
2027   \let\@tempa\@empty
2028   \def\do##1{%
2029     \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2030       \edef\@tempa{\@tempa \space\string##1}\fi}
2031   \bxjs@oldfontcmd@list
2032   \ifx\@tempa\@empty\else
2033     \ClassWarningNoLine\bxjs@clsname
2034     {Some old font commands were used in text\MessageBreak
2035      (see the log output for detail)}%
2036     \ClassInfo\bxjs@clsname
2037     {Some old font commands were used in text:\MessageBreak
2038      \space\@tempa\MessageBreak
2039      You should note, that since 1994 LaTeX2e provides a\MessageBreak
2040      new font selection scheme called NFSS2 with several\MessageBreak
2041      new, combinable font commands. The
2042      class provides\MessageBreak
2043      the old font commands
2044      only for compatibility%
2045      \@gobble}%
2046   \fi}
2047 \AtEndDocument{\bxjs@warnoldfontcmd@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2048 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2049 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 2050 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf 2051 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 2052 \jsc@DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

```

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

```

2053 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}

```

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

`\sl`

`\sc`

せん（警告メッセージを出力します）。通常のアップライト体に戻すコマンドは `\upshape` です。

```
2054 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2055 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2056 \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 2057 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
2058 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

9 相互参照

9.1 目次の類

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

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

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

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

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

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

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

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

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

という書式です。

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

インデント 左側の字下げ量です。

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

`\@pnumwidth` ページ番号の入る箱の幅です。

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

`\@dotsep` 点の間隔です（単位 μ ）。

`\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。

```
2059 \newcommand\@pnumwidth{1.55em}
2060 \newcommand\@tocrmarg{2.55em}
2061 \newcommand\@dotsep{4.5}
2062 %<!book&!report>\setcounter{tocdepth}{2}
2063 %<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)

```
2064 \newdimen\jsc@tocl@width
2065 \newcommand{\tableofcontents}{%
2066 %<*book|report>
2067 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2068 \settowidth\@tempdima{\headfont\appendixname}%
2069 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2070 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2071 \if@twocolumn
2072 \@restonecoltrue\onecolumn
2073 \else
2074 \@restonecolfalse
2075 \fi
2076 \chapter*{\contentsname}%
2077 \@mkboth{\contentsname}{}%
2078 %</book|report>
2079 %<*!book&!report>
2080 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2081 \settowidth\@tempdima{\headfont\appendixname}%
2082 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2083 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2084 \section*{\contentsname}%
2085 \@mkboth{\contentsname}{\contentsname}%
2086 %</!book&!report>
2087 \@starttoc{toc}%
2088 %<book|report> \if@restonecol\twocolumn\fi
2089 }
```

`\l@part` 部の目次です。

```
2090 \newcommand*{\l@part}[2]{%
2091 \ifnum \c@tocdepth >-2\relax
2092 %<!book&!report> \addpenalty\@secpenalty
2093 %<book|report> \addpenalty{-\@highpenalty}%
2094 \addvspace{2.25em \@plus\p@?}%
2095 \begingroup
2096 \parindent \z@
```

```

2097 %      \@pnumwidth should be \@tocrmarg
2098 %      \rightskip \@pnumwidth
2099      \rightskip \@tocrmarg
2100      \parfillskip -\rightskip
2101      {\leavevmode
2102       \large \headfont
2103       \setlength\@lnumwidth{4\jsZw}%
2104       #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2105      \nobreak
2106 %<book|report>      \global\@nobreaktrue
2107 %<book|report>      \everypar{\global\@nobreakfalse\everypar{}}%
2108      \endgroup
2109      \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)

```

2110 %<*book|report>
2111 \newcommand*{\l@chapter}[2]{%
2112   \ifnum \c@tocdepth >\m@ne
2113     \addpenalty{-\@highpenalty}%
2114     \addvspace{1.0em \@plus\p@?}
2115 %   \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
2116     \begingroup
2117       \parindent\z@
2118 %     \rightskip\@pnumwidth
2119     \rightskip\@tocrmarg
2120     \parfillskip-\rightskip
2121     \leavevmode\headfont
2122 %   % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2123     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2124     \advance\leftskip\@lnumwidth \hskip-\leftskip
2125     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2126     \penalty\@highpenalty
2127   \endgroup
2128   \fi}
2129 %</book|report>

```

\l@section 節の目次です。

```

2130 %<*!book&!report>
2131 \newcommand*{\l@section}[2]{%
2132   \ifnum \c@tocdepth >\z@
2133     \addpenalty{\@secpenalty}%
2134     \addvspace{1.0em \@plus\p@?}%
2135     \begingroup
2136       \parindent\z@
2137 %     \rightskip\@pnumwidth
2138     \rightskip\@tocrmarg
2139     \parfillskip-\rightskip

```

```

2140      \leavevmode\headfont
2141 %      % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2142      \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2143      \advance\leftskip\@lnumwidth \hskip-\leftskip
2144      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2145      \endgroup
2146      \fi}
2147 %</*!book&!report>

```

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

```

2148 %<book|report> % \newcommand*\l@section{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

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

```

\l@section さらに下位レベルの目次項目の体裁です。あまり使ったことがありませんので、要修正かも
 \l@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 2149 %</*!book&!report>
2150 % \newcommand*\l@section{\@dottedtocline{2}{1.5em}{2.3em}}
2151 % \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
2152 % \newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
2153 % \newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
2154 %
2155 % \newcommand*\l@subsubsection{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2156 % \newcommand*\l@paragraph{\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2157 % \newcommand*\l@subparagraph{\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2158 %
2159 \newcommand*\l@section{%
2160     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2161     \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2162 \newcommand*\l@subsubsection{%
2163     \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2164     \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2165 \newcommand*\l@paragraph{%
2166     \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2167     \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2168 \newcommand*\l@subparagraph{%
2169     \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2170     \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2171 %</*!book&!report>
2172 %<*book|report>
2173 % \newcommand*\l@section{\@dottedtocline{2}{3.8em}{3.2em}}
2174 % \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
2175 % \newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
2176 % \newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}
2177 \newcommand*\l@section{%
2178     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2179     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2180 \newcommand*\l@subsubsection{%

```

```

2181 \tempdima\jsc@tocl@width \advance\tempdima 2.683\jsZw
2182 \dottedtocline{2}{\tempdima}{3.5\jsZw}}
2183 \newcommand*\l@subsubsection}{%
2184 \tempdima\jsc@tocl@width \advance\tempdima 6.183\jsZw
2185 \dottedtocline{3}{\tempdima}{4.5\jsZw}}
2186 \newcommand*\l@paragraph}{%
2187 \tempdima\jsc@tocl@width \advance\tempdima 10.683\jsZw
2188 \dottedtocline{4}{\tempdima}{5.5\jsZw}}
2189 \newcommand*\l@subparagraph}{%
2190 \tempdima\jsc@tocl@width \advance\tempdima 16.183\jsZw
2191 \dottedtocline{5}{\tempdima}{6.5\jsZw}}
2192 %</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。

```

2193 \newdimen\lnumwidth
2194 \def\numberline#1{\hbext@\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。

```

2195 \def\dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2196 \vskip \z@ \@plus.2\p@?
2197 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2198 \parindent #2\relax\@afterindenttrue
2199 \interlinepenalty\@M
2200 \leavevmode
2201 \@lnumwidth #3\relax
2202 \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2203 {#4}\nobreak
2204 \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2205 mu$}\hfill \nobreak\hbext@\pnumwidth{%
2206 \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2207 \newcommand*\listoffigures}{%
2208 %<*book|report>
2209 \if@twocolumn\@restonecoltrue\onecolumn
2210 \else\@restonecolfalse\fi
2211 \chapter*{\listfigurename}%
2212 \mkboth{\listfigurename}{}%
2213 %</book|report>
2214 %<!*book&!report>
2215 \section*{\listfigurename}%
2216 \mkboth{\listfigurename}{\listfigurename}%

```

```

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

```

`\l@figure` 図目次の項目を出力します。

```

2221 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2222 \newcommand{\listoftables}{%
2223 %<*book|report>
2224 \if@twocolumn\@restonecoltrue\onecolumn
2225 \else\@restonecolfalse\fi
2226 \chapter*{\listtablename}%
2227 \@mkboth{\listtablename}{}%
2228 %</book|report>
2229 %<*!book&!report>
2230 \section*{\listtablename}%
2231 \@mkboth{\listtablename}{\listtablename}%
2232 %</!book&!report>
2233 \starttoc{lot}%
2234 %<book|report> \if@restonecol\twocolumn\fi
2235 }

```

`\l@table` 表目次は図目次と同じです。

```

2236 \let\l@table\l@figure

```

9.2 参考文献

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

```

2237 \newdimen\bibindent
2238 \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。

```

2239 \newenvironment{thebibliography}[1]{%
2240 \@jsc@warnoldfontcmdexceptiontrue
2241 \global\let\presectionname\relax
2242 \global\let\postsectionname\relax
2243 %<article|slide> \section*{\refname}\@mkboth{\refname}{\refname}%
2244 %<*kiyou>
2245 \vspace{1.5\baselineskip}
2246 \subsubsection*{\refname}\@mkboth{\refname}{\refname}%
2247 \vspace{0.5\baselineskip}
2248 %</kiyou>

```

```

2249 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2250 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2251 \list{\@biblabel{\@arabic{c@enumiv}}}%
2252 {\settowidth\labelwidth{\@biblabel{#1}}%
2253 \leftmargin\labelwidth
2254 \advance\leftmargin\labelsep
2255 \@openbib@code
2256 \usecounter{enumiv}%
2257 \let\p@enumiv\@empty
2258 \renewcommand\theenumiv{\@arabic{c@enumiv}}%
2259 %<kiyou> \small
2260 \sloppy
2261 \clubpenalty4000
2262 \@clubpenalty\clubpenalty
2263 \widowpenalty4000%
2264 \sfcode`.\@m}
2265 {\def\@noitemerr
2266 {\@latex@warning{Empty `thebibliography' environment}}}%
2267 \endlist}

```

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

```

2268 \newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

```

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

```

2269 \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。

```

2270 % \def\@biblabel#1{\jsInhibitGlue [#1] \jsInhibitGlue}

```

`\cite` 文献の番号を出力する部分は ltbibl.dtx で定義されていますが、コンマとカッコを和文 `\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必要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。

```

2271 % \def\@citex[#1]#2{%
2272 % \let\@citea\@empty
2273 % \@cite{\@for\@citeb:=#2\do
2274 % {\@citea\def\@citea{, \jsInhibitGlue\penalty\@m\ }%
2275 % \edef\@citeb{\expandafter\@firstofone\@citeb}%
2276 % \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2277 % \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
2278 % \G@refundefinedtrue
2279 % \@latex@warning
2280 % {Citation `'\@citeb' on page \thepage \space undefined}}%
2281 % {\hbox{\csname b@\@citeb\endcsname}}}{#1}}
2282 % \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。

```
2283 % \DeclareRobustCommand\cite{\unskip
2284 %   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]
2285 % \def\@cite#1#2{${\hbox{\scriptsize{#1}\if@tempswa
2286 %   , \jsInhibitGlue\ #2\fi}) }}$}
```

9.3 索引

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

```
2287 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2288   \if@twocolumn
2289     \onecolumn\@restonecolfalse
2290   \else
2291     \clearpage\@restonecoltrue
2292   \fi
2293   \columnseprule.4pt \columnsep 2\jsZw
2294   \ifx\multicols\@undefined
2295 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2296 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2297 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2298 %<!book&!report>    \twocolumn[\section*{\indexname}]%
2299   \else
2300     \ifdim\textwidth<\fullwidth
2301       \setlength{\evensidemargin}{\oddsidemargin}
2302       \setlength{\textwidth}{\fullwidth}
2303       \setlength{\linewidth}{\fullwidth}
2304 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
2305 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2306 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2307 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}]%
2308   \else
2309 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}%
2310 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}]%
2311 %<!book&!report>    \def\presectionname{}\def\postsectionname{}%
2312 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}]%
2313   \fi
2314   \fi
2315 %<book|report>      \@mkboth{\indexname}{}%
2316 %<!book&!report>    \@mkboth{\indexname}{\indexname}%
2317   \plainifnotempty % \thispagestyle{plain}
2318   \parindent\z@
2319   \parskip\z@ \@plus .3\p@?\relax
2320   \let\item\@idxitem
2321   \raggedright
2322   \footnotesize\narrowbaselines
```

```

2323 }{
2324   \ifx\multicols\undefined
2325     \if@restonecol\onecolumn\fi
2326   \else
2327     \end{multicols}
2328   \fi
2329   \clearpage
2330 }

```

`\@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 2331 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2332 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2333 \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。

```

2334 \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) などでいいでしょう。

```

2335 \newcommand\seename{\if@english see\else →\fi}
2336 \newcommand\alsoname{\if@english see also\else →\fi}

```

9.4 脚注

`\footnote` 和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため,

`\footnotemark` `\jsInhibitGlue` を入れることにします。

```

2337 \let\footnotes@ve=\footnote
2338 \def\footnote{\jsInhibitGlue\footnotes@ve}
2339 \let\footnotemarks@ve=\footnotemark
2340 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}

```

`\@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 `\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 pL^AT_EX の変更に従いました (Thanks: 角藤さん)。

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。

```
2341 \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。

```
2342 % \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。

```
2343 \renewcommand{\footnoterule}{%
```

```
2344   \kern-2.6\p@? \kern-.4\p@
```

```
2345   \hrule width .4\columnwidth
```

```
2346   \kern 2.6\p@?}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
2347 %<book|report>\@addtoreset{footnote}{chapter}
```

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

```
2348 \long\def\@footnotetext{%
```

```
2349   \insert\footins\bgroup
```

```
2350     \normalfont\footnotesize
```

```
2351     \interlinepenalty\interfootnotelinepenalty
```

```
2352     \splittopskip\footnotesep
```

```
2353     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
```

```
2354     \hsize\columnwidth \@parboxrestore
```

```
2355     \protected@edef\@currentlabel{%
```

```
2356       \csname p@footnote\endcsname\@thefnmark
```

```
2357   }%
```

```
2358   \color@begingroup
```

```
2359     \@makefnmark{%
```

```
2360       \rule\z@\footnotesep\ignorespaces}%
```

```
2361     \futurelet\next\fo@t}
```

```
2362 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@@
```

```
2363                                     \else \let\next\fo@t\fi \next}
```

```
2364 \def\fo@@t{\bgroup\aftergroup\@foot\let\next}
```

```
2365 \def\fo@t#1{#1\@foot}
```

```
2366 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup}
```

`\@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。

```
2367 \newcommand\@makefnmark[1]{%
```

```
2368   \advance\leftskip 3\jsZw
```

```
2369   \parindent 1\jsZw
```

```
2370   \noindent
```

```
2371   \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。

```
2372 % \def\@xfootnotenext[#1]{%
2373 %   \begingroup
2374 %     \ifnum#1>\z@
2375 %       \csname c@\@mpfn\endcsname #1\relax
2376 %       \unrestored@protected@xdef\@thefnmark{\thempfn}%
2377 %     \else
2378 %       \unrestored@protected@xdef\@thefnmark{}%
2379 %     \fi
2380 %   \endgroup
2381 %   \@footnotetext}
```

ここまでのコードは JS クラスを踏襲する。

10 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

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

```
2382 \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。

```
2383 \def\everyparhook{\jsInhibitGlueAtParTop}
2384 \AtBeginDocument{\everypar{\everyparhook}}
```

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

```
2385 \def\@inhibitglue{%
2386   \futurelet\@let@token\@inhibitglue}
2387 \def\@inhibitglue{%
2388   \ifx\@let@token 「%
2389     \jsInhibitGlue
2390   \else
2391     \ifx\@let@token (%
2392       \jsInhibitGlue
2393     \else
2394       \ifx\@let@token 『%
2395         \jsInhibitGlue
2396       \else
```

```

2397      \ifx\@let@token [%
2398      \jsInhibitGlue
2399      \fi
2400      \fi
2401      \fi
2402      \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。

```

2403 \def\@doendpe{%
2404   \@endpetrue
2405   \def\par{%
2406     \@restorepar\everypar{\everyparhook}\par\@endpefalse}%
2407   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

`\item` 命令の直後です。

```

2408 \def\@item[#1]{%
2409   \if@noperitem
2410     \@donoperitem
2411   \else
2412     \if@inlabel
2413       \indent \par
2414     \fi
2415     \ifhmode
2416       \unskip\unskip \par
2417     \fi
2418     \if@newlist
2419       \if@nobreak
2420         \@nbitem
2421       \else
2422         \addpenalty\@beginparpenalty
2423         \addvspace\@topsep
2424         \addvspace{-\parskip}%
2425       \fi
2426     \else
2427       \addpenalty\@itempenalty
2428       \addvspace\itemsep
2429     \fi
2430     \global\@inlabeltrue
2431   \fi
2432   \everypar{%
2433     \@minipagefalse
2434     \global\@newlistfalse
2435     \if@inlabel
2436       \global\@inlabelfalse
2437     {\setbox\z@\lastbox
2438       \ifvoid\z@

```

```

2439     \kern-\itemindent
2440     \fi}%
2441     \box\@labels
2442     \penalty\z@
2443     \fi
2444     \if@nobreak
2445         \@nobreakfalse
2446         \clubpenalty \@M
2447     \else
2448         \clubpenalty \@clubpenalty
2449         \everypar{\everyparhook}%
2450     \fi
2451     \bxjs@ltj@inhibitglue
2452     \everyparhook}%
2453 \if@noitemarg
2454     \@noitemargfalse
2455     \if@nmbrlist
2456         \refstepcounter\@listctr
2457     \fi
2458 \fi
2459 \sbox\@tempboxa{\makelabel{#1}}}%
2460 \global\setbox\@labels\hbox{%
2461     \unhbox\@labels
2462     \hskip \itemindent
2463     \hskip -\labelwidth
2464     \hskip -\labelsep
2465     \ifdim \wd\@tempboxa >\labelwidth
2466         \box\@tempboxa
2467     \else
2468         \hbox to\labelwidth {\unhbox\@tempboxa}%
2469     \fi
2470     \hskip \labelsep}%
2471 \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。

```

2472 \def\@afterheading{%
2473     \@nobreaktrue
2474     \everypar{%
2475         \if@nobreak
2476             \@nobreakfalse
2477             \clubpenalty \@M
2478             \if@afterindent \else
2479                 {\setbox\z@\lastbox}%
2480             \fi
2481         \else
2482             \clubpenalty \@clubpenalty
2483             \everypar{\everyparhook}%
2484         \fi\everyparhook}}

```

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

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

```
2485 \def\@gnewline #1{%
2486   \ifvmode
2487     \@nolnerr
2488   \else
2489     \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
2490     \jsInhibitGlue \ignorespaces
2491   \fi}
```

11 いろいろなロゴ

とりあえず削除。

12 amsmath との衝突の回避

最近の $\text{L}\text{a}\text{T}\text{E}\text{X}$ では該当の問題は対処されているので削除。

13 初期設定

■いろいろな語

```
\prepartname
\postpartname 2492 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername 2493 \newcommand{\postpartname}{\if@english\else 部\fi}
2494 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2495 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2496 \newcommand{\presectionname}{}% 第
\postsectionname 2497 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 2498 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2499 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
2500 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 2501 \newcommand{\refname}{\if@english References\else 参考文献\fi}
2502 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
\indexname
```

```

2503 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename

\tablename 2504 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
2505 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname

\abstractname 2506 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
2507 \newcommand{\appendixname}{\if@english \else 付録\fi}
2508 %<!book&!report>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

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

```

\today

2509 \@tempswafalse
2510 \if p\jsEngine \@tempswatrue \fi
2511 \if n\jsEngine \@tempswatrue \fi
2512 \if@tempswa \expandafter\@firstoftwo
2513 \else \expandafter\@secondoftwo
2514 \fi
2515 {%
2516 % 欧文 8bitTeX の場合
2517 \newif\ifjsSeireki \jsSeirekitrue
2518 \def\西暦{\jsSeirekitrue}
2519 \def\和暦{\jsSeirekifalse}
2520 \def\Seireki{\jsSeirekitrue}
2521 \def\Wareki{\jsSeirekifalse}
2522 \def\bxjs@if@use@seireki{%
2523 \ifjsSeireki \expandafter\@firstoftwo
2524 \else \expandafter\@secondoftwo \fi}
2525 }{%
2526 \newif\if 西暦 \西暦 true
2527 \def\西暦{\西暦 true}
2528 \def\和暦{\西暦 false}
2529 \def\Seireki{\西暦 true}
2530 \def\Wareki{\西暦 false}
2531 \def\bxjs@if@use@seireki{%
2532 \if 西暦 \expandafter\@firstoftwo
2533 \else \expandafter\@secondoftwo \fi}
2534 }
2535 % \bxjs@unxp
2536 \let\bxjs@unxp\@firstofone
2537 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2538 % \bxjs@iai
2539 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2540 \def\bxjs@iai{\noexpand~}
2541 \else \def\bxjs@iai{}

```

```

2542 \fi
2543 % \heisei
2544 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2545 % \today
2546 \edef\today{%
2547   \if@english
2548     \ifcase\month\or
2549       January\or February\or March\or April\or May\or June\or
2550       July\or August\or September\or October\or November\or December\fi
2551     \space\number\day, \number\year
2552   \else
2553     \noexpand\bxjs@if@use@seireki{%
2554       \number\year\bxjs@iai\bxjs@unxp{年}%
2555       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2556       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2557     }{%
2558       \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2559       \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2560       \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2561     }%
2562   \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）

```

2563 \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。

```

2564 %<slide>\pagestyle{empty}%
2565 %<article|report>\pagestyle{plain}%
2566 %<book>\pagestyle{headings}%
2567 \pagenumbering{arabic}
2568 \if@twocolumn
2569   \twocolumn
2570   \sloppy
2571   \flushbottom
2572 \else
2573   \onecolumn
2574   \raggedbottom
2575 \fi
2576 %<*slide>
2577 \renewcommand\familydefault{\sfdefault}
2578 \raggedright
2579 %</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。

```
2580 \catcode`\?=12
2581 \ifx\bxjs@jadriver\relax\else
2582 \input{bxjsja-\bxjs@jadriver.def}
2583 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
2584 \bxjs@restore@jltrcc
2585 %</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。

```
2586 %<*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

```
2587 %<*minimal>
```

```
2588 %% 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.
```

```
2589 \def\DeclareJaTextFontCommand#1#2{%
```

```
2590   \DeclareRobustCommand#1[1]{%
```

```
2591     \relax
```

```
2592     \ifmmode \expandafter\nfss@text \fi
```

```
2593     {#2##1}}%
```

```
2594 }
```

```
\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。
```

```
2595 \long\def\bxjs@@CSsfdefault{\sfdefault}%
```

```
2596 \@onlypreamble\bxjs@if@sf@default
```

```
2597 \def\bxjs@if@sf@default#1{%
```

```
2598   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
```

```
2599   \AtBeginDocument{%
```

```
2600     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
```

```
2601 }
```

`\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。

```
2602 \def\jsLetHeadChar#1#2{%
2603   \begingroup
2604     \escapechar=`\ %
2605     \let\bxjs@tmpa={% brace-match-hack
2606       \bxjs@let@hchar@exp#2}%
2607   \endgroup
2608   \let#1\bxjs@g@tmpa}
2609 \def\bxjs@let@hchar@exp{%
2610   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2611 \def\bxjs@let@hchar@exp@a{%
2612   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
2613     \bxjs@let@hchar@out\let\relax
2614   }{\bxjs@cond@ifcat\noexpand\@let@token\sptoken\fi{% 空白
2615     \bxjs@let@hchar@out\let\space%
2616   }{\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
2617     \bxjs@let@hchar@out\let\@backslashchar
2618   }{\bxjs@let@hchar@exp@b}}}}
2619 \def\bxjs@let@hchar@exp@b#1{%
2620   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2621 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2622   %\message{<#1#2>}%
2623   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴
2624     \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2625       \bxjs@let@hchar@out\let\relax
2626     }{%else
2627       \expandafter\bxjs@let@hchar@exp
2628     }%
2629   }{%else
2630     \bxjs@let@hchar@chr#1%
2631   }}
2632 \def\bxjs@let@hchar@chr#1{%
2633   \bxjs@let@hchar@out\def{{#1}}}
2634 \def\bxjs@let@hchar@out#1#2{%
2635   \global#1\bxjs@g@tmpa#2\relax
2636   \toks@\bgroup}% skip to right brace
```

UTF-8 のバイト列を扱うコード。

```
2637 \chardef\bxjs@let@hchar@csta=128
2638 \chardef\bxjs@let@hchar@cstb=192
2639 \chardef\bxjs@let@hchar@cstc=224
```

```

2640 \chardef\bxjs@let@hchar@cstd=240
2641 \chardef\bxjs@let@hchar@cste=248
2642 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2643 \def\bxjs@let@hchar@chr@ue#1{%
2644   \@tempcnta=#1\relax
2645   \%message{\the\@tempcnta}%
2646   \bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2647     \bxjs@let@hchar@chr@ue@a#1%
2648   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2649     \bxjs@let@hchar@out\let\relax
2650   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2651     \bxjs@let@hchar@chr@ue@b
2652   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2653     \bxjs@let@hchar@chr@ue@c
2654   }\bxjs@cond@ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2655     \bxjs@let@hchar@chr@ue@d
2656   }{%else
2657     \bxjs@let@hchar@out\let\relax
2658   }}}}
2659 \def\bxjs@let@hchar@chr@ue@a#1{%
2660   \bxjs@let@hchar@out\def{{#1}}}
2661 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2662   \bxjs@let@hchar@out\def{{#1#2}}}
2663 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2664   \bxjs@let@hchar@out\def{{#1#2#3}}}
2665 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2666   \bxjs@let@hchar@out\def{{#1#2#3#4}}}

```

B.2 (u)pTeX 用の設定

```
2667 \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。

```

2668 \def\bxjs@let@hchar@chr@pp#1{%
2669   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}
2670 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2671   \%message{(#1)}%
2672   \bxjs@cond@if#1t\fi{%
2673     \bxjs@let@hchar@chr@ue#3%
2674   }{%else
2675     \bxjs@let@hchar@out\def{{#3}}}%
2676   }}
2677 \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。

```
2678 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
2679 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
2680 \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。

```
2681 \@onlypreamble\bxjs@declarefontshape
2682 \ifjsWithupTeX
2683 \def\bxjs@declarefontshape{%
2684 \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
2685 \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
2686 \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
2687 \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
2688 }
2689 \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。

```
2690 \else
2691 \def\bxjs@declarefontshape{%
2692 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
2693 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
2694 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
2695 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
2696 }
2697 \def\bxjs@sizereference{jis}
2698 \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。

```
2699 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%
2700   \def\bxjs@y{#5}}
2701 \expandafter\expandafter\expandafter\bxjs@tmpa
2702 \expandafter\string\the\jfont\relax
2703 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
2704               \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
2705   {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
2706    \expandafter\let\csname\bxjs@x/\bxjs@y\endcsname=\@undefined}
```

■和文フォントスケールの補正 実は、`pTeX` の標準的な和文フォント（JFM のこと、例えば `jis`）では、指定された `\jsScale`（この値を s とする）をそのまま使って定義すると期待通りの大きさにならない。これらの JFM では `1zw` の大きさが指定されたサイズではなく

既にスケール（この値を f とする；jis では 0.962216 倍）が掛けられた値になっているからである。そのため、ここでは s/f を求めてその値をマクロ `\bxjs@scale` に保存する。

```

2707 \begingroup
2708 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2709 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2710 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2711 % 幅が丁度 10pt なら補正は不要
2712 \ifdim\wd\z@=10pt
2713   \global\let\bxjs@scale\jsScale
2714 \else
2715 % (10*s)/(10*f) として計算、\bxjs@invscale は BXJS で定義
2716 \edef\bxjs@tmpa{\strip@pt\wd\z@}
2717 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
2718 \bxjs@invscale\@tempdima\bxjs@tmpa
2719 \xdef\bxjs@scale{\strip@pt\@tempdima}
2720 \fi
2721 \endgroup
2722 %\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。

```

2723 \bxjs@declarefontshape
      フォント代替の明示的定義。
2724 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2725 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
2726 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
2727 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
2728 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
2729 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
2730 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
2731 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2732 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
2733 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
2734 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
2735 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
2736 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
2737 \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。

```

2738 \DeclareRobustCommand\rmfamily
2739 {\not@math@alphabet\rmfamily\mathrm
2740   \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
2741 \DeclareRobustCommand\sffamily
2742 {\not@math@alphabet\sffamily\mathsf
2743   \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
2744 \DeclareRobustCommand\ttfamily

```

```

2745 {\not@math@alphabet\ttfamily\mathtt
2746 \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
2747 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
2748 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
2749 \bxjs@if@sf@default{%
2750 \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

2751 \selectfont

```

■パラメタの設定

```

2752 \prebreakpenalty\jis"2147=10000
2753 \postbreakpenalty\jis"2148=10000
2754 \prebreakpenalty\jis"2149=10000
2755 \inhibitxspcode`!=1
2756 \inhibitxspcode`〒=2
2757 \xspcode`+=3
2758 \xspcode`\%=3

```

"80 || "FF の範囲の \spcode を 3 に変更。

```

2759 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
2760 \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

\jsInhibitGlueAtParTop の定義。

```

2761 \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。

```

2762 \begingroup
2763 \catcode`\!=0

```

\bxjs@ptex@dir 現在の組方向： t=縦、y=横、?=その他。

```

2764 \gdef\bxjs@ptex@dir{%
2765 !iftdir t%
2766 !else!ifydir y%
2767 !else ?%
2768 !fi!fi}

```

新版の pTeX で脚注番号の周囲の空きが過大になる現象への対処。

※現在の pLaTeX 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。

```

2769 % 古い \@makefnmark の定義
2770 \long\def\bxjs@tmpa{\hbox{%
2771 !ifydir \@textsuperscript{\normalfont\@thefnmark}%
2772 !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
2773 \ifx\@makefnmark\bxjs@tmpa

```

```

2774 \long\gdef\@makefnmark{%
2775   !ifdir \hbox{}\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}%
2776   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
2777 \fi
2778 \endgroup

```

B.3 pdf_TE_X 用の処理

```

2779 \else\ifx p\jsEngine
2780 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
2781 \@onlypreamble\bxjs@cjk@loaded
2782 \def\bxjs@cjk@loaded{%
2783   \def\@footnotemark{%
2784     \leavevmode
2785     \ifhmode
2786       \edef\@x@sf{\the\spacefactor}%
2787       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
2788         \unkern\unkern
2789         \ifdim\lastskip>\z@ \unskip \fi
2790       \fi\fi
2791       \nobreak
2792     \fi
2793     \@makefnmark
2794     \ifhmode \spacefactor\@x@sf \fi
2795     \relax}%
2796 \let\bxjs@cjk@loaded\relax
2797 }
2798 \AtBeginDocument{%
2799   \@ifpackageloaded{CJK}{%
2800     \bxjs@cjk@loaded
2801   }{}%
2802 }

```

B.4 X_YT_EX 用の処理

```

2803 \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。

2804 \def\bxjs@let@hchar@chr#1{%
2805   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
2806   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
2807     \bxjs@let@hchar@chr@xe
2808   }\bxjs@let@hchar@out\def{{#1}}}}
2809 \def\bxjs@let@hchar@chr@xe#1{%
2810   \lccode`0=`#1\relax
2811   \lowercase{\bxjs@let@hchar@out\def{{0}}}}

```

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

```

2812 \ifx\XeTeXgenerateactualtext\@undefined\else
2813   \def\bxjs@do@precisetext{%
2814     \XeTeXgenerateactualtext=\@ne}
2815 \fi

```

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

```

2816 \@onlypreamble\bxjs@do@simplejasetup
2817 \def\bxjs@do@simplejasetup{%
2818   \ifnum\XeTeXinterchartokenstate>\z@
2819   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
2820     \jsSimpleJaSetup
2821     \ClassInfo\bxjs@clsname
2822     {\string\jsSimpleJaSetup\space is applied\@gobble}%
2823   \fi\fi}

```

\jsSimpleJaSetup 日本語出力用の超簡易的な設定。

```

2824 \newcommand*{\jsSimpleJaSetup}{%
2825   \XeTeXlinebreaklocale "ja"\relax
2826   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
2827   \XeTeXlinebreakpenalty=0\relax}

```

B.5 後処理（エンジン共通）

```

2828 \fi\fi\fi

```

simplejasetup オプションの処理。

```

2829 \ifx\bxjs@do@simplejasetup\@undefined\else
2830   \AtBeginDocument{%
2831     \ifbxjs@simplejasetup
2832       \bxjs@do@simplejasetup
2833     \fi}
2834 \fi
2835 %
2836 % |precisetext| オプションの処理。
2837 %   \begin{macrocode}
2838 \ifbxjs@precisetext
2839   \ifx\bxjs@do@precisetext\@undefined
2840     \ClassWarning\bxjs@clsname
2841       {The current engine does not support the\MessageBreak
2842       'precisetext' option\@gobble}
2843   \else
2844     \bxjs@do@precisetext
2845   \fi
2846 \fi

```

以上で終わり。

```

2847 %</minimal>

```

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
2848 %<*standard>
2849 %% このファイルは日本語文字を含みます
2850 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
2851 \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 - $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 拡張が使えるか検査する。

```
2852 \ifjsWitheTeX
```

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

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

```
\jH2853 \@tempdima=0.25mm
2854 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
2855 \let\jH\jQ
```

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

```
\trueH2856 \ifjsc@mag
2857 \@tempdima=2.5mm
```

```

2858 \tempdimb=\jsBaseFontSize\relax
2859 \edef\bxjs@tmpa{\strip@pt\tempdimb}%
2860 \bxjs@invscale\tempdima\bxjs@tmpa
2861 \edef\trueQ{\dimexpr\the\tempdima\relax}
2862 \else \let\trueQ\jQ
2863 \fi
2864 \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} とすると、和文が 12Q になる。

```

2865 \tempdima\trueQ \bxjs@invscale\tempdima\jsScale
2866 \edef\ascQ{\dimexpr\the\tempdima\relax}
2867 \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。

```
2868 \def\bxjs@kanjiskip{0pt}
```

\setkanjiskip 和文間空白の量を設定する。

```

2869 \newcommand*\setkanjiskip[1]{%
2870 \edef\bxjs@kanjiskip{#1}%
2871 \bxjs@reset@kanjiskip}

```

\getkanjiskip 和文間空白の量を表すテキストに展開する。

```

2872 \newcommand*\getkanjiskip{%
2873 \bxjs@kanjiskip}

```

\ifbxjs@kanjiskip@enabled 和文間空白の挿入が有効か。ただし p_{TEX} では自身の \noautospacing の制御を用いるのでこの変数は常に真とする。

```
2874 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

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

```

\bxjs@disable@kanjiskip 2875 \bxjs@robust@def\bxjs@enable@kanjiskip{%
2876 \bxjs@kanjiskip@enabledtrue
2877 \bxjs@reset@kanjiskip}
2878 \bxjs@robust@def\bxjs@disable@kanjiskip{%
2879 \bxjs@kanjiskip@enabledfalse
2880 \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。

```

2881 \bxjs@robust@def\bxjs@reset@kanjiskip{%
2882 \ifbxjs@kanjiskip@enabled
2883 \setlength{\@tempskipa}{\bxjs@kanjiskip}%
2884 \else \@tempskipa\z@
2885 \fi
2886 \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

\getxkanjiskip

\ifbxjs@xkanjiskip@enabled

\bxjs@enable@xkanjiskip

\bxjs@disable@xkanjiskip

\bxjs@reset@xkanjiskip

```

2887 \def\bxjs@xkanjiskip{0pt}
2888 \newcommand*\setxkanjiskip[1]{%
2889   \edef\bxjs@xkanjiskip{#1}%
2890   \bxjs@reset@xkanjiskip}
2891 \newcommand*\getxkanjiskip{%
2892   \bxjs@xkanjiskip}
2893 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
2894 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
2895   \bxjs@xkanjiskip@enabledtrue
2896   \bxjs@reset@xkanjiskip}
2897 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
2898   \bxjs@xkanjiskip@enabledfalse
2899   \bxjs@reset@xkanjiskip}
2900 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
2901   \ifbxjs@xkanjiskip@enabled
2902     \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
2903   \else \@tempskipa\z@
2904   \fi
2905   \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。

```

2906 \g@addto@macro\jsResetDimen{%
2907   \bxjs@reset@kanjiskip
2908   \bxjs@reset@xkanjiskip}
2909 \let\bxjs@apply@kanjiskip\relax
2910 \let\bxjs@apply@xkanjiskip\relax

```

■和文フォント指定の扱い

\bxjs@adjust@jafont ムニヤムニヤ…。

```

2911 \@onlypreamble\bxjs@adjust@jafont
2912 \def\bxjs@adjust@jafont#1{%
2913   \ifx\jsJaFont\bxjs@auto
2914     \bxjs@get@kanjiEmbed
2915     \ifx\bxjs@kanjiEmbed\relax
2916       \let\bxjs@tmpa\@empty
2917     \else
2918       \let\bxjs@tmpa\bxjs@kanjiEmbed
2919     \fi
2920   \else
2921     \let\bxjs@tmpa\jsJaFont
2922   \fi
2923   \if f#1\ifx\bxjs@tmpa\bxjs@noEmbed
2924     \ClassWarning\bxjs@clsname
2925     {Option 'jafont=noEmbed' is ignored, because no-embed\MessageBreak
2926     is not available}%
2927     \let\bxjs@tmpa\@empty
2928   \fi\fi

```

```

2929 }
2930 \def\bxjs@@auto{auto}
2931 \def\bxjs@@noEmbed{noEmbed}

```

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

```

2932 \let\bxjs@kanjiEmbed\relax

```

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

```

2933 \@onlypreamble\bxjs@get@kanjiEmbed
2934 \def\bxjs@get@kanjiEmbed{%
2935   \begingroup\setbox\z@=\hbox{%
2936     \global\let\bxjs@g@tmpa\relax
2937     \endlinechar\m@ne
2938     \let\do\@makeother\dospecials
2939     \catcode32=10 \catcode12=10 %form-feed
2940     \let\bxjs@tmpa\@empty
2941     \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
2942     \ifeof\@inputcheck\else
2943       \read\@inputcheck to\bxjs@tmpa
2944       \closein\@inputcheck
2945     \fi
2946     \ifx\bxjs@tmpa\@empty\else
2947       \openin\@inputcheck="\bxjs@tmpa"\relax
2948       \@tempswatrue
2949       \loop\if@tempswa
2950         \read\@inputcheck to\bxjs@tmpa
2951         \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
2952       \ifx\bxjs@tmpa\relax\else
2953         \global\let\bxjs@g@tmpa\bxjs@tmpa
2954         \@tempswafalse
2955       \fi
2956       \ifeof\@inputcheck \@tempswafalse \fi
2957     \repeat
2958   \fi
2959 }\endgroup
2960 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
2961 }
2962 \@onlypreamble\bxjs@get@ke@a
2963 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
2964   \ifx$#1$\def\bxjs@tmpa{#2}%
2965   \else \let\bxjs@tmpa\relax
2966   \fi}

```

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

```

2967 \newcommand*\jachar[1]{%
2968   \begingroup

```

\jsLetHeadChar で先頭の “文字” を拾ってそれを \bxjs@jachar に渡す。

```

2969   \jsLetHeadChar\bxjs@tmpa{#1}%

```

```

2970 \ifx\bxjs@tmpa\relax
2971 \ClassWarningNoLine\bxjs@clsname
2972 {Illegal argument given to \string\jachar}%
2973 \else
2974 \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
2975 \fi
2976 \endgroup}

```

\jsJaChar を \jachar と等価にする。

```
2977 \let\jsJaChar\jachar
```

下請けの \bxjs@jachar の実装はエンジンにより異なる。

```
2978 \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。

```
2979 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```

2980 \@onlypreamble\bxjs@fix@hyperref@unicode
2981 \def\bxjs@fix@hyperref@unicode#1{%
2982 \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
2983 \@namedef{KV@Hyp@bxjs/hook}##1{%
2984 \KV@Hyp@unicode{##1}%
2985 \def\KV@Hyp@unicode####1{%
2986 \expandafter\ifx\csname if##1\expandafter\endcsname
2987 \csname if####1\endcsname\else
2988 \ClassWarningNoLine\bxjs@clsname
2989 {Blcoked hyperref option 'unicode=####1'}%
2990 \fi
2991 }%
2992 }%
2993 }

```

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

```

2994 \@onlypreamble\bxjs@urgent@special
2995 \def\bxjs@urgent@special#1{%
2996 \AtBeginDvi{\special{#1}}%
2997 \AtBeginDocument{%
2998 \ifpackageloaded{atbegshi}{%
2999 \begingroup
3000 \toks\z@{\special{#1}}%
3001 \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3002 \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3003 \endgroup
3004 }{}%
3005 }%
3006 }

```

C.2 pTeX 用設定

3007 \if j\jsEngine

■共通命令の実装

```
3008 \def\bxjs@apply@kanjiskip{%
3009   \kanjiskip\@tempskipa}
3010 \def\bxjs@apply@xkanjiskip{%
3011   \xkanjiskip\@tempskipa}
```

\jaJaChar のサブマクロ。

```
3012 \def\bxjs@jachar#1{%
3013   \bxjs@jachar@a#1...\@nil}
3014 \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。

```
3015   \ifx.#2#1%
```

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

```
3016   \else\ifx.#3%
3017     \@tempcnta`#1 \multiply\@tempcnta64
3018     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3019     \bxjs@jachar@b
3020   \else\ifx.#4%
3021     \@tempcnta`#1 \multiply\@tempcnta64
3022     \advance\@tempcnta`#2 \multiply\@tempcnta64
3023     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3024     \bxjs@jachar@b
3025   \else
3026     \@tempcnta`#1 \multiply\@tempcnta64
3027     \advance\@tempcnta`#2 \multiply\@tempcnta64
3028     \advance\@tempcnta`#3 \multiply\@tempcnta64
3029     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3030     \bxjs@jachar@b
3031   \fi\fi\fi}
```

符号値が \@tempcnta の和文文字を出力する処理。

```
3032 \ifjsWithupTeX
3033   \def\bxjs@jachar@b{\kchar\@tempcnta}
3034 \else
3035   \def\bxjs@jachar@b{%
3036     \ifx\bxUInt\@undefined\else
3037       \bxUInt{\@tempcnta}%
3038     \fi}
3039 \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。

```
3040 \let\bxjs@tmpa\jsJaFont
3041 \ifx\bxjs@tmpa\bxjs@@auto
3042   \let\bxjs@tmpa\@empty
3043 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3044   \def\bxjs@tmpa{noembed}
3045 \fi\fi
3046 \ifx\jsJaFont\@empty\else
3047   \edef\bxjs@nxt{%
3048     \noexpand\RequirePackage[\jsJaFont]
3049       {pxchfon}[2010/05/12]}% v0.5
3050   \bxjs@nxt
3051 \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……。)

```
3052 \begingroup
3053   \global\let\@gtempa\relax
3054   \catcode`\|=0 \catcode`\|=12
3055   |def|bxjs@check#1|@nil{%
3056     |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3057   |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3058     |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3059   |catcode`\|=0 \catcode`\|=12
3060   \def\bxjs@check@b#1keyval#2\@nnil{%
3061     \ifx$#2$\else
3062       \xdef\@gtempa{%
3063         \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3064     \fi}
3065 \@firstofone{%
3066   \catcode10=12 \endlinechar\m@ne
3067   \let\do\@makeother \dospecials \catcode32=10
3068   \openin\@inputcheck=otf.sty\relax
3069   \@tempswatrue
3070   \loop\if@tempswa
3071     \ifeof\@inputcheck \@tempswafalse \fi
3072     \if@tempswa
3073       \read\@inputcheck to\bxjs@line
3074       \expandafter\bxjs@check\bxjs@line\@nil
3075     \fi
3076   \repeat
3077   \closein\@inputcheck
3078 \endgroup}
3079 \@gtempa
```

■hyperref 対策 unicode にしてはいけない。

```
3080 \bxjs@fix@hyperref@unicode{false}
```

tounicode special 命令を出力する。

```
3081 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
3082   \ifnum\jis"2121"= "A1A1 %euc
3083     \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3084   \else\ifnum\jis"2121"= "8140 %sjis
3085     \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3086   \else\ifnum\jis"2121"= "3000 %uptex
3087     \ifbxjs@bigcode
3088       \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3089       \PassOptionsToPackage{bigcode}{pxjahyper}
3090     \else
3091       \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3092     \fi
3093   \fi\fi\fi
3094   \let\bxToUnicodeSpecialDone=t
3095 \fi
```

■microtype 対策

```
3096 \@namedef{ver@microtype.sty}{2000/01/01}
3097 \newcommand*{\UseMicrotypeSet}[2][]{}
```

C.3 pdf_TE_X 用設定 : CJK + bxcjkatype

```
3098 \else\if p\jsEngine
```

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

```
3099 \bxjs@adjust@jafont{f}
3100 \edef\bxjs@nxt{%
3101   \noexpand\RequirePackage[%
3102     \ifx\bxjs@tmpa\@empty\else \bxjs@tmpa,\fi
3103     whole,autotilde]{bxcjkatype}[2013/10/15]}% v0.2c
3104 \bxjs@nxt
3105 \bxjs@cjk@loaded
```

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

※取りあえず固定はしない。

```
3106 \PassOptionsToPackage{unicode}{hyperref}
```

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

※ bxcjkatype を whole 付きで使っていることが前提。

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

```
3107 \ifx\bxckjatypeHyperrefPatchDone\@undefined
3108 \begingroup
3109 \CJK@input{UTF8.bdg}
3110 \endgroup
3111 \g@addto@macro\pdfstringdefPreHook{%
3112 \@nameuse{CJK@UTF8Binding}%
3113 }
3114 \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。

```
3115 \ifx\bxckjatypeHyperrefPatchDone\@undefined
3116 \g@addto@macro\pdfstringdefPreHook{%
3117 \ifx~\bxjs@@CJKtilde
3118 \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3119 \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3120 \let~\@empty
3121 \fi
3122 }
3123 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3124 \def\bxjs@@tildecmd{~}
3125 \def\bxjs@LetUnexpandableSpace#1{%
3126 \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
3127 \bxjs@org@LetUnexpandableSpace#1%
3128 \fi}
3129 \fi
```

■ 共通命令の実装

```
3130 \newskip\jsKanjiSkip
3131 \newskip\jsXKanjiSkip
3132 \ifx\CJKecglue\@undefined
3133 \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3134 \fi
3135 \let\autospacing\bxjs@enable@kanjiskip
3136 \let\noautospacing\bxjs@disable@kanjiskip
3137 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3138 \def\bxjs@apply@kanjiskip{%
3139 \jsKanjiSkip\@tempskipa
3140 \let\CJKglue\bxjs@CJKglue}
3141 \let\autoxspacing\bxjs@enable@xkanjiskip
3142 \let\noautoxspacing\bxjs@disable@xkanjiskip
3143 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3144 \def\bxjs@apply@xkanjiskip{%
3145 \jsXKanjiSkip\@tempskipa
3146 \let\CJKecglue\bxjs@CJKecglue}
```

\jachar のサブマクロの実装。

```
3147 \def\bxjs@jachar#1{%
```

```
3148 \CJKforced{#1}}
```

C.4 Xe_{La}TeX 用設定 : xeCJK + zxjatype

```
3149 \else\if x\jsEngine
```

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

```
3150 \RequirePackage{zxjatype}
3151 \PassOptionsToPackage{no-math}{fontspec}%!
3152 \PassOptionsToPackage{xetex}{graphicx}%!
3153 \PassOptionsToPackage{xetex}{graphics}%!
3154 \ifx\zxJaFamilyName\undefined
3155 \ClassError\bxjs@clsname
3156 {xeCJK or zxjatype is too old}\@ehc
3157 \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。

```
3158 \bxjs@adjust@jafont{f}
3159 \ifx\bxjs@tmpa\@empty
3160 \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3161 \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3162 \else
3163 \edef\bxjs@nxt{%
3164 \noexpand\RequirePackage[\bxjs@tmpa]{
3165 {zxjafont}[2013/01/28]}% v0.2a
3166 \bxjs@nxt
3167 \fi
```

■hyperref 対策 unicode オプションの指定に関する話。

X_{La}TeX 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_{La}TeX 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_{La}TeX 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。

```
3168 \ifnum\stricmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3169 \PassOptionsToPackage{unicode}{hyperref}
3170 \fi
```

■段落頭でのグルー挿入禁止

```
3171 \let\jsInhibitGlueAtParTop\inhibitglue
```

■ 共通命令の実装

```
3172 \newskip\jsKanjiSkip
3173 \newskip\jsXKanjiSkip
3174 \ifx\CJKecglue\@undefined
3175   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3176 \fi
3177 \let\autospacing\bxjs@enable@kanjiskip
3178 \let\noautospacing\bxjs@disable@kanjiskip
3179 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3180 \def\bxjs@apply@kanjiskip{%
3181   \jsKanjiSkip\@tempskipa
3182   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3183 \let\autoxspacing\bxjs@enable@xkanjiskip
3184 \let\noautoxspacing\bxjs@disable@xkanjiskip
3185 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3186 \def\bxjs@apply@xkanjiskip{%
3187   \jsXKanjiSkip\@tempskipa
3188   \xeCJKsetup{CJKecglue={\bxjs@CJKecglue}}}
```

`\mcfamily`、`\gtfamily` は本来は `zxjatype` の方で定義すべきであろうが、現状は暫定的にここで定義する。

```
3189 \ifx\mcfamily\@undefined
3190   \protected\def\mcfamily{\CJKfamily{\CJKrmdefault}}
3191   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
3192 \fi
```

`\jachar` のサブマクロの実装。

```
3193 \def\bxjs@jachar#1{%
3194   \xeCJKDeclareCharClass{CJK}{`#1}\relax
3195   #1}
```

C.5 LuaTeX 用設定 : LuaTeX-ja

```
3196 \else\if 1\jsEngine
```

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

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

```
3197 \let\zw\@undefined
3198 \RequirePackage{luatexja}
3199 \RequirePackage{luatexja-fontspec}
3200 \PassOptionsToPackage{pdftex}{graphicx}%!
3201 \PassOptionsToPackage{pdftex}{graphics}%!
```

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

```
3202 \ExplSyntaxOn
3203 \fp_gset:Nn \g_ljtj_fontspeg_scale_fp { \jsScale }
3204 \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)。

```
3205 \bxjs@adjust@jafont{t}
3206 \ifx\bxjs@tmpa\bxjs@@noEmbed
3207   \def\bxjs@tmpa{noembed}
3208 \fi
3209 \ifx\bxjs@tmpa@empty
3210   \defaultjfontfeatures{ Kerning=Off }
3211   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3212   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3213 \else
3214   \edef\bxjs@nxt{%
3215     \noexpand\RequirePackage[\bxjs@tmpa]
3216       {luatexja-preset}}%
3217   \bxjs@nxt
3218 \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。

```
3219 \DeclareRobustCommand\rmfamily
3220   {\not@math@alphabet\rmfamily\mathrm
3221     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3222 \DeclareRobustCommand\sffamily
3223   {\not@math@alphabet\sffamily\mathsf
3224     \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
3225 \DeclareRobustCommand\ttfamily
3226   {\not@math@alphabet\ttfamily\mathtt
3227     \romanfamily\ttddefault\kanjifamily\gtdefault\selectfont}
3228 \AtBeginDocument{%
3229   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3230   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}}%
3231 \bxjs@if@sf@default{%
3232   \renewcommand\kanjifamilydefault{\gtdefault}}
```

■和文パラメタの設定

```
3233 % 次の3つは既定値の通り
3234 %\ltjsetparameter{prebreakpenalty}={` ,10000}
3235 %\ltjsetparameter{postbreakpenalty}={` “,10000}
3236 %\ltjsetparameter{prebreakpenalty}={` ”,10000}
3237 \ltjsetparameter{jaxspmde}={` ! ,1}
```

```

3238 \ltjsetparameter{jaxspmode={`〒,2}}
3239 \ltjsetparameter{alxspmode={`+,3}}
3240 \ltjsetparameter{alxspmode={`\%,3}}

```

■段落頭でのグルー挿入禁止

```

3241 \protected\def\@inhibitglue{%
3242   \directlua{%
3243     luatexja.jfmglue.create_beginpar_node()}}
3244 \let\bxjs@ltj@inhibitglue\@inhibitglue
3245 \let\@inhibitglue\@undefined

```

■hyperref 対策 unicode にするべき。

```

3246 \bxjs@fix@hyperref@unicode{true}

```

■共通命令の実装

```

3247 \protected\def\autospacing{%
3248   \ltjsetparameter{autospacing=true}}
3249 \protected\def\noautospacing{%
3250   \ltjsetparameter{autospacing=false}}
3251 \protected\def\autoxspacing{%
3252   \ltjsetparameter{autoxspacing=true}}
3253 \protected\def\noautoxspacing{%
3254   \ltjsetparameter{autoxspacing=false}}
3255 \def\bxjs@apply@kanjiskip{%
3256   \ltjsetparameter{kanjiskip={\@tempskipa}}}}
3257 \def\bxjs@apply@xkanjiskip{%
3258   \ltjsetparameter{xkanjiskip={\@tempskipa}}}}

```

\jachar のサブマクロの実装。

```

3259 \def\bxjs@jachar#1{%
3260   \ltjjachar`#1\relax}

```

C.6 共通処理 (2)

```

3261 \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 3262 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3263 \DeclareJaTextFontCommand{\textgt}{\gtfamily}

```

■和文・和欧文間空白の初期値

```

3264 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3265 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3266 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3267 \fi

```

以上で終わり。

```
3268 %</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。

```
3269 %<*modern>
```

```
3270 \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。

```
3271 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
```

```
3272 \def\encodingdefault{T1}%
```

```
3273 \input{t1enc.def}%
```

```
3274 \fontencoding\encodingdefault\selectfont
```

```
3275 \fi
```

基本フォントを Latin Modern フォントファミリーに変更する。

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

```
3276 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
```

```
3277 \renewcommand{\rmdefault}{lmr}
```

```
3278 \renewcommand{\sfdefault}{lms}
```

```
3279 \renewcommand{\ttdefault}{lmtt}
```

```
3280 \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。

```
3281 \DeclareFontShape{OMX}{cmex}{m}{n}{%
```

```
3282   <-7.5>cmex7<7.5-8.5>cmex8%
```

```
3283   <8.5-9.5>cmex9<9.5->cmex10}{}%
```

```
3284 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
```

`amsmath` 読込時に上書きされるのを防ぐ。

```
3285 \def\cmex@opt{10}
```

D.2 fixltx2e 読込

※ `fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```
3286 \ifx\@IncludeInRelease\@undefined
```

```
3287 \RequirePackage{fixltx2e}
```

```
3288 \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。

```
3289 \RequirePackage{bxjscjkat}
```

D.4 完了

おしまい。

```
3290 %</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。

```
3291 %<*pandoc>
```

```
3292 \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。

```
3293 \@onlypreamble\bxjs@set@dupload@proc
```

```
3294 \def\bxjs@set@dupload@proc#1{%
```

```
3295   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
3296 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
3297 \def\bxjs@set@dupload@proc@a#1{%
```

```
3298   \@onlypreamble#1\def#1##1}
```

`\@if@options` `\@if@options` の再定義。

```
3299 \@onlypreamble\bxjs@org@if@options
```

```
3300 \let\bxjs@org@if@options\@if@options
```

```
3301 \newif\ifbxjs@dlp
```

```
3302 \def\@if@options#1#2#3{%
```

```
3303   \bxjs@dlpfalse
```

```
3304   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentt}%
```

```
3305   \ifx\bxjs@tmpa\bxjs@tmpb
```

```
3306     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
```

```
3307       \bxjs@dlptrue \fi
```

```
3308   \fi
```

```

3309 \ifbxjs@dlp \expandafter\bxjs@do@upload@proc
3310 \else \expandafter\bxjs@org@if@ptions
3311 \fi {#1}{#2}{#3}}
3312 \AtBeginDocument{%
3313 \let\@if@ptions\bxjs@org@if@ptions}
3314 \@onlypreamble\bxjs@do@upload@proc
3315 \def\bxjs@do@upload@proc#1#2#3{%
3316 \csname bxjs@dlp/#2.#1\endcsname{#3}%
3317 \@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。

```

3318 \def\bxjs@mark@as@loaded#1{%
3319 \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3320 \def\bxjs@mal@a#1#2{%
3321 \ifx#1\relax
3322 \def#1{2001/01/01}%
3323 \ClassInfo\bxjs@clsname
3324 {File '#2' marked as loaded\@gobble}%
3325 \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。

```

3326 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0

```

Polyglossia について。

```

3327 \bxjs@mark@as@loaded{polyglossia.sty}
3328 \bxjs@set@upload@proc{polyglossia.sty}{%
3329 \ClassWarning\bxjs@clsname
3330 {Loading of polyglossia is blocked}}
3331 \ifx\setmainlanguage\@undefined
3332 \newcommand*\setmainlanguage[2][]{%
3333 \newcommand*\setotherlanguage[2][]{%
3334 \ifcat_#2_\else
3335 \expandafter\let\csname #2\endcsname\@empty
3336 \expandafter\let\csname end#2\endcsname\@empty
3337 \expandafter\let\csname text#2\endcsname\@firstofone
3338 \fi}
3339 \newcommand*\setotherlanguages[2][]{%
3340 \@for\bxjs@tmpa:={#2}\do{%
3341 \setotherlangauge{\bxjs@tmpa}}}
3342 \fi
3343 \else

```

Babel について。

```

3344 \bxjs@mark@as@loaded{babel.sty}
3345 \bxjs@set@dupload@proc{babel.sty}{%
3346   \ClassWarning\bxjs@clsname
3347     {Loading of babel is blocked}}
3348 \let\foreignlanguage\@secondoftwo
3349 \let\otherlanguage\@gobble
3350 \let\endotherlanguage\@empty
3351 \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。

```

3352 \bxjs@set@dupload@proc{geometry.sty}{%
3353   \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。

```

3354 \if 1\jsEngine
3355   \bxjs@mark@as@loaded{xeCJK.sty}
3356   \providecommand*\setCJKmainfont{\setmainfont}
3357 \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。

```

3358 \ifx\@IncludeInRelease\@undefined\else
3359   \bxjs@mark@as@loaded{fixltx2e.sty}
3360 \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。)

```

3361 \if j\jsEngine
3362   \bxjs@mark@as@loaded{cmap.sty}
3363 \fi

```

E.7 microtype パッケージ

警告が多すぎなので消す。

```
3364 \PassOptionsToPackage{verbose=silent}{microtype}
```

E.8 完了

おしまい。

```
3365 %</pandoc>
```

和文ドライバ実装はここまで。

```
3366 %</drv>
```

付録 F 補助パッケージ一覧

BXJS クラスの機能を実現するために用意されたものだが、他のクラスの文書で読み込んで利用することもできる。

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

```
3367 %<*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

```
3368 %<*compat>
```

```
3369 \def\bxac@pkgname{bxjscompat}
```

\bxjx@engine エンジンの種別。

```
3370 \let\bxac@engine=n
```

```
3371 \def\bxac@do#1#2{%
```

```
3372   \edef\bxac@tmpa{\string#1}%
```

```
3373   \edef\bxac@tmpb{\meaning#1}%
```

```
3374   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
3375 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
3376 \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。

```
3377 \ifx\jsAtEndOfClass\@undefined
```

```
3378   \let\bxac@delayed@if@bxjs\@firstofone
```

```
3379 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
3380 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 3381 \newif\ifbxac@in@old@behavior
3382 \let\ImposeOldLuaTeXBehavior\relax
3383 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 XeTeX 部分

```
3384 \ifx x\bxac@engine
    XeTeX 文字クラスのムニャムニャ。
3385 \@onlypreamble\bxac@adjust@charclass
3386 \bxac@delayed@if@bxjs{%
3387   \@ifpackageloaded{xCJK}{-}{%else
3388     \ifx\Xe@alloc@intercharclass\undefined\else
3389       \ifnum\Xe@alloc@intercharclass=\z@
3390         \PackageInfo\bxac@pkgname
3391           {Setting up interchar class for CJK...\@gobble}%
3392         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
3393           \Xe@alloc@intercharclass=3
3394         }{%else
3395           \PackageWarning\bxac@pkgname
3396             {Cannot find file 'load-unicode-xetex-classes.tex'%
3397             \@gobble}%
3398         }%
3399       \fi\fi
3400     \ifnum\XeTeXcharclass"3041=\z@
3401       \PackageInfo\bxac@pkgname
3402         {Adjusting interchar class for CJK...\@gobble}%
3403       \@for\bxac@x:={%
3404         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3405         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3406         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3407         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3408         31FF%
3409       }\do{\XeTeXcharclass"\bxac@x=\@ne}%
3410     \fi
3411   }%
3412 }

    以上。
3413 \fi
```

G.3 LuaTeX 部分

```
3414 \ifx l\bxac@engine
    ムニャムニャ。
3415 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
3416   \chardef\pdfTeXversion=200
3417   \def\pdfTeXrevision{0}
```

```

3418 \let\pdfptxheader\luatexheader
3419 \fi\fi

```

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

```

\RevokeOldLuaTeXBehavior 3420 \begingroup\expandafter\expandafter\expandafter\endgroup
3421 \expandafter\ifx\csname outputmode\endcsname\relax\else
3422 \def\bxac@ob@list{%
3423   \do{\let}\pdfoutput{\outputmode}%
3424   \do{\let}\pdfpagewidth{\pagewidth}%
3425   \do{\let}\pdfpageheight{\pageheight}%
3426   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
3427   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}%
3428 \def\bxac@ob@do#1#2{\begingroup
3429   \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
3430 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3431   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3432   \else \let#3#1\relax \let#1\@undefined
3433   \fi}
3434 \protected\def\ImposeOldLuaTeXBehavior{%
3435   \unless\ifbxac@in@old@behavior
3436     \bxac@in@old@behaviortrue
3437     \let\do\bxac@ob@do \bxac@ob@list
3438   \fi}
3439 \protected\def\RevokeOldLuaTeXBehavior{%
3440   \ifbxac@in@old@behavior
3441     \bxac@in@old@behaviorfalse
3442     \let\do\bxac@ob@do \bxac@ob@list
3443   \fi}
3444 \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。

```

3445 \ifnum\luatexversion>64 \directlua{
3446   local function range(cs, ce, cc, ff)
3447     if ff or not tex.getcatcode(cs) == cc then
3448       local setcc = tex.setcatcode
3449       for c = cs, ce do setcc(c, cc) end
3450     end
3451   end
3452   range(0x3400, 0x4DB5, 11, false)
3453   range(0x4DB5, 0x4DBF, 11, true)
3454   range(0x4E00, 0x9FCC, 11, false)
3455   range(0x9FCD, 0x9FFF, 11, true)
3456   range(0xAC00, 0xD7A3, 11, false)
3457   range(0x20000, 0x2A6D6, 11, false)
3458   range(0x2A6D7, 0x2A6FF, 11, true)
3459   range(0x2A700, 0x2B734, 11, false)
3460   range(0x2B735, 0x2B73F, 11, true)
3461   range(0x2B740, 0x2B81D, 11, false)
3462   range(0x2B81E, 0x2B81F, 11, true)

```

```

3463     range(0x2B820, 0x2CEA1, 11, false)
3464     range(0x2CEA2, 0x2FFFD, 11, true)
3465 } \fi

```

以上。

```

3466 \fi

```

G.4 完了

おしまい。

```

3467 %</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

```

3468 %<*cjkcat>
3469 \def\bxjx@pkgnam{bxjscjkat}
3470 \newcount\bxjx@cna

\bxjx@engine エンジンの種別。

3471 \let\bxjx@engine=n
3472 \def\bxjx@do#1#2{%
3473   \edef\bxjx@tmpa{\string#1}%
3474   \edef\bxjx@tmpb{\meaning#1}%
3475   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3476 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3477 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3478 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3479 \bxjx@do\pdfTeXversion{\let\bxjx@engine=p}
3480 \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。

```

3481 \def\bxjx@do#1#2{%
3482   \if#1\bxjx@engine
3483     \@ifpackageloaded{#2}{\}%else
3484     \PackageError\bxjx@pkgnam
3485       {Package '#2' must be loaded}%
3486       {Package loading is aborted.\MessageBreak\@ehc}%
3487     \endinput}
3488 \fi}
3489 \bxjx@do{p}{bxcjkatype}
3490 \bxjx@do{x}{xeCJK}
3491 \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。

```
3492 \ifx\TextOrMath\@undefined
3493   \RequirePackage{fixltx2e}
3494 \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-j_a と（ほぼ）等価なものに変更する。

※ LuaT_EX-j_a との相違点：A830、A960、1B000。

```
3495 \if u\bxjx@engine
3496 \@for\bxjx@x:={%
3497 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3498 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3499 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3500 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3501 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3502 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
3503 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3504 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
3505 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3506 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
3507 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3508 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3509 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3510 10480,10500,10530,10600,10800,10840,10860,10880,%
3511 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
3512 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3513 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3514 11280,112B0,11300,11480,11580,11600,11680,11700,%
3515 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3516 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3517 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3518 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
3519 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3520 }\do{\kcatcode"\bxjx@x=15 }
3521 \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。

```
3522 \def\bxjx@grkcyr@list{%
3523 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
3524 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
```

3525	<code>\do{0393}{LGR}{\textGamma}{\Gamma}%</code>	% GR. C. L. GAMMA
3526	<code>\do{0394}{LGR}{\textDelta}{\Delta}%</code>	% GR. C. L. DELTA
3527	<code>\do{0395}{LGR}{\textEpsilon}{\epsilon}%</code>	% GR. C. L. EPSILON
3528	<code>\do{0396}{LGR}{\textZeta}{\zeta}%</code>	% GR. C. L. ZETA
3529	<code>\do{0397}{LGR}{\textEta}{\eta}%</code>	% GR. C. L. ETA
3530	<code>\do{0398}{LGR}{\textTheta}{\theta}%</code>	% GR. C. L. THETA
3531	<code>\do{0399}{LGR}{\textIota}{\iota}%</code>	% GR. C. L. IOTA
3532	<code>\do{039A}{LGR}{\textKappa}{\kappa}%</code>	% GR. C. L. KAPPA
3533	<code>\do{039B}{LGR}{\textLambda}{\lambda}%</code>	% GR. C. L. LAMDA
3534	<code>\do{039C}{LGR}{\textMu}{\mu}%</code>	% GR. C. L. MU
3535	<code>\do{039D}{LGR}{\textNu}{\nu}%</code>	% GR. C. L. NU
3536	<code>\do{039E}{LGR}{\textXi}{\xi}%</code>	% GR. C. L. XI
3537	<code>\do{039F}{LGR}{\textOmicron}{\omicron}%</code>	% GR. C. L. OMICRON
3538	<code>\do{03A0}{LGR}{\textPi}{\pi}%</code>	% GR. C. L. PI
3539	<code>\do{03A1}{LGR}{\textRho}{\rho}%</code>	% GR. C. L. RHO
3540	<code>\do{03A3}{LGR}{\textSigma}{\sigma}%</code>	% GR. C. L. SIGMA
3541	<code>\do{03A4}{LGR}{\textTau}{\tau}%</code>	% GR. C. L. TAU
3542	<code>\do{03A5}{LGR}{\textUpsilon}{\Upsilon}%</code>	% GR. C. L. UPSILON
3543	<code>\do{03A6}{LGR}{\textPhi}{\phi}%</code>	% GR. C. L. PHI
3544	<code>\do{03A7}{LGR}{\textChi}{\chi}%</code>	% GR. C. L. CHI
3545	<code>\do{03A8}{LGR}{\textPsi}{\psi}%</code>	% GR. C. L. PSI
3546	<code>\do{03A9}{LGR}{\textOmega}{\omega}%</code>	% GR. C. L. OMEGA
3547	<code>\do{03B1}{LGR}{\textalpha}{\alpha}%</code>	% GR. S. L. ALPHA
3548	<code>\do{03B2}{LGR}{\textbeta}{\beta}%</code>	% GR. S. L. BETA
3549	<code>\do{03B3}{LGR}{\textgamma}{\gamma}%</code>	% GR. S. L. GAMMA
3550	<code>\do{03B4}{LGR}{\textdelta}{\delta}%</code>	% GR. S. L. DELTA
3551	<code>\do{03B5}{LGR}{\textepsilon}{\epsilon}%</code>	% GR. S. L. EPSILON
3552	<code>\do{03B6}{LGR}{\textzeta}{\zeta}%</code>	% GR. S. L. ZETA
3553	<code>\do{03B7}{LGR}{\texteta}{\eta}%</code>	% GR. S. L. ETA
3554	<code>\do{03B8}{LGR}{\texttheta}{\theta}%</code>	% GR. S. L. THETA
3555	<code>\do{03B9}{LGR}{\textiota}{\iota}%</code>	% GR. S. L. IOTA
3556	<code>\do{03BA}{LGR}{\textkappa}{\kappa}%</code>	% GR. S. L. KAPPA
3557	<code>\do{03BB}{LGR}{\textlambda}{\lambda}%</code>	% GR. S. L. LAMDA
3558	<code>\do{03BC}{LGR}{\textmu}{\mu}%</code>	% GR. S. L. MU
3559	<code>\do{03BD}{LGR}{\textnu}{\nu}%</code>	% GR. S. L. NU
3560	<code>\do{03BE}{LGR}{\textxi}{\xi}%</code>	% GR. S. L. XI
3561	<code>\do{03BF}{LGR}{\textomicron}{\omicron}%</code>	% GR. S. L. OMICRON
3562	<code>\do{03C0}{LGR}{\textpi}{\pi}%</code>	% GR. S. L. PI
3563	<code>\do{03C1}{LGR}{\textrho}{\rho}%</code>	% GR. S. L. RHO
3564	<code>\do{03C2}{LGR}{\textvarsigma}{\varsigma}%</code>	% GR. S. L. FINAL SIGMA
3565	<code>\do{03C3}{LGR}{\textsigma}{\sigma}%</code>	% GR. S. L. SIGMA
3566	<code>\do{03C4}{LGR}{\texttau}{\tau}%</code>	% GR. S. L. TAU
3567	<code>\do{03C5}{LGR}{\textupsilon}{\upsilon}%</code>	% GR. S. L. UPSILON
3568	<code>\do{03C6}{LGR}{\textphi}{\phi}%</code>	% GR. S. L. PHI
3569	<code>\do{03C7}{LGR}{\textchi}{\chi}%</code>	% GR. S. L. CHI
3570	<code>\do{03C8}{LGR}{\textpsi}{\psi}%</code>	% GR. S. L. PSI
3571	<code>\do{03C9}{LGR}{\textomega}{\omega}%</code>	% GR. S. L. OMEGA
3572	<code>\do{0401}{T2A}{\CYR0}{\cyr0}%</code>	% CY. C. L. IO
3573	<code>\do{0410}{T2A}{\CYRA}{\cyra}%</code>	% CY. C. L. A

3574 \do{0411}{T2A}{\CYRB}{}%	% CY. C. L. BE
3575 \do{0412}{T2A}{\CYRV}{}%	% CY. C. L. VE
3576 \do{0413}{T2A}{\CYRG}{}%	% CY. C. L. GHE
3577 \do{0414}{T2A}{\CYRD}{}%	% CY. C. L. DE
3578 \do{0415}{T2A}{\CYRE}{}%	% CY. C. L. IE
3579 \do{0416}{T2A}{\CYRZH}{}%	% CY. C. L. ZHE
3580 \do{0417}{T2A}{\CYRZ}{}%	% CY. C. L. ZE
3581 \do{0418}{T2A}{\CYRI}{}%	% CY. C. L. I
3582 \do{0419}{T2A}{\CYRISHRT}{}%	% CY. C. L. SHORT I
3583 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
3584 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
3585 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
3586 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
3587 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
3588 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
3589 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
3590 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
3591 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
3592 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
3593 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
3594 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
3595 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
3596 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
3597 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
3598 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
3599 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
3600 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
3601 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
3602 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
3603 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
3604 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
3605 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
3606 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
3607 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
3608 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
3609 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
3610 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
3611 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
3612 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
3613 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
3614 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
3615 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
3616 \do{043B}{T2A}{\cyrl}{}%	% CY. S. L. EL
3617 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
3618 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
3619 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
3620 \do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
3621 \do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
3622 \do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES

```

3623 \do{0442}{T2A}{\cyrt}{}%           % CY. S. L. TE
3624 \do{0443}{T2A}{\cyru}{}%           % CY. S. L. U
3625 \do{0444}{T2A}{\cyrf}{}%           % CY. S. L. EF
3626 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
3627 \do{0446}{T2A}{\cyrc}{}%           % CY. S. L. TSE
3628 \do{0447}{T2A}{\cyrch}{}%          % CY. S. L. CHE
3629 \do{0448}{T2A}{\cyrrh}{}%          % CY. S. L. SHA
3630 \do{0449}{T2A}{\cyrrhshch}{}%       % CY. S. L. SHCHA
3631 \do{044A}{T2A}{\cyrrhrdsn}{}%       % CY. S. L. HARD SIGN
3632 \do{044B}{T2A}{\cyrrery}{}%         % CY. S. L. YERU
3633 \do{044C}{T2A}{\cyrsftsn}{}%        % CY. S. L. SOFT SIGN
3634 \do{044D}{T2A}{\cyrrerev}{}%        % CY. S. L. E
3635 \do{044E}{T2A}{\cyryu}{}%          % CY. S. L. YU
3636 \do{044F}{T2A}{\cyrya}{}%          % CY. S. L. YA
3637 \do{0451}{T2A}{\cyryo}{}%          % CY. S. L. IO
3638 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
3639 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
3640 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
3641 \do{00B1}{TS1}{\textpm}{\pm}%       % PLUS-MINUS SIGN
3642 \do{00B4}{TS1}{\textasciicute}{}%   % ACUTE ACCENT
3643 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
3644 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
3645 \do{00F7}{TS1}{\textdiv}{\div}%     % DIVISION SIGN
3646 }

3647 \providecommand*{\mathdegree}{\circ}

```

\ifbxjx@gcc@cjkl [スイッチ] ギリシャ・キリル文字を和文扱いにするか。

```
3648 \newif\ifbxjx@gcc@cjkl
```

\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。

```

3649 \newcommand*\greekasCJK{%
3650   \bxjx@gcc@cjkltrue}
3651 \newcommand*\nogreekasCJK{%
3652   \bxjx@gcc@cjklfalse}

```

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

```

3653 \def\bxjx@do#1\relax{%
3654   \def\bxjx@fake@grkl##1##2{%
3655     \expandafter\bxjx@fake@grkl@a\meaning##2#1\@nil{##1}{##2}}%
3656   \def\bxjx@fake@grkl@a##1#1##2\@nil##3##4{%
3657     \ifx\##1\%
3658       \bxjx@canta##4\divide\bxjx@canta\@cclvi
3659       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
3660       \mathchar\bxjx@canta
3661     \else ##3\fi}
3662 } \expandafter\bxjx@do\string\mathchar\relax

```

■pdfLaTeX・upLaTeX の場合

```
3663 \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。

```
3664 \@ifpackageloaded{inputenc}{\fi}{%else
3665   \RequirePackage[utf8]{inputenc}}
3666 \def\bxjx@tmpa{utf8}
3667 \ifx\bxjx@tmpa\inputencdoingname
3668   \PackageWarningNoLine\bxjx@pkgname
3669     {Input encoding changed to utf8}%
3670   \inputencoding{utf8}%
3671 \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。

```
3672 \if u\bxjx@engine
3673 \kcatcode"0370=15
3674 \kcatcode"0400=15
3675 \kcatcode"0500=15
3676 \fi
```

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

```
3677 \def\do#1{%
3678   \@tempcnta="#1\relax
3679   \@tempcntb=\@tempcnta \divide\@tempcntb256
3680   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
3681   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
3682 \def\bxjx@do@a#1#2#3#4#5{%
3683   \ifx\#5\%
3684     \def\bxjx@tmpa{\@inmathwarn#4}%
3685   \else\ifcat A\noexpand#5%
3686     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3687       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
3688   \else \def\bxjx@tmpa{#5}%
3689   \fi\fi
3690   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
3691   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
3692 \if u\bxjx@engine
3693 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
3694 \def\bxjx@do@b#1#2#3#4#5{%
3695   \kchardef#1=\@tempcnta
3696   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
3697   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
3698 \else\if p\bxjx@engine
3699 \def\bxjx@do@b#1#2#3#4#5{%
3700   \mathchardef#1=\@tempcnta
3701   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
3702   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
3703 \fi\fi
3704 \bxjx@grkcyr@list
```

```

3705 \let\bxjx@do@a\undefined
3706 \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。

```

3707 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
3708 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
3709 \@onlypreamble\bxjx@DeclareUnicodeCharacter
3710 \def\bxjx@DeclareUnicodeCharacter#1#2{%
3711   \count@=#1\relax \bxjx@cnta\count@ \divide\bxjx@cnta256
3712   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cnta\endcsname\relax
3713     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3714   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
3715     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3716   \else
3717     \wlog{ \space\space skipped defining Unicode char U+#1}%
3718   \fi\fi}

```

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

```

3719 \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。

```

```

3720 \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。
3721 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
3722 \else \UseTextSymbol{#2}{#3}%
3723 \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。

```

3724 \begingroup
3725 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
3726 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
3727   \noexpand\bxjx@swap@DUC@cmd
3728   \the\toks@
3729   \noexpand\bxjx@swap@DUC@cmd}}
3730 \endgroup\next
3731 \def\bxjx@swap@DUC@cmd{%
3732   \let\bxjx@tmpa\DeclareUnicodeCharacter
3733   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
3734   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■ Xe_{La}TeX・Lua_{La}TeX の場合

```

3735 \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。

```

3736 \def\do#1{%
3737   \bxjx@cmta="#1\relax
3738   \begingroup
3739     \lccode`~=\bxjx@cmta
3740   \lowercase{\endgroup
3741     \bxjx@do@a{~}}{#1}}
3742 \def\bxjx@do@a#1#2#3#4#5{%
3743   \ifx\#5\\\let\bxjx@tmpa\relax
3744   \else\ifcat A\noexpand#5%
3745     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3746       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
3747   \else \def\bxjx@tmpa{#5}%
3748   \fi\fi
3749   \ifx\bxjx@tmpa\relax\else
3750     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
3751   \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。

```

3752 \mathchardef\bxjx@tmpa="119
3753 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
3754 \let\bxjx@do@a\undefined

```

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

```

3755 \if l\bxjx@engine
3756   \protected\def\greekasCJK{%
3757     \bxjx@gcc@cjctrue
3758     \ltjsetparameter{jacharrange={+2, +8}}
3759   \protected\def\nogreekasCJK{%
3760     \bxjx@gcc@cjkfalse
3761     \ltjsetparameter{jacharrange={-2, -8}}
3762 \fi

```

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

```

3763 \if x\bxjx@engine
3764   \protected\def\greekasCJK{%
3765     \bxjx@gcc@cjctrue
3766     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
3767     \bxjx@grkcyr@list}
3768   \protected\def\nogreekasCJK{%
3769     \bxjx@gcc@cjkfalse
3770     \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
3771     \bxjx@grkcyr@list}
3772 \fi

```

以上。

```

3773 \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。

```
3774 \nogreekasCJK
```

H.5 完了

おしまい。

```
3775 %</cjkcat>
```

補助パッケージ実装はここまで。

```
3776 %</anc>
```