

# Package **gentombow** v0.9k

Hironobu Yamashita

2019/07/21

English version is available: see [gentombow.pdf](#)!

日本の出版では、たとえば仕上がり寸法が A4 サイズの場合、一回り大きな B4 サイズの用紙の中央にトンボ付きで印刷することがあるようです。このドキュメントは、実際に A4 用紙を B4 用紙の中央にトンボ付きで配置している事例です。

`gentombow` パッケージは、クラスファイルなどが想定している用紙サイズ情報 (`\paperwidth`, `\paperheight`) を検知し、それより一回り大きなサイズの用紙の中央にトンボ付きで出力するために必要な機能を提供します。使い方の例は

```
\documentclass[a4j]{jarticle}
\usepackage{gentombow}
\usepackage{bounddvi}
\begin{document}
...本文...
\end{document}
```

です。

任意の  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  フォーマットで利用可能です。具体的には

- $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ,  $\text{upL}^{\text{A}}\text{T}_{\text{E}}\text{X}$  (縦組にも対応)
- $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$  ( $\text{LuaT}_{\text{E}}\text{X}$ -ja 使用の場合は縦組にも対応)
- $\text{pdfL}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- $\text{XeL}^{\text{A}}\text{T}_{\text{E}}\text{X}$

での動作を確認しています。また、`jsclasses` のような版面拡大 (`\mag`) が使われた場合にも対応しています。

ただし、本パッケージは実際の出力サイズ指定を発行しませんので、以下のいずれかのパッケージと併用するとよいでしょう：

- `bounddvi` パッケージ (`gentombow` バンドル)
- `bxpapersize` パッケージ (ZR さん作)
- `graphicx` パッケージの 2016 年以降の版 (`setpagesize` 機能付きのもの)

`bounddvi` パッケージは本パッケージと同じく `gentombow` バンドルに収録されていますが、名前のとおり DVI を経由する場合にしか利用できません。一方、`bxpapersize` パッケージは汎用で、 $\text{LuaT}_{\text{E}}\text{X}$  のような PDF 直接出力の場合にも利用できます。また、`graphicx` パッケージは画像挿入用のパッケージですが、2016 年以降は

自動的に出力サイズ指定も行うようです ( $\text{\mag} \neq 1000$  の場合には機能しません)。

本パッケージは、gentombow バンドルの一部として配布されています：

<https://github.com/aminophen/gentombow>

2017 年までは platex-tools バンドルの一部でしたが、2018 年以降は独立させてあります。また、v0.9e までは (u)pL<sup>A</sup>T<sub>E</sub>X と LuaT<sub>E</sub>X-ja 専用の pxgentombow パッケージを用意していましたが、v0.9f 以降は gentombow に一本化しました。

## 1 簡単な使用法

はじめに、パッケージをオプション指定なしに使用する場合の機能を説明します。

### 1.1 用紙サイズの自動検知による出力サイズ決定

パッケージにはあらかじめ A 系列 (a0–a10)、B 系列 (b0–b10)、C 系列 (c0–c10) と letter, legal, executive の用紙サイズが定義されています。ここで、B 系列は ISO ではなく JIS です。また、変型版として a4var (a4 の変型) と b5var (b5 の変型) も定義されています。

これらのうちいずれか (ただし a0, b0, c0 を除く) の用紙サイズを検知すると、出力サイズが次の規則で自動的に決定されます<sup>\*1</sup>。

- 用紙サイズが A 系列のとき：出力サイズは一回り大きな B 系列
- 用紙サイズが B, C 系列または letter, legal, executive のとき：出力サイズは一回り大きな A 系列

この場合、パッケージを読み込んだだけでトンボが付きます。なお、用紙サイズが横長の場合は自動的に出力も横長になり、縦長の場合は自動的に縦長になります。

よく使われる用紙サイズの例を挙げます。

用紙サイズ	出力サイズ
a6	b6
b6	a5
a5	b5
b5	a4
a4	b4
b4	a3
a3	b3
b3	a2
c6	a5
c5	a4
c4	a3

<sup>\*1</sup> なお、C 系列と letter, legal, executive については日本での慣習が不明のため、現時点では A 系列のサイズで出力することになっています。

c3	a2
letter	a3
legal	a3
executive	a4

### 参考：用紙サイズの自動検知に失敗した場合の出力サイズ決定

仮に用紙サイズが定義済みのいずれとも異なる場合は、デフォルトでは用紙の天地左右に 1 インチずつのノビを付けたサイズで出力します。たとえば、幅 100 mm、高さ 200 mm の用紙の場合、出力サイズは幅 100 mm + 2 in、高さ 200 mm + 2 in になります。

## 1.2 トンボへのバナーの出力

デフォルトでは gentombow-ja (2020-09-25 21:18) のような形式で、トンボにジョブ情報（処理ファイル名と日付）を示すバナーが出力されます。これは、pL<sup>A</sup>T<sub>E</sub>X 向けの日本語クラスファイルの多くで利用可能な tombow オプションと同様です。ただし、クラスオプションに tombo（ジョブ情報を出力しない）または mentuke（トンボの線を表示しない）が指定されている場合は、この設定が引き継がれます。

## 2 パッケージオプション

パッケージオプションによって、出力サイズやバナーのジョブ情報の有無を指定する方法を説明します。

### 2.1 出力サイズの明示指定

自動決定されるサイズと異なるサイズに出力したい場合、パッケージオプションで明示的に指定することができます。たとえば

```
\documentclass[a4j]{jarticle}
\usepackage[tombow-a3]{gentombow}
\begin{document}
...本文...
\end{document}
```

とすると、出力サイズは（自動決定の b4 は無視されて）a3 に変わります。指定可能なサイズは、定義済みの用紙サイズと同じものです。すなわち、A 系列 (a0–a10)、B 系列 (b0–b10)、C 系列 (c0–c10) と a4var, b5var, letter, legal, executive です。なお、ここでも用紙サイズが横長の場合は自動的に出力も横長になり、縦長の場合は自動的に縦長になります。

オプションは〈トンボ形式〉-〈出力サイズ〉の書式で与えます。トンボ形式は、pL<sup>A</sup>T<sub>E</sub>X の標準クラスと同じで tombow, tombo, mentuke のいずれかを選びます（tombow はジョブ情報を表示し、tombo は表示しません。また、mentuke はトンボの線を表示しません）。

## 2.2 トンボに表示するジョブ情報の有無

用紙サイズを明示的に指定する場合は、前節のようにトンボ形式と出力サイズを同時にオプションで与えることができます。逆に出力サイズを自動決定に任せつつ、ジョブ情報の有無だけを指定したい場合は、トンボへのバナー出力を抑制する `notombowbanner` オプションを使用してください。

## 3 詳細なカスタマイズ

`gentombow` パッケージは、トンボの見た目やバナーの内容をカスタマイズする機能も提供します。

### 3.1 トンボに表示するバナーの内容

`\settombowbanner` 命令を用います。例えば  $\epsilon$ -pTeX の `\pdfcreationdate` プリミティブを使って

```
\documentclass[a4j]{jarticle}
\usepackage{gentombow}
\settombowbanner{\jobname\space (\pdfcreationdate)}
\begin{document}
…本文…
\end{document}
```

とすると、バナーの表示内容は `gentombow-ja (D:20200925211826+09'00')` になります。命令の引数には任意のトークン列を与えることができます。

### 3.2 バナーの文字のフォント

`\settombowbannerfont` 命令を用います。例えば

```
\settombowbannerfont{cmss10 at 9pt}
```

のような指定が可能です。内部的には `\font` プリミティブを利用しており、使用している TeX エンジンがサポートする任意のフォントを指定可能です\*2。

### 3.3 トンボの線の太さ

デフォルトではトンボの線の太さは 0.1 pt ですが、`\settombowwidth{1pt}` とすると 1 pt になります。命令の引数には任意の寸法を与えることができますが、実用上は数 pt 以内が良いでしょう。

### 3.4 塗り足し（ドブ）の幅

デフォルトでは塗り足し（ドブ）の幅は 3 mm ですが、`\settombowbleed{5mm}` とすると 5 mm になります。命令の引数には任意の寸法を与えることができますが、実用上は数 mm でしょう。

---

\*2 LuaTeX や XeTeX (`gentombow` パッケージの場合) では、OpenType フォントを直接利用することもできます。

### 3.5 トンボの色

トンボの色を指定します。実際に色をつける場合は `xcolor` パッケージ（推奨）または `color` パッケージが必要です。`\settombowcolor{\color[cm]{0,1,0,0}}` とするとマゼンタになります。CMYK の版ごとにトンボを作成する場合や、特色のトンボが必要な場合に便利です。

## 4 PDF のデジタルトンボ (page box) の設定

この機能はドライバに依存するため、オプション機能としています。gentombow パッケージ読み込み時に `pdfbox` オプションを指定します。DVI を経由する場合は、さらにドライバオプションも必須です\*3。

この機能を有効化すると、PDF に以下のバウンディングボックス情報（いわゆるデジタルトンボ）が書き込まれ、さらに用紙サイズ設定（/MediaBox 相当）も行われます。

- /TrimBox: 仕上がり紙面サイズ。
- /BleedBox: 裁ち落とし領域を含んだサイズ。
- /CropBox, /ArtBox: 明示しない。

注意: pdfx パッケージが PDF/X モードで読み込まれていれば、このオプション機能は自動的に有効化されます。また、/CropBox も明示します。

## 5 その他の注意点

特定のクラスファイルやパッケージと一緒に使用する場合、いくつかの注意点があります。

### 5.1 jsclasses で使用する場合の注意

奥村晴彦氏による `jsclasses` のクラス（2016 年以降は日本語  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  開発コミュニティが管理）を使用している、10pt 以外のサイズオプションを指定する場合は、以下のいずれかの方法をとってください。

- クラスオプションに「トンボオプション」（`tombow` または `tombo`）を追加する。
- クラスオプションに「`\mag` を使わないオプション」（`nomag` または `nomag*`）を追加する。

これは、`jsclasses` クラス内で行われる `\oddsidemargin` と `\topmargin` の計算の都合からくる制約です。たとえば

```
\documentclass[a4j,14pt]{jsarticle}
\usepackage{gentombow}
```

という使い方は誤りです（このままでは誤った余白設定が適用されますので、安全のため gentombow パッケージがエラーを出すようにしてあります）。代わりに

```
\documentclass[a4j,14pt,tombow]{jsarticle}
```

\*3 例外として、gentombow が読み込まれた時点で `graphics/color` のいずれかが読み込まれていた場合は、そのドライバ設定を引き継ぐことができます。

```
\usepackage{gentombow}
```

と書くようにしてください。

また、塗り足しの幅などの寸法指定は `true` 付きの単位を使用してください。例えば `\settombowbleed{3truemm}` と指定すれば、`\mag` による版面拡大（縮小）後の塗り足し幅を 3 mm にできます。

同様に、ZR 氏による `BXjscls` のクラスを使用していて、10pt 以外のサイズオプションを指定する場合は

- クラスオプションに「`\mag` を使わないオプション」（`nomag` または `nomag*`）を追加する。

という対処が必要です（`BXjscls` クラスでは「トンボオプション」が機能しないダミーになっていることに注意）。

## 5.2 レイアウト設定の注意

余白などのレイアウト設定でありがちですが、`\hoffset` や `\voffset` の値を 0 以外に設定するのは避けてください。このような設定は `gentombow` パッケージのトンボ出力機能と衝突するため、正しい出力が得られません。レイアウト設定のために変更すべきなのはこれらの寸法ではなく、`\oddsidemargin` や `\topmargin` であることがほとんどです。したがって、それらを適切な値に設定するか、レイアウトの設定すべてを `geometry` パッケージに任せてしまうのも一つの方法です。

たとえば左右の余白を 25 mm に、上下の余白を 30 mm にしたいとき<sup>\*4</sup>、まず「 $\mathrm{TeX}$  の 1 インチ」を削除してから `\oddsidemargin` や `\topmargin` を変更すると、「見かけ上は」期待どおりの結果になることがあります。

```
\setlength{\hoffset}{-1in}% <== NG
\setlength{\voffset}{-1in}% <== NG
\setlength{\oddsidemargin}{25mm}
\setlength{\topmargin}{30mm}
\setlength{\textwidth}{\paperwidth}
\addtolength{\textwidth}{-2\oddsidemargin}
\setlength{\textheight}{\paperheight}
\addtolength{\textheight}{-2\topmargin}
\addtolength{\topmargin}{-\headheight}
\addtolength{\topmargin}{-\headsep}
```

しかし、この設定では `gentombow` パッケージがトンボを追加するとき、正しい余白を維持することができません。

一方、`\hoffset` や `\voffset` は 0 のままで、以下のように `\oddsidemargin` や `\topmargin` を設定していれば問題ありません。

```
\setlength{\oddsidemargin}{-0.4mm}% 25mm = 1inch - 0.4mm
\setlength{\topmargin}{4.6mm}%      30mm = 1inch + 4.6mm
\setlength{\textwidth}{\paperwidth}
\addtolength{\textwidth}{-50mm}
\addtolength{\topmargin}{-\headheight}
\addtolength{\topmargin}{-\headsep}
\setlength{\textheight}{\paperheight}
```

<sup>\*4</sup> ここでは「本文の領域以外」を余白と定義します。すなわち、ヘッダとフッタは余白の一部です。

```
\addtolength{\textheight}{-60mm}
```

これと同等のレイアウト設定は、以下のように `geometry` パッケージで行うのが簡単です。

```
\usepackage[lmargin=25mm,rmargin=25mm,
            tmargin=30mm,bmargin=30mm]{geometry}
```

なお、`geometry` パッケージは用紙サイズ (`\paperwidth`, `\paperheight`) を変更する場合があります。こうした場合に備え、`gentombow` パッケージは `geometry` パッケージよりも後に読み込むようにしてください。

## 雑記

発端はこの話です。

- 齋藤修三郎 (@psi\_tau) on Twitter, 2017 年 2 月 9 日  
[https://twitter.com/psi\\_tau/status/829873082911248386](https://twitter.com/psi_tau/status/829873082911248386)

また、現在検討中の事項を挙げておきます。

- 現状では、用紙の横長・縦長がそのまま出力に反映されるので、これを逆転させるオプションの実装。また、任意の出力サイズを指定できるインタフェースの実現。
- 自動で決定できる出力サイズの拡張。現状では用紙サイズが定義値に完全一致する場合のみ自動決定されるが、中間のサイズでもその一回り大きなサイズに出力したい場合があるのではないか。
- 出版用途ではカラー印刷の場合に、CMYK の版ごとにトンボを作る必要がある。`color` パッケージが利用可能な場合に、オプション次第で CK など必要な色を選べるようにするとよいのではないか<sup>\*5</sup>。
- `\mag ≠ 1000` の場合について、`jsclasses` 以外での動作は未確認。特に、`geometry` とは共存しない可能性が高い。
- `\stockwidth/\stockheight` が `\paperwidth/\paperheight` より小さい場合の動作。現在は警告を出すだけとしているが、さらに天地左右 1in にフォールバックしたほうが無難だろうか。

## 変更履歴

- 2017/02/10 v0.1 最初の公開版 (`pxgentombow` として)
- 2017/03/01 v0.4 トンボ形式の修正など
- 2017/05/05 v0.5 `jsclasses` の `\mag ≠ 1000` に対応、最初の CTAN リリース版
- 2017/05/06 v0.6 `jsclasses` との共存時のチェック強化、`LuaTeX-ja` での動作確認
- 2017/07/23 v0.7 ドキュメント更新
- 2017/12/17 v0.9 `gentombow` パッケージの新設
- 2018/03/16 v0.9e クラスオプションの `tombow` 等の引き継ぎ、トンボ書式のカスタマイズ機能の追加 (`\settombowbanner`, `\settombowbannerfont`, `\settombowwidth` の各命令)
- 2018/05/17 v0.9g `gentombow` に `pxgentombow` を統合、`pdfbox` オプションの追加、`\settombowbleed`,

---

<sup>\*5</sup> ただし作者の本業は出版ではないので、商用を含む実用には程遠いかもしれない。

`\settombowcolor` の追加

- 2018/08/20 v0.9h 用紙サイズ検出に `hagaki` を追加、`pdfpages` 対策の改良
- 2018/08/30 v0.9j `jsclasses` の `\mag \neq 1000` でも `pdfbox` オプションが正常動作するように改良、`\settombowbleed` が `TrimBox` に反映されなかったバグの修正
- 2019/07/21 v0.9k `pdfx` パッケージとの整合性をとる改良