

Package `bounddvi` v7.2

Koichi Inoue & Hironobu Yamashita

2016/12/28

Package `bounddvi` sets papersize special into DVI file. This package can be used in both *tate* (vertical) and *yoko* (horizontal) writing directions of Japanese $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ / $\text{u}\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$, and both `dvipdfmx` and `dvips` drivers are supported. The `tombow` option defined in Japanese $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ kernel is also supported. Of course, this package can be used also with the original $\text{L}\text{A}\text{T}\text{E}\text{X}$ in DVI output mode.

This package (since v7.0) is part of `gentombow` bundle:

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

Originally it was part of `platex-tools` bundle, but it has been moved since March 2018.

Usage

Load this package in preamble.

```
\documentclass[a5paper]{article}
\usepackage{bounddvi}
...
```

Process the `.tex` file using `latex + dvips` chain or `latex + dvipdfmx` chain.

Known limitations

1. The compatibility with `geometry` package may not be perfect, as both `geometry` and `bounddvi` embeds papersize special into a DVI file. The loading order of these two packages may affect the actual size of output.
2. This package supports “`jsclasses`-like employment” of `\mag`, because it’s more widely used in Japan. This may be incompatible with some classes or packages which employ `\mag` in other ways.

The details are described in the sections below.

The behavior of multiple papersize specials

Sometimes multiple papersize specials may be embedded into a DVI file. Among these specials, the specification which appears *at last* in DVI takes effect when

`dvipdfmx` or `dvips` (T_EX Live 2017 or later) is used¹. For example, when the following source is processed with `dvipdfmx`,

```
% latex + dvipdfmx
\documentclass{...}
\usepackage{bounddvi}
\usepackage[dvipdfm]{geometry}
```

the specification by `geometry` wins.

Note about `\mag` handling

Among the packages in CTAN, there are two types of implementation in terms of `\mag` employment. It seems that there is no (official or practical) “standard” in `\mag` treatment.

When the output is going to the physical size of A4 (210 mm × 297 mm) with the setting of `\mag=2000`, there are two ways: some classes/packages can set

```
\mag=2000
\paperwidth=210mm (= 420 truemm)
\paperheight=297mm (= 594 truemm)
```

and others can set

```
\mag=2000
\paperwidth=105mm (= 210 truemm)
\paperheight=148.5mm (= 297 truemm)
```

The first way is adopted by `geometry` package etc, and it’s (probably) based on the behavior of the `papersize` special of `dvips`. It does not handle true units properly, and accepts only non-true units and evaluates them as if they were true units. The second way is adopted by `jsclasses` document class etc, and is also supported by `pdf:pagesize` special of `dvipdfm(x)`. This can be more consistent with L^AT_EX, since all other layout parameters (e.g. `\textwidth`) are set according to the unit `truemm`.

The `bounddvi` supports the latter, so some classes/packages which are based on the former may or may not work properly when using `bounddvi` package.

References

- Setting paper size using `dvips` & `dvipdfm` (description in Japanese)
<https://www.ma.ns.tcu.ac.jp/Pages/TeX/bounddvi.sty.html>

ChangeLog

- 2002/03/10 v1.0 (KI) First version
- 2002/10/30 v2.0 (KI) Add `dvipdfm pdf:pagesize` special

¹When `dvips` in T_EX Live 2016 or earlier is used, the specification which appears *first* in DVI took effect, but the default behavior was changed in T_EX Live r42420 to be compatible with `dvipdfmx`. The new option `-LO` can be used to recover the old behavior.

- 2003/03/22 v3.2 (KI) Compatibility with `hyperref`
- 2004/05/08 v4.0 (KI) Support for `\mag ≠ 1000`
- 2004/12/08 v5.2 (KI) Compatibility with `geometry`
- 2004/12/15 v6.0 (KI) Not to use `dvipdfm(x) pdf:pagesize special`
- 2016/10/25 v7.1 (HY) Support for pL^AT_ΕX 2_ε `tombow` option, compatibility with `graphics/color` packages
- 2016/12/28 v7.2 (HY) Documentation for the new `dvips` behavior