

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”)

v1.4 [2017/02/03]

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

目次

1	はじめに	3
2	オプション	7
3	和文フォントの変更	32
4	フォントサイズ	32
5	レイアウト	38
5.1	ページレイアウト	39
6	ページスタイル	46
7	文書のマークアップ	49
7.1	表題	49
7.2	章・節	54
7.3	リスト環境	66
7.4	パラメータの設定	73
7.5	フロート	74
7.6	キャプション	75
8	フォントコマンド	77
9	相互参照	79

9.1	目次の類	79
9.2	参考文献	84
9.3	索引	86
9.4	脚注	87
10	段落の頭へのグルー挿入禁止	90
11	いろいろなロゴ	93
12	amsmath との衝突の回避	94
13	初期設定	94
付録 A	和文ドライバの仕様 ㊦	98
付録 B	和文ドライバ : minimal ㊦	99
B.1	補助マクロ	99
B.2	(u)pTeX 用の設定	101
B.3	pdfTeX 用の処理	105
B.4	X _Y TeX 用の処理	106
B.5	後処理 (エンジン共通)	106
付録 C	和文ドライバ : standard ㊦	108
C.1	共通処理 (1)	109
C.2	pTeX 用設定	113
C.3	pdfTeX 用設定 : CJK + bxcjkatype	116
C.4	X _Y TeX 用設定 : xeCJK + zxjatype	118
C.5	LuaTeX 用設定 : LuaTeX-ja	120
C.6	共通処理 (2)	122
付録 D	和文ドライバ : modern ㊦	123
D.1	フォント設定	123
D.2	fixltx2e 読込	123
D.3	和文カテゴリコード	124
D.4	完了	124
付録 E	和文ドライバ : pandoc ㊦	124
E.1	dupload システム	124
E.2	lang 変数	125
E.3	geometry 変数	126
E.4	CJKmainfont 変数	126
E.5	fixltx2e パッケージ	126
E.6	cmap パッケージ	126

E.7	microtype パッケージ	127
E.8	完了	127
付録 F	補助パッケージ一覧 𐄂	127
付録 G	補助パッケージ : bxjscompat 𐄂	127
G.1	準備	127
G.2	X _Y TeX 部分	128
G.3	LuaTeX 部分	128
G.4	完了	130
付録 H	補助パッケージ : bxjscjkat 𐄂	130
H.1	準備	130
H.2	和文カテゴリコードの設定	131
H.3	ギリシャ・キリル文字の扱い	131
H.4	初期設定	138
H.5	完了	138

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 upL^AT_EX 対応パッチを取り込みました。

以下では実際のコードに即して説明します。

■ BXJS クラス特有の設定 𐄂

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

```

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

\jsDocClass [トークン] 文書クラスの種別。以下の定値トークンの何れかと同等：\jsArticle=bxjsarticle、\jsBook=bxjsbook、\jsReport=bxjsreport、\jsSlide=bxjsslide。
9 \let\jsArticle=a
10 \let\jsBook=b
11 \let\jsReport=r
12 \let\jsSlide=s
13 %<article>\let\jsDocClass\jsArticle
14 %<article>\def\bxjs@clsname{bxjsarticle}
15 %<book>\let\jsDocClass\jsBook
16 %<book>\def\bxjs@clsname{bxjsbook}
17 %<report>\let\jsDocClass\jsReport
18 %<report>\def\bxjs@clsname{bxjsreport}
19 %<slide>\let\jsDocClass\jsSlide
20 %<slide>\def\bxjs@clsname{bxjsslide}

\jsEngine [暗黙文字トークン] エンジン (TEX の種類) の種別：j = pTEX 系、x = XYTEX、p = pdfTEX (含 DVI モード)、l = LuaTEX、J = NTT jTEX、O = Omega 系、n = 以上の何れでもない。
21 \let\jsEngine=n
22 \def\bxjs@test@engine#1#2{%
23   \edef\bxjs@tmpa{\string#1}%
24   \edef\bxjs@tmpb{\meaning#1}%
25   \ifx\bxjs@tmpa\bxjs@tmpb #2\fi}
26 \bxjs@test@engine\kanjiskip{\let\jsEngine=j}
27 \bxjs@test@engine\jintercharskip{\let\jsEngine=J}
28 \bxjs@test@engine\Omegaversion{\let\jsEngine=O}
29 \bxjs@test@engine\XeTeXversion{\let\jsEngine=x}
30 \bxjs@test@engine\pdftexversion{\let\jsEngine=p}
31 \bxjs@test@engine\luatexversion{\let\jsEngine=l}

\ifjsWithupTeX [スイッチ] エンジンが (内部漢字コードが Unicode の) upTEX であるか。
32 \newif\ifjsWithupTeX
33 \ifx\ucs\undefined\else \ifnum\ucs"3000="3000
34   \jsWithupTeXtrue

```

```

35 \fi\fi
36 \let\if@jsc@uplatex\ifjsWithupTeX

\ifjsWithpTeXng 〔スイッチ〕 エンジンが pTeX-ng であるか。
37 \newif\ifjsWithpTeXng
38 \bxjs@test@engine\ngbanner{\jsWithpTeXngtrue}

\ifjsWitheTeX 〔スイッチ〕 エンジンが  $\epsilon$ -TeX 拡張をもつか。
39 \newif\ifjsWitheTeX
40 \bxjs@test@engine\eTeXversion{\jsWitheTeXtrue}

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

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

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

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

\bxjs@cond \bxjs@cond\ifXXX……\fi{〈真〉}{〈偽〉}

```

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

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

`\jsAtEndOfClass` このクラスの読込終了時に対するフック。(補助パッケージ中で用いられる。)

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

LuaTeX の場合、原版のコード中のコントロールワード中に現れる日本語文字のカテゴリコードを一時的に 11 に変更する。クラス読込終了時点で元に戻される。

※現在の LuaLaTeX では、漢字のカテゴリコードは最初から 11 になっているので、この処理は特段の意味を持たない。しかし、昔は 12 になっていて、この場合、日本語文字のコントロールワードの命令を使用するには、カテゴリコードを 11 に変更する必要がある。

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

`\jsInhibitGlue` は `\inhibitglue` が定義されていればそれを実行し、未定義ならば何もしない。

```
84 \bxjs@robust\def\jsInhibitGlue{%
85   \ifx\inhibitglue\undefined\else \inhibitglue \fi}
```

万が一「2.09 互換モード」になっていた場合は、これ以上進むと危険なので強制終了させる。

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

2 オプション

これらのクラスは `\documentclass{jsarticle}` あるいは `\documentclass[オプション]{jsarticle}` のように呼び出します。

まず、オプションに関連するいくつかのコマンドやスイッチ（論理変数）を定義します。

```
\if@restonecol 段組のときに真になる論理変数です。
96 \newif\if@restonecol

\if@titlepage これを真にすると表題，概要を独立したページに出力します。
97 \newif\if@titlepage

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

\if@mainmatter 真なら本文，偽なら前付け・後付けです。偽なら \chapter で章番号が出ません。
99 %<book|report>\newif\if@mainmatter \@mainmattertrue

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

JS クラスと異なり、初期値は偽とする。

```
100 \newif\if@enablejfam \@enablejfamfalse
```

以下で各オプションを宣言します。

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

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

デフォルトは `a4paper` です。

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

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

```
101 \def\bxjs@setpaper#1{\def\bxjs@param@paper{#1}}
102 \DeclareOption{a3paper}{\bxjs@setpaper{a3paper}}
103 \DeclareOption{a4paper}{\bxjs@setpaper{a4paper}}
104 \DeclareOption{a5paper}{\bxjs@setpaper{a5paper}}
```

```

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

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

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

```

■slide オプション slide を新設しました。

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

```

121 \newif\if@slide

```

BXJS ではスライド用のクラス bxjsslide を用意しているので、本来はこのスイッチは不要なはずである。しかし、JS クラスの一部のコードをそのまま使うために保持している。※この \if@slide という制御綴は、ユニークでないにも関わらず、衝突した場合に正常動作が保たれない、という問題を抱えている。

```

122 %<!slide>\@slidefalse
123 %<slide>\@slidetrue

```

■サイズオプション 10pt, 11pt, 12pt のほかに、8pt, 9pt, 14pt, 17pt, 21pt, 25pt, 30pt, 36pt, 43pt を追加しました。これは等比数列になるように選んだものです（従来の 20pt も残しました）。\@ptsize の定義が変だったのでご迷惑をおかけしましたが、標準的なドキュメントクラスと同様にポイント数から 10 を引いたものに直しました。

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

[2016-07-08] \mag を使わずに各種寸法をスケールさせるためのオプション nomag を新設しました。usemag オプションの指定で従来通りの動作となります。デフォルトは usemag です。

[2016-07-24] オプティカルサイズを調整するために NFSS ヘパッチを当てるオプション nomag* を新設しました。

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

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

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

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

`\bxjs@setbasefontlength` base, jbase で指定される長さ (式) のための特別な `\setlength`。与えられた式が“(実数)Q”の形の場合、Q 単位の長さを代入する (この場合“式”は使えない)。
※クラスオプションのトークン列の中に展開可能なトークンがある場合、 \LaTeX はファイルの読込の前にそれを展開しようとする。このため、この位置で `\jq` をサポートすることは原理的に不可能である。

```
134 \def\bxjs@setbasefontlength#1#2{%
```

ここで `true` の長さが使われるのは不合理なので、式が“`true`”を含む場合には警告を出す。

```
135   \bxjs@setbasefontlength@a#2true\@nnil
```

式の末尾が“Q”である時は特別に扱い、それ以外は `\setlength` に移譲する。

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

`\ifjsc@mag` は「`\mag` を使うか」を表すスイッチ。

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

```
152 \newif\ifjsc@mag
153 \newif\ifjsc@mag@xreal
```

```

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

```

JS クラス互換の magstyle 設定オプション。

```

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

```

■ トンボオプション

欧文 L^AT_EX のカーネルではサポートされないため削除。

ただし JS クラスとの互換のため以下の変数を用意する。

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

```

177 \newdimen\stockwidth \newdimen\stockheight

```

■ 面付け

欧文 L^AT_EX のカーネルではサポートされないため削除。

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

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

```

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

```

```
180 \DeclareOption{vartwoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 twocolumn で二段組になります。

```
181 \DeclareOption{onecolumn}{\@twocolumnfalse}
182 \DeclareOption{twocolumn}{\@twocolumntrue}
```

■表題ページ titlepage で表題・概要を独立したページに出力します。

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

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

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

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

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

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

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

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

```

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

```

■文献リスト 文献リストを open 形式（著者名や書名の後に改行が入る）で出力します。
これは使われることはないのでコメントアウトしてあります。

```

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

```

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

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

```

237 %\let\bxjs@enablejfam\undefined

fancyhdr オプションの処理。
238 \def\bxjs@kv@enablejfam@true{\let\bxjs@enablejfam=t}
239 \def\bxjs@kv@enablejfam@false{\let\bxjs@enablejfam=f}
240 \def\bxjs@kv@enablejfam@default{\let\bxjs@enablejfam\undefined}
241 \define@key{bxjs}{enablejfam}{%

```

```
242 \bxjs@set@keyval{enablejfam}{#1}{}}
```

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

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

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

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

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

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

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

■和文フォントメトリックの選択

ここは和文ドライバの管轄。

■papersize スペシャルの利用

`geometry` パッケージが行う。

`\ifbxjs@papersize` [スイッチ] `papersize` スペシャルを出力するか。既定で有効であるが、`nopapersize` オプションで無効にできる。

※ JS クラスでは `\ifpapersize` という制御綴だが、これは採用しない。

```
255 \newif\ifbxjs@papersize
256 \bxjs@papersizetrue
257 \DeclareOption{nopapersize}{\bxjs@papersizefalse}
```

```
258 \DeclareOption{papersize}{\bxjs@papersizetrue}
```

■英語化 オプション `english` を新設しました。

```
259 \newif\if@english
260 \@englishfalse
261 \DeclareOption{english}{\@englishttrue}
```


■jsreport 相当 オプション `report` を新設しました。

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

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

BXJS クラスでは、`nojslogo` を既定とする。

```
262 \newif\if@jslogo \@jslogofalse
263 \DeclareOption{jslogo}{\@jslogotrue}
264 \DeclareOption{nojslogo}{\@jslogofalse}
```

■BXJS 特有のオプションの一覧 

- ・ エンジンオプション： `xelatex` 等。
- ・ ドライバオプション： `dvipdfmx` 等。
- ・ 複合設定オプション： `pandoc` 等。
- ・ `nopapersize`： `papersize`（既定で有効）の否定。
- ・ `zw` / `nozw`： `\jsZw` と等価な命令として `\zw` を定義する / しない。
- ・ `js` / `nojs`： JS クラスを読込済として扱う / 扱わない。
- ・ `precisetext` / `noprecisetext`： X_YL^AT_EX の “generateactualtext” を有効 / 無効にする。
- ・ `simplejasetup` / `nosimplejasetup`： X_YL^AT_EX の “linebreaklocale” を有効 / 無効にする。
- ・ `bigcode` / `nobigcode`： up_TE_X で CMap として UTF8-UCS2 の代わりに UTF8-UTF16 を使う / 使わない。
- ・ `oldfontcommands` / `nooldfontcommands`： 古い “二文字フォント命令” に対する警告を抑止する / しない。
- ・ `base` = $\langle \textit{dimen} \rangle$ ： 基底フォントサイズを直接指定する。（`xxpt` オプションの代用なので、既定値は 10pt である。）

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

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

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

```

265 \mathchardef\bxjs@csta=259
266 \def\bxjs@invscale#1#2{%
267   \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
268     \@tempcnta\@tempdima \multiply\@tempcnta\@cclvi
269     \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@cclvi
270     \@tempcntb\p@ \divide\@tempcntb\@tempdimb
271     \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
272     \@tempdimb\@tempcnta\@ne
273     \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
274     \advance\@tempcnta\bxjs@csta \@tempdimc\@tempcnta\@ne
275     \@whiledim\@tempdimb<\@tempdimc\do{%
276       \@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
277       \advance\@tempcntb\@ne \divide\@tempcntb\tw@
278       \ifdim #2\@tempcntb>\@tempdima
279         \advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne

```

```

280     \else \@tempdimb=\@tempcntb\@ne \fi}%
281     \xdef\bxjs@gtmpa{\the\@tempdimb}%
282     \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。

```
283 \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。

```
284 \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。

```

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

```

289   \def\bxjs@driver@opt{dvipdfmx}%
290   \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。

```
291 %\let\bxjs@engine@given\@undefined
```

`\bxjs@engine@opt` 明示されたエンジンのオプション名。

```
292 %\let\bxjs@engine@opt\@undefined
```

エンジン明示指定のオプションの処理。

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

```

293 \DeclareOption{autodetect-engine}{%
294   \let\bxjs@engine@given=*}

```



```

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

```

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

```

320 %\let\bxjs@driver@given\@undefined
321 \let\bxjs@driver@@dvimode=0
322 \let\bxjs@driver@@dvipdfmx=1
323 \let\bxjs@driver@@pdfmode=2
324 \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。

```

325 %\let\bxjs@driver@opt\@undefined

326 \DeclareOption{dvips}{%
327   \def\bxjs@driver@opt{dvips}%
328   \let\bxjs@driver@given\bxjs@driver@@dvimode}
329 \DeclareOption{dviout}{%
330   \def\bxjs@driver@opt{dviout}%
331   \let\bxjs@driver@given\bxjs@driver@@dvimode}
332 \DeclareOption{xdvi}{%
333   \def\bxjs@driver@opt{xdvi}%
334   \let\bxjs@driver@given\bxjs@driver@@dvimode}
335 \DeclareOption{dvipdfmx}{%
336   \def\bxjs@driver@opt{dvipdfmx}%
337   \let\bxjs@driver@given\bxjs@driver@@dvipdfmx}
338 \DeclareOption{pdftex}{%

```

```

339 \def\bxjs@driver@opt{pdftex}%
340 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
341 \DeclareOption{luatex}{%
342 \def\bxjs@driver@opt{luatex}%
343 \let\bxjs@driver@given\bxjs@driver@@pdfmode}
344 \DeclareOption{xetex}{%
345 \def\bxjs@driver@opt{xetex}%
346 \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。

```

347 \DeclareOption{dvipdfmx-if-dvi}{%
348 \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。

```

349 \newif\ifbxjs@usezw \bxjs@usezwtrue

```

`zw`、`noz` オプションの定義。

```

350 \DeclareOption{noz}{%
351 \bxjs@usezwfalse}
352 \DeclareOption{zw}{%
353 \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。

```

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

```

`nojs`、`js` オプションの定義。

```

355 \DeclareOption{nojs}{%
356 \bxjs@disguise@jsfalse}
357 \DeclareOption{js}{%
358 \bxjs@disguise@jstrue}

```

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

```

359 \newif\ifbxjs@precisetext

```

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

```

360 \DeclareOption{noprecisetext}{%
361 \bxjs@precisetextfalse}
362 \DeclareOption{precisetext}{%
363 \bxjs@precisetexttrue}

```

`\ifbxjs@simplejasetup` \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）。

```

364 \newif\ifbxjs@simplejasetup \bxjs@simplejasetuptrue

```

nosimplejasetup / simplejasetup オプションの定義。

```
365 \DeclareOption{nosimplejasetup}{%
366   \bxjs@simplejasetupfalse}
367 \DeclareOption{simplejasetup}{%
368   \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。

```
369 \newif\ifbxjs@bigcode \bxjs@bigcodefalse
```

その上で、「 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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、「 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ のバージョンが 3.14159265 (2014 年 1 月) 以上であるか」を採用する。

```
370 \edef\bxjs@tmpa{\expandafter\noexpand\csname\endcsname}
371 \def\bxjs@tmpb#1 #2#3\@nil{%
372   \ifx1#2\bxjs@bigcodetrue \fi}
373 \expandafter\bxjs@tmpb\meaning\bxjs@tmpa1 0\@nil
```

nobigcode / bigcode オプションの定義。

```
374 \DeclareOption{nobigcode}{%
375   \bxjs@bigcodefalse}
376 \DeclareOption{bigcode}{%
377   \bxjs@bigcodetrue}
```

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

```
378 \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。

```
379 \DeclareOption{nooldfontcommands}{%
380   \bxjs@oldfontcommandsfalse}
381 \DeclareOption{oldfontcommands}{%
382   \bxjs@oldfontcommandstrue}
```

■keyval 型のオプション

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

```
387 \def\bxjs@safe@setkeys#1#2{%
388   \let\bxjs@KV@errx\KV@errx
```

```

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

\bxjs@set@keyval \bxjs@set@keyval{<key>}{<value>}{<error>}
    \bxjs@kv@<key>@<value> が定義済ならそれを実行し、未定義ならエラーを出す。
393 \def\bxjs@set@keyval#1#2#3{%
394 \expandafter\let\expandafter\bxjs@next\csname bxjs@kv@#1@#2\endcsname
395 \ifx\bxjs@next\relax
396 \bxjs@error@keyval{#1}{#2}%
397 #3%
398 \else \bxjs@next
399 \fi}
400 \@onlypreamble\bxjs@error@keyval
401 \def\bxjs@error@keyval#1#2{%
402 \ClassError\bxjs@clsname
403 {Invalid value '#2' for option #1}\@ehc}

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

\jsScale 〔実数値マクロ〕 和文スケール値。
405 \def\jsScale{0.924715}

    base オプションの処理。
406 \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。
407 \@onlypreamble\bxjs@do@opt@jbase
408 \let\bxjs@do@opt@jbase\relax
409 \define@key{bxjs}{jbase}{\bxjs@setjbasefontsize{#1}}
410 \def\bxjs@setjbasefontsize#1{%
411 \def\bxjs@do@opt@jbase{%
412 \bxjs@setbasefontlength\@tempdima{#1}%
413 \bxjs@invscale\@tempdima\jsScale
414 \bxjs@setbasefontsize{\@tempdima}}

    scale オプションの処理。
415 \define@key{bxjs}{scale}{%
416 \bxjs@scalesettrue
417 \edef\jsScale{#1}}

    noscale オプションの処理。
418 \DeclareOption{noscale}{%
419 \bxjs@scalesettrue
420 \def\jsScale{1}}

\bxjs@param@mag mag オプションの値。

```

```

421 \let\bxjs@param@mag\relax

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

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

\bxjs@jadriver 和文ドライバの名前。
424 \let\bxjs@jadriver\relax
425 %\let\bxjs@jadriver@given\undefined

ja オプションの処理。
※ jadriver は 0.9 版で用いられた旧称。
※単なる ja という指定は無視される (Pandoc 対策)。
426 \define@key{bxjs}{jadriver}{\edef\bxjs@jadriver{#1}}
427 \define@key{bxjs}{ja}[\relax]{%
428   \ifx\relax#1\else\edef\bxjs@jadriver{#1}\fi}

\jsJaFont 和文フォント設定の名前。
429 \let\jsJaFont\empty

jafont オプションの処理。
430 \define@key{bxjs}{jafont}{\edef\jsJaFont{#1}}

\jsJaParam 和文ドライバパラメタの文字列。
431 \let\jsJaParam\empty

japaram オプションの処理。
432 \define@key{bxjs}{japaram}{\edef\jsJaParam{#1}}

\bxjs@magstyle magstyle 設定値。(古いイマイチな名前。)
433 \let\bxjs@magstyle@mag=m
434 \let\bxjs@magstyle@real=r
435 \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。
436 \let\bxjs@magstyle@usemag\bxjs@magstyle@mag
437 \let\bxjs@magstyle@nomag\bxjs@magstyle@real
438 \expandafter\let\csname bxjs@magstyle@nomag*\endcsname\bxjs@magstyle@xreal

\bxjs@magstyle@default は既定の値を表す。
439 \let\bxjs@magstyle@default\bxjs@magstyle@usemag
440 \ifx l\jsEngine \ifnum\luatexversion>86
441   \let\bxjs@magstyle@default\bxjs@magstyle@xreal
442 \fi\fi
443 \ifjsWithpTeXng

```

```

444 \let\bxjs@magstyle@default\bxjs@magstyle@xreal
445 \fi
446 \let\bxjs@magstyle\bxjs@magstyle@default

magstyle オプションの処理。
447 \define@key{bxjs}{magstyle}{%
448 \expandafter\let\expandafter\bxjs@magstyle\csname
449 bxjs@magstyle@#1\endcsname
450 \ifx\bxjs@magstyle\relax
451 \ClassError\bxjs@clsname
452 {Invalid value '#1' for option magstyle}\@ehc
453 \let\bxjs@magstyle\bxjs@magstyle@default
454 \fi}

```

\bxjs@geometry geometry オプションの値。

```

455 \let\bxjs@geometry@class=c
456 \let\bxjs@geometry@user=u
457 \let\bxjs@geometry\bxjs@geometry@class

geometry オプションの処理。
458 \define@key{bxjs}{geometry}{%
459 \expandafter\let\expandafter\bxjs@geometry\csname
460 bxjs@geometry@#1\endcsname
461 \ifx\bxjs@geometry\relax
462 \ClassError\bxjs@clsname
463 {Invalid value '#1' for option geometry}\@ehc
464 \let\bxjs@geometry\bxjs@geometry@class
465 \fi}

```

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

```

466 \newif\ifbxjs@fancyhdr \bxjs@fancyhdrtrue

fancyhdr オプションの処理。
467 \let\bxjs@kv@fancyhdr@true\bxjs@fancyhdrtrue
468 \let\bxjs@kv@fancyhdr@false\bxjs@fancyhdrfalse
469 \define@key{bxjs}{fancyhdr}{%
470 \bxjs@set@keyval{fancyhdr}{#1}{}}

```

\ifbxjs@dvi@opt dvi オプションが指定されたか。

```

471 \newif\ifbxjs@dvi@opt

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

dvi オプションの処理。
476 \define@key{bxjs}{dvi}{%

```

```

477 \expandafter\let\expandafter\bxjs@tmpa\csname
478   bxjs@dvidriver@@#1\endcsname
479 \ifx\bxjs@tmpa\relax
480   \ClassError\bxjs@clsname
481     {Invalid value '#1' for option dvi}\@ehc
482 \else
483   \bxjs@driver@given を未定義にしていることに注意。
484   \def\bxjs@driver@opt{#1}%
485   \let\bxjs@driver@given\@undefined
486   \bxjs@dvi@opttrue
487 \fi}

```

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

※既定はアレだが 1.3 版で非アレになる予定。

```

487 \newif\ifbxjs@layout@buggyhmargin \bxjs@layout@buggyhmarginfalse

      layout オプションの処理。
488 \@namedef{bxjs@kv@layout@v1}{%
489   \bxjs@layout@buggyhmargintrue}
490 \@namedef{bxjs@kv@layout@v2}{%
491   \bxjs@layout@buggyhmarginfalse}
492 \define@key{bxjs}{layout}{%
493   \bxjs@set@keyval{layout}{#1}{}}

```

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

```

494 %\let\bxjs@textwidth@limit\@undefined
495 \define@key{bxjs}{textwidth-limit}{%
496   \edef\bxjs@textwidth@limit{#1}}

```

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

```

497 %\let\bxjs@paragraph@mark\@undefined
498 \define@key{bxjs}{paragraph-mark}{%
499   \edef\bxjs@paragraph@mark{#1}}

```

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

```

500 \newif\ifbxjs@whole@zw@lines \bxjs@whole@zw@linestrue
501 \let\bxjs@kv@wholezwlines@true\bxjs@whole@zw@linestrue
502 \let\bxjs@kv@wholezwlines@false\bxjs@whole@zw@linesfalse
503 \define@key{bxjs}{whole-zw-lines}{\bxjs@set@keyval{wholezwlines}{#1}{}}

```

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

```

504 \newif\ifbxjs@dummy@microtype \bxjs@dummy@microtypetrue
505 \let\bxjs@kv@dummymicrotype@true\bxjs@dummy@microtypetrue
506 \let\bxjs@kv@dummymicrotype@false\bxjs@dummy@microtypefalse
507 \define@key{bxjs}{dummy-microtype}{\bxjs@set@keyval{dummymicrotype}{#1}{}}

```

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

```

508 \newif\ifbxjs@hyperref@enc \bxjs@hyperref@enctrue

```

```

509 \let\bxjs@kv@hyperrefenc@true\bxjs@hyperref@enctrue
510 \let\bxjs@kv@hyperrefenc@false\bxjs@hyperref@encfalse
511 \define@key{bxjs}{hyperref-enc}{\bxjs@set@keyval{hyperrefenc}{#1}{}}

```

■ オプションの実行

L^AT_EX の実装では、クラスやパッケージのオプションのトークン列の中に { } が含まれると正常に処理ができない。これに対処する為 \@removeelement の実装に少し手を加える(仕様は変わらない)。

※クラスに \DeclareOption* がある場合は \@unusedoptions は常に空のままであることを利用している。

```

512 \let\bxjs@ltx@removeelement\@removeelement
513 \def\@removeelement#1#2#3{%
514   \def\reserved@a{#2}%
515   \ifx\reserved@a\@empty \let#3\@empty
516   \else \bxjs@ltx@removeelement{#1}{#2}{#3}%
517   \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。

```

518 %<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
519 %<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
520 %<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
521 %<slide>\ExecuteOptions{36pt,a4paper,landscape,oneside,onecolumn,titlepage,final}
522 \ProcessOptions\relax
523 \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。

```

524 \@onlypreamble\bxjs@purge@brace@elts
525 \def\bxjs@purge@brace@elts{%
526   \def\bxjs@tmpa{\gobble}%
527   \expandafter\bxjs@purge@be@a\@classoptionslist,\@nil,%
528   \let\@classoptionslist\bxjs@tmpa}
529 \@onlypreamble\bxjs@purge@be@a
530 \def\bxjs@purge@be@a#1,{%
531   \ifx\@nil#1\relax\else
532     \bxjs@purge@be@b#1{\@nil
533     \if@tempswa \edef\bxjs@tmpa{\bxjs@tmpa,#1}\fi
534     \expandafter\bxjs@purge@be@a
535   \fi}

```



```

536 \@onlypreamble\bxjs@purge@be@b
537 \def\bxjs@purge@be@b#1#{\bxjs@purge@be@c}
538 \@onlypreamble\bxjs@purge@be@c
539 \def\bxjs@purge@be@c#1\@nil{%
540   \ifx\@nil#1\@nil \@tempwattrue \else \@tempwafalse \fi}
541 \bxjs@purge@brace@elts

```

`papersize`、`10pt`、`noscale` の各オプションは他のパッケージと衝突を起こす可能性があるため、グローバルオプションから外す。

```

542 \@expandtwoargs\@removeelement
543 {papersize}\@classoptionslist\@classoptionslist
544 \@expandtwoargs\@removeelement
545 {10pt}\@classoptionslist\@classoptionslist
546 \@expandtwoargs\@removeelement
547 {noscale}\@classoptionslist\@classoptionslist

```

現在の（正規化前の）和文ドライバの値を `\bxjs@jadriver@given` に保存する。

```

548 \ifx\bxjs@jadriver\relax\else
549   \let\bxjs@jadriver@given\bxjs@jadriver
550 \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。

```

551 \let\bxjs@tmpb\jsEngine
552 \ifx j\bxjs@tmpb\ifjsWithpTeXng
553   \let\bxjs@tmpb=g
554 \fi\fi
555 \ifx j\bxjs@tmpb\ifjsWithupTeX
556   \let\bxjs@tmpb=u
557 \fi\fi
558 \ifx p\bxjs@tmpb\ifjsInPdfMode\else
559   \let\bxjs@tmpb=n
560 \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。）

```

561 \ifx *\bxjs@engine@given
562   \let\bxjs@engine@given\bxjs@tmpb

```

エンジン指定が `autodetect-engine` であり、かつ実際のエンジンが (u)pL^AT_EX だった場合は、本来のエンジンオプションをグローバルオプションに加える。

```

563 \ifx j\bxjs@engine@given
564   \g@addto@macro\@classoptionslist{,latex}
565 \else\ifx u\bxjs@engine@given
566   \g@addto@macro\@classoptionslist{,uplatex}
567 \fi\fi
568 \fi
569 \ifx\bxjs@engine@given\@undefined\else
570   \ifx\bxjs@engine@given\bxjs@tmpb\else
571     \ClassError\bxjs@clsname

```

```

572     {Option '\bxjs@engine@opt' used on wrong engine}\@ehc
573   \fi
574 \fi

```

エンジンが pT_EX-ng の場合、グローバルオプションに upl_atex を追加する。

```

575 \ifjsWithpTeXng
576   \g@addto@macro\@classoptionslist{,uplatex}
577 \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。

```

578 \@tempswatrue
579 \ifx \bxjs@driver@given\undefined\else
580   \ifjsInPdfMode
581     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode\else
582       \@tempswafalse
583     \fi
584   \else\ifx x\jsEngine
585     \ifx\bxjs@driver@given\bxjs@driver@@xetex\else
586       \@tempswafalse
587     \fi
588   \else
589     \ifx\bxjs@driver@given\bxjs@driver@@pdfmode
590       \@tempswafalse
591     \else\ifx\bxjs@driver@given\bxjs@driver@@xetex
592       \@tempswafalse
593     \fi\fi
594     \ifjsWithpTeXng\ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx\else
595       \@tempswafalse
596     \fi\fi
597   \fi\fi
598 \fi
599 \if@tempswa\else
600   \ClassError\bxjs@clsname
601   {Option '\bxjs@driver@opt' used on wrong engine}\@ehc
602 \fi

```

DVI 出力のエンジンである場合の追加処理。

```

603 \ifjsInPdfMode \@tempswafalse
604 \else\ifx x\jsEngine \@tempswafalse
605 \else\ifjsWithpTeXng \@tempswafalse
606 \else \@tempswarue
607 \fi\fi\fi
608 \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)。

```

609 \ifx\bxjs@driver@opt\undefined \ifx\bxjs@jadriver@given\undefined\else
610   \ClassWarningNoLine\bxjs@clsname
611   {No driver option is given}
612 \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。)

```

613 \ifbxjs@dvi@opt
614 \edef\bxjs@nxt{%
615 \let\noexpand\bxjs@driver@given
616 \csname bxjs@dvidriver@@\bxjs@driver@opt\endcsname
617 \noexpand\g@addto@macro\noexpand\@classoptionslist
618 {,\bxjs@driver@opt}%
619 }\bxjs@nxt
620 \fi
621 \fi

```

エンジンが pTeX-ng の場合、グローバルオプションに dvipdfmx を追加する。ただし、エンジンオプションが latex-ng* (*付) の場合、および既に dvipdfmx が指定されている場合を除く。

```

622 \ifjsWithpTeXng
623 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
624 \let\bxjs@platexng@nodrv\@undefined
625 \else\ifx t\bxjs@platexng@nodrv\else
626 \g@addto@macro\@classoptionslist{,dvipdfmx}
627 \fi\fi
628 \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。

```

629 \def\bxjs@@minimal{minimal}
630 \ifx\bxjs@jadriver\relax
631 \ifx j\jsEngine
632 \def\bxjs@jadriver{standard}
633 \else
634 \let\bxjs@jadriver\bxjs@@minimal
635 \fi
636 \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。

```

637 \ifx\bxjs@jadriver@given\@undefined\else
638 \ifx\bxjs@engine@given\@undefined
639 \ClassError\bxjs@clsname
640 {An engine option must be explicitly given}%
641 {When you use a Japanese-driver you must specify a correct\MessageBreak
642 engine option.\MessageBreak\@ehc}
643 \fi\fi

```

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

```

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

```

```

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

```

653 \bxjs@do@opt@jbase

```

\Cjascale 和文クラス共通仕様（※ただし ZR 氏提唱）における、和文スケール値の変数。

```

654 \let\Cjascale\jsScale

```

後処理

```

655 \if@slide
656 \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}{}}
657 \fi
658 \if@landscape
659 \setlength\@tempdima {\paperheight}
660 \setlength\paperheight{\paperwidth}
661 \setlength\paperwidth {\@tempdima}
662 \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。）

```

663 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
664 \@tempcnta="80 \loop \ifnum\@tempcnta<"100
665 \catcode\@tempcnta\active
666 \advance\@tempcnta\@ne
667 \repeat
668 \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。

```

669 \ifbxjs@disguise@js
670 %<book|report>\def\bxjs@js@clsname{jsbook}
671 %<!book&!report>\def\bxjs@js@clsname{jsarticle}
672 \@namedef{ver@\bxjs@js@clsname.cls}{2001/01/01 (bxjs)}
673 \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。

```

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

```

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

```
675 \ifbxjs@oldfontcommands
676   \AtEndOfClass{\allowoldfontcommands}
677 \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。

```
678 %<slide>\def\n@baseline{13}%
679 %<!slide>\ifdim\bxjs@param@basefontsize<10pt \def\n@baseline{15}%
680 %<!slide>\else \def\n@baseline{16}\fi
```

■拡大率の設定

\bxjs@magstyle の値に応じてスイッチ jsc@mag と jsc@mag@xreal を設定する。

```
681 \ifx\bxjs@magstyle\bxjs@magstyle@mag
682   \jsc@magtrue
683 \else\ifx\bxjs@magstyle\bxjs@magstyle@xreal
684   \jsc@mag@xrealtrue
685 \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。

```
686 \ifx\bxjs@param@mag\relax
687   \@tempdima=\bxjs@param@basefontsize
688   \advance\@tempdima.001pt \multiply\@tempdima25
689   \divide\@tempdima16384\relax \@tempcnta\@tempdima\relax
690   \edef\bxjs@param@mag{\the\@tempcnta}
691 \else
```

```

692 % mag 値が直接指定された場合
693 \let\c@bxjs@cmta\@tempcmta
694 \setcounter{bxjs@cmta}{\bxjs@param@mag}
695 \ifnum\@tempcmta<\z@ \@tempcmta=\z@ \fi
696 % 有効な mag 値の範囲は 1--32768
697 \edef\bxjs@param@mag{\the\@tempcmta}
698 \advance\@tempcmta100000
699 \def\bxjs@tmpa#1#2#3#4#5\@nil{\@tempdima=#2#3#4.#5\p@}
700 \expandafter\bxjs@tmpa\the\@tempcmta\@nil
701 \edef\bxjs@param@basefontsize{\the\@tempdima}
702 \fi
703 \@tempcmta\bxjs@param@mag \advance\@tempcmta100000
704 \def\bxjs@tmpa#1#2#3#4\@nil{\@tempdima=#2#3.#4\p@}
705 \expandafter\bxjs@tmpa\the\@tempcmta\@nil
706 \edef\jsc@magscale{\strip@pt\@tempdima}
707 \let\jsBaseFontSize\bxjs@param@basefontsize
708 %\typeout{\string\jsDocClass: \meaning\jsDocClass}
709 %\typeout{\string\jsEngine: \meaning\jsEngine}
710 %\typeout{\string\jsBaseFontSize: \jsBaseFontSize}
711 %\typeout{\string\bxjs@param@mag: \bxjs@param@mag}
712 %\typeout{\string\jsc@magscale: \jsc@magscale}
713 %\typeout{\string\ifjsc@mag: \meaning\ifjsc@mag}
714 %\typeout{\string\ifjsc@mag@xreal: \meaning\ifjsc@mag@xreal}

```

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

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

```

715 \ifjsc@mag
716 \let\jsc@mpt\p@
717 \newdimen\jsc@mmm \jsc@mmm=1mm
718 \catcode`\?=9 % \p@? read as \p@
719 \else
720 \newdimen\jsc@mpt \jsc@mpt=\jsc@magscale \p@
721 \newdimen\jsc@mmm \jsc@mmm=\jsc@magscale mm
722 \catcode`\?=11 \let\p@?\jsc@mpt
723 \fi
724 \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。

```

725 \newdimen\jsZw

```

```

726 \jsZw=10\jsc@empt \jsZw=\jsScale\jsZw
727 \ifbxjs@usezw
728 \providecommand*\zw{\jsZw}
729 \fi

```

\zwspace 全角幅の水平空き。

```

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

```

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

```

731 \ifjsc@mag@xreal
732 \RequirePackage{type1cm}
733 \let\jsc@invscale\bxjs@invscale

```

ムニャムニャムニャ……。

```

734 \expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
735 \expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
736 \let\jsc@get@external@font\get@external@font
737 \def\get@external@font{%
738   \jsc@preadjust@extract@font
739   \jsc@get@external@font}
740 \def\jsc@fstrunc#1{%
741   \edef\jsc@tmpa{\strip@pt#1}%
742   \expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
743 \def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
744   \if#5*\else
745     \edef\jsc@tmpa{#1%
746       \ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
747   \fi}
748 \def\jsc@preadjust@extract@font{%
749   \let\jsc@req@size\f@size
750   \dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
751   \advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
752   \let\jsc@ref@size\jsc@tmpa
753   \let\f@size\jsc@ref@size}
754 \def\execute@size@function#1{%
755   \let\jsc@cref@size\f@size
756   \let\f@size\jsc@req@size
757   \csname s@fct@#1\endcsname}
758 \let\jsc@DeclareErrorFont\DeclareErrorFont
759 \def\DeclareErrorFont#1#2#3#4#5{%
760   \@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
761   \edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
762   \expandafter\jsc@DeclareErrorFont\jsc@tmpa}
763 \def\gen@sfcnt{%
764   \edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
765   \empty@sfcnt}
766 \def\genb@sfcnt{%
767   \edef\mandatory@arg{%

```

```

768      \mandatory@arg\expandafter\genb@x\jsc@cref@size..\@{}%
769      \empty@sfcnt}
770      \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
771 \fi

```

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

```

\jsc@smallskip
\jsc@medskip 772 \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip 773 %\def\jsc@medskip{\vspace\jsc@medskipamount}
              774 %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount 775 \newskip\jsc@smallskipamount
\jsc@bigskipamount 776 \jsc@smallskipamount=3\jsc@mpt plus 1\jsc@mpt minus 1\jsc@mpt
                    777 %\newskip\jsc@medskipamount
                    778 %\jsc@medskipamount =6\jsc@mpt plus 2\jsc@mpt minus 2\jsc@mpt
                    779 %\newskip\jsc@bigskipamount
                    780 %\jsc@bigskipamoun =12\jsc@mpt plus 4\jsc@mpt minus 4\jsc@mpt

```

■pagesize スペシャルの出力

削除。

3 和文フォントの変更

和文フォントの設定は和文ドライバの管轄。

\@ 欧文といえば、 \LaTeX の `\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。

```

781 \chardef\bxjs@periodchar=~\ .
782 \bxjs@protected\def\bxjs@SE{\spacefactor\sfcodes\bxjs@periodchar}
783 \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}`

のように行います。これは

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

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

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

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

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

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

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

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

`\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。

```

784 \def\bxjs@tmpa{\def\set@fontsize##1##2##3}
785 \expandafter\bxjs@tmpa\expandafter{%
786   \set@fontsize{#1}{#2}{#3}%
787 % 末尾にコードを追加
788   \expandafter\def\expandafter\size@update\expandafter{%
789     \size@update
790     \jsFontSizeChanged}%
791 }
```

`\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。

```

792 \newcommand*\jsFontSizeChanged{%
793   \jsZw=\f@size\p@
794   \jsZw=\jsScale \jsZw
795   \ifdim\parindent>\z@
```

```

796     \if@english \parindent=1em
797     \else        \parindent=1\jsZw
798     \fi
799     \fi\relax
800     \jsResetDimen}

```

`\jsResetDimen` ユーザ定義用のフック。

```

801 \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。

```

802 \ifjsc@mag
803   \let\jsc@setfontsize\@setfontsize
804 \else
805   \def\jsc@setfontsize#1#2#3{%
806     \@setfontsize#1{#2\jsc@mpt}{#3\jsc@mpt}}
807 \fi

```

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

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

```

808 \emergencystretch 3\jsZw

```

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

`\narrowbaselines` [2003-06-30] 数式に入るところで `\narrowbaselines` を実行しているので `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対して、しば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 (`\@currsiz` の実体) が呼ばれた」ことになり警告が出る。JS クラスでは、`\@setfontsize` 中の `\@nomath` 実行を消して「そもそもサイズ命令で警告が出ない」ようにしている。警告が常に出ないのも望ましくないので、BXJS クラスの実装では、`\narrowbaselines` の時だけ警告が出ないようにする。

```

809 \newif\ifnarrowbaselines
810 \if@english
811   \narrowbaselinestrue
812 \fi

```

```

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

```

829 \def\bxjs@if@narrowbaselines{%
830   \ifnarrowbaselines\expandafter\@firstoftwo
831   \else \expandafter\@secondoftwo
832   \fi
833 }

```

`\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。

```

834 \renewcommand{\normalsize}{%
835   \bxjs@if@narrowbaselines{%
836     \jsc@setfontsize\normalsize\@xpt\@xiipt
837   }{%else
838     \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
839   }%

```

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 (`\belowdisplayskip`) 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。

```
840 \abovedisplayskip 11\p@? \@plus3\p@? \@minus4\p@?
841 \abovedisplayshortskip \z@ \@plus3\p@?
842 \belowdisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
843 \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。

```
844 \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。

```
845 \normalsize
```

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

`\Cwd`

`\Cvs` まず `\Cwd` 等の変数を定義する。

```
\Chs 846 \ifx\Cht\@undefined \newdimen\Cht \fi
847 \ifx\Cdp\@undefined \newdimen\Cdp \fi
848 \ifx\Cwd\@undefined \newdimen\Cwd \fi
849 \ifx\Cvs\@undefined \newdimen\Cvs \fi
850 \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。

```
851 \setlength\Cht{0.88\jsZw}
852 \setlength\Cdp{0.12\jsZw}
853 \setlength\Cwd{1\jsZw}
854 \setlength\Cvs{\baselineskip}
855 \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。

```
856 \newcommand{\small}{%
857   \bxjs@if@narrowbaselines{%
858     %<!kiyou> \jsc@setfontsize\small\@ixpt{11}%
859     %<kiyou> \jsc@setfontsize\small{8.8888}{11}%
```

```

860 }{%else
861 %<!kiyou> \jsc@setfontsize\small\@ixpt{13}%
862 %<kiyou> \jsc@setfontsize\small{8.8888}{13.2418}%
863 }%
864 \abovedisplayskip 9\p@? \@plus3\p@? \@minus4\p@?
865 \abovedisplayshortskip \z@ \@plus3\p@?
866 \belowdisplayskip \abovedisplayskip
867 \belowdisplayshortskip \belowdisplayskip
868 \def\@listi{\leftmargin\leftmargini
869 \topsep \z@
870 \parsep \z@
871 \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。

```

872 \newcommand{\footnotesize}{%
873 \bxjs@if@narrowbaselines{%
874 %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{9.5}%
875 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{11}%
876 }{%else
877 %<!kiyou> \jsc@setfontsize\footnotesize\@viipt{11}%
878 %<kiyou> \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
879 }%
880 \abovedisplayskip 6\p@? \@plus2\p@? \@minus3\p@?
881 \abovedisplayshortskip \z@ \@plus2\p@?
882 \belowdisplayskip \abovedisplayskip
883 \belowdisplayshortskip \belowdisplayskip
884 \def\@listi{\leftmargin\leftmargini
885 \topsep \z@
886 \parsep \z@
887 \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 888 \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
889 \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\LARGE 890 \if@twocolumn
891 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{\n@baseline}}
892 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
893 \else
894 %<!kiyou> \newcommand{\large}{\jsc@setfontsize\large\@xipt{17}}
895 %<kiyou> \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
896 \fi
897 %<!kiyou>\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
898 %<kiyou>\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}

```

```

899 \newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xvipt{25}}
900 \newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
901 \newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
902 \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。

```

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

```

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

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

```

904 % \newcommand{\headfont}{\bfseries}
905 \newcommand{\headfont}{\sffamily}
906 % \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。
907 %<kiyou>\setlength\columnsep{2\Cwd}
908 %<kiyou>\setlength\columnsep{28truebp}
909 \setlength\columnseprule{0\p@}

```

■段落

```

\lineskip 上下の行の文字が \lineskiplimit より接近したら、\lineskip より近づかないようにし
\normallineskip ます。元は 0pt ですが 1pt に変更しました。normal... の付いた方は保存用です。
\lineskiplimit 910 \setlength\lineskip{1\p@?}
\normallineskiplimit 911 \setlength\normallineskip{1\p@?}
912 \setlength\lineskiplimit{1\p@?}
913 \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。

```
914 \renewcommand{\baselinestretch}{}
```

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

```
915 \setlength\parskip{0\p@}  
916 \if@slide  
917   \setlength\parindent{0\p@}  
918 \else  
919   \setlength\parindent{1\Cwd}  
920 \fi
```

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

```
\@highpenalty 921 \@lowpenalty    51  
922 \@medpenalty 151  
923 \@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。

```
924 % \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。

```
925 % \brokenpenalty 100
```

5.1 ページレイアウト

BXJS ではページレイアウトの処理は `geometry` パッケージが担当している。

■ 準備

現状ではここで `\mag` を設定している。

`\topskip` も指定する。

```
926 \ifjsc@mag  
927 \mag=\bxjs@param@mag  
928 \fi  
929 \setlength{\topskip}{10\p@?}
```

`\bxjs@param@paper` が長さ指定 (`{W}{H}`) の場合、`geometry` の形式 (`papersize={W,H}`) に変換する。

```
930 \def\bxjs@read@a{\futurelet\bxjs@tmpa\bxjs@read@b}  
931 \def\bxjs@read@b{%  
932   \ifx\bxjs@tmpa\bgroup \expandafter\bxjs@read@c
```

```

933 \else \expandafter\bxjs@read@d \fi}
934 \def\bxjs@read@c#1#2#3\@nil{\def\bxjs@param@paper{papersize={#1,#2}}}
935 \def\bxjs@read@d#1\@nil{}
936 \expandafter\bxjs@read@a\bxjs@param@paper\@nil

```

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

```

937 \edef\bxjs@layout@paper{%
938 \ifjsc@mag truedimen,\fi
939 \if@landscape landscape,\fi
940 \bxjs@param@paper}

```

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

```

941 %<*article|report>
942 \def\bxjs@layout{%
943 headheight=\topskip,footskip=0.03367\paperheight,%
944 headsep=\footskip-\topskip,includeheadfoot,%
945 hscale=0.76,hmarginratio=1:1,%
946 vscale=0.83,vmarginratio=1:1,%
947 }
948 %</article|report>
949 %<*book>
950 \ifbxjs@layout@buggyhmargin %---
951 % アレ
952 \def\bxjs@layout{%
953 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
954 hmargin=36\jsc@mmm,hmarginratio=1:1,%
955 vscale=0.83,vmarginratio=1:1,%
956 }
957 \else %---
958 % 非アレ
959 \def\bxjs@layout{%
960 headheight=\topskip,headsep=6\jsc@mmm,nofoot,includeheadfoot,%
961 hmargin=18\jsc@mmm,%
962 vscale=0.83,vmarginratio=1:1,%
963 }
964 \fi %---
965 %</book>
966 %<*slide>
967 \def\bxjs@layout{%
968 noheadfoot,%
969 hscale=0.9,hmarginratio=1:1,%
970 vscale=0.944,vmarginratio=1:1,%
971 }
972 %</slide>

```

`\fullwidth` [寸法レジスタ] ヘッダ・フッタ領域の横幅。

```

973 \newdimen\fullwidth

```

`\jsTextWidthLimit` [実数値マクロ] `bxjsbook` における、`\textwidth` の上限の全角単位での値。


```

974 %<*book>
975 \newcommand\jsTextWidthLimit{40}
976 \ifx\bxjs@textwidth@limit\@undefined\else
977   \let\c@bxjs@cmta\@tempcmta
978   \setcounter{bxjs@cmta}{\bxjs@textwidth@limit}
979   \long\edef\jsTextWidthLimit{\the\@tempcmta}
980 \fi
981 %</book>

```

\bxjs@postproc@layout geometry の後処理。

```

982 \def\bxjs@postproc@layout{%
983   % ドライバ再設定
984   \ifx\bxjs@geometry@driver\relax\else
985     \let\Gm@driver\bxjs@geometry@driver
986   \fi
987   % textwidth 調整
988   \@tempdima=\textwidth
989   \if@twocolumn \@tempdima=2\Cwd \else \@tempdima=1\Cwd \fi
990   \divide\textwidth\@tempdima \multiply\textwidth\@tempdima
991   \advance\@tempdima-\textwidth
992   \advance\oddsidemargin 0.5\@tempdima
993   \advance\evensidemargin 0.5\@tempdima
994   \fullwidth=\textwidth
995 %<*book>
996   \ifdim\textwidth>\jsTextWidthLimit\Cwd
997     \textwidth=\jsTextWidthLimit\Cwd
998     \addtolength\evensidemargin{\fullwidth-\textwidth}
999   \fi
1000 %</book>
1001 % textheight 調整
1002   \@tempdima=\textheight
1003   \advance\textheight-\topskip
1004   \divide\textheight\baselineskip \multiply\textheight\baselineskip
1005   \advance\textheight\topskip
1006   \advance\@tempdima-\textheight
1007   \advance\topmargin0.5\@tempdima
1008 % headheight 調整
1009   \@tempdima=\topskip
1010   \advance\headheight\@tempdima
1011   \advance\topmargin-\@tempdima
1012 % marginpar 設定
1013   \setlength\marginparsep{\columnsep}
1014   \setlength\marginparpush{\baselineskip}
1015   \setlength\marginparwidth{\paperwidth-\oddsidemargin-1truein%
1016     -\textwidth-10\jsc@mm-\marginparsep}
1017   \divide\marginparwidth\Cwd \multiply\marginparwidth\Cwd
1018 % 連動する変数
1019   \maxdepth=.5\topskip
1020   \stockwidth=\paperwidth

```

```

1021 \stockheight=\paperheight
1022 }

```

`\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。

```

1023 \edef\jsGeometryOptions{%
1024 \bxjs@layout@paper,\bxjs@layout}

```

■`geometry` パッケージ読込

`geomentry` オプションの値に応じて分岐する。

まずは `geometry=class` の場合。

```

1025 \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。

```

1026 \@onlypreamble\bxjs@revert
1027 \let\bxjs@revert\@empty
1028 \edef\bxjs@tmpa{\the\ht\strutbox}
1029 \ht\strutbox=10\p@?
1030 \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。

```

1031 \ifbxjs@papersize
1032 \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx
1033 \PassOptionsToPackage{dvipdfm}{geometry}
1034 \fi
1035 \let\bxPapersizeSpecialDone=t
1036 \else
1037 \PassOptionsToPackage{driver=none}{geometry}
1038 \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。

```

1039 \edef\bxjs@nxt{%
1040 \noexpand\RequirePackage[\bxjs@layout@paper,\bxjs@layout]{geometry}}
1041 \AtBeginDocument{\bxjs@pre@geometry@hook}
1042 \AtBeginDocument{\ImposeOldLuaTeXBehavior}
1043 \bxjs@nxt \bxjs@revert
1044 \AtBeginDocument{\RevokeOldLuaTeXBehavior}

```

`\bxjs@geometry@driver` `geometry` が用いるドライバの名前。

※この値は一度決めた後は変わってほしくないので、`\bxjs@postproc@layout` において書き戻す処理を入れている。

```
1045 \let\bxjs@geometry@driver\Gm@driver
1046 \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。

```
1047 \@onlypreamble\bxjs@pre@geometry@hook
1048 \def\bxjs@pre@geometry@hook{%
1049   \@ifpackageloaded{geometry}{%
1050     \@ifpackagelater{geometry}{2010/02/12}{\}%else
1051     \PackageWarningNoLine\bxjs@clsname
1052     {The 'geometry' package installed\MessageBreak
1053      is too old (< v5.0)}%
1054     \if x\jsEngine \ifnum\mag=\@m\else
1055       \def\bxjs@Gm@driver{pdftex}
1056       \ifx\pdfhorigin\@undefined \newdimen\pdfhorigin \fi
1057       \ifx\pdfvorigin\@undefined \newdimen\pdfvorigin \fi
1058     \fi\fi
1059   }%
1060   \ifjsWithpTeXng
1061     \ifx\Gm@driver\@empty
1062       \def\Gm@driver{pdftex}%
1063     \fi
1064   \fi
1065 }{}}
```

`\setpagelayout` ページレイアウト設定のためのユーザ命令。

- `\setpagelayout{<text>}` : 現在の `geometry` の設定の一部を修正する。
- `\setpagelayout*{<text>}` : 用紙以外の設定をリセットして、改めて設定を行う。

どちらも設定の後で後処理 `\bxjs@postproc@layout` を実行する。

```
1066 \def\setpagelayout{\@ifstar
1067   {\bxjs@reset@layout}{\bxjs@modify@layout}}
1068 \def\bxjs@modify@layout#1{%
1069   \edef\bxjs@nxt{\noexpand\geometry{#1,truedimen}}%
1070   \bxjs@nxt\bxjs@postproc@layout}
1071 \def\bxjs@reset@layout#1{%
1072   \edef\bxjs@nxt{\noexpand\geometry{reset,\bxjs@layout@paper,#1,truedimen}}%
1073   \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。

```
1074 \else\ifx\bxjs@geometry\bxjs@geometry@user
```

この場合はユーザが何らかの方法（例えば `geometry` を読み込む）でページレイアウトを設定する必要がある。もし、本体開始時に `\textwidth` がカーネル設定の値（`.5\maxdimen`）のままになっている場合はエラーを出す。

※\jsUseMinimalPageLayout は動作テスト用。

```
1075 \AtBeginDocument{\bxjs@check@page@layout}
1076 \@onlypreamble\bxjs@check@page@layout
1077 \def\bxjs@check@page@layout{%
1078   \ifdim\textwidth=.5\maxdimen
1079     \ClassError\bxjs@clsname
1080       {Page layout is not properly set}%
1081       {\@ehd}
1082   \fi}
1083 \def\jsUseMinimalPageLayout{%
1084   \setlength{\textwidth}{6.5in}%
1085   \setlength{\textheight}{8in}}

\setpagelayout はとりあえず無効にしておく。

1086 \let\bxjs@geometry@driver\relax
1087 \def\setpagelayout{\@ifstar
1088   {\bxjs@pagelayout@a}{\bxjs@pagelayout@a}}
1089 \def\bxjs@pagelayout@a#1{%
1090   \ClassError\bxjs@clsname
1091     {Command '\string\setpagelayout' is not supported,\MessageBreak
1092     because 'geometry' value is not 'class'}\@eha}

geometry=user の場合の処理はここで終わり。
1093 \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。

```
1094 \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。

1095 `\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 \LaTeX 2_ϵ 本体で定義されていますが，ここで設定変更します。本文ページ（本文とフロートが共存するページ）とフロートだけのページで設定が異なります。ちなみに，カウンタは内部では `\c@` を名前に冠したマクロになっています。

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

[2003-08-23] ちょっと増やしました。

1096 `\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。

1097 `\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。

1098 `\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。

1099 `\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。

1100 `\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。

1101 `\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。

1102 `\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。

1103 `\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。

1104 `\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。

1105 `\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。

```
1106 \setlength\floatsep      {12\p@? \@plus 2\p@? \@minus 2\p@?}
1107 \setlength\textfloatsep{20\p@? \@plus 2\p@? \@minus 4\p@?}
1108 \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 1109 \setlength\dblfloatsep      {12\p@? \@plus 2\p@? \@minus 2\p@?}
1110 \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 1111 \setlength\@fptop{0\p@? \@plus 1fil}
1112 \setlength\@fpsep{8\p@? \@plus 2fil}
1113 \setlength\@fpbot{0\p@? \@plus 1fil}
```

`\@dblfpsep` 段抜きフロートについての値です。

```
\@dblfpsep 1114 \setlength\@dblfpsep{0\p@? \@plus 1fil}
1115 \setlength\@dblfpsep{8\p@? \@plus 2fil}
\@dblfpbot 1116 \setlength\@dblfpbot{0\p@? \@plus 1fil}
```

6 ページスタイル

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

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

ページスタイルは `\ps@...` の形のマクロで定義されています。

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

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

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

柱を設定する命令は, 右の柱が左の柱の下位にある場合は十分まともに動作します。たとえば左マークを `\chapter`, 右マークを `\section` で変更する場合はこれにあたります。し

かし、同一ページに複数の `\markboth` があると、おかしい結果になることがあります。

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

`\ps@empty` `empty` ページスタイルの定義です。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。

```
1117 % \def\ps@empty{%
1118 %   \let\mkboth\gobbletwo
1119 %   \let\@oddhead\@empty
1120 %   \let\@oddfoot\@empty
1121 %   \let\@evenhead\@empty
1122 %   \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。

```
1123 \def\ps@plainfoot{%
1124   \let\mkboth\gobbletwo
1125   \let\@oddhead\@empty
1126   \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
1127   \let\@evenhead\@empty
1128   \let\@evenfoot\@oddfoot}
1129 \def\ps@plainhead{%
1130   \let\mkboth\gobbletwo
1131   \let\@oddfoot\@empty
1132   \let\@evenfoot\@empty
1133   \def\@evenhead{%
1134     \ifmparswitch \hss \fi
1135     \hbox to \fullwidth{\textbf{\thepage}\hfil}%
1136     \ifmparswitch\else \hss \fi}%
1137   \def\@oddhead{%
1138     \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
1139 %<book>\let\ps@plain\ps@plainhead
1140 %<!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。

```
1141 %<*article|slide>
1142 \if@twoside
1143   \def\ps@headings{%
1144     \let\@oddfoot\@empty
1145     \let\@evenfoot\@empty
1146     \def\@evenhead{\ifmparswitch \hss \fi
1147       \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
1148     \ifmparswitch\else \hss \fi}%
1149   \def\@oddhead{%
```

```

1150     \underline{%
1151     \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1152     \let\@mkboth\markboth
1153     \def\sectionmark##1{\markboth{%
1154     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1155     ##1}{}}%
1156     \def\subsectionmark##1{\markright{%
1157     \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1\jsZw\fi
1158     ##1}}%
1159 }
1160 \else % if not twoside
1161     \def\ps@headings{%
1162     \let\@oddfoot\@empty
1163     \def\@oddhead{%
1164     \underline{%
1165     \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
1166     \let\@mkboth\markboth
1167     \def\sectionmark##1{\markright{%
1168     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi
1169     ##1}}%
1170 \fi
1171 %</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）。

```

1172 %<*book|report>
1173 \newif\if@omit@number
1174 \def\ps@headings{%
1175     \let\@oddfoot\@empty
1176     \let\@evenfoot\@empty
1177     \def\@evenhead{%
1178     \if@mparswitch \hss \fi
1179     \underline{\hbox to \fullwidth{\autoxspacing
1180     \textbf{\thepage}\hfil\leftmark}}}%
1181     \if@mparswitch\else \hss \fi}%
1182     \def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
1183     {\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
1184     \let\@mkboth\markboth
1185     \def\chaptermark##1{\markboth{%
1186     \ifnum \c@secnumdepth >\m@ne
1187     \if@mainmatter
1188     \if@omit@number\else
1189     \@chapapp\thechapter\@chappos\hskip1\jsZw
1190     \fi
1191     \fi
1192     \fi
1193     ##1}{}}%
1194     \def\sectionmark##1{\markright{%
1195     \ifnum \c@secnumdepth >\z@ \thesection \hskip1\jsZw\fi

```



```

1196     ##1}}}%
1197 %</book|report>

    最後は学会誌の場合です。

1198 %<*jspf>
1199 \def\ps@headings{%
1200   \def\@oddfoot{\normalfont\hfil\thepage\hfil}
1201   \def\@evenfoot{\normalfont\hfil\thepage\hfil}
1202   \def\@oddhead{\normalfont\hfil \@title \hfil}
1203   \def\@evenhead{\normalfont\hfil プラズマ・核融合学会誌\hfil}
1204 %</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。

```

1205 \def\ps@myheadings{%
1206   \let\@oddfoot\@empty\let\@evenfoot\@empty
1207   \def\@evenhead{%
1208     \if@mparswitch \hss \fi%
1209     \hbox to \fullwidth{\thepage\hfil\leftmark}%
1210     \if@mparswitch\else \hss \fi}%
1211   \def\@oddhead{%
1212     \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
1213   \let\@mkboth\@gobbletwo
1214 %<book|report> \let\chaptermark\@gobble
1215 \let\sectionmark\@gobble
1216 %<!book&!report> \let\subsectionmark\@gobble
1217 }

```

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
\date
1218 % \newcommand*{\title}[1]{\gdef\@title{#1}}
1219 % \newcommand*{\author}[1]{\gdef\@author{#1}}
1220 % \newcommand*{\date}[1]{\gdef\@date{#1}}
1221 % \date{\today}

```

`\subtitle` 副題を設定する。

`\jsSubtitle` ※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。

```

1222 \newcommand*{\jsSubtitle}[1]{\gdef\bxjs@subtitle{#1}}
1223 %\let\bxjs@subtitle\@undefined

    \title にフックを入れる。

1224 \renewcommand*{\title}[1]{\bxjs@decl@subtitle\gdef\@title{#1}}
1225 \AtBeginDocument{\bxjs@decl@subtitle}
1226 \def\bxjs@decl@subtitle{%
1227     \global\let\bxjs@decl@subtitle\relax
1228     \ifx\subtitle\@undefined
1229         \global\let\subtitle\jsSubtitle
1230     \fi}

```

`\bxjs@annihilate@subtitle` `\subtitle` 命令を無効化する。

※独自の `\subtitle` が使われている場合は無効化しない。

```

1231 \def\bxjs@annihilate@subtitle{%
1232     \ifx\subtitle\jsSubtitle \global\let\subtitle\relax \fi
1233     \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 1234 %<*jspf>
\keywords 1235 \newcommand*{\etitle}[1]{\gdef\@etitle{#1}}
1236 \newcommand*{\author}[1]{\gdef\@author{#1}}
1237 \newcommand*{\keywords}[1]{\gdef\@keywords{#1}}
1238 \newcommand*{\email}[1]{\gdef\authors@mail{#1}}
1239 \newcommand*{\AuthorsEmail}[1]{\gdef\authors@mail{author's e-mail:\ #1}}
1240 %</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。

```

1241 \def\plainifnotempty{%
1242     \ifx \@oddhead \@empty
1243         \ifx \@oddfoot \@empty
1244             \else
1245                 \thispagestyle{plainfoot}%
1246             \fi
1247         \else
1248             \thispagestyle{plainhead}%
1249         \fi}

```

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

[2016-11-16] 新設された `nomag` および `nomag*` オプションの場合をデフォルト (`usemag` 相当) に合わせるため，`\smallskip` を `\jsc@smallskip` に置き換えました。`\smallskip`

のままでは `nomag(*)` の場合にスケールしなくなり、レイアウトが変わってしまいます。

```
1250 %<*article|book|report|slide>
1251 \if@titlepage
1252   \newcommand{\maketitle}{%
1253     \begin{titlepage}%
1254       \let\footnotesize\small
1255       \let\footnoterule\relax
1256       \let\footnote\thanks
1257       \null\vfil
1258       \if@slide
1259         {\footnotesize \@date}%
1260         \begin{center}
1261           \mbox{} \\\[1\jsZw]
1262           \large
1263           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1264           \jsc@smallskip
1265           \@title
1266           \ifx\bxjs@subtitle\@undefined\else
1267             \par\vskip\z@
1268             {\small \bxjs@subtitle\par}
1269           \fi
1270           \jsc@smallskip
1271           {\maybeblue\hrule height0\p@? depth2\p@?\relax}\par
1272           \vfill
1273           {\small \@author}%
1274         \end{center}
1275       \else
1276       \vskip 60\p@?
1277       \begin{center}%
1278         {\LARGE \@title \par}%
1279         \ifx\bxjs@subtitle\@undefined\else
1280           \vskip5\p@?
1281           {\normalsize \bxjs@subtitle\par}
1282         \fi
1283         \vskip 3em%
1284         {\large
1285           \lineskip .75em
1286           \begin{tabular}[t]{c}%
1287             \@author
1288           \end{tabular}\par}%
1289         \vskip 1.5em
1290         {\large \@date \par}%
1291       \end{center}%
1292       \fi
1293       \par
1294       \@thanks\vfil\null
1295     \end{titlepage}%
1296     \setcounter{footnote}{0}%
```

```

1297 \global\let\thanks\relax
1298 \global\let\maketitle\relax
1299 \global\let\@thanks\@empty
1300 \global\let\@author\@empty
1301 \global\let\@date\@empty
1302 \global\let\@title\@empty
1303 \global\let\title\relax
1304 \global\let\author\relax
1305 \global\let\date\relax
1306 \global\let\and\relax
1307 \bxjs@annihilate@subtitle
1308 }%
1309 \else
1310 \newcommand{\maketitle}{\par
1311 \begingroup
1312 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1313 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1314 \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1315 \parindent 1\jsZw\noindent
1316 \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1317 \if@twocolumn
1318 \ifnum \col@number=\@ne
1319 \maketitle
1320 \else
1321 \twocolumn[\maketitle]%
1322 \fi
1323 \else
1324 \newpage
1325 \global\@topnum\z@ % Prevents figures from going at top of page.
1326 \maketitle
1327 \fi
1328 \plainifnotempty
1329 \@thanks
1330 \endgroup
1331 \setcounter{footnote}{0}%
1332 \global\let\thanks\relax
1333 \global\let\maketitle\relax
1334 \global\let\@thanks\@empty
1335 \global\let\@author\@empty
1336 \global\let\@date\@empty
1337 \global\let\@title\@empty
1338 \global\let\title\relax
1339 \global\let\author\relax
1340 \global\let\date\relax
1341 \global\let\and\relax
1342 \bxjs@annihilate@subtitle
1343 }

```

`\@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。

```

1344 \def\@maketitle{%
1345     \newpage\null
1346     \vskip 2em
1347     \begin{center}%
1348         \let\footnote\thanks
1349         {\LARGE \@title \par}%
1350         \ifx\bxjs@subtitle\@undefined\else
1351             \vskip3\p@?
1352             {\normalsize \bxjs@subtitle\par}
1353         \fi
1354         \vskip 1.5em
1355         {\large
1356             \lineskip .5em
1357             \begin{tabular}[t]{c}%
1358                 \@author
1359             \end{tabular}\par}%
1360         \vskip 1em
1361         {\large \@date}%
1362     \end{center}%
1363     \par\vskip 1.5em
1364 %<article|slide> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
1365 }
1366 \fi
1367 %</article|book|report|slide>
1368 %<*jspf>
1369 \newcommand{\maketitle}{\par
1370     \begingroup
1371         \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
1372         \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
1373         \long\def\@makefntext##1{\advance\leftskip 3\jsZw
1374             \parindent 1\jsZw\noindent
1375             \llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3\jsZw}##1}%
1376             \twocolumn[\@maketitle]%
1377         \plainifnotempty
1378         \@thanks
1379     \endgroup
1380     \setcounter{footnote}{0}%
1381     \global\let\thanks\relax
1382     \global\let\maketitle\relax
1383     \global\let\@thanks\@empty
1384     \global\let\@author\@empty
1385     \global\let\@date\@empty
1386 % \global\let\@title\@empty % \@title は柱に使う
1387     \global\let\title\relax
1388     \global\let\author\relax
1389     \global\let\date\relax
1390     \global\let\and\relax

```

```

1391 \ifx\authors@mail\@undefined\else{%
1392   \def\@makefnstext{\advance\leftskip 3\jsZw \parindent -3\jsZw}%
1393   \footnotetext[0]{\itshape\authors@mail}%
1394 } \fi
1395 \global\let\authors@mail\@undefined}
1396 \def\@maketitle{%
1397   \newpage\null
1398   \vskip 6em % used to be 2em
1399   \begin{center}
1400     \let\footnote\thanks
1401     \ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi
1402     \lineskip .5em
1403     \ifx\@author\@undefined\else
1404       \vskip 1em
1405       \begin{tabular}[t]{c}%
1406         \@author
1407       \end{tabular}\par
1408     \fi
1409     \ifx\@etitle\@undefined\else
1410       \vskip 1em
1411       {\large \@etitle \par}%
1412     \fi
1413     \ifx\@eauthor\@undefined\else
1414       \vskip 1em
1415       \begin{tabular}[t]{c}%
1416         \@eauthor
1417       \end{tabular}\par
1418     \fi
1419     \vskip 1em
1420     \@date
1421   \end{center}
1422   \vskip 1.5em
1423   \centerline{\box\@abstractbox}
1424   \ifx\@keywords\@undefined\else
1425     \vskip 1.5em
1426     \centerline{\parbox{157mm}{\textsf{Keywords:}}\ \small\@keywords}}
1427   \fi
1428   \vskip 1.5em}
1429 %</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。

```
1430 \def\@startsection#1#2#3#4#5#6{%
1431   \if@noskipsec \leavevmode \fi
1432   \par
1433 % 見出し上の空きを \@tempskipa にセットする
1434   \@tempskipa #4\relax
1435 % \@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
1436   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1437 % 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
1438   \ifdim \@tempskipa <\z@
1439     \@tempskipa -\@tempskipa \@afterindentfalse
1440   \fi
1441   \if@nobreak
1442 %   \everypar{\everyparhook}% これは間違い
1443     \everypar{}%
1444   \else
1445     \addpenalty\@secpenalty
1446 % 次の行は削除
1447 %   \addvspace\@tempskipa
1448 % 次の \noindent まで追加
1449     \ifdim \@tempskipa >\z@
1450       \if@slide\else
1451         \null
1452         \vspace*{-\baselineskip}%
1453       \fi
1454       \vskip\@tempskipa
1455     \fi
```

```

1456 \fi
1457 \noindent
1458 % 追加終わり
1459 \@ifstar
1460     {\@ssect{#3}{#4}{#5}{#6}}%
1461     {\@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。

1462 \def\@sect#1#2#3#4#5#6[#7]#8{%
1463     \ifnum #2>\c@secnumdepth
1464         \let\@svsec\empty
1465     \else
1466         \refstepcounter{#1}%
1467         \protected@edef\@svsec{\@secntformat{#1}\relax}%
1468     \fi
1469 % 見出し後の空きを \@tempskipa にセット
1470 \@tempskipa #5\relax
1471 % 条件判断の順序を入れ替えました
1472 \ifdim \@tempskipa<\z@
1473     \def\@svsechd{%
1474         #6{\hskip #3\relax
1475             \@svsec #8}%
1476         \csname #1mark\endcsname{#7}%
1477         \addcontentsline{toc}{#1}{%
1478             \ifnum #2>\c@secnumdepth \else
1479                 \protect\numberline{\csname the#1\endcsname}%
1480             \fi
1481             #7}}% 目次にフルネームを載せるなら #8
1482     \else
1483         \begingroup
1484             \interlinepenalty \@M % 下から移動
1485             #6{%
1486                 \@hangfrom{\hskip #3\relax\@svsec}%
1487             % \interlinepenalty \@M % 上に移動
1488             #8\@@par}%
1489         \endgroup
1490         \csname #1mark\endcsname{#7}%
1491         \addcontentsline{toc}{#1}{%
1492             \ifnum #2>\c@secnumdepth \else
1493                 \protect\numberline{\csname the#1\endcsname}%
1494             \fi
1495             #7}}% 目次にフルネームを載せるならここは #8
1496     \fi
1497     \@xsect{#5}}

```

二つ挿入した `\everyparhook` のうち後者が `\paragraph` 類の後で 2 回実行され、それ以降は前者が実行されます。

[2016-07-28] `slide` オプションと `twocolumn` オプションを同時に指定した場合の罫線の

位置を微調整しました。

```
1498 \def\xsect#1{%
1499 % 見出しの後ろの空きを \@tempkipa にセット
1500 \@tempkipa #1\relax
1501 % 条件判断の順序を変えました
1502 \ifdim \@tempkipa<\z@
1503   \@nbreakfalse
1504   \global\@noskipsecttrue
1505   \everypar{%
1506     \if@noskipsec
1507       \global\@noskipsecfalse
1508       {\setbox\z@\lastbox}%
1509       \clubpenalty\@M
1510       \begingroup \@svsechd \endgroup
1511       \unskip
1512       \@tempkipa #1\relax
1513       \hskip -\@tempkipa
1514       \bxjs<ltj\inhibitglue
1515     \else
1516       \clubpenalty \@clubpenalty
1517       \everypar{\everyparhook}%
1518     \fi\everyparhook}%
1519 \else
1520   \par \nobreak
1521   \vskip \@tempkipa
1522   \@afterheading
1523 \fi
1524 \if@slide
1525   {\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
1526    \maybeblue\hrule height0\jsc@empt depth1\jsc@empt
1527    \vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
1528 \fi
1529 \par % 2000-12-18
1530 \ignorespaces}
1531 \def@ssect#1#2#3#4#5{%
1532 \@tempkipa #3\relax
1533 \ifdim \@tempkipa<\z@
1534   \def\@svsechd{#4{\hskip #1\relax #5}}%
1535 \else
1536   \begingroup
1537   #4{%
1538     \@hangfrom{\hskip #1}%
1539     \interlinepenalty \@M #5\@par}%
1540   \endgroup
1541 \fi
1542 \xsect{#3}}
```

上記の定義中の `\bxjs@ltj@inhibitglue` は Lua_T_EX-j_a で用いられるフック。
1543 `\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 1544 \newcommand*\chaptermark[1]{}
                  1545 % \newcommand*\sectionmark[1]{}
\subsubsectionmark 1546 % \newcommand*\subsectionmark[1]{}
                  1547 % \newcommand*\subsubsectionmark[1]{}
\paragraphmark 1548 % \newcommand*\paragraphmark[1]{}
\subparagraphmark 1549 % \newcommand*\subparagraphmark[1]{}

```

■カウンタの定義

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

```
1550 %<!book&!report>\setcounter{secnumdepth}{3}
1551 %<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 1552 \newcounter{part}
                  1553 %<book|report>\newcounter{chapter}
\c@subsubsection 1554 %<book|report>\newcounter{section}[chapter]
                  1555 %<!book&!report>\newcounter{section}
\c@paragraph 1556 \newcounter{subsection}[section]
\c@subparagraph 1557 \newcounter{subsubsection}[subsection]
                  1558 \newcounter{paragraph}[subsubsection]
                  1559 \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。

```
1560 \renewcommand{\thepart}{\@Roman\c@part}
1561 %<!book&!report>% \renewcommand{\thesection}{\@arabic\c@section}
1562 %<!book&!report>\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
1563 %<!book&!report>\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
```

```

1564 %<*book|report>
1565 \renewcommand{\thechapter}{\@arabic\c@chapter}
1566 \renewcommand{\thesection}{\thechapter.\@arabic\c@section}
1567 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
1568 %</book|report>
1569 \renewcommand{\thesubsubsection}{%
1570   \thesubsection.\@arabic\c@subsubsection}
1571 \renewcommand{\theparagraph}{%
1572   \thesubsubsection.\@arabic\c@paragraph}
1573 \renewcommand{\thesubparagraph}{%
1574   \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。

```

1575 %<book|report>\newcommand{\@chapapp}{\prechaptername}
1576 %<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。

```

1577 %<*book|report>
1578 \newcommand\frontmatter{%
1579   \if@openright
1580     \cleardoublepage
1581   \else
1582     \clearpage
1583   \fi
1584   \@mainmatterfalse
1585   \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。

```

1586 \newcommand\mainmatter{%
1587 % \if@openright
1588   \cleardoublepage
1589 % \else
1590 %   \clearpage
1591 % \fi
1592   \@mainmattertrue
1593   \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。

```

1594 \newcommand\backmatter{%
1595   \if@openright
1596     \cleardoublepage
1597   \else
1598     \clearpage

```

```

1599 \fi
1600 \@mainmatterfalse}
1601 %</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。

```

1602 %<*&book&!report>
1603 \newcommand\part{%
1604   \if@noskipsec \leavevmode \fi
1605   \par
1606   \addvspace{4ex}%
1607   \if@english \@afterindentfalse \else \@afterindenttrue \fi
1608   \secdef\@part\@spart}
1609 %</!book&!report>

```

book スタイルの場合は、少し複雑です。

```

1610 %<*&book|report>
1611 \newcommand\part{%
1612   \if@openright
1613     \cleardoublepage
1614   \else
1615     \clearpage
1616   \fi
1617   \thispagestyle{empty}% 欧文用標準スタイルでは plain
1618   \if@twocolumn
1619     \onecolumn
1620     \@restonecoltrue
1621   \else
1622     \@restonecolfalse
1623   \fi
1624   \null\vfil
1625   \secdef\@part\@spart}
1626 %</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。

```
1627 %<*!book&!report>
1628 \def\@part[#1]#2{%
1629   \ifnum \c@secnumdepth >\m@ne
1630     \refstepcounter{part}%
1631     \addcontentsline{toc}{part}{%
1632       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1633   \else
1634     \addcontentsline{toc}{part}{#1}%
1635   \fi
1636   \markboth{}{}%
1637   {\parindent\z@
1638     \raggedright
1639     \interlinepenalty \@M
1640     \normalfont
1641     \ifnum \c@secnumdepth >\m@ne
1642       \Large\headfont\prepartname\thepart\postpartname
1643       \par\nobreak
1644     \fi
1645     \huge \headfont #2%
1646     \markboth{}{}\par}%
1647   \nobreak
1648   \vskip 3ex
1649   \@afterheading}
1650 %</!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。

```
1651 %<*book|report>
1652 \def\@part[#1]#2{%
1653   \ifnum \c@secnumdepth >-2\relax
1654     \refstepcounter{part}%
1655     \addcontentsline{toc}{part}{%
1656       \prepartname\thepart\postpartname\hspace{1\jsZw}#1}%
1657   \else
1658     \addcontentsline{toc}{part}{#1}%
1659   \fi
1660   \markboth{}{}%
1661   {\centering
1662     \interlinepenalty \@M
1663     \normalfont
1664     \ifnum \c@secnumdepth >-2\relax
1665       \huge\headfont \prepartname\thepart\postpartname
1666       \par\vskip20\p@?
1667     \fi
1668     \Huge \headfont #2\par}%
1669   \@endpart}
1670 %</book|report>
```

\@spart 番号を付けない部です。

```

1671 %<*!book&!report>
1672 \def\@spart#1{%
1673     \parindent \z@ \raggedright
1674     \interlinepenalty \@M
1675     \normalfont
1676     \huge \headfont #1\par}%
1677 \nobreak
1678 \vskip 3ex
1679 \@afterheading}
1680 %</!book&!report>
1681 %<*book|report>
1682 \def\@spart#1{%
1683     \centering
1684     \interlinepenalty \@M
1685     \normalfont
1686     \Huge \headfont #1\par}%
1687 \@endpart}
1688 %</book|report>

```

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

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

```

1689 %<*book|report>
1690 \def\@endpart{\vfil\newpage
1691     \if@twoside
1692     \if@openright %% added (2016/12/13)
1693     \null
1694     \thispagestyle{empty}%
1695     \newpage
1696     \fi %% added (2016/12/13)
1697 \fi
1698 \if@restonecol
1699     \twocolumn
1700 \fi}
1701 %</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。

```

1702 %<*book|report>
1703 \newcommand{\chapter}{%
1704     \if@openright\cleardoublepage\else\clearpage\fi
1705     \plainifnotempty % 元: \thispagestyle{plain}
1706     \global\@topnum\z@

```

```

1707 \if@english \@afterindentfalse \else \@afterindenttrue \fi
1708 \secdef
1709   {\@omit@numberfalse\@chapter}%
1710   {\@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。

```

1711 \def\@chapter[#1]#2{%
1712   \ifnum \c@secnumdepth >\m@ne
1713     \if@mainmatter
1714       \refstepcounter{chapter}%
1715       \typeout{\@chapapp\thechapter\@chappos}%
1716       \addcontentsline{toc}{chapter}%
1717         {\protect\numberline
1718 %       %{\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
1719         {\@chapapp\thechapter\@chappos}%
1720         #1}%
1721     \else\addcontentsline{toc}{chapter}{#1}\fi
1722   \else
1723     \addcontentsline{toc}{chapter}{#1}%
1724   \fi
1725   \chaptermark{#1}%
1726   \addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
1727   \addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
1728   \if@twocolumn
1729     \@topnewpage[\@makechapterhead{#2}]%
1730   \else
1731     \@makechapterhead{#2}%
1732     \@afterheading
1733   \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。

```

1734 \def\@makechapterhead#1{%
1735   \vspace*{2\Cvs}% 欧文は 50pt
1736   {\parindent \z@ \raggedright \normalfont
1737     \ifnum \c@secnumdepth >\m@ne
1738       \if@mainmatter
1739         \huge\headfont \@chapapp\thechapter\@chappos
1740         \par\nobreak
1741         \vskip \Cvs % 欧文は 20pt
1742       \fi
1743     \fi
1744     \interlinepenalty\@M
1745     \Huge \headfont #1\par\nobreak
1746     \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。

```

1747 \def\@schapter#1{%
1748   \chaptermark{#1}%

```

```

1749 \if@twocolumn
1750 \topnewpage[\@makeschapterhead{#1}]%
1751 \else
1752 \@makeschapterhead{#1}\@afterheading
1753 \fi}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

1754 \def\@makeschapterhead#1{%
1755 \vspace*{2\Cvs}% 欧文は 50pt
1756 {\parindent \z@ \raggedright
1757 \normalfont
1758 \interlinepenalty\@M
1759 \Huge \headfont #1\par\nobreak
1760 \vskip 3\Cvs}} % 欧文は 40pt
1761 %</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。

```

1762 \if@twocolumn
1763 \newcommand{\section}{%
1764 %<jspf>\ifx\maketitle\relax\else\maketitle\fi
1765 \@startsection{section}{1}{\z@}%
1766 %<kiyou> {0.6\Cvs}{0.4\Cvs}%
1767 %<kiyou> {\Cvs}{0.5\Cvs}%
1768 % {\normalfont\large\headfont\@secapp}}
1769 {\normalfont\large\headfont\raggedright}}
1770 \else
1771 \newcommand{\section}{%
1772 \if@slide\clearpage\fi
1773 \@startsection{section}{1}{\z@}%
1774 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1775 {.5\Cvs \@plus.3\Cdp}% 後アキ
1776 % {\normalfont\Large\headfont\@secapp}}
1777 {\normalfont\Large\headfont\raggedright}}
1778 \fi

```

`\subsection` 同上です。

```

1779 \if@twocolumn
1780 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1781 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1782 {\normalfont\normalsize\headfont}}
1783 \else
1784 \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
1785 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
1786 {.5\Cvs \@plus.3\Cdp}% 後アキ

```



```
1787 {\normalfont\large\headfont}}
1788 \fi
```

\subsubsection [2016-07-22] slide オプション指定時に \subsubsection の文字列と罫線が重なる問題に
対処しました (forum:1982)。

```
1789 \if@twocolumn
1790 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1791 {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
1792 {\normalfont\normalsize\headfont}}
1793 \else
1794 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
1795 {\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1796 {\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
1797 {\normalfont\normalsize\headfont}}
1798 \fi
```

\paragraph 見出しの後ろで改行されません。

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

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

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

※ BXJS クラスでは、1.1 版 [2016-02-14] から \jsParagraphMark をサポートしている。
段落のマーク（■）が必ず和文フォントで出力されるようにする。

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

```
1799 \newcommand\jsParagraphMark{\jsJaChar{■}}
1800 \ifx\bxjs@paragraph@mark\undefined\else
1801 \long\edef\jsParagraphMark{\noexpand\jsJaChar{\bxjs@paragraph@mark}}
1802 \fi
1803 \let\jsJaChar\@empty
1804 \if@twocolumn
1805 \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1806 {\z@}{\if@slide .4\Cvs \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1807 %<jspf> {\normalfont\normalsize\headfont}}
1808 %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
1809 \else
1810 \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
1811 {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
1812 {\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}% 改行せず 1\jsZw のアキ
1813 %<jspf> {\normalfont\normalsize\headfont}}
1814 %<!jspf> {\normalfont\normalsize\headfont\jsParagraphMark}}
```

1815 \fi

\subparagraph 見出しの後ろで改行されません。

```
1816 \if@twocolumn
1817   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1818     {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1819     {\normalfont\normalsize\headfont}}
1820 \else
1821   \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
1822     {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1\jsZw\fi}%
1823     {\normalfont\normalsize\headfont}}
1824 \fi
```

7.3 リスト環境

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

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

```
1825 \if@slide
1826   \setlength\leftmargini{1\jsZw}
1827 \else
1828   \if@twocolumn
1829     \setlength\leftmargini{2\jsZw}
1830   \else
1831     \setlength\leftmargini{3\jsZw}
1832   \fi
1833 \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 1834 \if@slide
\leftmarginv 1835   \setlength\leftmarginii {1\jsZw}
1836   \setlength\leftmarginiii{1\jsZw}
\leftmarginvi 1837   \setlength\leftmarginiv {1\jsZw}
1838   \setlength\leftmarginv {1\jsZw}
1839   \setlength\leftmarginvi {1\jsZw}
1840 \else
1841   \setlength\leftmarginii {2\jsZw}
1842   \setlength\leftmarginiii{2\jsZw}
1843   \setlength\leftmarginiv {2\jsZw}
1844   \setlength\leftmarginv {1\jsZw}
1845   \setlength\leftmarginvi {1\jsZw}
1846 \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。

```
1847 \setlength \labelsep {0.5\jsZw} % .5em
1848 \setlength \labelwidth{\leftmargini}
1849 \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。

```
1850 \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 1851 \@beginparpenalty -\@lowpenalty
\@itempenalty 1852 \@endparpenalty -\@lowpenalty
1853 \@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。

```
1854 \def\@listi{\leftmargin\leftmargini
1855 \parsep \z@
1856 \topsep 0.5\baselineskip
1857 \itemsep \z@ \relax}
1858 \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）。

```
1859 \@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 1860 \def\@listii{\leftmargin\leftmarginii
\@listiv 1861 \labelwidth\leftmarginii \advance\labelwidth-\labelsep
1862 \topsep \z@
\@listv 1863 \parsep \z@
\@listvi 1864 \itemsep\parsep}
1865 \def\@listiii{\leftmargin\leftmarginiii
1866 \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
1867 \topsep \z@
1868 \parsep \z@
1869 \itemsep\parsep}
1870 \def\@listiv {\leftmargin\leftmarginiv
1871 \labelwidth\leftmarginiv
1872 \advance\labelwidth-\labelsep}
1873 \def\@listv {\leftmargin\leftmarginv
```

```

1874 \labelwidth\leftmarginv
1875 \advance\labelwidth-\labelsep}
1876 \def\@listvi {\leftmargin\leftmarginvi
1877 \labelwidth\leftmarginvi
1878 \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。

```

1879 \renewcommand{\theenumi}{\@arabic\c@enumi}
1880 \renewcommand{\theenumii}{\@alph\c@enumii}
1881 \renewcommand{\theenumiii}{\@roman\c@enumiii}
1882 \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。

```

1883 \def\bxjs@dust{\vrule\@width\z@\@height\z@\@depth\z@}
1884 \newcommand*\jsInJaParen[1]{%
1885 \bxjs@dust\jsInhibitGlue (\theenumii) \jsInhibitGlue\bxjs@dust}
1886 \newcommand{\labelenumi}{\theenumi.}
1887 \newcommand{\labelenumii}{\jsInJaParen{ (\theenumii) }}
1888 \newcommand{\labelenumiii}{\theenumiii.}
1889 \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 1890 \renewcommand{\p@enumii}{\theenumi}
1891 \renewcommand{\p@enumiii}{\theenumi\jsInhibitGlue (\theenumii) }
1892 \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`

```

1893 \newcommand\labelitemi{\textbullet}
1894 \newcommand\labelitemii{\normalfont\bfseries \textendash}
1895 \newcommand\labelitemiii{\textasteriskcentered}
1896 \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。

```

1897 \newenvironment{description}{%
1898   \list{}{%
1899     \labelwidth=\leftmargin
1900     \labelsep=1\jsZw
1901     \advance \labelwidth by -\labelsep
1902     \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。

```

1903 \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。

```

1904 %<*book|report>
1905 \newenvironment{abstract}{%
1906   \begin{list}{}{%
1907     \listparindent=1\jsZw
1908     \itemindent=\listparindent
1909     \rightmargin=\z@
1910     \leftmargin=5\jsZw}\item[]}{\end{list}\vspace{\baselineskip}}
1911 %</book|report>
1912 %<*article|slide>
1913 \newbox\@abstractbox
1914 \if@titlepage
1915   \newenvironment{abstract}{%
1916     \titlepage
1917     \null\vfil
1918     \@beginparpenalty\@lowpenalty
1919     \begin{center}%
1920       \headfont \abstractname
1921       \@endparpenalty\@M
1922     \end{center}}{%
1923     {\par\vfil\null\endtitlepage}

```

```

1924 \else
1925   \newenvironment{abstract}{%
1926     \if@twocolumn
1927       \ifx\maketitle\relax
1928         \section*{\abstractname}%
1929       \else
1930         \global\setbox\@abstractbox\hbox\bgroup
1931         \begin{minipage}[b]{\textwidth}
1932           \small\parindent1\jsZw
1933           \begin{center}%
1934             {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1935           \end{center}%
1936           \list{}{%
1937             \listparindent\parindent
1938             \itemindent \listparindent
1939             \rightmargin \leftmargin}%
1940           \item\relax
1941         \fi
1942       \else
1943         \small
1944         \begin{center}%
1945           {\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
1946         \end{center}%
1947         \list{}{%
1948           \listparindent\parindent
1949           \itemindent \listparindent
1950           \rightmargin \leftmargin}%
1951         \item\relax
1952       \fi}{\if@twocolumn
1953         \ifx\maketitle\relax
1954         \else
1955           \endlist\end{minipage}\egroup
1956         \fi
1957       \else
1958         \endlist
1959       \fi}
1960 \fi
1961 %</article|slide>
1962 %<*jspf>
1963 \newbox\@abstractbox
1964 \newenvironment{abstract}{%
1965   \global\setbox\@abstractbox\hbox\bgroup
1966   \begin{minipage}[b]{157mm}{\sffamily Abstract}\par
1967     \small
1968     \if@english \parindent6mm \else \parindent1\jsZw \fi}%
1969   {\end{minipage}\egroup}
1970 %</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。

```
1971 %<*jspf>
1972 %\newbox\@keywordsbox
1973 %\newenvironment{keywords}{%
1974 % \global\setbox\@keywordsbox\hbox\bgroup
1975 % \begin{minipage}[b]{157mm}{\sffamily Keywords:}\par
1976 % \small\parindent0\jsZw}%
1977 % {\end{minipage}\egroup}
1978 %</jspf>
```

■verse 環境

verse 詩のための `verse` 環境です。

```
1979 \newenvironment{verse}{%
1980 \let \=\@centercr
1981 \list{}{%
1982 \itemsep \z@
1983 \itemindent -2\jsZw % 元: -1.5em
1984 \listparindent\itemindent
1985 \rightmargin \z@
1986 \advance\leftmargin 2\jsZw}% 元: 1.5em
1987 \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。

```
1988 \newenvironment{quotation}{%
1989 \list{}{%
1990 \listparindent\parindent
1991 \itemindent\listparindent
1992 \rightmargin \z@}%
1993 \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。

```
1994 \newenvironment{quote}%
1995 {\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。

```
1996 \def\@begintheorem#1#2{\trivlist\labelsep=1\jsZw
1997   \item[\hskip \labelsep{\headfont #1\ #2}]}
1998 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\jsZw
1999   \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。

```
2000 \newenvironment{titlepage}{%
2001   %<book|report> \cleardoublepage
2002   \if@twocolumn
2003     \@restonecoltrue\onecolumn
2004   \else
2005     \@restonecolfalse\newpage
2006   \fi
2007   \thispagestyle{empty}%
2008   \setcounter{page}\@ne
2009 }%
2010 {\if@restonecol\twocolumn \else \newpage \fi
2011   \if@twoside\else
2012     \setcounter{page}\@ne
2013   \fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
2014 %<!*book&!report>
2015 \newcommand{\appendix}{\par
2016   \setcounter{section}{0}%
2017   \setcounter{subsection}{0}%
2018   \gdef\presectionname{\appendixname}%
2019   \gdef\postsectionname{}}%
2020 % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
2021 \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
2022 \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
2023 %</!*book&!report>
2024 %<*book|report>
2025 \newcommand{\appendix}{\par
2026   \setcounter{chapter}{0}%
2027   \setcounter{section}{0}%
2028   \gdef\@chapapp{\appendixname}%
2029   \gdef\@chappos{}}%
2030 \gdef\thechapter{\@Alph\c@chapter}}
2031 %</book|report>

```


7.4 パラメータの設定

■array と tabular 環境

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

```
2032 \setlength\arraycolsep{5\p@?}
```

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

```
2033 \setlength\tabcolsep{6\p@?}
```

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

```
2034 \setlength\arrayrulewidth{.4\p@}
```

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

```
2035 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` `\'` コマンドで入るアキです。

```
2036 \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。

```
2037 \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。

```
2038 \setlength\fboxsep{3\p@?}
```

```
2039 \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。

```
2040 %<!book&!report>\renewcommand \theequation {\@arabic\c@equation}
```

```
2041 %<*book|report>
```

```
2042 \@addtoreset{equation}{chapter}
```

```
2043 \renewcommand\theequation
```

```
2044 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

```
2045 %</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。

```
2046 % \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。

```
2047 % \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。

```
2048 % \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。

```
2049 %<!*book&!report>
2050 \newcounter{figure}
2051 \renewcommand \thefigure {\@arabic\c@figure}
2052 %</!*book&!report>
2053 %<!*book|report>
2054 \newcounter{figure}[chapter]
2055 \renewcommand \thefigure
2056     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
2057 %</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 2058 \def\fps@figure{tbp}
2059 \def\ftype@figure{1}
\fnum@figure 2060 \def\ext@figure{lof}
2061 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 2062 \newenvironment{figure}%
2063     {\@float{figure}}%
2064     {\end@float}
```

```

2065 \newenvironment{figure*}%
2066         {\@dblfloat{figure}}%
2067         {\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。

```

2068 %<!*book&!report>
2069 \newcounter{table}
2070 \renewcommand\thetable{\@arabic\c@table}
2071 %</!*book&!report>
2072 %<*book|report>
2073 \newcounter{table}[chapter]
2074 \renewcommand \thetable
2075         {\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@table}
2076 %</book|report>

```

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

```

\ext@table 2077 \def\fps@table{tbp}
2078 \def\ftype@table{2}
\fnun@table 2079 \def\ext@table{lot}
2080 \def\fnun@table{\tablename\nobreak\thetable}

```

`table` * は段抜きのフロートです。

```

table* 2081 \newenvironment{table}%
2082         {\@float{table}}%
2083         {\end@float}
2084 \newenvironment{table*}%
2085         {\@dblfloat{table}}%
2086         {\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
うのを直しました。

```

2087 \newlength\abovecaptionskip
2088 \newlength\belowcaptionskip
2089 \setlength\abovecaptionskip{5\p@?} % 元: 10\p@
2090 \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。

```
2091 %<*!jspf>
2092 % \long\def\@makecaption#1#2{{\small
2093 % \advance\leftskip1cm
2094 % \advance\rightskip1cm
2095 % \vskip\abovecaptionskip
2096 % \sbox\@tempboxa{#1\hskip1\jsZw\relax #2}%
2097 % \ifdim \wd\@tempboxa >\hsize
2098 % #1\hskip1\jsZw\relax #2\par
2099 % \else
2100 % \global \@minipagefalse
2101 % \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2102 % \fi
2103 % \vskip\belowcaptionskip}}
2104 \long\def\@makecaption#1#2{{\small
2105 \advance\leftskip .0628\linewidth
2106 \advance\rightskip .0628\linewidth
2107 \vskip\abovecaptionskip
2108 \sbox\@tempboxa{#1\zwspace#2}%
2109 \ifdim \wd\@tempboxa <\hsize \centering \fi
2110 #1\zwspace#2\par
2111 \vskip\belowcaptionskip}}
2112 %</!jspf>
2113 %<*jspf>
2114 \long\def\@makecaption#1#2{%
2115 \vskip\abovecaptionskip
2116 \sbox\@tempboxa{\small\sffamily #1\quad #2}%
2117 \ifdim \wd\@tempboxa >\hsize
2118 {\small\sffamily
2119 \list{#1}{%
2120 \renewcommand{\makelabel}[1]{##1\hfil}
2121 \itemsep \z@
2122 \itemindent \z@
2123 \labelsep \z@
2124 \labelwidth 11mm
2125 \listparindent\z@
2126 \leftmargin 11mm}\item\relax #2\endlist}
2127 \else
2128 \global \@minipagefalse
2129 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
2130 \fi
2131 \vskip\belowcaptionskip}
2132 %</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 `\scrDeclareOldFontCommand` に倣い、これらの命令を使うときには警告を発することにしました。

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

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

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

```
2133 \newif\if@jsc@warnoldfontcmd
2134 \@jsc@warnoldfontcmdtrue
2135 \newif\if@jsc@warnoldfontcmdexception
2136 \@jsc@warnoldfontcmdexceptionfalse

\jsc@DeclareOldFontCommand

2137 \newcommand*{\jsc@DeclareOldFontCommand}[3]{%
2138   \g@addto@macro\bxjs@oldfontcmd@list{\do#1}%
2139   \DeclareOldFontCommand{#1}{%
2140     \bxjs@oldfontcmd{#1}#2%
2141   }{%
2142     \bxjs@oldfontcmd{#1}#3%
2143   }%
2144 }
2145 \DeclareRobustCommand*{\jsc@warnoldfontcmd}[1]{%
2146   \ClassInfo\bxjs@clsname
2147   {Old font command '\string#1' is used!!\MessageBreak
2148     The first occurrence is}%
2149 }
```

`\allowoldfontcommands` “二文字フォント命令”の使用を許可する（警告しない）。
`\disallowoldfontcommands` “二文字フォント命令”の使用に対して警告を出す。

```
2150 \newcommand*{\allowoldfontcommands}{%
2151   \@jsc@warnoldfontcmdexceptiontrue}
2152 \newcommand*{\disallowoldfontcommands}{%
2153   \@jsc@warnoldfontcmdexceptionfalse}
```

※ 1.x 版では Warning ではなく Info に留めておく。

```

2154 \let\bxjs@oldfontcmd@list\@empty
2155 \def\bxjs@oldfontcmd#1{%
2156   \expandafter\bxjs@oldfontcmd@a\csname bxjs@ofc/\string#1\endcsname#1}
2157 \def\bxjs@oldfontcmd@a#1#2{%
2158   \if@jsc@warnoldfontcmdexception\else
2159     \global\@jsc@warnoldfontcmdfalse
2160     \ifx#1\relax
2161       \global\let#1=t%
2162       \jsc@warnoldfontcmd{#2}%
2163     \fi
2164   \fi}
2165 \def\bxjs@warnoldfontcmd@final{%
2166 % \par
2167 \global\let\bxjs@warnoldfontcmd@final\@empty
2168 \let\@tempa\@empty
2169 \def\do##1{%
2170   \expandafter\ifx\csname bxjs@ofc/\string##1\endcsname\relax\else
2171     \edef\@tempa{\@tempa \space\string##1}\fi}
2172 \bxjs@oldfontcmd@list
2173 \ifx\@tempa\@empty\else
2174   \ClassWarningNoLine\bxjs@clsname
2175   {Some old font commands were used in text\MessageBreak
2176    (see the log file for detail)}%
2177   \ClassInfo\bxjs@clsname
2178   {Some old font commands were used in text:\MessageBreak
2179    \space\@tempa\MessageBreak
2180    You should note, that since 1994 LaTeX2e provides a\MessageBreak
2181    new font selection scheme called NFSS2 with several\MessageBreak
2182    new, combinable font commands. The
2183    class provides\MessageBreak
2184    the old font commands
2185    only for compatibility%
2186    \@gobble}%
2187 \fi}

```

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

```

2188 \def\bxjs@warnoldfontcmd@kick@final{%
2189   \g@addto@macro\clearpage{\bxjs@warnoldfontcmd@final}}
2190 \AtEndDocument{\bxjs@warnoldfontcmd@kick@final}

```

`\mc` フォントファミリーを変更します。

```

\gt 2191 \jsc@DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 2192 \jsc@DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
2193 \jsc@DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sfb
\tt

```

```
2194 \jsc@DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
2195 \jsc@DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
```

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

```
2196 \jsc@DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

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

```
2197 \jsc@DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
2198 \jsc@DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
2199 \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 2200 \DeclareRobustCommand*{\cal}{\@fontswitch\relax\mathcal}
2201 \DeclareRobustCommand*{\mit}{\@fontswitch\relax\mathnormal}
```

9 相互参照

9.1 目次の類

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

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

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

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

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

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

この「番号」は `\thefigure` コマンドで生成された図番号です。

`table` 環境も同様です。

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

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

という書式です。

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

インデント 左側の字下げ量です。

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

`\@pnumwidth` ページ番号の入る箱の幅です。

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

`\@dotsep` 点の間隔です (単位 mu)。

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

```
2202 \newcommand\@pnumwidth{1.55em}
2203 \newcommand\@tocrmarg{2.55em}
2204 \newcommand\@dotsep{4.5}
2205 %<!book&!report>\setcounter{tocdepth}{2}
2206 %<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)

```
2207 \newdimen\jsc@tocl@width
2208 \newcommand{\tableofcontents}{%
2209 %<*book|report>
2210 \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
2211 \settowidth\@tempdima{\headfont\appendixname}%
2212 \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
2213 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2214 \if@twocolumn
2215 \@restonecoltrue\onecolumn
2216 \else
2217 \@restonecolfalse
2218 \fi
2219 \chapter*{\contentsname}%
2220 \@mkboth{\contentsname}{}%
2221 %</book|report>
2222 %<*!book&!report>
2223 \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
2224 \settowidth\@tempdima{\headfont\appendixname}%
2225 \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
2226 \ifdim\jsc@tocl@width<2\jsZw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1\jsZw\fi
2227 \section*{\contentsname}%
2228 \@mkboth{\contentsname}{\contentsname}%
2229 %</!book&!report>
2230 \@starttoc{toc}%
2231 %<book|report> \if@restonecol\twocolumn\fi
2232 }
```


`\l@part` 部の目次です。

```
2233 \newcommand*{\l@part}[2]{%
2234   \ifnum \c@tocdepth >-2\relax
2235   %<!book&!report>   \addpenalty\@secpenalty
2236   %<book|report>     \addpenalty{-\@highpenalty}%
2237   \addvspace{2.25em \@plus\p@?}%
2238   \begingroup
2239     \parindent \z@
2240   %   \@pnumwidth should be \@tocrmarg
2241   %   \rightskip \@pnumwidth
2242   \rightskip \@tocrmarg
2243   \parfillskip -\rightskip
2244   {\leavevmode
2245     \large \headfont
2246     \setlength\@lnumwidth{4\jsZw}%
2247     #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
2248   \nobreak
2249 %<book|report>   \global\@nobreaktrue
2250 %<book|report>   \everypar{\global\@nobreakfalse\everypar{}}%
2251   \endgroup
2252   \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)

```
2253 %<*book|report>
2254 \newcommand*{\l@chapter}[2]{%
2255   \ifnum \c@tocdepth >\m@ne
2256     \addpenalty{-\@highpenalty}%
2257     \addvspace{1.0em \@plus\p@?}
2258   %   \vskip 1.0em \@plus\p@   % book.cls では↑がこうなっている
2259   \begingroup
2260     \parindent\z@
2261   %   \rightskip\@pnumwidth
2262   \rightskip\@tocrmarg
2263   \parfillskip-\rightskip
2264   \leavevmode\headfont
2265   %   % \if@english\setlength\@lnumwidth{5.5em}\else\setlength\@lnumwidth{4.683\jsZw}\fi
2266   \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2.683\jsZw
2267   \advance\leftskip\@lnumwidth \hskip-\leftskip
2268   #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2269   \penalty\@highpenalty
2270   \endgroup
2271   \fi}
2272 %</book|report>
```

`\l@section` 節の目次です。

```
2273 %<*!book&!report>
```

```

2274 \newcommand*{\l@section}[2]{%
2275   \ifnum \c@tocdepth >\z@
2276     \addpenalty{\@secpenalty}%
2277     \addvspace{1.0em \@plus\p@?}%
2278     \begingroup
2279       \parindent\z@
2280 %     \rightskip\@pnumwidth
2281     \rightskip\@tocrmarg
2282     \parfillskip-\rightskip
2283     \leavevmode\headfont
2284 %     % \setlength\@lnumwidth{4\jsZw}% 元 1.5em [2003-03-02]
2285     \setlength\@lnumwidth{\jsc@tocl@width}\advance\@lnumwidth 2\jsZw
2286     \advance\leftskip\@lnumwidth \hskip-\leftskip
2287     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
2288   \endgroup
2289   \fi}
2290 %</*!book&!report>

```

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

```

2291 %<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] ここの \jsc@tocl@width から決めるようにしてみました。(by ts)

```

\l@subparagraph 2292 %</*!book&!report>
2293 % \newcommand*{\l@subsection} {\@dottedtocline{2}{1.5em}{2.3em}}
2294 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{3.8em}{3.2em}}
2295 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{7.0em}{4.1em}}
2296 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{10em}{5em}}
2297 %
2298 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{2\jsZw}{3\jsZw}}
2299 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{3\jsZw}{3\jsZw}}
2300 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{4\jsZw}{3\jsZw}}
2301 %
2302 \newcommand*{\l@section}{%
2303   \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2304   \@dottedtocline{2}{\@tempdima}{3\jsZw}}
2305 \newcommand*{\l@subsubsection}{%
2306   \@tempdima\jsc@tocl@width \advance\@tempdima 0\jsZw
2307   \@dottedtocline{3}{\@tempdima}{4\jsZw}}
2308 \newcommand*{\l@paragraph}{%
2309   \@tempdima\jsc@tocl@width \advance\@tempdima 1\jsZw
2310   \@dottedtocline{4}{\@tempdima}{5\jsZw}}
2311 \newcommand*{\l@subparagraph}{%
2312   \@tempdima\jsc@tocl@width \advance\@tempdima 2\jsZw
2313   \@dottedtocline{5}{\@tempdima}{6\jsZw}}
2314 %</*!book&!report>

```

```

2315 %<*book|report>
2316 % \newcommand*{\l@section} {\@dottedtocline{2}{3.8em}{3.2em}}
2317 % \newcommand*{\l@subsubsection}{\@dottedtocline{3}{7.0em}{4.1em}}
2318 % \newcommand*{\l@paragraph} {\@dottedtocline{4}{10em}{5em}}
2319 % \newcommand*{\l@subparagraph} {\@dottedtocline{5}{12em}{6em}}
2320 \newcommand*{\l@section}{%
2321     \@tempdima\jsc@tocl@width \advance\@tempdima -1\jsZw
2322     \@dottedtocline{1}{\@tempdima}{3.683\jsZw}}
2323 \newcommand*{\l@subsubsection}{%
2324     \@tempdima\jsc@tocl@width \advance\@tempdima 2.683\jsZw
2325     \@dottedtocline{2}{\@tempdima}{3.5\jsZw}}
2326 \newcommand*{\l@subsubsection}{%
2327     \@tempdima\jsc@tocl@width \advance\@tempdima 6.183\jsZw
2328     \@dottedtocline{3}{\@tempdima}{4.5\jsZw}}
2329 \newcommand*{\l@paragraph}{%
2330     \@tempdima\jsc@tocl@width \advance\@tempdima 10.683\jsZw
2331     \@dottedtocline{4}{\@tempdima}{5.5\jsZw}}
2332 \newcommand*{\l@subparagraph}{%
2333     \@tempdima\jsc@tocl@width \advance\@tempdima 16.183\jsZw
2334     \@dottedtocline{5}{\@tempdima}{6.5\jsZw}}
2335 %</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。

```

2336 \newdimen\@lnumwidth
2337 \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。

```

2338 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
2339     \vskip \z@ \@plus.2\p@?
2340     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
2341         \parindent #2\relax\@afterindenttrue
2342         \interlinepenalty\@M
2343         \leavevmode
2344         \@lnumwidth #3\relax
2345         \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
2346         {#4}\nobreak
2347         \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
2348             mu$}\hfill \nobreak\hbext@{\@pnumwidth{#5}}
2349         \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

2350 \newcommand{\listoffigures}{%

```

```

2351 %<*book|report>
2352 \if@twocolumn\@restonecoltrue\onecolumn
2353 \else\@restonecolfalse\fi
2354 \chapter*{\listfigurename}%
2355 \@mkboth{\listfigurename}{}%
2356 %</book|report>
2357 %<*&book&!report>
2358 \section*{\listfigurename}%
2359 \@mkboth{\listfigurename}{\listfigurename}%
2360 %<*&book&!report>
2361 \@starttoc{lof}%
2362 %<book|report> \if@restonecol\twocolumn\fi
2363 }

```

`\l@figure` 図目次の項目を出力します。

```

2364 \newcommand*{\l@figure}{\@dottedtocline{1}{1\jsZw}{3.683\jsZw}}

```

`\listoftables` 表目次を出力します。

```

2365 \newcommand{\listoftables}{%
2366 %<*book|report>
2367 \if@twocolumn\@restonecoltrue\onecolumn
2368 \else\@restonecolfalse\fi
2369 \chapter*{\listtablename}%
2370 \@mkboth{\listtablename}{}%
2371 %</book|report>
2372 %<*&book&!report>
2373 \section*{\listtablename}%
2374 \@mkboth{\listtablename}{\listtablename}%
2375 %<*&book&!report>
2376 \@starttoc{lot}%
2377 %<book|report> \if@restonecol\twocolumn\fi
2378 }

```

`\l@table` 表目次は図目次と同じです。

```

2379 \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。

```

2380 \newdimen\bibindent
2381 \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。

```

2382 \newenvironment{thebibliography}[1]{%

```

```

2383 \jsc@warnoldfontcmdexceptiontrue
2384 \global\let\presectionname\relax
2385 \global\let\postsectionname\relax
2386 %<article|slide> \section*{\refname}\mkboth{\refname}{\refname}%
2387 %<*kiyou>
2388 \vspace{1.5\baselineskip}
2389 \subsubsection*{\refname}\mkboth{\refname}{\refname}%
2390 \vspace{0.5\baselineskip}
2391 %</kiyou>
2392 %<book|report> \chapter*{\bibname}\mkboth{\bibname}{}%
2393 %<book|report> \addcontentsline{toc}{chapter}{\bibname}%
2394 \list{\@biblabel{\@arabic\c@enumiv}}%
2395     {\settowidth\labelwidth{\@biblabel{#1}}%
2396     \leftmargin\labelwidth
2397     \advance\leftmargin\labelsep
2398     \@openbib@code
2399     \usecounter{enumiv}%
2400     \let\p@enumiv\@empty
2401     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
2402 %<kiyou> \small
2403 \sloppy
2404 \clubpenalty4000
2405 \@clubpenalty\clubpenalty
2406 \widowpenalty4000%
2407 \sfcode`.\@m}
2408 {\def\@noitemerr
2409     {\@latex@warning{Empty `thebibliography' environment}}}%
2410 \endlist}

```

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

```

2411 \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。

```

2412 \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。

```

2413 % \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。

```

2414 % \def\@citex[#1]#2{\leavevmode
2415 %     \let\@citea\@empty
2416 %     \@cite{\@for\@citeb:=#2\do

```

```

2417 %      {\citea\def\@citea{, \inhibitglue\penalty\@m\ }%
2418 %      \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
2419 %      \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
2420 %      \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
2421 %      \G@refundefinedtrue
2422 %      \@latex@warning
2423 %      {Citation `@\@citeb' on page \thepage \space undefined}}%
2424 %      {\@citeofmt{\csname b@\@citeb\endcsname}}}{#1}}
2425 % \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。

```

2426 % \DeclareRobustCommand\cite{\unskip
2427 %      \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}]
2428 % \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
2429 %      , \jsInhibitGlue\ #2\fi}} }$}

```

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くにマージンがずれる現象を直しました (Thanks: 藤村さん)。

```

2430 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
2431     \if@twocolumn
2432         \onecolumn\@restonecolfalse
2433     \else
2434         \clearpage\@restonecoltrue
2435     \fi
2436     \columnseprule.4pt \columnsep 2\jsZw
2437     \ifx\multicols\@undefined
2438 %<book|report>      \twocolumn[\@makeschapterhead{\indexname}%
2439 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}}%
2440 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2441 %<!book&!report>    \twocolumn[\section*{\indexname}}%
2442     \else
2443         \ifdim\textwidth<\fullwidth
2444             \setlength{\evensidemargin}{\oddsidemargin}
2445             \setlength{\textwidth}{\fullwidth}
2446             \setlength{\linewidth}{\fullwidth}
2447 %<book|report>      \begin{multicols}{3}[\chapter*{\indexname}%
2448 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}}%
2449 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2450 %<!book&!report>    \begin{multicols}{3}[\section*{\indexname}}%
2451     \else
2452 %<book|report>      \begin{multicols}{2}[\chapter*{\indexname}%
2453 %<book|report>      \addcontentsline{toc}{chapter}{\indexname}}%
2454 %<!book&!report>    \def\presectionname{}\def\postsectionname{%
2455 %<!book&!report>    \begin{multicols}{2}[\section*{\indexname}}%
2456     \fi

```

```

2457 \fi
2458 %<book|report> \mkboth{\indexname}{}%
2459 %<!book&!report> \mkboth{\indexname}{\indexname}%
2460 \plainifnotempty % \thispagestyle{plain}
2461 \parindent\z@
2462 \parskip\z@ \@plus .3\p@?\relax
2463 \let\item\@idxitem
2464 \raggedright
2465 \footnotesize\narrowbaselines
2466 }{
2467 \ifx\multicols\@undefined
2468 \if@restonecol\onecolumn\fi
2469 \else
2470 \end{multicols}
2471 \fi
2472 \clearpage
2473 }

```

`\@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 2474 \newcommand{\@idxitem}{\par\hangindent 4\jsZw} % 元 40pt
\subsubitem 2475 \newcommand{\subitem}{\@idxitem \hspace*{2\jsZw}} % 元 20pt
2476 \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。

```
2477 \newcommand{\indexspace}{\par \vskip 10\p@? \@plus5\p@? \@minus3\p@?\relax}
```

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

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

```

2478 \newcommand\seename{\if@english see\else →\fi}
2479 \newcommand\alsoname{\if@english see also\else →\fi}

```

9.4 脚注

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

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

```

2480 \ifx\pltx@foot@penalty\@undefined
2481 \let\footnotes@ve=\footnote
2482 \def\footnote{\jsInhibitGlue\footnotes@ve}
2483 \let\footnotemarks@ve=\footnotemark
2484 \def\footnotemark{\jsInhibitGlue\footnotemarks@ve}

```

2485 \fi

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

[2002-04-09] インプリメントの仕方を変えたため消しました。

[2013-04-23] 新しい p_TE_X 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 p_LA_TE_X の変更に追随しました (Thanks: 角藤さん)。p_LA_TE_X の日付が 2016/04/17 より新しい場合は、このパッチが不要なのであてません。

p_TE_X 依存のコードなので、minimal 和文ドライバ実装に移動。

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

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

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

2486 `\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。

2487 `% \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。

2488 `\renewcommand{\footnoterule}{%`

2489 `\kern-2.6\p@? \kern-.4\p@`

2490 `\hrule width .4\columnwidth`

2491 `\kern 2.6\p@?}`

`\c@footnote` 脚注番号は章ごとにリセットされます。

2492 `%<book|report>\@addtoreset{footnote}{chapter}`

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

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

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

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

2493 `\long\def\@footnotetext{%`


```

2494 \insert\footins\bgroup
2495 \normalfont\footnotesize
2496 \interlinepenalty\interfootnotelinepenalty
2497 \splittopskip\footnotesep
2498 \splitmaxdepth \dp\strutbox \floatingpenalty \MM
2499 \hsize\columnwidth \@parboxrestore
2500 \protected@edef\@currentlabel{%
2501 \csname p@footnote\endcsname\@thefnmark
2502 }%
2503 \color@begingroup
2504 \@makefnmark{%
2505 \rule{z@\footnotesep\ignorespaces}%
2506 \futurelet\next\fo@t}
2507 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
2508 \else \let\next\fo@t\fi \next}
2509 \def\fo@@t{\bgroup\aftergroup\@foot\let\next}
2510 \def\fo@t#1{#1\@foot}
2511 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
2512 \ifx\pltx@foot@penalty\undefined\else
2513 \ifhmode\null\fi
2514 \ifnum\pltx@foot@penalty=z@\else
2515 \penalty\pltx@foot@penalty
2516 \pltx@foot@penaltyz@
2517 \fi
2518 \fi}

```

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

```

2519 \newcommand\@makefnmark[1]{%
2520 \advance\leftskip 3\jsZw
2521 \parindent 1\jsZw
2522 \noindent
2523 \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。

```

2524 % \def\@xfootnotenext[#1]{%
2525 % \begingroup
2526 % \ifnum#1>z@
2527 % \csname c@\@mpfn\endcsname #1\relax
2528 % \unrestored@protected@xdef\@thefnmark{\thempfn}%
2529 % \else
2530 % \unrestored@protected@xdef\@thefnmark{}%
2531 % \fi

```

```
2532 % \endgroup
2533 % \@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

段落頭のかぎカッコなどを見かけ 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。

```
2534 \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。

```
2535 \def\everyparhook{\jsInhibitGlueAtParTop}
2536 \AtBeginDocument{\everypar{\everyparhook}}
```

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

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

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

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

```
2537 \def\@inhibitglue{%
2538   \futurelet\@let@token\@@inhibitglue}
2539 \begingroup
2540 \let\GDEF=\gdef
2541 \let\CATCODE=\catcode
2542 \let\ENDGROUP=\endgroup
2543 \CATCODE`k=12
2544 \CATCODE`a=12
2545 \CATCODE`n=12
2546 \CATCODE`j=12
2547 \CATCODE`i=12
2548 \CATCODE`c=12
2549 \CATCODE`h=12
2550 \CATCODE`r=12
```

```

2551 \CATCODE`t=12
2552 \CATCODE`e=12
2553 \GDEF\KANJI@CHARACTER{kanji character }
2554 \ENDGROUP
2555 \def\@@inhibitglue{%
2556   \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
2557 \expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
2558   \def\bxjs@ig@temp{#1}%
2559   \ifx\bxjs@ig@temp\@empty
2560     \ifnum\the\inhibitxspcode`#2=2\relax
2561       \inhibitglue
2562     \fi
2563   \fi}

```

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

まず、環境の直後の段落です。

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

```

2564 \def\@doendpe{%
2565   \@endpetrue
2566   \def\par{%
2567     \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
2568   \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

`\item` 命令の直後です。

```

2569 \def\@item[#1]{%
2570   \if@noparitem
2571     \@donoparitem
2572   \else
2573     \if@inlabel
2574       \indent \par
2575     \fi
2576     \ifhmode
2577       \unskip\unskip \par
2578     \fi
2579     \if@newlist
2580       \if@nobreak
2581         \@nbitem
2582       \else
2583         \addpenalty\@beginparpenalty
2584         \addvspace\@topsep
2585         \addvspace{-\parskip}%
2586       \fi
2587     \else
2588       \addpenalty\@itempenalty
2589       \addvspace\itemsep

```

```

2590 \fi
2591 \global\@inlabeltrue
2592 \fi
2593 \everypar{%
2594 \@minipagefalse
2595 \global\@newlistfalse
2596 \if@inlabel
2597 \global\@inlabelfalse
2598 {\setbox\z@\lastbox
2599 \ifvoid\z@
2600 \kern-\itemindent
2601 \fi}%
2602 \box\@labels
2603 \penalty\z@
2604 \fi
2605 \if@nobreak
2606 \@nobreakfalse
2607 \clubpenalty \@M
2608 \else
2609 \clubpenalty \@clubpenalty
2610 \everypar{\everyparhook}%
2611 \fi
2612 \bxjs@ltj@inhibitglue
2613 \everyparhook}%
2614 \if@noitemarg
2615 \@noitemargfalse
2616 \if@nmbrlist
2617 \refstepcounter\@listctr
2618 \fi
2619 \fi
2620 \sbox\@tempboxa{\makelabel{#1}}}%
2621 \global\setbox\@labels\hbox{%
2622 \unhbox\@labels
2623 \hskip \itemindent
2624 \hskip -\labelwidth
2625 \hskip -\labelsep
2626 \ifdim \wd\@tempboxa >\labelwidth
2627 \box\@tempboxa
2628 \else
2629 \hbox to\labelwidth {\unhbox\@tempboxa}%
2630 \fi
2631 \hskip \labelsep}%
2632 \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。

```

2633 \def\@afterheading{%
2634 \@nobreaktrue
2635 \everypar{%

```

```

2636 \if@nobreak
2637 \@nobreakfalse
2638 \clubpenalty \@M
2639 \if@afterindent \else
2640 {\setbox\z@\lastbox}%
2641 \fi
2642 \else
2643 \clubpenalty \@clubpenalty
2644 \everypar{\everyparhook}%
2645 \fi\everyparhook}}

```

`\@gnewline` についてはちょっと複雑な心境です。もともとの p \LaTeX 2_ε 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。

```

2646 \def\@gnewline #1{%
2647 \ifvmode
2648 \@nolnerr
2649 \else
2650 \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
2651 \jsInhibitGlue \ignorespaces
2652 \fi}

```

11 いろいろなロゴ

\LaTeX 関連のロゴを作り直します。

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

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

※`\小`、`\上小` の制御綴は定義しない。

```

2653 \if@jslogo
2654 \IfFileExists{jslogo.sty}{%
2655 \RequirePackage{jslogo}%
2656 }{%
2657 \ClassWarningNoLine{bxjs@clsname}
2658 {The package 'jslogo' is not installed.\MessageBreak
2659 It is included in the recent release of\MessageBreak
2660 the 'jsclasses' bundle}
2661 }
2662 \fi

```

12 amsmath との衝突の回避

最近の \LaTeX では該当の問題は対処されているので削除。

13 初期設定

■いろいろな語

```
\prepartname
\postpartname 2663 \newcommand{\prepartname}{\if@english Part~\else 第\fi}
                2664 \newcommand{\postpartname}{\if@english\else 部\fi}
\prechaptername
                2665 %<book|report>\newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postchaptername 2666 %<book|report>\newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname 2667 \newcommand{\presectionname}{}% 第
\postsectionname 2668 \newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename 2669 \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename 2670 \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
                2671 \newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname 2672 \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname 2673 \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
                2674 \newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename 2675 \newcommand{\figurename}{\if@english Fig.~\else 図\fi}
                2676 \newcommand{\tablename}{\if@english Table~\else 表\fi}

\appendixname
\abstractname 2677 % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
                2678 \newcommand{\appendixname}{\if@english\else 付録\fi}
                2679 %<!book&!report>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}
```

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

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

※ “新元号” への対応？

```

\today

2680 \@tempswafalse
2681 \if p\jsEngine \@tempswatrue \fi
2682 \if n\jsEngine \@tempswatrue \fi
2683 \if@tempswa \expandafter\@firstoftwo
2684 \else \expandafter\@secondoftwo
2685 \fi
2686 {%
2687 % 欧文 8bitTeX の場合
2688 \newif\ifjsSeireki \jsSeirekitrue
2689 \def\bxjs@decl@Seireki@cmds{%
2690   \def\西暦{\jsSeirekitrue}%
2691   \def\和暦{\jsSeirekifalse}}
2692 \def\Seireki{\jsSeirekitrue}
2693 \def\Wareki{\jsSeirekifalse}
2694 \def\bxjs@if@use@seireki{%
2695   \ifjsSeireki \expandafter\@firstoftwo
2696   \else \expandafter\@secondoftwo \fi}
2697 }{%
2698 \newif\if 西暦 \西暦 true
2699 \def\bxjs@decl@Seireki@cmds{%
2700   \def\西暦{\西暦 true}%
2701   \def\和暦{\西暦 false}}
2702 \def\Seireki{\西暦 true}
2703 \def\Wareki{\西暦 false}
2704 \def\bxjs@if@use@seireki{%
2705   \if 西暦 \expandafter\@firstoftwo
2706   \else \expandafter\@secondoftwo \fi}
2707 }
2708 \bxjs@decl@Seireki@cmds
2709 % \bxjs@unxp
2710 \let\bxjs@unxp\@firstofone
2711 \bxjs@test@engine\unexpanded{\let\bxjs@unxp\unexpanded}
2712 % \bxjs@iai
2713 \if \if p\jsEngine T\else\if n\jsEngine T\else F\fi\fi T
2714   \def\bxjs@iai{\noexpand~}
2715 \else \def\bxjs@iai{}
2716 \fi
2717 % \heisei
2718 \newcount\heisei \heisei\year \advance\heisei-1988\relax
2719 % \today
2720 \edef\bxjs@today{%
2721   \if@english
2722     \ifcase\month\or
2723       January\or February\or March\or April\or May\or June\or
2724       July\or August\or September\or October\or November\or December\fi
2725     \space\number\day, \number\year
2726   \else

```

```

2727 \ifnum\heisei>\@ne
2728 \expandafter\noexpand\expandafter\bxjs@if@use@seireki
2729 \else \expandafter\@firstoftwo
2730 \fi {%
2731 \number\year\bxjs@iai\bxjs@unxp{年}%
2732 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2733 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2734 }{%
2735 \bxjs@unxp{平成}\bxjs@iai\number\heisei\bxjs@iai\bxjs@unxp{年}%
2736 \bxjs@iai\number\month\bxjs@iai\bxjs@unxp{月}%
2737 \bxjs@iai\number\day\bxjs@iai\bxjs@unxp{日}%
2738 }%
2739 \fi}
2740 \let\today\bxjs@today

```

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

```

2741 \AtBeginDocument{%
2742 \ifx\bbl@jpn@Seirekitrue\undefined\else
2743 \bxjs@decl@Seireki@cmds
2744 \g@addto@macro\datejapanese{%
2745 \let\today\bxjs@today}%
2746 \fi}

```

■ハイフネーション例外 TeX のハイフネーションルールの補足です (ペンディング: eng-lish)

```

2747 \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。

```

2748 %<slide>\pagestyle{empty}%
2749 %<article|report>\pagestyle{plain}%
2750 %<book>\pagestyle{headings}%
2751 \pagenumbering{arabic}
2752 \if@twocolumn
2753 \twocolumn
2754 \sloppy
2755 \flushbottom
2756 \else
2757 \onecolumn
2758 \raggedbottom
2759 \fi
2760 %<*/slide>
2761 \renewcommand\familydefault{\sfdefault}
2762 \raggedright
2763 %</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。

```
2764 \catcode`\?=12
2765 \ifx\bxjs@jadriver\relax\else
2766 \input{bxjsja-\bxjs@jadriver.def}
2767 \fi
```

最後に日本語文字のカテゴリコードを元に戻す。

```
2768 \bxjs@restore@jltrcc
2769 %</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。

```
2770 %<*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

```
2771 %<*minimal>
```

```
2772 %% 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.
```

```
2773 \def\DeclareJaTextFontCommand#1#2{%
```

```
2774   \DeclareRobustCommand#1[1]{%
```

```
2775     \relax
```

```
2776     \ifmmode \expandafter\nfss@text \fi
```

```
2777     {#2##1}}%
```

```
2778 }
```

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

```
2779 \def\DeclareJaMathFontCommand#1#2{%
```

```
2780   \DeclareRobustCommand#1[1]{%
```

```
2781     \relax
```

```
2782     \ifmmode\else \non@alpherr{#1\space}\fi
```

```
2783     \nfss@text{\fontfamily\familydefault
```

```
2784       \fontseries{m}\fontshape{n}\selectfont\relax
```

```
2785       #2##1}}%
```

```

2786 }%
2787 }

```

`\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。

```

2788 \long\def\bxjs@@CSsfdefault{\sfdefault}%
2789 \@onlypreamble\bxjs@if@sf@default
2790 \def\bxjs@if@sf@default#1{%
2791   \ifx\familydefault\bxjs@@CSsfdefault#1\fi
2792   \AtBeginDocument{%
2793     \ifx\familydefault\bxjs@@CSsfdefault#1\fi}%
2794 }

```

`\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。

```

2795 \def\jsLetHeadChar#1#2{%
2796   \begingroup
2797   \escapechar=`\ %
2798   \let\bxjs@tmpa={% brace-match-hack
2799   \bxjs@let@hchar@exp#2}%
2800   \endgroup
2801   \let#1\bxjs@g@tmpa}
2802 \def\bxjs@let@hchar@exp{%
2803   \futurelet\@let@token\bxjs@let@hchar@exp@a}
2804 \def\bxjs@let@hchar@exp@a{%
2805   \bxjs@cond@ifcat\noexpand\@let@token\bgroup\fi{% 波括弧
2806     \bxjs@let@hchar@out\let\relax
2807   }\bxjs@cond@ifcat\noexpand\@let@token\@sptoken\fi{% 空白
2808     \bxjs@let@hchar@out\let\space%
2809   }\bxjs@cond@if\noexpand\@let@token\@backslashchar\fi{% バックスラッシュ
2810     \bxjs@let@hchar@out\let\@backslashchar
2811   }\bxjs@let@hchar@exp@b}}}%
2812 \def\bxjs@let@hchar@exp@b#1{%
2813   \expandafter\bxjs@let@hchar@exp@c\string#1?\@nil#1}
2814 \def\bxjs@let@hchar@exp@c#1#2\@nil{%
2815   %\message{<#1#2>}%
2816   \bxjs@cond@if#1\@backslashchar\fi{% 制御綴
2817     \bxjs@cond\expandafter\ifx\noexpand\@let@token\@let@token\fi{%
2818       \bxjs@let@hchar@out\let\relax
2819     }{%else
2820       \expandafter\bxjs@let@hchar@exp
2821     }%
2822   }{%else
2823     \bxjs@let@hchar@chr#1%

```

```

2824   }}
2825 \def\bxjs@let@hchar@chr#1{%
2826   \bxjs@let@hchar@out\def{{#1}}}
2827 \def\bxjs@let@hchar@out#1#2{%
2828   \global#1\bxjs@g@tmpa#2\relax
2829   \toks@{\bgroup}% skip to right brace

UTF-8 のバイト列を扱うコード。

2830 \chardef\bxjs@let@hchar@csta=128
2831 \chardef\bxjs@let@hchar@cstb=192
2832 \chardef\bxjs@let@hchar@cstc=224
2833 \chardef\bxjs@let@hchar@cstd=240
2834 \chardef\bxjs@let@hchar@cste=248
2835 \let\bxjs@let@hchar@chr@ue@a\bxjs@let@hchar@chr
2836 \def\bxjs@let@hchar@chr@ue#1{%
2837   \@tempcnta=#1\relax
2838   \%message{\the\@tempcnta}%
2839   \bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@csta\fi{%
2840     \bxjs@let@hchar@chr@ue@a#1%
2841   }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstb\fi{%
2842     \bxjs@let@hchar@out\let\relax
2843   }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstc\fi{%
2844     \bxjs@let@hchar@chr@ue@b
2845   }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cstd\fi{%
2846     \bxjs@let@hchar@chr@ue@c
2847   }\{\bxjs@cond\ifnum\@tempcnta<\bxjs@let@hchar@cste\fi{%
2848     \bxjs@let@hchar@chr@ue@d
2849   }\%else
2850     \bxjs@let@hchar@out\let\relax
2851   }}}}}
2852 \def\bxjs@let@hchar@chr@ue@a#1{%
2853   \bxjs@let@hchar@out\def{{#1}}}
2854 \def\bxjs@let@hchar@chr@ue@b#1#2{%
2855   \bxjs@let@hchar@out\def{{#1#2}}}
2856 \def\bxjs@let@hchar@chr@ue@c#1#2#3{%
2857   \bxjs@let@hchar@out\def{{#1#2#3}}}
2858 \def\bxjs@let@hchar@chr@ue@d#1#2#3#4{%
2859   \bxjs@let@hchar@out\def{{#1#2#3#4}}}

```

B.2 (u)pTeX 用の設定

```
2860 \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。

```

2861 \def\bxjs@let@hchar@chr@pp#1{%
2862   \expandafter\bxjs@let@hchar@chr@pp@a\meaning#1\relax#1}

```

```

2863 \def\bxjs@let@hchar@chr@pp@a#1#2\relax#3{%
2864 %\message{(#1)}%
2865 \bxjs@cond\if#1t\fi{%
2866 \bxjs@let@hchar@chr@ue#3%
2867 }{%else
2868 \bxjs@let@hchar@out\def{#3}%
2869 }}
2870 \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。

```

2871 \edef\jsc@JYn{\ifjsWithupTeX JY2\else JY1\fi}
2872 \edef\jsc@JTn{\ifjsWithupTeX JT2\else JT1\fi}
2873 \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。

```

2874 \@onlypreamble\bxjs@declarefontshape
2875 \ifjsWithupTeX
2876 \def\bxjs@declarefontshape{%
2877 \DeclareFontShape{JY2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-h}{}%
2878 \DeclareFontShape{JY2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-h}{}%
2879 \DeclareFontShape{JT2}{mc}{m}{n}{<->s*[\bxjs@scale]upjpnrm-v}{}%
2880 \DeclareFontShape{JT2}{gt}{m}{n}{<->s*[\bxjs@scale]upjpngt-v}{}%
2881 }
2882 \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。

```

2883 \else
2884 \def\bxjs@declarefontshape{%
2885 \DeclareFontShape{JY1}{mc}{m}{n}{<->s*[\bxjs@scale]jis}{}%
2886 \DeclareFontShape{JY1}{gt}{m}{n}{<->s*[\bxjs@scale]jisg}{}%
2887 \DeclareFontShape{JT1}{mc}{m}{n}{<->s*[\bxjs@scale]tmin10}{}%
2888 \DeclareFontShape{JT1}{gt}{m}{n}{<->s*[\bxjs@scale]tgoth10}{}%
2889 }
2890 \def\bxjs@sizereference{jis}
2891 \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。

```

2892 \def\bxjs@tmpa#1/#2/#3/#4/#5\relax{%

```

```

2893 \def\bxjs@y{#5}}
2894 \ifjsWithpTeXng \def\bxjs@y{10}%
2895 \else
2896 \expandafter\expandafter\expandafter\bxjs@tmpa
2897 \expandafter\string\the\jfont\relax
2898 \fi
2899 \@for\bxjs@x:={\jsc@JYn/mc/m/n,\jsc@JYn/gt/m/n,%
2900             \jsc@JTn/mc/m/n,\jsc@JTn/gt/m/n}\do
2901 {\expandafter\let\csname\bxjs@x/10\endcsname=\@undefined
2902 \expandafter\let\csname\bxjs@x/\bxjs@y\endcsname=\@undefined}

```

■和文フォントスケールの補正 実、pTeX の標準的な和文フォント（JFM のこと、例えば jis）では、指定された `\jsScale`（この値を s とする）をそのまま使って定義すると期待通りの大きさにならない。これらの JFM では 1zw の大きさが指定されたサイズではなく既にスケール（この値を f とする；jis では 0.962216 倍）が掛けられた値になっているからである。そのため、ここでは s/f を求めてその値をマクロ `\bxjs@scale` に保存する。

```

2903 \begingroup
2904 % 参照用フォント (\bxjs@sizereference) の全角空白の幅を取得
2905 \font\bxjs@tmpa=\bxjs@sizereference\space at 10pt
2906 \setbox\z@\hbox{\bxjs@tmpa\char\jis"2121\relax}
2907 % 幅が丁度 10pt なら補正は不要
2908 \ifdim\wd\z@=10pt
2909 \global\let\bxjs@scale\jsScale
2910 \else
2911 %  $(10*s)/(10*f)$  として計算、\bxjs@invscale は BXJS で定義
2912 \edef\bxjs@tmpa{\strip@pt\wd\z@}
2913 \@tempdima=10pt \@tempdima=\jsScale\@tempdima
2914 \bxjs@invscale\@tempdima\bxjs@tmpa
2915 \xdef\bxjs@scale{\strip@pt\@tempdima}
2916 \fi
2917 \endgroup
2918 %\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。

```

2919 \bxjs@declarefontshape

```

フォント代替の明示的定義。

```

2920 \DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2921 \DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}
2922 \DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{-}
2923 \DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{-}
2924 \DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{-}
2925 \DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{-}
2926 \DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{-}
2927 \DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{-}
2928 \DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{-}

```

```

2929 \DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{ }
2930 \DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{ }
2931 \DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{ }
2932 \DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{ }
2933 \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。

```

2934 \DeclareRobustCommand\rmfamily
2935   {\not@math@alphabet\rmfamily\mathrm
2936     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
2937 \DeclareRobustCommand\sffamily
2938   {\not@math@alphabet\sffamily\mathsf
2939     \romanfamily\sfddefault\kanjifamily\gtdefault\selectfont}
2940 \DeclareRobustCommand\ttfamily
2941   {\not@math@alphabet\ttfamily\mathtt
2942     \romanfamily\ttddefault\kanjifamily\gtdefault\selectfont}
2943 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
2944 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
2945 \bxjs@if@sf@default{%
2946   \renewcommand\kanjifamilydefault{\gtdefault}}

```

念のため。

```

2947 \selectfont

```

■パラメタの設定

```

2948 \prebreakpenalty\jis"2147=10000
2949 \postbreakpenalty\jis"2148=10000
2950 \prebreakpenalty\jis"2149=10000
2951 \inhibitxspcode`!=1
2952 \inhibitxspcode`ㄱ=2
2953 \xspcode`+=3
2954 \xspcode`\%=3

```

"80～"FF の範囲の \spcode を 3 に変更。

```

2955 \@tempcnta="80 \@whilenum\@tempcnta<"100 \do{%
2956   \xspcode\@tempcnta=3\advance\@tempcnta\@ne}

```

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

```

2957 \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。

```

2958 \begingroup
2959 \catcode`\!=0

```

\bxjs@ptex@dir 現在の組方向： t=縦、y=横、?=その他。


```

2960 \gdef\bxjs@ptex@dir{%
2961   !iftdir t%
2962   !else!ifydir y%
2963   !else ?%
2964   !fi!fi}

```

新版の p_TE_X で脚注番号の周囲の空きが過大になる現象への対処。

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

```

2965 % 古い \@makefnmark の定義
2966 \long\def\bxjs@tmpa{\hbox{%
2967   !ifydir \@textsuperscript{\normalfont\@thefnmark}%
2968   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}}
2969 \ifx\@makefnmark\bxjs@tmpa
2970 \long\gdef\@makefnmark{%
2971   !ifydir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
2972   !else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}!fi}
2973 \fi
2974 \endgroup

```

B.3 pdf_TE_X 用の処理

```

2975 \else\ifx p\jsEngine
2976 \let\bxjs@let@hchar@chr\bxjs@let@hchar@chr@ue
2977 \@onlypreamble\bxjs@cjk@loaded
2978 \def\bxjs@cjk@loaded{%
2979   \def\@footnotemark{%
2980     \leavevmode
2981     \ifhmode
2982       \edef\@x@sf{\the\spacefactor}%
2983       \ifdim\lastkern>\z@\ifdim\lastkern<5sp\relax
2984         \unkern\unkern
2985         \ifdim\lastskip>\z@ \unskip \fi
2986       \fi\fi
2987       \nobreak
2988     \fi
2989     \@makefnmark
2990     \ifhmode \spacefactor\@x@sf \fi
2991     \relax}%
2992 \let\bxjs@cjk@loaded\relax
2993 }
2994 \AtBeginDocument{%
2995   \@ifpackageloaded{CJK}{%
2996     \bxjs@cjk@loaded
2997   }{}%
2998 }

```

B.4 XeTeX 用の処理

```
2999 \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。

```
3000 \def\bxjs@let@hchar@chr#1{%
3001   \@tempcnta`#1\relax \divide\@tempcnta"800\relax
3002   \bxjs@cond\ifnum\@tempcnta=27 \fi{%
3003     \bxjs@let@hchar@chr@xe
3004   }\bxjs@let@hchar@out\def{{#1}}}}
3005 \def\bxjs@let@hchar@chr@xe#1{%
3006   \lccode`0=`#1\relax
3007   \lowercase{\bxjs@let@hchar@out\def{{0}}}}
```

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

```
3008 \ifx\XeTeXgenerateactualtext\@undefined\else
3009   \def\bxjs@do@precisetext{%
3010     \XeTeXgenerateactualtext=\@ne}
3011 \fi
```

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

```
3012 \@onlypreamble\bxjs@do@simplejasetup
3013 \def\bxjs@do@simplejasetup{%
3014   \ifnum\XeTeXinterchartokenstate>\z@
3015   \else\ifnum\strcmp{\the\XeTeXlinebreakskip}{\the\z@}=\z@
3016     \jsSimpleJaSetup
3017     \ClassInfo\bxjs@clsname
3018     {'\string\jsSimpleJaSetup' is applied\@gobble}%
3019   \fi\fi}
```

`\jsSimpleJaSetup` 日本語出力用の超簡易的な設定。

```
3020 \newcommand*{\jsSimpleJaSetup}{%
3021   \XeTeXlinebreaklocale "ja"\relax
3022   \XeTeXlinebreakskip=0pt plus 1pt minus 0.1pt\relax
3023   \XeTeXlinebreakpenalty=0\relax}
```

B.5 後処理（エンジン共通）

```
3024 \fi\fi\fi
```

`simplejasetup` オプションの処理。

```
3025 \ifx\bxjs@do@simplejasetup\@undefined\else
3026   \AtBeginDocument{%
3027     \ifbxjs@simplejasetup
3028       \bxjs@do@simplejasetup
3029     \fi}
3030 \fi
```

`precisetext` オプションの処理。

```

3031 \ifbxjs@precisetext
3032   \ifx\bxjs@do@precisetext\@undefined
3033     \ClassWarning\bxjs@clsname
3034     {The current engine does not support the\MessageBreak
3035     'precisetext' option\@gobble}
3036   \else
3037     \bxjs@do@precisetext
3038   \fi
3039 \fi

```

■fancyhdr 対策 fancyhdr オプションの値が true であり、かつ fancyhdr が使用された場合に以下の対策を行う。

- デフォルトの書式設定に含まれる“二文字フォント命令”を除去する。
- bxjsbook において、ヘッダ・フッタの横幅を \fullwidth に変える。

```

3040 \ifbxjs@fancyhdr

```

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

```

3041 \@onlypreamble\bxjs@adjust@fancyhdr
3042 \def\bxjs@adjust@fancyhdr{%

```

ヘッダ・フッタの要素の書式について、それが既定のままであれば、“二文字フォント命令”を除去したものに置き換える。

※和文なので \sl は無い方がよいはず。

```

3043 \def\bxjs@tmpa{\fancyplain{}{\sl\rightmark}\strut}%
3044 \def\bxjs@tmpb{\fancyplain{}{\rightmark}\strut}%
3045 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3046 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3047 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3048 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3049 \def\bxjs@tmpa{\fancyplain{}{\sl\leftmark}\strut}%
3050 \def\bxjs@tmpb{\fancyplain{}{\leftmark}\strut}%
3051 \ifx\f@ncyelh\bxjs@tmpa \global\let\f@ncyelh\bxjs@tmpb \fi
3052 \ifx\f@ncyerh\bxjs@tmpa \global\let\f@ncyerh\bxjs@tmpb \fi
3053 \ifx\f@ncyolh\bxjs@tmpa \global\let\f@ncyolh\bxjs@tmpb \fi
3054 \ifx\f@ncyorh\bxjs@tmpa \global\let\f@ncyorh\bxjs@tmpb \fi
3055 \def\bxjs@tmpa{\rm\thepage\strut}%
3056 \def\bxjs@tmpb{\thepage\strut}%
3057 \ifx\f@ncyecf\bxjs@tmpa \global\let\f@ncyecf\bxjs@tmpb \fi
3058 \ifx\f@ncyocf\bxjs@tmpa \global\let\f@ncyocf\bxjs@tmpb \fi

```

\fullwidth が（定義済で）\textwidth よりも大きい場合、ヘッダ・フッタの横幅を \fullwidth に合わせる。

```

3059 \ifx\fullwidth\@undefined\else \ifdim\textwidth<\fullwidth
3060   \setlength{\@tempdima}{\fullwidth-\textwidth}%
3061   \edef\bxjs@tmpa{\noexpand\fancyhfoffset[EL,OR]{\the\@tempdima}%
3062   }\bxjs@tmpa

```

```

3063 \fi\fi
3064 \PackageInfo\bxjs@clsname
3065 {Patch to fancyhdr is applied\@gobble}}

```

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

```

3066 \def\bxjs@pagestyle@hook{%
3067 \ifpackageloaded{fancyhdr}{%
3068 \bxjs@adjust@fancyhdr
3069 \global\let\bxjs@adjust@fancyhdr\relax
3070 }{}}

```

`\pagestyle` にフックを入れ込む。

```

3071 \let\bxjs@org@pagestyle\pagestyle
3072 \def\pagestyle{%
3073 \bxjs@pagestyle@hook \bxjs@org@pagestyle}

```

`begin-document` フック。

※これ以降に `fancyhdr` が読み込まれることはあり得ない。

```

3074 \AtBeginDocument{%
3075 \bxjs@pagestyle@hook
3076 \global\let\bxjs@pagestyle@hook\relax}

```

終わり。

```

3077 \fi

```

以上で終わり。

```

3078 %</minimal>

```

付録 C 和文ドライバ : standard

標準のドライバ。

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

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

コマンドで使う“ファミリー”と同じにすることを想定する。特別な値として、`auto` は `kanji-config-updmap` で現在指定されているファミリーを表す。

C.1 共通処理 (1)

まず `minimal` ドライバを読み込む。

```
3079 %<*standard>
3080 %% このファイルは日本語文字を含みます
3081 \input{bxjsja-minimal.def}
```

`simplejasetup` は `standard` では無効になる。

```
3082 \bxjs@simplejasetupfalse
```

■共通命令の実装 `\jQ` 等の「単位」系の共通命令を実装する。まず ε -TeX 拡張が使えるか 検査する。

```
3083 \ifjsWithTeX
```

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

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

```
\jH3084 \@tempdima=0.25mm
3085 \protected\edef\jQ{\dimexpr\the\@tempdima\relax}
3086 \let\jH\jQ
```

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

```
\trueH3087 \ifjsc@mag
3088 \@tempdimb=\jsBaseFontSize\relax
3089 \edef\bxjs@tmpa{\strip@pt\@tempdimb}%
3090 \@tempdima=2.5mm
3091 \bxjs@invscale\@tempdima\bxjs@tmpa
3092 \protected\edef\trueQ{\dimexpr\the\@tempdima\relax}
3093 \@tempdima=10pt
3094 \bxjs@invscale\@tempdima\bxjs@tmpa
3095 \protected\edef\bxjs@truept{\dimexpr\the\@tempdima\relax}
3096 \else \let\trueQ\jQ \let\bxjs@truept\p@
3097 \fi
3098 \let\trueH\trueQ
```

`\ascQ` `\ascQ` は `\trueQ` を和文スケール値で割った値。例えば、`\fontsize{12\ascQ}{16\trueH}` `\ascpt` とすると、和文が 12Q になる。

同様に、`\ascpt` は `truept` を和文スケールで割った値。

```
3099 \@tempdima\trueQ \bxjs@invscale\@tempdima\jsScale
3100 \protected\edef\ascQ{\dimexpr\the\@tempdima\relax}
3101 \@tempdima\bxjs@truept \bxjs@invscale\@tempdima\jsScale
3102 \protected\edef\ascpt{\dimexpr\the\@tempdima\relax}
3103 \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。

```
3104 \def\bxjs@kanjiskip{0pt}
```

`\setkanjiskip` 和文間空白の量を設定する。

```
3105 \newcommand*\setkanjiskip[1]{%
```

```
3106   \edef\bxjs@kanjiskip{#1}%
```

```
3107   \bxjs@reset@kanjiskip}
```

`\getkanjiskip` 和文間空白の量を表すテキストに展開する。

```
3108 \newcommand*\getkanjiskip{%
```

```
3109   \bxjs@kanjiskip}
```

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

```
3110 \newif\ifbxjs@kanjiskip@enabled \bxjs@kanjiskip@enabledtrue
```

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

```
\bxjs@disable@kanjiskip 3111 \bxjs@robust@def\bxjs@enable@kanjiskip{%
```

```
3112   \bxjs@kanjiskip@enabledtrue
```

```
3113   \bxjs@reset@kanjiskip}
```

```
3114 \bxjs@robust@def\bxjs@disable@kanjiskip{%
```

```
3115   \bxjs@kanjiskip@enabledfalse
```

```
3116   \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。

```
3117 \bxjs@robust@def\bxjs@reset@kanjiskip{%
```

```
3118   \ifbxjs@kanjiskip@enabled
```

```
3119     \setlength{\@tempskipa}{\bxjs@kanjiskip}%
```

```
3120   \else \@tempskipa\z@
```

```
3121   \fi
```

```
3122   \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 3123 \def\bxjs@xkanjiskip{0pt}
```

```
\getxkanjiskip 3124 \newcommand*\setxkanjiskip[1]{%
```

```
3125   \edef\bxjs@xkanjiskip{#1}%
```

```
\ifbxjs@xkanjiskip@enabled 3126   \bxjs@reset@xkanjiskip}
```

```
\bxjs@enable@xkanjiskip 3127 \newcommand*\getxkanjiskip{%
```

```
\bxjs@disable@xkanjiskip 3128   \bxjs@xkanjiskip}
```

```
3129 \newif\ifbxjs@xkanjiskip@enabled \bxjs@xkanjiskip@enabledtrue
```

```
\bxjs@reset@xkanjiskip 3130 \bxjs@robust@def\bxjs@enable@xkanjiskip{%
```

```
3131   \bxjs@xkanjiskip@enabledtrue
```

```
3132   \bxjs@reset@xkanjiskip}
```

```
3133 \bxjs@robust@def\bxjs@disable@xkanjiskip{%
```

```
3134   \bxjs@xkanjiskip@enabledfalse
```

```
3135   \bxjs@reset@xkanjiskip}
```

```
3136 \bxjs@robust@def\bxjs@reset@xkanjiskip{%
```

```

3137 \ifbxjs@xkanjiskip@enabled
3138   \setlength{\@tempskipa}{\bxjs@xkanjiskip}%
3139 \else \@tempskipa\z@
3140 \fi
3141 \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。

```

3142 \g@addto@macro\jsResetDimen{%
3143   \bxjs@reset@kanjiskip
3144   \bxjs@reset@xkanjiskip}
3145 \let\bxjs@apply@kanjiskip\relax
3146 \let\bxjs@apply@xkanjiskip\relax

```

■和文フォント指定の扱い

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

```

3147 \@onlypreamble\bxjs@adjust@jafont
3148 \def\bxjs@adjust@jafont#1{%
3149   \ifx\jsJaFont\bxjs@@auto
3150     \bxjs@get@kanjiEmbed
3151     \ifx\bxjs@kanjiEmbed\relax
3152       \let\bxjs@tmpa\@empty
3153     \else
3154       \let\bxjs@tmpa\bxjs@kanjiEmbed
3155     \fi
3156   \else
3157     \let\bxjs@tmpa\jsJaFont
3158   \fi
3159   \if f#1\ifx\bxjs@tmpa\bxjs@@noEmbed
3160     \ClassWarningNoLine\bxjs@clsname
3161       {Option 'jafont=noEmbed' is ignored, because it is\MessageBreak
3162        not available on the current situation}%
3163     \let\bxjs@tmpa\@empty
3164   \fi\fi
3165 }
3166 \def\bxjs@@auto{auto}
3167 \def\bxjs@@noEmbed{noEmbed}

```

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

```

3168 \let\bxjs@kanjiEmbed\relax

```

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

```

3169 \@onlypreamble\bxjs@get@kanjiEmbed
3170 \def\bxjs@get@kanjiEmbed{%

```

```

3171 \begingroup\setbox\z@=\hbox{%
3172   \global\let\bxjs@g@tmpa\relax
3173   \endlinechar\m@ne
3174   \let\do\@makeother\dospecials
3175   \catcode32=10 \catcode12=10 %form-feed
3176   \let\bxjs@tmpa\@empty
3177   \openin\@inputcheck="|kpsewhich updmap.cfg"\relax
3178   \ifeof\@inputcheck\else
3179     \read\@inputcheck to\bxjs@tmpa
3180     \closein\@inputcheck
3181   \fi
3182   \ifx\bxjs@tmpa\@empty\else
3183     \openin\@inputcheck="\bxjs@tmpa"\relax
3184     \@tempswatrue
3185     \loop\if@tempswa
3186       \read\@inputcheck to\bxjs@tmpa
3187       \expandafter\bxjs@get@ke@a\bxjs@tmpa\@nil kanjiEmbed \@nil\@nnil
3188       \ifx\bxjs@tmpa\relax\else
3189         \global\let\bxjs@g@tmpa\bxjs@tmpa
3190         \@tempswafalse
3191       \fi
3192       \ifeof\@inputcheck \@tempswafalse \fi
3193     \repeat
3194   \fi
3195 } \endgroup
3196 \let\bxjs@kanjiEmbed\bxjs@g@tmpa
3197 }
3198 \@onlypreamble\bxjs@get@ke@a
3199 \def\bxjs@get@ke@a#1kanjiEmbed #2\@nil#3\@nnil{%
3200   \ifx$#1$\def\bxjs@tmpa{#2}%
3201   \else \let\bxjs@tmpa\relax
3202   \fi}

```

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

```

3203 \newcommand*\jachar[1]{%
3204   \begingroup
3205     \jsLetHeadChar\bxjs@tmpa{#1}%
3206     \ifx\bxjs@tmpa\relax
3207       \ClassWarningNoLine\bxjs@clsname
3208         {Illegal argument given to \string\jachar}%
3209     \else
3210       \expandafter\bxjs@jachar\expandafter{\bxjs@tmpa}%
3211     \fi
3212   \endgroup}

```

`\jsJaChar` を `\jachar` と等価にする。

```

3213 \let\jsJaChar\jachar

```


下請けの `\bxjs@jachar` の実装はエンジンにより異なる。

```
3214 \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。

```
3215 \PassOptionsToPackage{setpagesize=false}{hyperref}
```

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

```
3216 \@onlypreamble\bxjs@fix@hyperref@unicode
3217 \def\bxjs@fix@hyperref@unicode#1{%
3218   \PassOptionsToPackage{bxjs/hook=#1}{hyperref}%
3219   \@namedef{KV@Hyp@bxjs/hook}##1{%
3220     \KV@Hyp@unicode{##1}%
3221     \def\KV@Hyp@unicode####1{%
3222       \expandafter\ifx\csname if##1\expandafter\endcsname
3223       \csname if####1\endcsname\else
3224       \ClassWarningNoLine{bxjs@clsname
3225       {Blcoked hyperref option 'unicode=####1'}}%
3226     \fi
3227   }%
3228 }%
3229 }
```

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

```
3230 \@onlypreamble\bxjs@urgent@special
3231 \def\bxjs@urgent@special#1{%
3232   \AtBeginDvi{\special{#1}}%
3233   \AtBeginDocument{%
3234     \@ifpackageloaded{atbegshi}{%
3235       \begingroup
3236       \toks\z@{\special{#1}}%
3237       \toks\tw@\expandafter{\AtBegShi@HookFirst}%
3238       \xdef\AtBegShi@HookFirst{\the\toks@\the\toks\tw@}%
3239     \endgroup
3240   }{}%
3241 }%
3242 }
```

C.2 pTeX 用設定

```
3243 \if j\jsEngine
```

■共通命令の実装

```
3244 \def\bxjs@apply@kanjiskip{%
3245   \kanjiskip\@tempskipa}
3246 \def\bxjs@apply@xkanjiskip{%
3247   \xkanjiskip\@tempskipa}
```

`\jaJaChar` のサブマクロ。

```
3248 \def\bxjs@jachar#1{%
3249   \bxjs@jachar@a#1...\@nil}
3250 \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。

```
3251   \ifx.#2#1%
```

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

```
3252   \else\ifx.#3%
3253     \@tempcnta`#1 \multiply\@tempcnta64
3254     \advance\@tempcnta`#2 \advance\@tempcnta-"3080
3255     \bxjs@jachar@b
3256   \else\ifx.#4%
3257     \@tempcnta`#1 \multiply\@tempcnta64
3258     \advance\@tempcnta`#2 \multiply\@tempcnta64
3259     \advance\@tempcnta`#3 \advance\@tempcnta-"E2080
3260     \bxjs@jachar@b
3261   \else
3262     \@tempcnta`#1 \multiply\@tempcnta64
3263     \advance\@tempcnta`#2 \multiply\@tempcnta64
3264     \advance\@tempcnta`#3 \multiply\@tempcnta64
3265     \advance\@tempcnta`#4 \advance\@tempcnta-"3C82080
3266     \bxjs@jachar@b
3267   \fi\fi\fi}
```

符号値が `\@tempcnta` の和文文字を出力する処理。

```
3268 \ifjsWithupTeX
3269   \def\bxjs@jachar@b{\kchar\@tempcnta}
3270 \else
3271   \def\bxjs@jachar@b{%
3272     \ifx\bxUInt\@undefined\else
3273       \bxUInt{\@tempcnta}%
3274     \fi}
3275 \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。

```
3276 \let\bxjs@tmpa\jsJaFont
3277 \ifx\bxjs@tmpa\bxjs@@auto
3278   \let\bxjs@tmpa\@empty
3279 \else\ifx\bxjs@tmpa\bxjs@@noEmbed
3280   \def\bxjs@tmpa{noembed}
3281 \fi\fi
3282 \ifx\jsJaFont\@empty\else
3283   \edef\bxjs@nxt{%
3284     \noexpand\RequirePackage[\jsJaFont]
```

```

3285 {pxchfon}[2010/05/12]]% v0.5
3286 \bxjs@nxt
3287 \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……。)

```

3288 \begingroup
3289 \global\let\@gtempa\relax
3290 \catcode`\|=0 \catcode`\|=12
3291 |def|bxjs@check#1|@nil{%
3292 |bxjs@check@a#1|@nil\RequirePackage|@nnil}%
3293 |def|bxjs@check@a#1\RequirePackage#2|@nnil{%
3294 |ifx$#1$|bxjs@check@b#2|@nil keyval|@nnil |fi}%
3295 |catcode`\|=0 \catcode`\|=12
3296 \def|bxjs@check@b#1keyval#2\@nnil{%
3297 \ifx$#2$\else
3298 \xdef\@gtempa{%
3299 \noexpand\PassOptionsToPackage{scale=\jsScale}{otf}}%
3300 \fi}
3301 \@firstofone{%
3302 \catcode10=12 \endlinechar\m@ne
3303 \let\do\@makeother \dospecials \catcode32=10
3304 \openin\@inputcheck=otf.sty\relax
3305 \@tempswatrue
3306 \loop\if@tempswa
3307 \ifeof\@inputcheck \@tempswafalse \fi
3308 \if@tempswa
3309 \read\@inputcheck to\bxjs@line
3310 \expandafter\bxjs@check\bxjs@line\@nil
3311 \fi
3312 \repeat
3313 \closein\@inputcheck
3314 \endgroup}
3315 \@gtempa

```

■hyperref 対策 unicode にしてはいけない。

```

3316 \bxjs@fix@hyperref@unicode{false}

```

tounicode special 命令を出力する。

```

3317 \if \ifx\bxjs@driver@given\bxjs@driver@@dvipdfmx T%
3318 \else\ifjsWithpTeXng T\else F\fi\fi T%
3319 \ifnum\jis"2121="A1A1 %euc
3320 \bxjs@urgent@special{pdf:tounicode EUC-UCS2}
3321 \else\ifnum\jis"2121="8140 %sjis
3322 \bxjs@urgent@special{pdf:tounicode 90ms-RKSJ-UCS2}
3323 \else\ifnum\jis"2121="3000 %uptex

```

```

3324 \ifbxjs@bigcode
3325 \bxjs@urgent@special{pdf:tounicode UTF8-UTF16}
3326 \PassOptionsToPackage{bigcode}{pxjahyper}
3327 \else
3328 \bxjs@urgent@special{pdf:tounicode UTF8-UCS2}
3329 \fi
3330 \fi\fi\fi
3331 \let\bxToUnicodeSpecialDone=t
3332 \fi

```

■microtype 対策

```

3333 \namedef{ver@microtype.sty}{2000/01/01}
3334 \newcommand*{\UseMicrotypeSet}[2][{}]{

```

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

```

3335 \ifx f\bxjs@enablejfam\else
3336 \@enablejfamtrue
3337 \fi

```

実際に和文用の数式ファミリーの設定を行う。

```

3338 \if@enablejfam
3339 \DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
3340 \DeclareSymbolFontAlphabet{\mathmc}{mincho}
3341 \SetSymbolFont{mincho}{bold}{\jsc@JYn}{gt}{m}{n}
3342 \jfam\symmincho
3343 \DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
3344 \AtBeginDocument{%
3345 \ifx\reDeclareMathAlphabet\undefined\else
3346 \reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}%
3347 \reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}%
3348 \reDeclareMathAlphabet{\mathsf}{\@mathsf}{\@mathgt}%
3349 \fi}
3350 \fi

```

C.3 pdfTeX 用設定 : CJK + bxcjkatype

```

3351 \else\if p\jsEngine

```

■bxcjkatype パッケージの読み込み `\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。

```

3352 \bxjs@adjust@jafont{f}
3353 \edef\bxjs@nxt{%
3354 \noexpand\RequirePackage[
3355 \ifx\bxjs@tmpa\@empty\else \bxjs@tmpa,\fi
3356 whole,autotilde]{bxcjkatype}[2013/10/15]}% v0.2c

```

```

3357 \bxjs@nxt
3358 \bxjs@cjk@loaded

```

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

※取りあえず固定はしない。

```

3359 \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。

※ bxcjkjatype を whole 付きで使っていることが前提。

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

```

3360 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3361 \begingroup
3362   \CJK@input{UTF8.bdg}
3363 \endgroup
3364 \g@addto@macro\pdfstringdefPreHook{%
3365   \@nameuse{CJK@UTF8Binding}%
3366 }
3367 \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。

```

3368 \ifx\bxcjkjatypeHyperrefPatchDone\@undefined
3369 \g@addto@macro\pdfstringdefPreHook{%
3370   \ifx~\bxjs@@CJKtilde
3371     \let\bxjs@org@LetUnexpandableSpace\HyPsd@LetUnexpandableSpace
3372     \let\HyPsd@LetUnexpandableSpace\bxjs@LetUnexpandableSpace
3373     \let~\@empty
3374   \fi
3375 }
3376 \def\bxjs@@CJKtilde{\CJKecglue\ignorespaces}
3377 \def\bxjs@@tildecmd{~}
3378 \def\bxjs@LetUnexpandableSpace#1{%
3379   \def\bxjs@tmpa{#1}\ifx\bxjs@tmpa\bxjs@@tildecmd\else
3380     \bxjs@org@LetUnexpandableSpace#1%
3381   \fi}
3382 \fi

```

■共通命令の実装

```

3383 \newskip\jsKanjiSkip
3384 \newskip\jsXKanjiSkip
3385 \ifx\CJKecglue\@undefined
3386   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3387 \fi
3388 \let\autospacing\bxjs@enable@kanjiskip
3389 \let\noautospacing\bxjs@disable@kanjiskip
3390 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3391 \def\bxjs@apply@kanjiskip{%

```

```

3392 \jsKanjiSkip\@tempskipa
3393 \let\CJKglue\bxjs@CJKglue}
3394 \let\autoxspacing\bxjs@enable@xkanjiskip
3395 \let\noautoxspacing\bxjs@disable@xkanjiskip
3396 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3397 \def\bxjs@apply@xkanjiskip{%
3398   \jsXKanjiSkip\@tempskipa
3399   \let\CJKecglue\bxjs@CJKecglue}

```

\jachar のサブマクロの実装。

```

3400 \def\bxjs@jachar#1{%
3401   \CJKforced{#1}}

```

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

```

3402 \ifx t\bxjs@enablejfam
3403   \ClassWarningNoLine\bxjs@clsname
3404     {You cannot use 'enablejfam=true', since the\MessageBreak
3405      CJK package does not support Japanese math}
3406 \fi

```

C.4 Xe_{La}TeX 用設定：xeCJK + zxjatype

```

3407 \else\if x\jsEngine

```

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

```

3408 \RequirePackage{zxjatype}
3409 \PassOptionsToPackage{no-math}{fontspec}%!
3410 \PassOptionsToPackage{xetex}{graphicx}%!
3411 \PassOptionsToPackage{xetex}{graphics}%!
3412 \ifx\zxJaFamilyName\undefined
3413   \ClassError\bxjs@clsname
3414     {xeCJK or zxjatype is too old}\@ehc
3415 \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。

```

3416 \bxjs@adjust@jafont{f}
3417 \ifx\bxjs@tmpa\empty
3418   \setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}
3419   \setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}
3420 \else
3421   \edef\bxjs@nxt{%
3422     \noexpand\RequirePackage[\bxjs@tmpa]{
3423       {zxjafont}[2013/01/28]}% v0.2a
3424   \bxjs@nxt
3425 \fi

```

■hyperref 対策 unicode オプションの指定に関する話。

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

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

※取りあえず固定はしない。

```
3426 \ifnum\strcmp{\the\XeTeXversion\XeTeXrevision}{0.99992}>\m@ne
3427   \PassOptionsToPackage{unicode}{hyperref}
3428 \fi
```

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

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

```
3429 \let\jsInhibitGlueAtParTop\@empty
```

■共通命令の実装

```
3430 \newskip\jsKanjiSkip
3431 \newskip\jsXKanjiSkip
3432 \ifx\CJKecglue\@undefined
3433   \def\CJKtilde{\CJK@global\def~{\CJKecglue\ignorespaces}}
3434 \fi
3435 \let\autospacing\bxjs@enable@kanjiskip
3436 \let\noautospacing\bxjs@disable@kanjiskip
3437 \protected\def\bxjs@CJKglue{\hskip\jsKanjiSkip}
3438 \def\bxjs@apply@kanjiskip{%
3439   \jsKanjiSkip\@tempskipa
3440   \xeCJKsetup{CJKglue={\bxjs@CJKglue}}}
3441 \let\autoxspacing\bxjs@enable@xkanjiskip
3442 \let\noautoxspacing\bxjs@disable@xkanjiskip
3443 \protected\def\bxjs@CJKecglue{\hskip\jsXKanjiSkip}
3444 \def\bxjs@apply@xkanjiskip{%
3445   \jsXKanjiSkip\@tempskipa
3446   \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。

```
3447 \ifx\mcfamily\@undefined
3448   \protected\def\mcfamily{\CJKfamily{\CJKrmddefault}}
3449   \protected\def\gtfamily{\CJKfamily{\CJKsfdefault}}
```

```
3450 \fi
```

`\jachar` のサブマクロの実装。

```
3451 \def\bxjs@jachar#1{%
3452   \xeCJKDeclareCharClass{CJK}{\#1}\relax
3453   #1}
```

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

```
3454 \ifx t\bxjs@enablejfam
3455   \@enablejfamtrue
3456 \fi
```

実際に和文用の数式ファミリーの設定を行う。

※ FIXME: 要検討。

```
3457 \if@enablejfam
3458   \xeCJKsetup{CJKmath=true}
3459 \fi
```

C.5 Lua_T_EX 用設定 : Lua_T_EX-ja

```
3460 \else\if l\jsEngine
```

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

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

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

```
3461 \let\zw\undefined
3462 \RequirePackage{luatexja}
3463 \RequirePackage{luatexja-fontspec}
```

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

```
3464 \ExplSyntaxOn
3465 \fp_gset:Nn \g_ljtj_fontspec_scale_fp { \jsScale }
3466 \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)。

```
3467 \bxjs@adjust@jafont{t}
3468 \ifx\bxjs@tmpa\bxjs@@noEmbed
3469   \def\bxjs@tmpa{noembed}
3470 \fi
```



```

3471 \ifx\bxjs@tmpa\@empty
3472   \defaultjfontfeatures{ Kerning=Off }
3473   \setmainfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexMincho}
3474   \setsansfont[BoldFont=IPAexGothic,JFM=ujis]{IPAexGothic}
3475 \else
3476   \edef\bxjs@nxt{%
3477     \noexpand\RequirePackage[\bxjs@tmpa]
3478       {luatexja-preset}}%
3479   \bxjs@nxt
3480 \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。

```

3481 \DeclareRobustCommand\rmfamily
3482   {\not@math@alphabet\rmfamily\mathrm
3483     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
3484 \DeclareRobustCommand\sffamily
3485   {\not@math@alphabet\sffamily\mathsf
3486     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
3487 \DeclareRobustCommand\ttfamily
3488   {\not@math@alphabet\ttfamily\mathtt
3489     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}
3490 \AtBeginDocument{%
3491   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
3492   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}%
3493   \reDeclareMathAlphabet{\mathsf}{\mathsf}{\mathgt}}%
3494 \bxjs@if@sf@default{%
3495   \renewcommand\kanjifamilydefault{\gtdefault}}

```

■和文パラメタの設定

```

3496 % 次の3つは既定値の通り
3497 %\ltjsetparameter{prebreakpenalty={` ,10000}}
3498 %\ltjsetparameter{postbreakpenalty={` ",10000}}
3499 %\ltjsetparameter{prebreakpenalty={` ” ,10000}}
3500 \ltjsetparameter{jaxspmde={` ! ,1}}
3501 \ltjsetparameter{jaxspmde={` 〒 ,2}}
3502 \ltjsetparameter{alxspmde={` + ,3}}
3503 \ltjsetparameter{alxspmde={` \% ,3}}

```

■段落頭でのグルー挿入禁止

```

3504 \protected\def\@inhibitglue{%
3505   \directlua{%
3506     luatexja.jfmglue.create_beginpar_node()}}
3507 \let\bxjs@ltj@inhibitglue\@inhibitglue
3508 \let\@@inhibitglue\@undefined

```

■hyperref 対策 unicode にするべき。

```

3509 \bxjs@fix@hyperref@unicode{true}

```

■共通命令の実装

```
3510 \protected\def\autospacing{%
3511   \ltjsetparameter{autospacing=true}}
3512 \protected\def\noautospacing{%
3513   \ltjsetparameter{autospacing=false}}
3514 \protected\def\autoxspacing{%
3515   \ltjsetparameter{autoxspacing=true}}
3516 \protected\def\noautoxspacing{%
3517   \ltjsetparameter{autoxspacing=false}}
3518 \def\bxjs@apply@kanjiskip{%
3519   \ltjsetparameter{kanjiskip={\@tempskipa}}}}
3520 \def\bxjs@apply@xkanjiskip{%
3521   \ltjsetparameter{xkanjiskip={\@tempskipa}}}}
```

\jachar のサブマクロの実装。

```
3522 \def\bxjs@jachar#1{%
3523   \ltjjachar`#1\relax}
```

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

```
3524 \ifx f\bxjs@enablejfam
3525   \ClassWarningNoLine\bxjs@clsname
3526     {You cannot use 'enablejfam=false', since the\MessageBreak
3527     LuaTeX-jd always provides Japanese math families}
3528 \fi
```

C.6 共通処理 (2)

```
3529 \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 3530 \DeclareJaTextFontCommand{\textmc}{\mcfamily}
3531 \DeclareJaTextFontCommand{\textgt}{\gtfamily}
```

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

\mathgt バックの定義を行う。

```
3532 \ifx\mathmc\@undefined
3533   \DeclareJaMathFontCommand{\mathmc}{\mcfamily}
3534   \DeclareJaMathFontCommand{\mathgt}{\gtfamily}
3535 \fi
```

■和文・和欧文間空白の初期値

```
3536 \setkanjiskip{0pt plus.1\jsZw minus.01\jsZw}
3537 \ifx\jsDocClass\jsSlide \setxkanjiskip{0.1em}
3538 \else \setxkanjiskip{0.25em plus 0.15em minus 0.06em}
3539 \fi
```

以上で終わり。

```
3540 %</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。

```
3541 %<*modern>
```

```
3542 \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。

```
3543 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
```

```
3544 \def\encodingdefault{T1}%
```

```
3545 \input{t1enc.def}%
```

```
3546 \fontencoding\encodingdefault\selectfont
```

```
3547 \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。

```
3548 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi=\z@
```

```
3549 \renewcommand{\rmdefault}{lmr}
```

```
3550 \renewcommand{\sfdefault}{lms}
```

```
3551 \renewcommand{\ttdefault}{lmtt}
```

```
3552 \fi
```

大型演算子用の数式フォントの設定。

※ `amsmath` パッケージと同等にする。

```
3553 \DeclareFontShape{OMX}{cmex}{m}{n}{%
```

```
3554   <-7.5>cmex7<7.5-8.5>cmex8%
```

```
3555   <8.5-9.5>cmex9<9.5->cmex10}{}%
```

```
3556 \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。

```
3557 \def\cmex@opt{10}
```

D.2 fixltx2e 読込

※ `fixltx2e` 廃止前の L^AT_EX カーネルの場合。

```
3558 \ifx\@IncludeInRelease\@undefined
```

```
3559 \RequirePackage{fixltx2e}
```

```
3560 \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。

```
3561 \RequirePackage{bxjscjkat}
```

D.4 完了

おしまい。

```
3562 %</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。

```
3563 %<*pandoc>
```

```
3564 \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。

```
3565 \@onlypreamble\bxjs@set@dupload@proc
```

```
3566 \def\bxjs@set@dupload@proc#1{%
```

```
3567   \expandafter\bxjs@set@dupload@proc@a\csname bxjs@dlp/#1\endcsname}
```

```
3568 \@onlypreamble\bxjs@set@dupload@proc@a
```

```
3569 \def\bxjs@set@dupload@proc@a#1{%
```

```
3570   \@onlypreamble#1\def#1##1}
```

`\@if@options` `\@if@options` の再定義。

```
3571 \@onlypreamble\bxjs@org@if@options
```

```
3572 \let\bxjs@org@if@options\@if@options
```

```
3573 \newif\ifbxjs@dlp
```

```
3574 \def\@if@options#1#2#3{%
```

```
3575   \bxjs@dlpfalse
```

```
3576   \def\bxjs@tmpa{#1}\def\bxjs@tmpb{\@currentx}%
```

```
3577   \ifx\bxjs@tmpa\bxjs@tmpb
```

```
3578     \expandafter\ifx\csname bxjs@dlp/#2.#1\endcsname\relax\else
```

```
3579       \bxjs@dlptrue \fi
```

```
3580 \fi
```

```

3581 \ifbxjs@dlp \expandafter\bxjs@do@upload@proc
3582 \else \expandafter\bxjs@org@if@ptions
3583 \fi {#1}{#2}{#3}}
3584 \AtBeginDocument{%
3585 \let\@if@ptions\bxjs@org@if@ptions}
3586 \@onlypreamble\bxjs@do@upload@proc
3587 \def\bxjs@do@upload@proc#1#2#3{%
3588 \csname bxjs@dlp/#2.#1\endcsname{#3}%
3589 \@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。

```

3590 \def\bxjs@mark@as@loaded#1{%
3591 \expandafter\bxjs@mal@a\csname ver@#1\endcsname{#1}}
3592 \def\bxjs@mal@a#1#2{%
3593 \ifx#1\relax
3594 \def#1{2001/01/01}%
3595 \ClassInfo\bxjs@clsname
3596 {File '#2' marked as loaded\@gobble}%
3597 \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。

```

3598 \ifnum0\if x\jsEngine1\fi\if l\jsEngine1\fi>0

```

Polyglossia について。

```

3599 \bxjs@mark@as@loaded{polyglossia.sty}
3600 \bxjs@set@upload@proc{polyglossia.sty}{%
3601 \ClassWarning\bxjs@clsname
3602 {Loading of polyglossia is blocked}}
3603 \ifx\setmainlanguage\@undefined
3604 \newcommand*\setmainlanguage[2][]{%
3605 \newcommand*\setotherlanguage[2][]{%
3606 \ifcat_#2_\else
3607 \expandafter\let\csname #2\endcsname\@empty
3608 \expandafter\let\csname end#2\endcsname\@empty
3609 \expandafter\let\csname text#2\endcsname\@firstofone
3610 \fi}
3611 \newcommand*\setotherlanguages[2][]{%
3612 \@for\bxjs@tmpa:={#2}\do{%
3613 \setotherlangauge{\bxjs@tmpa}}}
3614 \fi
3615 \else

```

Babel について。

```

3616 \bxjs@mark@as@loaded{babel.sty}
3617 \bxjs@set@dupload@proc{babel.sty}{%
3618   \ClassWarning\bxjs@clsname
3619     {Loading of babel is blocked}}
3620 \let\foreignlanguage\@secondoftwo
3621 \let\otherlanguage\@gobble
3622 \let\endotherlanguage\@empty
3623 \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。

```

3624 \bxjs@set@dupload@proc{geometry.sty}{%
3625   \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。

```

3626 \if 1\jsEngine
3627   \bxjs@mark@as@loaded{xeCJK.sty}
3628   \providecommand*\setCJKmainfont{\setmainfont}
3629 \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。

```

3630 \ifx\@IncludeInRelease\@undefined\else
3631   \bxjs@mark@as@loaded{fixltx2e.sty}
3632 \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。)

```

3633 \if j\jsEngine
3634   \bxjs@mark@as@loaded{cmap.sty}
3635 \fi

```

E.7 microtype パッケージ

警告が多すぎなので消す。

```
3636 \PassOptionsToPackage{verbose=silent}{microtype}
```

E.8 完了

おしまい。

```
3637 %</pandoc>
```

和文ドライバ実装はここまで。

```
3638 %</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。

```
3639 %<*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

```
3640 %<*compat>
```

```
3641 \def\bxac@pkgname{bxjscompat}
```

\bxjx@engine エンジンの種別。

```
3642 \let\bxac@engine=n
```

```
3643 \def\bxac@do#1#2{%
```

```
3644   \edef\bxac@tmpa{\string#1}%
```

```
3645   \edef\bxac@tmpb{\meaning#1}%
```

```
3646   \ifx\bxac@tmpa\bxac@tmpb #2\fi}
```

```
3647 \bxac@do\XeTeXversion{\let\bxac@engine=x}
```

```
3648 \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。

```
3649 \ifx\jsAtEndOfClass\@undefined
```

```
3650   \let\bxac@delayed@if@bxjs\@firstofone
```

```
3651 \else \let\bxac@delayed@if@bxjs\jsAtEndOfClass
```

```
3652 \fi
```

`\ImposeOldLuaTeXBehavior` ムニャムニャ。

```
\RevokeOldLuaTeXBehavior 3653 \newif\ifbxac@in@old@behavior
3654 \let\ImposeOldLuaTeXBehavior\relax
3655 \let\RevokeOldLuaTeXBehavior\relax
```

G.2 XeTeX 部分

```
3656 \ifx x\bxac@engine

XeTeX 文字クラスのムニャムニャ。

3657 \@onlypreamble\bxac@adjust@charclass
3658 \bxac@delayed@if@bxjs{%
3659   \@ifpackageloaded{xCJK}{}{%else
3660     \ifx\XeTeX@alloc@intercharclass\undefined\else
3661       \ifnum\XeTeX@alloc@intercharclass=\z@
3662         \PackageInfo\bxac@pkgname
3663           {Setting up interchar class for CJK...\@gobble}%
3664         \InputIfFileExists{load-unicode-xetex-classes.tex}{%
3665           \XeTeX@alloc@intercharclass=3
3666         }{%else
3667           \PackageWarning\bxac@pkgname
3668             {Cannot find file 'load-unicode-xetex-classes.tex'%
3669             \@gobble}%
3670         }%
3671       \fi\fi
3672     \ifnum\XeTeXcharclass"3041=\z@
3673       \PackageInfo\bxac@pkgname
3674         {Adjusting interchar class for CJK...\@gobble}%
3675       \@for\bxac@x:={%
3676         3041,3043,3045,3047,3049,3063,3083,3085,3087,308E,%
3677         3095,3096,30A1,30A3,30A5,30A7,30A9,30C3,30E3,30E5,%
3678         30E7,30EE,30F5,30F6,30FC,31F0,31F1,31F2,31F3,31F4,%
3679         31F5,31F6,31F7,31F8,31F9,31FA,31FB,31FC,31FD,31FE,%
3680         31FF%
3681       }\do{\XeTeXcharclass"\bxac@x=\@ne}%
3682     \fi
3683   }%
3684 }

以上。
3685 \fi
```

G.3 LuaTeX 部分

```
3686 \ifx l\bxac@engine

ムニャムニャ。

3687 \unless\ifnum\luatexversion<80 \ifnum\luatexversion<85
3688   \chardef\pdfTeXversion=200
3689   \def\pdfTeXrevision{0}
```



```

3690 \let\pdfptxheader\luatexheader
3691 \fi\fi

\ImposeOldLuaTeXBehavior ムニヤムニヤ。

\RevokeOldLuaTeXBehavior 3692 \begingroup\expandafter\expandafter\expandafter\endgroup
3693 \expandafter\ifx\csname outputmode\endcsname\relax\else
3694 \def\bxac@ob@list{%
3695   \do{\let}\pdfoutput{\outputmode}%
3696   \do{\let}\pdfpagewidth{\pagewidth}%
3697   \do{\let}\pdfpageheight{\pageheight}%
3698   \do{\protected\edef}\pdfhorigin{\pdfvariable horigin}}%
3699   \do{\protected\edef}\pdfvorigin{\pdfvariable vorigin}}}
3700 \def\bxac@ob@do#1#2{\begingroup
3701   \expandafter\bxac@ob@do@a\csname bxac@\string#2\endcsname{#1}#2}
3702 \def\bxac@ob@do@a#1#2#3#4{\endgroup
3703   \ifbxac@in@old@behavior \let#1#3\relax #2#3#4\relax
3704   \else \let#3#1\relax \let#1\@undefined
3705   \fi}
3706 \protected\def\ImposeOldLuaTeXBehavior{%
3707   \unless\ifbxac@in@old@behavior
3708     \bxac@in@old@behaviortrue
3709     \let\do\bxac@ob@do \bxac@ob@list
3710   \fi}
3711 \protected\def\RevokeOldLuaTeXBehavior{%
3712   \ifbxac@in@old@behavior
3713     \bxac@in@old@behaviorfalse
3714     \let\do\bxac@ob@do \bxac@ob@list
3715   \fi}
3716 \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。

```

3717 \ifnum\luatexversion>64 \directlua{
3718   local function range(cs, ce, cc, ff)
3719     if ff or not tex.getcatcode(cs) == cc then
3720       local setcc = tex.setcatcode
3721       for c = cs, ce do setcc(c, cc) end
3722     end
3723   end
3724   range(0x3400, 0x4DB5, 11, false)
3725   range(0x4DB5, 0x4DBF, 11, true)
3726   range(0x4E00, 0x9FCC, 11, false)
3727   range(0x9FCD, 0x9FFF, 11, true)
3728   range(0xAC00, 0xD7A3, 11, false)
3729   range(0x20000, 0x2A6D6, 11, false)
3730   range(0x2A6D7, 0x2A6FF, 11, true)
3731   range(0x2A700, 0x2B734, 11, false)
3732   range(0x2B735, 0x2B73F, 11, true)
3733   range(0x2B740, 0x2B81D, 11, false)
3734   range(0x2B81E, 0x2B81F, 11, true)

```

```

3735     range(0x2B820, 0x2CEA1, 11, false)
3736     range(0x2CEA2, 0x2FFFD, 11, true)
3737 } \fi

```

以上。

```

3738 \fi

```

G.4 完了

おしまい。

```

3739 %</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

```

3740 %<*cjkat>
3741 \def\bxjx@pkgnam{bxjscjkat}
3742 \newcount\bxjx@cna

\bxjx@engine エンジンの種別。

3743 \let\bxjx@engine=n
3744 \def\bxjx@do#1#2{%
3745   \edef\bxjx@tmpa{\string#1}%
3746   \edef\bxjx@tmpb{\meaning#1}%
3747   \ifx\bxjx@tmpa\bxjx@tmpb #2\fi}
3748 \bxjx@do\kanjiskip{\let\bxjx@engine=j}
3749 \bxjx@do\enablecjktoken{\let\bxjx@engine=u}
3750 \bxjx@do\XeTeXversion{\let\bxjx@engine=x}
3751 \bxjx@do\pdftexversion{\let\bxjx@engine=p}
3752 \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。

```

3753 \def\bxjx@do#1#2{%
3754   \if#1\bxjx@engine
3755     \@ifpackageloaded{#2}{\}%else
3756     \PackageError\bxjx@pkgnam
3757       {Package '#2' must be loaded}%
3758       {Package loading is aborted.\MessageBreak\@ehc}%
3759     \endinput}
3760 \fi}
3761 \bxjx@do{p}{bxckjatype}
3762 \bxjx@do{x}{xeCJK}
3763 \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。

```
3764 \ifx\TextOrMath\@undefined
3765   \RequirePackage{fixltx2e}
3766 \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。

```
3767 \if u\bxjx@engine
3768 \@for\bxjx@x:={%
3769 0080,0100,0180,0250,02B0,0300,0500,0530,0590,0600,%
3770 0700,0750,0780,07C0,0800,0840,08A0,0900,0980,0A00,%
3771 0A80,0B00,0B80,0C00,0C80,0D00,0D80,0E00,0E80,0F00,%
3772 1000,10A0,1200,1380,13A0,1400,1680,16A0,1700,1720,%
3773 1740,1760,1780,1800,18B0,1900,1950,1980,19E0,1A00,%
3774 1A20,1AB0,1B00,1B80,1BC0,1C00,1C50,1CC0,1CD0,1D00,%
3775 1D80,1DC0,1E00,2440,27C0,27F0,2800,2A00,2C00,2C60,%
3776 2C80,2D00,2D30,2D80,2DE0,2E00,4DC0,A4D0,A500,A640,%
3777 A6A0,A700,A720,A800,A830,A840,A880,A8E0,A900,A930,%
3778 A980,A9E0,AA00,AA60,AA80,AAE0,AB00,AB30,AB70,ABC0,%
3779 D800,DB80,DC00,E000,FB00,FB50,FE00,FE70,%
3780 10000,10080,10100,10140,10190,101D0,10280,102A0,%
3781 102E0,10300,10330,10350,10380,103A0,10400,10450,%
3782 10480,10500,10530,10600,10800,10840,10860,10880,%
3783 108E0,10900,10920,10980,109A0,10A00,10A60,10A80,%
3784 10AC0,10B00,10B40,10B60,10B80,10C00,10C80,10E60,%
3785 11000,11080,110D0,11100,11150,11180,111E0,11200,%
3786 11280,112B0,11300,11480,11580,11600,11680,11700,%
3787 118A0,11AC0,12000,12400,12480,13000,14400,16800,%
3788 16A40,16AD0,16B00,16F00,1BC00,1BCA0,1D000,1D100,%
3789 1D200,1D300,1D360,1D400,1D800,1E800,1EE00,1F000,%
3790 1F030,1F0A0,1F100,1F200,1F300,1F600,1F650,1F680,%
3791 1F700,1F780,1F800,1F900,E0000,F0000,100000%
3792 }\do{\kcatcode"\bxjx@x=15 }
3793 \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。

```
3794 \def\bxjx@grkcyr@list{%
3795 \do{0391}{LGR}{\textAlpha}{A}%           % GR. C. L. ALPHA
3796 \do{0392}{LGR}{\textBeta}{B}%           % GR. C. L. BETA
```

3797	\do{0393}{LGR}{\textGamma}{\Gamma}%	% GR. C. L. GAMMA
3798	\do{0394}{LGR}{\textDelta}{\Delta}%	% GR. C. L. DELTA
3799	\do{0395}{LGR}{\textEpsilon}{\epsilon}%	% GR. C. L. EPSILON
3800	\do{0396}{LGR}{\textZeta}{\zeta}%	% GR. C. L. ZETA
3801	\do{0397}{LGR}{\textEta}{\eta}%	% GR. C. L. ETA
3802	\do{0398}{LGR}{\textTheta}{\theta}%	% GR. C. L. THETA
3803	\do{0399}{LGR}{\textIota}{\iota}%	% GR. C. L. IOTA
3804	\do{039A}{LGR}{\textKappa}{\kappa}%	% GR. C. L. KAPPA
3805	\do{039B}{LGR}{\textLambda}{\lambda}%	% GR. C. L. LAMDA
3806	\do{039C}{LGR}{\textMu}{\mu}%	% GR. C. L. MU
3807	\do{039D}{LGR}{\textNu}{\nu}%	% GR. C. L. NU
3808	\do{039E}{LGR}{\textXi}{\xi}%	% GR. C. L. XI
3809	\do{039F}{LGR}{\textOmicron}{\omicron}%	% GR. C. L. OMICRON
3810	\do{03A0}{LGR}{\textPi}{\pi}%	% GR. C. L. PI
3811	\do{03A1}{LGR}{\textRho}{\rho}%	% GR. C. L. RHO
3812	\do{03A3}{LGR}{\textSigma}{\sigma}%	% GR. C. L. SIGMA
3813	\do{03A4}{LGR}{\textTau}{\tau}%	% GR. C. L. TAU
3814	\do{03A5}{LGR}{\textUpsilon}{\Upsilon}%	% GR. C. L. UPSILON
3815	\do{03A6}{LGR}{\textPhi}{\phi}%	% GR. C. L. PHI
3816	\do{03A7}{LGR}{\textChi}{\chi}%	% GR. C. L. CHI
3817	\do{03A8}{LGR}{\textPsi}{\psi}%	% GR. C. L. PSI
3818	\do{03A9}{LGR}{\textOmega}{\omega}%	% GR. C. L. OMEGA
3819	\do{03B1}{LGR}{\textalpha}{\alpha}%	% GR. S. L. ALPHA
3820	\do{03B2}{LGR}{\textbeta}{\beta}%	% GR. S. L. BETA
3821	\do{03B3}{LGR}{\textgamma}{\gamma}%	% GR. S. L. GAMMA
3822	\do{03B4}{LGR}{\textdelta}{\delta}%	% GR. S. L. DELTA
3823	\do{03B5}{LGR}{\textepsilon}{\epsilon}%	% GR. S. L. EPSILON
3824	\do{03B6}{LGR}{\textzeta}{\zeta}%	% GR. S. L. ZETA
3825	\do{03B7}{LGR}{\texteta}{\eta}%	% GR. S. L. ETA
3826	\do{03B8}{LGR}{\texttheta}{\theta}%	% GR. S. L. THETA
3827	\do{03B9}{LGR}{\textiota}{\iota}%	% GR. S. L. IOTA
3828	\do{03BA}{LGR}{\textkappa}{\kappa}%	% GR. S. L. KAPPA
3829	\do{03BB}{LGR}{\textlambda}{\lambda}%	% GR. S. L. LAMDA
3830	\do{03BC}{LGR}{\textmu}{\mu}%	% GR. S. L. MU
3831	\do{03BD}{LGR}{\textnu}{\nu}%	% GR. S. L. NU
3832	\do{03BE}{LGR}{\textxi}{\xi}%	% GR. S. L. XI
3833	\do{03BF}{LGR}{\textomicron}{\omicron}%	% GR. S. L. OMICRON
3834	\do{03C0}{LGR}{\textpi}{\pi}%	% GR. S. L. PI
3835	\do{03C1}{LGR}{\textrho}{\rho}%	% GR. S. L. RHO
3836	\do{03C2}{LGR}{\textvarsigma}{\varsigma}%	% GR. S. L. FINAL SIGMA
3837	\do{03C3}{LGR}{\textsigma}{\sigma}%	% GR. S. L. SIGMA
3838	\do{03C4}{LGR}{\texttau}{\tau}%	% GR. S. L. TAU
3839	\do{03C5}{LGR}{\textupsilon}{\upsilon}%	% GR. S. L. UPSILON
3840	\do{03C6}{LGR}{\textphi}{\phi}%	% GR. S. L. PHI
3841	\do{03C7}{LGR}{\textchi}{\chi}%	% GR. S. L. CHI
3842	\do{03C8}{LGR}{\textpsi}{\psi}%	% GR. S. L. PSI
3843	\do{03C9}{LGR}{\textomega}{\omega}%	% GR. S. L. OMEGA
3844	\do{0401}{T2A}{\CYR0}{\cyr0}%	% CY. C. L. IO
3845	\do{0410}{T2A}{\CYRA}{\cyra}%	% CY. C. L. A

3846 \do{0411}{T2A}{\CYRB}{}%	% CY. C. L. BE
3847 \do{0412}{T2A}{\CYRV}{}%	% CY. C. L. VE
3848 \do{0413}{T2A}{\CYRG}{}%	% CY. C. L. GHE
3849 \do{0414}{T2A}{\CYRD}{}%	% CY. C. L. DE
3850 \do{0415}{T2A}{\CYRE}{}%	% CY. C. L. IE
3851 \do{0416}{T2A}{\CYRZH}{}%	% CY. C. L. ZHE
3852 \do{0417}{T2A}{\CYRZ}{}%	% CY. C. L. ZE
3853 \do{0418}{T2A}{\CYRI}{}%	% CY. C. L. I
3854 \do{0419}{T2A}{\CYRISHRT}{}%	% CY. C. L. SHORT I
3855 \do{041A}{T2A}{\CYRK}{}%	% CY. C. L. KA
3856 \do{041B}{T2A}{\CYRL}{}%	% CY. C. L. EL
3857 \do{041C}{T2A}{\CYRM}{}%	% CY. C. L. EM
3858 \do{041D}{T2A}{\CYRN}{}%	% CY. C. L. EN
3859 \do{041E}{T2A}{\CYRO}{}%	% CY. C. L. O
3860 \do{041F}{T2A}{\CYRP}{}%	% CY. C. L. PE
3861 \do{0420}{T2A}{\CYRR}{}%	% CY. C. L. ER
3862 \do{0421}{T2A}{\CYRS}{}%	% CY. C. L. ES
3863 \do{0422}{T2A}{\CYRT}{}%	% CY. C. L. TE
3864 \do{0423}{T2A}{\CYRU}{}%	% CY. C. L. U
3865 \do{0424}{T2A}{\CYRF}{}%	% CY. C. L. EF
3866 \do{0425}{T2A}{\CYRH}{}%	% CY. C. L. HA
3867 \do{0426}{T2A}{\CYRC}{}%	% CY. C. L. TSE
3868 \do{0427}{T2A}{\CYRCH}{}%	% CY. C. L. CHE
3869 \do{0428}{T2A}{\CYRSH}{}%	% CY. C. L. SHA
3870 \do{0429}{T2A}{\CYRSHCH}{}%	% CY. C. L. SHCHA
3871 \do{042A}{T2A}{\CYRHRDSN}{}%	% CY. C. L. HARD SIGN
3872 \do{042B}{T2A}{\CYRERY}{}%	% CY. C. L. YERU
3873 \do{042C}{T2A}{\CYRSFTSN}{}%	% CY. C. L. SOFT SIGN
3874 \do{042D}{T2A}{\CYREREV}{}%	% CY. C. L. E
3875 \do{042E}{T2A}{\CYRYU}{}%	% CY. C. L. YU
3876 \do{042F}{T2A}{\CYRYA}{}%	% CY. C. L. YA
3877 \do{0430}{T2A}{\cyra}{}%	% CY. S. L. A
3878 \do{0431}{T2A}{\cyrb}{}%	% CY. S. L. BE
3879 \do{0432}{T2A}{\cyrv}{}%	% CY. S. L. VE
3880 \do{0433}{T2A}{\cyrg}{}%	% CY. S. L. GHE
3881 \do{0434}{T2A}{\cyrd}{}%	% CY. S. L. DE
3882 \do{0435}{T2A}{\cyre}{}%	% CY. S. L. IE
3883 \do{0436}{T2A}{\cyrzh}{}%	% CY. S. L. ZHE
3884 \do{0437}{T2A}{\cyrz}{}%	% CY. S. L. ZE
3885 \do{0438}{T2A}{\cyri}{}%	% CY. S. L. I
3886 \do{0439}{T2A}{\cyrishrt}{}%	% CY. S. L. SHORT I
3887 \do{043A}{T2A}{\cyrk}{}%	% CY. S. L. KA
3888 \do{043B}{T2A}{\cyrl}{}%	% CY. S. L. EL
3889 \do{043C}{T2A}{\cyrm}{}%	% CY. S. L. EM
3890 \do{043D}{T2A}{\cyrn}{}%	% CY. S. L. EN
3891 \do{043E}{T2A}{\cyro}{}%	% CY. S. L. O
3892 \do{043F}{T2A}{\cyrp}{}%	% CY. S. L. PE
3893 \do{0440}{T2A}{\cyrr}{}%	% CY. S. L. ER
3894 \do{0441}{T2A}{\cyrs}{}%	% CY. S. L. ES

```

3895 \do{0442}{T2A}{\cyrt}{}%           % CY. S. L. TE
3896 \do{0443}{T2A}{\cyru}{}%           % CY. S. L. U
3897 \do{0444}{T2A}{\cyrf}{}%           % CY. S. L. EF
3898 \do{0445}{T2A}{\cyrh}{}%           % CY. S. L. HA
3899 \do{0446}{T2A}{\cyrc}{}%           % CY. S. L. TSE
3900 \do{0447}{T2A}{\cyrch}{}%          % CY. S. L. CHE
3901 \do{0448}{T2A}{\cyrsh}{}%          % CY. S. L. SHA
3902 \do{0449}{T2A}{\cyrshch}{}%        % CY. S. L. SHCHA
3903 \do{044A}{T2A}{\cyrhrdsn}{}%       % CY. S. L. HARD SIGN
3904 \do{044B}{T2A}{\cyrery}{}%         % CY. S. L. YERU
3905 \do{044C}{T2A}{\cyrstsn}{}%       % CY. S. L. SOFT SIGN
3906 \do{044D}{T2A}{\cyrrrev}{}%       % CY. S. L. E
3907 \do{044E}{T2A}{\cyrYu}{}%         % CY. S. L. YU
3908 \do{044F}{T2A}{\cyrYa}{}%         % CY. S. L. YA
3909 \do{0451}{T2A}{\cyrYo}{}%         % CY. S. L. IO
3910 \do{00A7}{TS1}{\textsection}{\mathsection}% SECTION SYMBOL
3911 \do{00A8}{TS1}{\textasciidieresis}{}% % DIAERESIS
3912 \do{00B0}{TS1}{\textdegree}{\mathdegree}% % DEGREE SIGN
3913 \do{00B1}{TS1}{\textpm}{\pm}%      % PLUS-MINUS SIGN
3914 \do{00B4}{TS1}{\textasciicute}{}%  % ACUTE ACCENT
3915 \do{00B6}{TS1}{\textparagraph}{\mathparagraph}% PILCROW SIGN
3916 \do{00D7}{TS1}{\texttimes}{\times}% % MULTIPLICATION SIGN
3917 \do{00F7}{TS1}{\textdiv}{\div}%    % DIVISION SIGN
3918 }
3919 \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。

```
3920 \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。

```

3921 \newcommand*\greekasCJK{%
3922   \bxjx@gcc@cjkltrue}
3923 \newcommand*\nogreekasCJK{%
3924   \bxjx@gcc@cjklfalse}

```

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

```

3925 \def\bxjx@do#1\relax{%
3926   \def\bxjx@fake@grkl##1##2{%
3927     \expandafter\bxjx@fake@grkl@a\meaning##2#1\@nil{##1}{##2}}%
3928   \def\bxjx@fake@grkl@a##1#1##2\@nil##3##4{%
3929     \ifx\##1\%
3930       \bxjx@canta##4\divide\bxjx@canta\@cclvi
3931       \multiply\bxjx@canta\@cclvi \advance\bxjx@canta`##3\relax
3932       \mathchar\bxjx@canta
3933     \else ##3\fi}
3934 } \expandafter\bxjx@do\string\mathchar\relax

```

■pdfLaTeX・upLaTeX の場合

```
3935 \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。

```
3936 \@ifpackageloaded{inputenc}{\%else
3937   \RequirePackage[utf8]{inputenc}}
3938 \def\bxjx@tmpa{utf8}
3939 \ifx\bxjx@tmpa\inputencdoingname
3940   \PackageWarningNoLine\bxjx@pkgname
3941     {Input encoding changed to utf8}%
3942   \inputencoding{utf8}%
3943 \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。

```
3944 \if u\bxjx@engine
3945 \kcatcode"0370=15
3946 \kcatcode"0400=15
3947 \kcatcode"0500=15
3948 \fi
```

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

```
3949 \def\do#1{%
3950   \@tempcnta="#1\relax
3951   \@tempcntb=\@tempcnta \divide\@tempcntb256
3952   \expandafter\let\csname bxjx@KCR/\the\@tempcntb\endcsname=t%
3953   \expandafter\bxjx@do@a\csname bxjx@KC/\the\@tempcnta\endcsname{#1}}
3954 \def\bxjx@do@a#1#2#3#4#5{%
3955   \ifx\#5\%
3956     \def\bxjx@tmpa{\@inmathwarn#4}%
3957   \else\ifcat A\noexpand#5%
3958     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
3959       {\ifnum\uccode`#5=`#5\noexpand\Pi\else\noexpand\pi\fi}}%
3960   \else \def\bxjx@tmpa{#5}%
3961   \fi\fi
3962   \def\bxjx@tmpb{\bxjx@do@b{#1}{#2}{#3}{#4}}%
3963   \expandafter\bxjx@tmpb\expandafter{\bxjx@tmpa}}
3964 \if u\bxjx@engine
3965 % {\bxjx@KC/NN}{XXXX}{ENC}{\textCS}{\mathCS}
3966 \def\bxjx@do@b#1#2#3#4#5{%
3967   \kchardef#1=\@tempcnta
3968   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{#1}{#3}{#4}}%
3969   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
3970 \else\if p\bxjx@engine
3971 \def\bxjx@do@b#1#2#3#4#5{%
3972   \mathchardef#1=\@tempcnta
3973   \DeclareTextCommandDefault{#4}{\bxjx@ja@or@not{\UTF{#2}}{#3}{#4}}%
3974   \DeclareUnicodeCharacter{#2}{\TextOrMath{#4}{#5}}%
3975 \fi\fi
3976 \bxjx@grkcyr@list
```

```

3977 \let\bxjx@do@a\undefined
3978 \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。

```

3979 \@onlypreamble\bxjx@org@DeclareUnicodeCharacter
3980 \let\bxjx@org@DeclareUnicodeCharacter\DeclareUnicodeCharacter
3981 \@onlypreamble\bxjx@DeclareUnicodeCharacter
3982 \def\bxjx@DeclareUnicodeCharacter#1#2{%
3983   \count@="#1\relax \bxjx@cnta\count@ \divide\bxjx@cnta256
3984   \expandafter\ifx\csname bxjx@KCR/\the\bxjx@cnta\endcsname\relax
3985     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3986   \else\expandafter\ifx\csname bxjx@KC/\the\count@\endcsname\relax
3987     \bxjx@org@DeclareUnicodeCharacter{#1}{#2}%
3988   \else
3989     \wlog{ \space\space skipped defining Unicode char U+#1}%
3990   \fi\fi}

```

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

```

3991 \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。

```

3992 \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。

```

3993 \else\expandafter\ifx\csname T@#2\endcsname\relax #1%
3994 \else \UseTextSymbol{#2}{#3}%
3995 \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。

```

3996 \begingroup
3997 \toks@{\expandafter{\DeclareFontEncoding@{#1}{#2}{#3}}}
3998 \xdef\next{\def\noexpand\DeclareFontEncoding@##1##2##3{%
3999   \noexpand\bxjx@swap@DUC@cmd
4000   \the\toks@
4001   \noexpand\bxjx@swap@DUC@cmd}}
4002 \endgroup\next
4003 \def\bxjx@swap@DUC@cmd{%
4004   \let\bxjx@tmpa\DeclareUnicodeCharacter
4005   \let\DeclareUnicodeCharacter\bxjx@DeclareUnicodeCharacter
4006   \let\bxjx@DeclareUnicodeCharacter\bxjx@tmpa}

```

以上。

■ Xe_{La}TeX・Lua_{La}TeX の場合

```

4007 \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。

```

4008 \def\do#1{%
4009   \bxjx@cmta="#1\relax
4010   \begingroup
4011     \lccode`~=\bxjx@cmta
4012   \lowercase{\endgroup
4013     \bxjx@do@a{~}}{#1}}
4014 \def\bxjx@do@a#1#2#3#4#5{%
4015   \ifx\#5\\\let\bxjx@tmpa\relax
4016   \else\ifcat A\noexpand#5%
4017     \edef\bxjx@tmpa{\noexpand\bxjx@fake@grk{#5}%
4018       {\ifnum\uccode`#5=#5\noexpand\Pi\else\noexpand\pi\fi}}%
4019   \else \def\bxjx@tmpa{#5}%
4020   \fi\fi
4021   \ifx\bxjx@tmpa\relax\else
4022     \mathcode\bxjx@cmta"8000 \let#1\bxjx@tmpa
4023   \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。

```

4024 \mathchardef\bxjx@tmpa="119
4025 \ifx\bxjx@tmpa\pi \bxjx@grkcyr@list \fi
4026 \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。

```

4027 \if l\bxjx@engine
4028   \protected\def\greekasCJK{%
4029     \bxjx@gcc@cjctrue
4030     \ltjsetparameter{jacharrange={+2, +8}}
4031   \protected\def\nogreekasCJK{%
4032     \bxjx@gcc@cjkfalse
4033     \ltjsetparameter{jacharrange={-2, -8}}
4034   \fi

```

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

```

4035 \if x\bxjx@engine
4036   \protected\def\greekasCJK{%
4037     \bxjx@gcc@cjctrue
4038     \def\do##1##2##3##4{\XeTeXcharclass"##1\@ne}%
4039     \bxjx@grkcyr@list}
4040   \protected\def\nogreekasCJK{%
4041     \bxjx@gcc@cjkfalse
4042     \def\do##1##2##3##4{\XeTeXcharclass"##1\z@}%
4043     \bxjx@grkcyr@list}
4044   \fi

```

以上。

```

4045 \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。

```
4046 \nogreekasCJK
```

H.5 完了

おしまい。

```
4047 %</cjkcat>
```

補助パッケージ実装はここまで。

```
4048 %</anc>
```