BXjscls パッケージ
（BXJS 文書クラス集）
ユーザマニュアル

八登崇之 （Takayuki YATO; aka. “ZR”）

v1.4　[2017/02/03]

目次

注意

BXJS 文書クラスについては、“TEX Wiki”中の記事、[*1]およびそこからたどれるサイトにある情報も併せて参照してほしい。

1 概要

本パッケージに含まれる文書クラス集（以下では **BXJS（文書）クラス**と呼ぶ）は、奥村晴彦氏および“日本語 TEX 開発コミュニティ”により作製された「pLATEX 2ε 新ドキュメントクラス」（以下では **JS（文書）クラス**と呼ぶ）の拡張版に相当する。JS クラスのレイアウトデザインと機能をほぼ踏襲しているが、以下の点で改良が加えられている。

- JS クラスは pLATEX と upLATEX のみをサポートするが、BXJS クラスはこれらに加えて pdfLATEX と XeLATEX と LuaLATEX をサポートしており主要エンジンの全てで使用可能である。
- (u)pLATEX 以外では各々のエンジンの日本語処理パッケージを利用するが、“標準設定”を用いることで、それらのパッケージの設定を書かずに済ませられるので、pLATEX 並に簡単に日本語の文書を書き始めることができる。
- JS クラスでは、フォントのオプティカルサイズを最適にするため、（基底フォントサイズが 10pt 以外の時に）TEX の版面拡大（mag）機能を利用しているが、これが他のパッケージと衝突して不具合を起こすことがある。BXJS クラスでは mag 機能を使う他に別の方式を選べるようにしている。
  ※ JS クラスについても新しい（2016/07/11 以降の）版では同様の機能が提供されている。
- 用紙サイズや基底フォントサイズについて、任意の値を指定することができる。

2 最も基本的な使い方

ここでは、BXJS クラスを“標準設定”（standard 和文ドライバ）で用いる場合について解説する。この場合、`\documentclass` 命令を次のように書く。[*2]

```
\documentclass[⟨エンジン⟩,⟨ドライバ⟩,ja=standard,jafont=⟨フォント指定⟩,⟨他オプション⟩]
{⟨クラス名⟩}
```

- ⟨エンジン⟩ の指定は必須で、実際に使っている「LATEX のコマンド名」を書く。`latex`、`platex`、`uplatex`、`pdflatex`、`xelatex`、`lualatex` が指定できる。
- DVI 出力のエンジンを使う場合は、⟨ドライバ⟩ の指定が必須で、これは実際に使っている「DVI ウェ

[*1] https://texwiki.texjp.org/?BXjscls
[*2] もちろんクラスオプションの順序は任意である。

アの名前」を書く。dvips、dvipdfmx、dviout、xdvi が指定できる。PDF 出力のエンジンの場合は ⟨ドライバ⟩ の指定は不要である。

- “標準設定”を適用するので ja=standard を指定する。（ja の代わりに jadriver と書いてもよい。）
- 既定以外のフォント設定を利用する場合は、⟨フォント指定⟩ にその名前を書く。既定の設定を用いる場合は jafont=... 自体を省略する。
- その他のクラスオプション（a4paper 等）については、多くの場合 JS クラスと同じものが使える。
- BXJS クラスについて、⟨クラス名⟩ は以下のものがある。
  - bxjsarticle： 章のないレポート（jsarticle に相当する）
  - bxjsreport： 章のあるレポート（jsbook＋report に相当する）
  - bxjsbook： 書籍（jsbook に相当する）
  - bxjsslide： スライド（jsarticle＋slide に相当する）

XeLaTeX で bxjsarticle クラスを用いた文書の例を示す。[*3]

```
\documentclass[a4paper,xelatex,ja=standard]{bxjsarticle}
\usepackage[unicode,colorlinks,
    pdftitle={いきなり日本語}]{hyperref}
\title{いきなり日本語}
\author{七篠 権兵衛}
\begin{document}
\maketitle

\section{日本語で{\LaTeX}する}
中身はまだない。

\end{document}
```

以下では各エンジンについて、挙動を少し詳しく説明する。

2.1 pLaTeX の場合

例えば次の設定は：

```
\documentclass[a4paper,platex,dvipdfmx,ja=standard]{bxjsarticle}
```

対応する JS クラスを用いた次の設定と（ほぼ）等価になる[*4]：

```
\documentclass[a4paper,dvipdfmx]{jsarticle}
```

次のように jafont を指定した場合は：

```
\documentclass[a4paper,platex,dvipdfmx,ja=standard,jafont=ms]{bxjsarticle}
```

その値をプリセットオプションとして pxchfon が読み込まれる：

[*3] 組版結果における日付の出力は JS クラスと同様の「2015 年 7 月 3 日」の形式になる。

[*4] すなわち、論理フォントは明朝が jis、ゴシックが jisg が使われる。なお、BXJS では mingoth 等の論理フォント変更のオプションはサポートされていない。

```
\documentclass[a4paper,dvipdfmx]{jsarticle}
\usepackage[ms]{pxchfon}
```

2.2 upLaTeX の場合

例えば次の設定は：

```
\documentclass[a5paper,uplatex,dvipdfmx,ja=standard]{bxjsarticle}
```

次の設定と（ほぼ）等価になる[*5]：

```
\documentclass[uplatex,a4paper,dvipdfmx]{jsarticle}
```

jafont オプションの扱いは pLaTeX の場合と同じである。

2.3 pdfLaTeX の場合

エンジン指定が pdflatex の場合、日本語処理パッケージとして bxcjkjatype（これ自体は内部で CJK パッケージを読み込む）を利用する。

例えば次の設定は：

```
\documentclass[a4paper,pdflatex,ja=standard]{bxjsarticle}
```

次の設定と大体同じであるが、ただし文書レイアウトは article でなく jsarticle とほぼ同じになっている：

```
\documentclass[a4paper]{article}
\usepackage[whole,autotilde]{bxcjkjatype}
```

jafont を指定した場合は：

```
\documentclass[a4paper,pdflatex,ja=standard,jafont=ipaex]{bxjsarticle}
```

その値が bxcjkjatype のフォントプリセットになる。

```
\documentclass[a4paper]{article}
\usepackage[whole,autotilde,ipaex]{bxcjkjatype}
```

※補足：

- 自動的に文書本体が CJK* 環境[*6]で囲まれかつ \CJKtilde が有効な状態になっている。従っていきなり日本語を書き始めることができる。ただし和欧文間空白（四分空き）は手動で ~ を入れる必要がある。[*7]日本語出力の挙動の詳細については bxcjkjatype のマニュアルを参照してほしい。以下に完全な文書ソースの例を示す：

```
\documentclass[a4paper,pdflatex,ja=standard]{bxjsarticle}
\begin{document}
```

[*5] 論理フォントについては、従来のもの（明朝が upjisr-h、ゴシックが upjisg-h）に代わって、BMP 外の文字に対応したもの（明朝が upjpnrm-h、ゴシックが upjpngt-h）を採用した。組み方は従来のものと変わらない。

[*6] CJKspace パッケージが読み込まれた下での CJK* 環境である。

[*7] CJK パッケージには自動で和欧文間空白を入れる機能はない。

```
日本語で~pdf{\LaTeX}~するテスト。
\end{document}
```

- bxcjkjatype パッケージにおけるフォントの既定設定は「Type1 形式の IPAex フォント」（ipaex-type1 パッケージ）である。一方、ipaex プリセットを指定した場合は「TrueType 形式の IPAex フォント」が使われるので、両者の出力は“PDF データとしては”異なる（見かけは同じのはずだが）。[*8]

2.4 XeLaTeX の場合

エンジン指定が xelatex の場合、日本語処理パッケージとして zxjatype （これ自体は内部で xeCJK パッケージを読み込む）を利用する。

例えば次の設定は：

```
\documentclass[a4paper,twocolumn,xelatex,ja=standard]{bxjsarticle}
```

次の設定と大体同じであるが、ただし文書レイアウトは jsarticle とほぼ同じになっている：

```
\documentclass[a4paper,twocolumn]{article}
\usepackage{zxjatype}
\setCJKmainfont[BoldFont=IPAexGothic]{IPAexMincho}% 明朝→ IPAex 明朝
\setCJKsansfont[BoldFont=IPAexGothic]{IPAexGothic}% ゴシック→ IPAex ゴシック
```

jafont を指定した場合は：

```
\documentclass[a4paper,xelatex,ja=standard,jafont=ms]{bxjsarticle}
```

その値が zxjafont のプリセットになる。

```
\usepackage{zxjatype}
\usepackage[ms]{zxjafont}
```

2.5 LuaLaTeX の場合

エンジン指定が lualatex の場合、日本語処理パッケージとして LuaTeX-ja を利用する。

例えば次の設定は：

```
\documentclass[b5paper,9pt,lualatex,ja=standard]{bxjsarticle}
```

次の設定と（ほぼ）等価になる（ただし luatexja-preset は実際には読み込まれない）：

```
\documentclass[b5paper,9pt]{ltjsarticle}
\usepackage{luatexja-fontspec}
\usepackage[ipaex]{luatexja-preset}
```

jafont を指定した場合は：

```
\documentclass[b5paper,lualatex,ja=standard,jafont=ms]{bxjsarticle}
```

[*8] ちなみに、bxcjkjatype には ipaex-type1 というオプションもあるが、この設定と既定設定（オプション無し）も動作は異なる。BXJS クラスが用いるのは既定設定の方である。

次の設定と（ほぼ）等価になる：

```
\documentclass[b5paper]{ltjsarticle}
\usepackage{luatexja-fontspec}
\usepackage[ms]{luatexja-preset}
```

※補足：

- luatexja-preset パッケージの読込が行われるのは `jafont` を指定した場合に限られる。

2.6 注意事項

主に JS クラスとの違いについての注意。

- ページレイアウトについて、JS クラスの設計思想を受け継いでいるが、全く同じになるわけではない。
- JS クラスの一部のオプションで、BXJS クラスでは使用不可能なものがある。（4.3 節参照。）
- BXJS クラスではページレイアウトを設定するために内部で geometry パッケージを読み込んでいる。そのため、後からユーザが geometry を読み込むことはできない。ページレイアウトを変更する場合は、BXJS クラスが用意している再設定用の命令（6.1 節参照）か、または geometry パッケージが提供する再設定用命令（`\geometry` 等）を利用する。
- `papersize` オプションは既定で有効になっていて、出力用紙サイズはクラスオプションで指定したものに自動的に設定される。この処理を無効にするには `nopapersize` オプションを指定すればよい。
- `papersize` オプションの処理は geometry パッケージの機能により行われる。hyperref パッケージや（最近の）graphics ／ color パッケージがもつ出力用紙サイズ設定の機能はこれと干渉する可能性があるので、BXJS クラスにおいては自動的に無効化される。
- hyperref パッケージにおける“PDF の文字コード”の設定はエンジンごとに適切な値が異なっていて複雑であり、これが不適切であるために PDF 文書情報（しおり等）が文字化けしてしまう事例が数知れない。そこで、文書クラス側でエンジン毎に適切な設定を予め行うようにした。[*9]（ただし文書クラスが hyperref を読み込むわけではない。）
- pdfLaTeX 上で hyperref で `pdftitle` 等の文書情報に和文文字を含めたい場合は、`\hypersetup` 命令を通常通り使うことができる。[*10]

  ```
  \documentclass[pdflatex,a4paper,ja=standard]{bxjsarticle}
  \usepackage[colorlinks]{hyperref}
  \hypersetup{pdftitle={日本語タイトル}}
  ```

  hyperref のパッケージオプションで和文文字を含む文書情報を指定することはできない。
- `jafont` が無い場合の“既定のフォント設定”は多くのエンジンにおいて「IPAex フォント使用」であるが、(u)pLaTeX だけは異なっていて「何も指定しない状態」（JS クラスと同様）である。すなわち実際に使われる物理フォントの選択は DVI ウェアの設定に委ねられている。

[*9] 従って、(u)pLaTeX において、ほとんどの場合に pxjahyper パッケージを読み込む必要がない。ただし読み込んでも構わないし、必要な場合もある。

[*10] ちなみに、普通に CJKutf8 パッケージを用いた場合は、この方法では失敗してしまう。恐らく `\hypersetup` 命令全体を `CJK*` 環境で囲う必要があるのだと思われる。

- JS クラスは `\ifdraft` という公開名のスイッチ[*11]を用いているが、これは ifdraft パッケージと衝突する。そこで BXJS クラスでは代わりに `\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している。
  ※ JS クラスの `\ifdraft` は 2016/07/13 版で廃止された。従って、BXJS クラスにおいて `\ifdraft` スイッチの使用を非推奨とし、将来的に廃止を予定する。
- 1.2 版より、`\bf` や `\it` 等の旧式のフォント選択命令の使用が非推奨となり、これらの命令を使うと警告が出るようになった。詳細は 8 節を参照。

3 Pandoc モードの使い方

「Pandoc モード」は文書形式変換ツールである Pandoc[*12]での日本語 LaTeX 文書生成（および LaTeX 経由の PDF 生成）のために調整された設定である。

`\documentclass[pandoc,`⟨ドライバ⟩`,jafont=`⟨フォント指定⟩`,`⟨他オプション⟩`]{`⟨クラス名⟩`}`

- クラスオプションに `pandoc` を指定し、代わりに「エンジン」と「和文ドライバ（`ja`）」のオプションを省く。[*13]
- エンジンが DVI 出力である場合のドライバの既定値が `dvipdfmx` になる。ただし明示的にドライバオプションを与えることで `dvips` などに変更できる。
- 「和文フォント（`jafont`）」および他のクラスオプションは従来通り使用できる。

以下で Pandoc で BXJS クラスを用いる例を示す。

- bxjsarticle クラスを使用する一例。
  - XeLaTeX 経由
  - 用紙サイズは A4 判
  - 和文フォント設定は `ipaex`

  コマンド行は次のようになる[*14]（Pandoc 1.16 版の場合[*15]）：

  ```
  pandoc ⟨入力ファイル名⟩-o ⟨出力ファイル名⟩.pdf --latex-engine=xelatex
  -V papersize=a4 -V documentclass=bxjsarticle -V classoption=pandoc
  -V classoption=jafont=ipaex
  ```

  なおこの場合、途中で生成される LaTeX 文書のクラス指定は以下のようになる：

  ```
  \documentclass[a4paper,jafont=ipaex,pandoc]{bxjsarticle}
  ```
- bxjsbook クラスを用いる場合の一例。
  - 節番号を出力する
  - LuaLaTeX 経由

[*11] スイッチなので LaTeX レベルの命令ではない。標準クラスではこれに相当するものは `\if@draft` という非公開の制御綴で定義されている。
[*12] http://pandoc.org/
[*13] `pandoc` が指定された場合は、エンジンオプションの値は `autodetect-engine`、和文ドライバの値は `pandoc` に固定される。
[*14] もちろん、実際には改行を含めず 1 行で書く。
[*15] 少し古い版の場合、変数 `papersize` の値は `a4` ではなく `a4paper` と書くことになる。

  - 用紙サイズは B5 判
  - 和文の基底フォントサイズは 11Q
  - 和文フォント設定は `ipaex`
  - 欧文フォントを Pandoc の機能で設定

  コマンド行は次のようになる：

  ```
  pandoc ⟨入力ファイル名⟩-o ⟨出力ファイル名⟩.pdf --chapters -N
  --latex-engine=lualatex -V papersize=b5 -V documentclass=bxjsbook
  -V classoption=pandoc -V classoption=jbase=11Q
  -V classoption=jafont=ipaex
  -V mainfont="TeX Gyre Termes" -V sansfont="TeX Gyre Heros"
  ```

  注意事項：

  - bxjsbook クラスは「章（`\chapter`）」をもつクラスなので、Pandoc で `--chapters` の指定が必要。
- Pandoc で（LaTeX 経由で）PDF を出力する場合、エンジン指定（`--latex-engine`）は `pdflatex`、`xelatex`、`lualatex` のみがサポートされる。しかし、Pandoc の出力を「単体の LaTeX 文書」とすることで、LaTeX エンジンに (u)pLaTeX を使用することができる。出力された LaTeX 文書は通常の方法で PDF や PostScript 形式に変換できる。

  ```
  pandoc mydoc.md -o mydoc.tex -s -V documentclass=bxjsarticle -V classoption=pandoc
  uplatex mydoc
  uplatex mydoc
  dvipdfmx mydoc
  ```

4 クラスオプション

4.1 BXJS クラスに特有のオプション

JS クラスには無く BXJS クラスで追加されたクラスオプション。

- エンジンオプション： 実際に使用するエンジン（LaTeX コマンド名）を指定する。有効な値は `latex`、`platex`、`uplatex`、`pdflatex`、`xelatex`、`lualatex` である。エンジンオプション（と次項の `autodetect-engine` の何れか）の指定は必須である。
- `autodetect-engine`： 使用しているエンジンを判定して、自動的に適切なエンジンオプションを設定する。[*16]

  ※ BXJS クラスの設計の思想としては、「LaTeX 文書がどのエンジンでコンパイルすべきものかはソース中に明示されるべき」と考えていて、従って、“人間が普通に”文書を作る際にはこのオプションの使用は推奨されない。このオプションは“LaTeX ソースの自動生成”が絡む処理を念頭において用意されている。
- ドライバオプション： DVI 出力のエンジンを用いる場合に、実際に使用する DVI ウェアの名前を指定する。有効な値は `dvips`、`dvipdfmx`、`dviout`、`xdvi` である。ドライバオプションの指定は必須で

[*16] 実はエンジンの判定は常に行っていて、エンジンオプションが指定された場合はそれが正しいかを検査して、誤りの場合はエラーを出すようにしている。

ある。[*17]

- dvi=⟨ドライバ名⟩： エンジンが DVI 出力の場合に限り、指定のドライバオプションを有効にする。[*18]autodetect-engine と一緒に使うことが想定されている。
- dvipdfmx-if-dvi： dvi=dvipdfmx と同値。
  ※ dvi オプションは 1.2 版で新設されたもので、以前はこのオプションのみが存在した。
- pandoc： 「Pandoc モード」（3 節）を指定する。以下の設定が行われる：
  – エンジンオプションが autodetect-engine に固定される。
  – 和文ドライバが pandoc に固定される。
  – ドライバオプションについて dvi=dvipdfmx が既定になる（明示指定で上書可能）。
- nopapersize：“papersize special 出力”を抑止する。（JS クラスとは異なり、special 出力のオプション papersize は既定で有効である。）
  ※ papersize special を出力する他のパッケージとの干渉に対する対策。
- ja=⟨名前⟩： 使用する和文ドライバの名前を指定する。（詳細は 5 節を参照。）標準で提供されている和文ドライバには minimal、standard、pandoc がある。既定値は、エンジンが platex か uplatex の時は standard、それ以外は minimal。
- jadriver=⟨名前⟩： 「ja=⟨名前⟩」の（0.9 版以前で使われていた）別名。
- jafont=⟨名前⟩： “和文フォントプリセット指定”の名雨を設定する。
- japaram=⟨値⟩： “和文ドライバパラメタ”の値を設定する。
  ※ jafont と japaram の値がどのように解釈されるかは和文ドライバの仕様次第である。minimal 和文ドライバではこの 2 つの値は全く参照されない。2 節で解説した通り、standard 和文ドライバでは jafont の値が利用される。現状では japaram は参照されない。
- base=⟨長さ⟩： 基底フォントサイズ（\normalsize のフォントのサイズ）を指定する。JS クラスの 10pt、11pt 等と同じ役割で、任意の値を指定できる。基底フォントサイズの既定値は 10 pt である。
  ※##pt の形のオプションには名前と実際に設定される値がずれているものが多く、例えば 11pt は 10.95 pt、14pt は 14.4 pt が実際の設定値である。[*19]これに対して base=14pt は文字通り 14 pt を設定する。
  ※ ⟨長さ⟩ には calc パッケージの式が使用できる。この他に、特別に、(u)pLaTeX 以外のエンジンでも 11Q のように Q 単位で指定することができる（この場合は“⟨実数⟩Q”の形に限られ式は使えない）。[*20]
- jbase=⟨長さ⟩： 和文を基準にして基底フォントサイズを指定する。すなわち和文フォントの \normalsize のサイズを指定の長さとする。[*21]
  ※ ⟨長さ⟩ の書式は base オプションと同様。
- scale=⟨実数⟩： 和文スケール値を設定する。既定値は 0.924715（= 13 Q/10 pt）である。[*22]
- noscale： scale=1 と同値。
- paper={⟨横幅⟩}{⟨縦幅⟩}： 用紙サイズ設定。a4paper 等と同じ役割で、任意の値を指定できる。用紙

[*17] ただし現状では、ドライバオプションが無い場合にはエラーではなく警告が出る。
[*18] 実際にドライバオプションが有効化された場合は、そのオプションがグローバルオプションとしても働く。例えば、DVI 出力のエンジンで dvi=dvipdfmx が指定された場合は、dvipdfmx がグローバルオプションに追加される。
[*19] これは昔の LaTeX の“magstep”の習慣に由来する。
[*20] なお、(u)pLaTeX ではエンジンが単位 Q をサポートするので、11Q も 10Q+1Q も calc の式として受け付けられる。
[*21] この場合に決定される mag 値は和文スケール値にも依存することに注意。
[*22] これは JS クラスの設計に基づく値である。ただし実装の都合で、JS クラスの実際のスケール値はこれから僅かだけずれている。

サイズの既定値は A4 縦（210 mm × 294 mm）である。

- `enablejfam=`⟨値⟩： 数式中の和文出力をサポートするか否か。値は `true`（有効）／`false`（無効）／`default`（既定値に従う）の何れかである。エンジンや和文ドライバの種類により、既定値が有効・無効の何れになるかは異なるし、また、そもそも有効・無効の一方しか選択できない場合もある。詳細については 7 節を参照されたい。
- `disablejfam`： `enablejfam=false` と同値。
  ※ JS クラスとの互換のため存在する。

以下に挙げるのは上級者向けのオプションである。

- `zw`（既定）： `\jsZw` と等価な命令として `\zw` を定義する。
- `nozw`： `zw` の否定。
  ※命令名の衝突に対する対策。
- `js`（既定）： JS クラス（例えば bxjsreport の場合は jsbook）が読込済であるように振舞う。
  ※「JS クラスであるかによって挙動を変える」パッケージに対する対策。
- `nojs`： `js` の否定。
  ※つまり「JS クラスの一種である」と判定されると不都合な場合にこのオプションを指定する。
- `bigcode`： upTEX エンジンと dvipdfmx の組合せで hyperref パッケージを利用する時に適用される ToUnicode CMap として `UTF8-UTF16` を指定する。PDF の文書情報の文字列に BMP 外の文字が含まれる場合にはこの指定が必要である。`UTF8-UTF16` のファイルがインストールされていないと、dvipdfmx の処理が失敗する。
- `nobigcode`： `bigcode` の否定。ToUnicode CMap として `UTF8-UCS2` を指定する。この場合は文書情報の文字列に BMP 外の文字を使用できない。
  ※ `UTF8-UTF16` のファイルが利用できるかを確実に判定するのは困難なため、`bigcode` の既定値は次のようなアドホックな方法で決めている：TEX エンジンのバージョンが 3.14159265 以上[*23]である場合は、（TEX 環境がある程度新しく、`UTF8-UTF16` が利用可能と思われるので）`bigcode` を既定とし、それ以外は `nobigcode` を既定とする。
- `precisetext`： XHTEX エンジンにおいて、「ActualText 生成機能」を有効化する。[*24]
- `noprecisetext`（既定）： `precisetext` の否定。
- `simplejasetup`（既定）： XHTEX エンジン自体の行組版機能（`\XeTeXlinebreaklocale` 等）を利用した、簡易的な日本語用組版設定を行う。
  ※ XHTEX エンジン以外では無効である。また、xeCJK や zhspacing 等の日本語（CJK）組版用パッケージが読み込まれた場合も無効化される。特に、和文ドライバが standard である場合は、必ず xeCJK が読み込まれるため、このオプションは無意味である。
- `nosimplejasetup`： `simplejasetup` の否定。
- `mag=`⟨整数⟩： 版面拡大率（mag 値）の直接設定。既定は `base` から算出する。
  ※ mag 値が n の場合、版面が $n/1000$ 倍に拡大される。
- `magstyle=`⟨値⟩： “版面拡大”の実現方法を指定する。有効な値は `usemag`、`nomag`、`nomag*` の何れ

[*23] TEX のバージョン 3.14159265 は 2014 年 1 月にリリースされた。
[*24] つまり、`\XeTeXgenerateactualtext=1` を行う。ActualText 生成機能と日本語処理は相性が悪いため、これを使うと出力 PDF のサイズが増大する（1.5〜2 倍）ことに注意。

か。詳細は 4.5 節を参照。

- geometry=⟨値⟩： geometry パッケージの読込に対する制御。
  - class（既定）： 通常通り、文書クラスが geometry パッケージを読み込む。ユーザは geometry を後から読み込むことはできない。
  - user： 文書クラスによる geometry パッケージの読込をスキップする。この場合、ユーザが自分で geometry を読み込むことが想定される。[*25]

  ※“どうしても geometry パッケージを自分で読み込みたい”という人のための設定。
- oldfontcommands： \bf 等の“二文字フォント命令”の使用を許容する。
- nooldfontcommands（既定）： “二文字フォント命令”の使用に対して警告を出す。詳細については 8 節を参照されたい。
- fancyhdr=⟨真偽値⟩[*26]： fancyhdr パッケージの機能に対する補正を行うか。真の場合、以下の補正が行われる。既定値は真。
  - ヘッダ・フッタ書式の既定値に含まれる“二文字フォント命令”を除去する。
  - bxjsbook クラスでヘッダ・フッタの横幅を（\textwidth ではなく）\fullwidth に一致させる。
- textwidth-limit=⟨整数⟩： bxjsbook クラスにおける、\textwidth の上限の長さ（全角単位）。[*27] 既定値は 40。
- paragraph-mark=⟨文字 1 つ⟩： パラグラフ（\paragraph）の見出し先頭に付く記号。既定値は“■”。
- layout=⟨値⟩： レイアウトの変種を選択する。有効な値は以下の通り。
  - v2（既定）： 現版の既定のレイアウト。
  - v1： bxjsbook クラスについて、1.2a 版以前の（本来は不適切な）水平マージンの設定を適用する。（詳細は 6.1 節を参照。）その他のクラスについては v2 と全く同じ。

4.2 JS クラスのオプションで使用可能なもの

これらについては名前だけ列挙するに留める。ただし、“JS クラス特有”（標準クラスに無い）オプションの一部については解説を加える。

■用紙サイズ指定 a3paper、a4paper、a5paper、a6paper、b4paper、b5paper、b6paper、a4j、a5j、b4j、b5j、a4var、b5var、letterpaper、legalpaper、executivepaper。
※ a4var は A4 変判（210 mm × 283 mm）、b5var は B5 変判（182 mm × 230 mm）。
※（BX）JS クラスでは a4j は a4paper と全く等価である。（他の b4j 等も同様。）

■横置き landscape。

■基底フォントサイズ 8pt、9pt、10pt、11pt、12pt、14pt、17pt、20pt、21pt、25pt、30pt、36pt、43pt、12Q、14Q、10ptj、10.5ptj、11ptj、12ptj。

[*25] geometry の読込は必須ではなく、ページレイアウトのパラメタを自分で設定しても構わない。ただし geometry の読込が強く推奨される。（geometry 非読込時の動作テストはほとんど行っていない。）
[*26] 真偽値は true（真）または false（偽）で指定する。
[*27] つまり、bxjsbook クラスにおいては、geometry で指定する“width”の値は \fullwidth の値と見なされ、それがこのオプションで指定する上限値を超えている場合は、\textwidth は上限値になる。この場合の本文領域の配置の様式は jsbook クラスと同様である。

※ 10pt、11pt、12pt、14pt、17pt、21pt、25pt、30pt、36pt、43pt はそれぞれ magstep の 0、0.5、1、2、3、4、5、6、7、8 である。8pt、9pt、20pt は文字通りの値。##Q ／##ptj は jbase=##Q ／ jbase=##pt を表す（つまり和文規準）。[*28]

■両面用レイアウト　oneside、twoside、vartwoside。
※ vartwoside は twoside と同様だが傍注が常に右側余白に出力される。

■段組み　onecolumn、twocolumn。

■表題ページ　titlepage、notitlepage。

■起こし　openright、openany。
※ jsbook のみ（BXJS では bxjsreport と bxjsbook）にのみ存在するオプション。

■数式配置　leqno、fleqn。

■オーバーフル警告　final、draft。

■papersize special 出力　papersize。
※ BXJS クラスでは papersize は既定で有効。

■英語化　english。

4.3　JS クラスのオプションで使用不可能なもの

- クラス変種指定： report、slide。
  ※ report 相当は bxjsreport、slide 相当は bxjsslide と別クラスになっている。
- トンボ出力： tombow、tombo、mentuke
  ※これは pLaTeX のカーネル命令を利用しているのでとりあえず除外。
- 和文フォントメトリック指定： jis、winjis、mingoth。
  ※異なるエンジンで汎用的に扱うのが難しい。

4.4　クラスオプション設定の既定値

- BXJS クラス共通： a4paper、onecolumn、final、ja=minimal、jafont は空、japaram は空、scale=0.924715、magstyle=usemag[*29]
- bxjsarticle： 10pt、oneside、notitlepage
- bxjsreport： 10pt、oneside、titlepage、openany
- bxjsbook： 10pt、twoside、titlepage、openright
- bxjsslide： 36pt、oneside、notitlepage

[*28] ちなみに JS クラスの（固定の）和文スケール値に従うと 10pt が jbase=13Q に相当するので 13Q というオプションは無い。
[*29] LuaTeX の 0.87 版以降では magstyle=nomag* が既定となる。4.5 節参照。

4.5 magstyle オプション

JS クラスにおけるページレイアウト決定の過程では、基底フォントサイズが 10 pt 以外の場合に、“版面を拡大縮小する”という処理を採用している。[*30] これには、「フォントのオプティカルサイズの選択を最適にするため[*31]」という理由があり、またこれにより、多種の基底フォントサイズへの対応が容易になるという利点もある。[*32] ところがここで、JS クラスではこの“版面の拡大”を実現するために TEX エンジンが持つ版面拡大機能（仮に「mag 設定」と呼称する）を用いていて、これについて批判されることが多い。また、現実問題として、mag 設定が LATEX で用いられる機会は少ないため、実際に用いられた時にそれを想定していないパッケージが誤動作するという問題もある。

これらの問題を緩和するため、BXJS クラスでは“版面拡大”について他の実現方法を提供している。それを選択するのが以下に挙げる「magstyle オプション」である。[*33]

- `usemag`： JS クラスと同様に、“版面拡大”のために mag 設定を用いる。
- `nomag`： mag 設定を一切用いず、代わりに、全てのページレイアウトのパラメタの値をスケールさせる。`\normalsize` や `\large` 等の高位フォントサイズ命令で指定されるフォントサイズもスケールさせるが、“オプティカルサイズの調整”は行わない。いわゆる「基本 35 書体」のようなオプティカルサイズでない[*34] フォントのみを用いるのであれば、この設定が最も適切である。
- `nomag*`： `nomag` と同様に、全てのページレイアウトのパラメタの値をスケールさせる。さらに、“オプティカルサイズの調整”を実現するために、NFSS の実装コードにパッチを当てる。[*35] この場合、mag 設定による不具合は起こらなくなるが、当然、NFSS のパッチのせいで別の不具合が起こる可能性はある。

※ LuaTEX の 0.87 版以降では（PDF 出力時の）「mag 設定」の機能が廃止されている。そのため、そのようなエンジンでは `usemag` はサポートされない（エラーになる）。
※ `magstyle` オプションの既定値は `usemag` である。ただし例外として、LuaTEX の 0.87 版以降では（`mag` が使えないため）`nomag*` が既定値となる。

5 和文ドライバ

BXJS クラスでは様々なエンジンについて、そのエンジンおよびそれに対応するパッケージが提供する日本語処理機能を活用することで、日本語用の文書クラスとしての機能を実現している。そこでの汎用性を確保す

[*30] 例えば、基底フォントサイズが 20 pt だとすると、まずは指定されたものの半分の縦横幅をもつ用紙に対して基底フォントサイズが 10 pt としてレイアウトを決定し、それを縦横 2 倍に拡大する、という過程をとっている。
[*31] LATEX の既定の欧文フォントである Computer Modern フォントがオプティカルサイズの性質をもつことは有名であるが、少々癖が強くて、本文を 10 pt（cmr10）で組んだ場合と 12 pt（cmr12）で組んだ場合でかなり異なった印象を受ける場合がある。JS クラスではそれを嫌って、本文（`\normalsize` のフォント）が必ず「cmr10 を拡大縮小したもの」で組まれることを企図しているのである。
[*32] BXJS で「任意の」基底フォントサイズが設定できるのもこの利点があるため。
[*33] 「magstyle オプション」の値は、`magstyle` をキー名にした keyval 形式（例えば `magstyle=nomag*`）で書くこともできる。1.1e 版以前では、keyval 形式のみがサポートされ、しかも値の識別子として `usemag` ／ `nomag` ／ `nomag*`の代わりに `mag` ／ `real` ／ `xreal` を用いて（つまり `magstyle=xreal` と書いて）いた。
[*34] 或いは、オプティカルサイズに“変な癖”のない。
[*35] 要するに、`OT1/cmr/m/n/12` が要求された時に、`cmr12` でなくて `cmr10 at 12pt` が選ばれるようにする。

るため、“日本語処理機能と連携する部分”の実装をモジュールとして分離していて、これを**和文ドライバ**と呼ぶ。[*36]BXjscls のバンドルでは次の 2 つの和文ドライバを提供している。

- standard 和文ドライバ： 各エンジンについて、最も一般的に用いられる特定の“日本語処理機能”（例えば `lualatex` なら LuaTEX-ja）を連携対象とした和文ドライバ。(u)pLATEX 上の JS クラスと同じくらい容易に日本語が書き始められることを目指している。
- minimal 和文ドライバ： “何も実装されていない”和文ドライバ。上級ユーザがプレアンブルや自作パッケージ等にアドホックな連携コードを書いて、好きな“日本語処理機能”との連携を実現するために用いることを想定している。
- pandoc 和文ドライバ：「Pandoc モード」で仕様される和文ドライバ。基本的に standard と同じ設定を用いるが、「Pandoc の既定の latex テンプレート」が使われることを前提として、それと BXJS の設定を整合させるための措置を加えている。

和文ドライバは自分で作製することも可能である。[*37]`bxjsja-XXX.def`（`XXX` は任意の文字列[*38]）の名前のファイルに実装コードを書いてそのファイルを配置すると、`ja=XXX` のオプション指定でその和文ドライバを利用できる。

なお、和文ドライバ指定オプション（`ja`）の既定値は `minimal` である。現実には、ほとんど全ての場合に `standard` が用いられると思われるが、種々の理由があって、これを既定値にはしていない。
※ただし、(u)pLATEX については、日本語処理機能がエンジン自体に備わっていて不可分なため少し異なる扱いになっている。[*39]`minimal` を用いる意義がほとんどないため、`standard` が既定値になっている。

6 ユーザ用命令

原則として、BXJS クラスで追加されたものだけを説明する。

6.1 レイアウト設定関連

BXJS クラスではページレイアウトの設定に geometry パッケージを用いて次の手順で行っている。

1. （基底フォントサイズにより決定された mag 値を実際に設定する。）
2. geometry で次のパラメタを設定する。
   （a）クラスオプションで指定された用紙サイズ、および `truedimen`。
   （b）bxjsarticle ／ bxjsreport の場合は次のパラメタ値。
   ```
   headheight=10pt, footskip=0.03367\paperheight,
   headsep=\footskip-\topskip, includeheadfoot,
   hscale=0.76, hmarginratio=1:1, vscale=0.83, vmarginratio=1:1
   ```
   （c）bxjsbook の場合は次のパラメタ値。
   ```
   headheight=10pt, headsep=6mm, nofoot, includeheadfoot,
   hmargin=18mm, vscale=0.83, vmarginratio=1:1
   ```

[*36] graphicx パッケージ等の「ドライバ」と類似した概念のためこの名称を用いた。
[*37] 和文ドライバの実装に必要な連携仕様の情報については、ソースコード説明書（bxjscls.pdf）の付録 A を参照してほしい。
[*38] カテゴリコード 11 または 12 の文字からなる必要がある。
[*39] JS クラスの実装から分離した「日本語処理関連」のコードを minimal に配置している。

※ bxjsbook の 1.2a 版以前では、この設定の中の“hmargin=18mm”の代わりに“hmargin=36mm, hmarginratio=1:1”を用いていた。これでは jsbook の水平マージン設定と同等にならないため 1.3 版で現在の設定に修正された。もし 1.2a 版以前との互換性を保ちたい場合は、クラスオプションに layout=v1 を指定してほしい。

（d）bxjsslide の場合は次のパラメタ値。

```
noheadfoot, hscale=0.9, hmarginratio=1:1,
vscale=0.944, vmarginratio=1:1
```

3. 後処理を行う。以下の処理が含まれる。
   - textwidth を全角幅の整数倍に、textheight を整数行分の自然長になるように丸める。
   - marginpar 関連の設定を行う。

ページレイアウトの再設定のために次の命令が用意されている。

- \setpagelayout{⟨設定⟩}： 現在のページレイアウトの設定の一部を修正する。⟨設定⟩ は geometry のパラメタの記述であり、現在の設定に追記して geometry が再設定を行った後、再び 3 の後処理が行われる。
- \setpagelayout*{⟨設定⟩}： 用紙以外の設定をリセットして改めてページレイアウトの設定を行う。具体的には、2a と ⟨設定⟩ の内容を用いて geometry が再設定を行った後、再び 3 の後処理が行われる。

なお、\geometry 命令を直接呼び出すことも可能である。当然この場合は 3 の後処理は行われない。

6.2 構造マークアップ関連

- \subtitle{⟨テキスト⟩}： サブタイトルを設定する。
  ※\maketitle の出力にサブタイトルが含まれるようになる。

6.3 和文用設定関連

- \jsZw： 和文の全角幅を表す。
- \zw： \jsZw の別名。[*40]ただし nozw 指定時は定義されない。
- \zwspace： 全角（\jsZw）幅の水平空き。

6.3.1 standard 和文ドライバの場合

standard 和文ドライバでは和文に関連する文書ソース記述をエンジンに依らずに共通になることを目指している。従って、和文関連の組版パラメタの設定*1 についても「共通の命令」が提供される。[*41]

- 和文ファミリ変更命令： pLaTeX と同様に、\mcfamily で「明朝」、\gtfamily で「ゴシック」に変更される。\textmc、\textgt も使える。

[*40] LuaTeX-ja では「実際の全角幅」を表す命令 \zw（pLaTeX の zw と本当に等価）が規定されている。lualatex エンジン指定かつ和文ドライバが standard の場合はこの \zw の定義がそのまま使われる。（従って nozw は実質的に無効である。）なお、\jsZw は「規約上の全角幅」であり、「実際の全角幅」と本来は一致するはずだが、実際には計算誤差のせいで僅かに値が異なる。

[*41] pandoc 和文ドライバは standard 和文ドライバの設定を引き継ぐため、pandoc モードでもこれらの命令を使用できる。

- 欧文ファミリ変更命令での和文の連動: JS クラスと同様[*42]に、`\rmfamily` で和文が「明朝」、`\sffamily` および `\ttfamily` で和文が「ゴシック」に変更される。
- `\zw`： 和文の全角幅を表す。例えば `2\zw` が pLaTeX の `2zw` に相当する。
- `\jQ`、`\jH`、`\trueQ`、`\trueH`： それぞれ pLaTeX の単位 Q、H、trueQ、trueH に相当する長さ。
- `\ascQ`： `\trueQ` を和文スケール値で割った長さ。[*43]例えば、`\fontsize{10\ascQ}{16\trueH}` で和文のサイズが 10 Q になる。
- `\ascpt`： 1 true pt を和文スケール値で割った長さ。例えば、`\fontsize{9\ascpt}{13truept}` で和文のサイズが 9 ポイントになる。
- `\setxkanjiskip{`⟨長さ⟩`}`: 和欧文間空白の量を指定する。pLaTeX での `\setlength{\xkanjiskip}{`⟨長さ⟩`}` に相当する。
- `\getxkanjiskip`： 現在の和欧文間空白の量を表す文字列に展開される。pLaTeX での `\xkanjiskip` の読出[*44]に相当する。
- `\autoxspacing` ／`\noautoxspacing`： 和欧文間空白の挿入を有効／無効にする。pLaTeX の同名の命令と同等。
- `\setkanjiskip{`⟨長さ⟩`}`： 和文間空白の量を指定する。pLaTeX での `\setlength{\kanjiskip}{`⟨長さ⟩`}` に相当する。
- `\getkanjiskip`： 現在の和文間空白の量を表す文字列に展開される。pLaTeX での `\kanjiskip` の読出[*45]に相当する。
- `\autospacing` ／`\noautospacing`： 和文間空白の挿入を有効／無効にする。pLaTeX の同名の命令と同等。
- `\jachar{`⟨文字 1 つ⟩`}`： 指定の文字を和文文字として（現在の和文フォントで）出力する。
- 和文数式フォント命令： JS クラスと同様に、`\mathmc` で「明朝」、`\mathgt` で「ゴシック」の和文数式フォントを指定する。
- 欧文数式フォント命令での和文の連動： `\mathrm` で和文が「明朝」、`\mathsf` および `\mathtt` で和文が「ゴシック」に指定される。
  ※ JS クラスとは異なり、“連動の組合せ”はテキストと同一であることに注意。

例えば、pLaTeX において、次のように「和文間空白」を利用して均等割りを行うという技法が知られている。

```
%% \kintouwari{<整数 n>}{<テキスト>}
% n 全角の幅にテキストを均等割りで出力する.
\newcommand\kintouwari[2]{{%
  \setlength{\kanjiskip}{\fill}%
  \makebox[#1zw][s]{#2}}}
```

これと同等のものを、次のようにエンジン非依存な形で書くことができる。

```
\newcommand\kintouwari[2]{{%
```

[*42] ちなみに、(u)pLaTeX の既定ではこの連動は起こらない。
[*43] 命令名は“anti-scaled Q”の略。
[*44] TeX 言語でいうと `\the\xkanjiskip`。
[*45] TeX 言語でいうと `\the\kanjiskip`。

表 1　standard 和文ドライバにおける数式中の和文出力のサポート

エンジン	enablejfam	直書き	\mathmc/gt	和欧文連動
(u)platex	true（既定）	可	サポート有り	有り
	false	不可	フォールバック	—
lualatex	true	可	サポート有り	有り
xelatex	true	可	フォールバック	無し
	false（既定）	不可	フォールバック	—
pdflatex	false	不可	フォールバック	—

```
\setkanjiskip{\fill}%
\makebox[#1\zw][s]{#2}}}
```

7　数式中の和文出力について

minimal 和文ドライバは数式中の和文出力の機能を何も提供しない。従って、そのような機能を提供する他のパッケージを併用するのでない限り、数式中に和文を書いたときの挙動は未定義である。

standard 和文ドライバ（およびそれを継承する和文ドライバ）における数式中の和文出力の扱いは、エンジンと enablejfam オプションの値の組合せにより異なり、表 1 に示すようになる。以下でこの表に関する補足説明を行う。

- この表にある以外のエンジンと enablejfam 値の組合せは許容されない。この場合、警告が出て、enablejfam が可能な値に自動的に変更される。
- 「直書き」が「可」の場合、数式フォント命令（\mathXX{}）の外に書いた和文文字は明朝体で出力される。「不可」の場合、そのような和文文字の扱いは未定義である。
- 「\mathmc/gt」が「サポート有り」の場合、これらの命令は“本物”の数式フォント命令として働く。「フォールバック」の場合は、これらの命令は内部で一旦テキストモードに切り替えて非数式として出力される。このフォールバック機能を実用したい場合は、amstext（または amsmath）パッケージの併用が望ましい。[*46]

8　“二文字フォント命令”に対する警告

ここでいう“二文字フォント命令”というのは、\bf や \it 等の LaTeX 2.09 で標準であったフォント選択命令のことである。[*47] LaTeX 2ε においては、これらに代わって、\bfseries 等の（NFSS に基づく）新しい命令群が標準となり、古い二文字フォント命令はカーネルではもはサポートされなくなった。しかし同時に、二文字フォント命令を利用したパッケージを動作させるための“当面の”[*48]互換性対策として、「標準の文書クラス（article、book 等）で二文字フォント命令のサポートを行う」という方針がとられた。これに倣って、

[*46] amstext を読み込まない場合、添字中で \mathmc/gt を用いたときに文字サイズが非添字のものに戻ってしまうという不具合が出る。

[*47] なお、\em は「二文字の名前のフォント命令」であるが、これは LaTeX 2ε でも標準命令であり、“二文字フォント命令”には含まれない。

[*48] ちなみに、LaTeX 2ε が最初にリリースされたのは 1994 年のことである。

他の文書クラスの多くもクラスのレベルで二文字フォント命令をサポートしていて、BXJS クラスもその例に含まれる。

ところが最近になって、一部の文書クラス（KOMA-Script クラス群や memoir クラス等）において、二文字フォント命令を明示的に非推奨の扱いにした上で、その使用に制限を設ける（警告を出す、オプションを指定しないと使えない、等）という措置が取られるようになっている。

これに合わせて、BXJS クラスでは 1.2 版より二文字フォント命令を非推奨とし、また、既定でその使用に対して警告を出すようにした。

8.1 警告の内容

現状では警告はとても控えめに出される。すなわち、文書中で `\bf` などの二文字フォント命令が呼び出された場合、コンパイルの最後に（一度だけ）以下の警告メッセージが表示される。

```
Class bxjsarticle Warning: Some old font commands were used in text
(bxjsarticle)              (see the log output for detail).
```

少し詳細な注意メッセージが、ログファイルのみに書き出される。

なお、この警告は、パッケージの機能の実装として用いられたものも含めて全ての二文字フォント命令の呼出が対象になる。ただし例外として、thebibliography 環境の内部で呼び出されたものだけは対象から除外される。BibTeX の文献スタイルファイル（`.bst`）や文献データベース（`.bib`）のファイルは（パッケージと比較しても）極めて古いものが割と普通に使い続けられることが多い。そういった極めて古いファイルに由来する二文字フォント命令を警告したとしても、多くの場合、ユーザ側には対処する方法が存在しない。これが文献リスト環境中で警告を抑止する理由である。

8.2 警告の制御

二文字フォント命令に対する警告の有無はクラスオプションで制御できる。

- `oldfontcommands`： 二文字フォント命令を警告の対象にしない。
- `nooldfontcommands`（既定）： 二文字フォント命令を警告の対象にする。

また、以下の命令により、文書中で一時的に警告の設定を変えられる。[*49]

- `\alloweoldfontcommands`： これ以降に実行される二文字フォント命令を警告の対象にしない。
- `\disalloweoldfontcommands`： これ以降に実行される二文字フォント命令を警告の対象にする。

8.3 将来的な二文字フォント命令の扱い

※以下の記述は現在存在する和文ドライバを使用する場合についてのものである。

- 二文字フォント命令に対する警告の様式は、今後変更される可能性がある。
- しかし、将来に二文字フォント命令のサポートが廃止されることはない。

[*49] これらの命令の効果はグルーピングの影響を受ける。

- `oldfontcommands` オプションよおび `\alloweoldfontcommands` 命令は継続して提供され、これらの機能を用いた場合は、二文字フォント命令に関する警告が端末に表示されることは一切無い。