The twoup Package*

© Mogens Lemvig Hansen†
mogens@kayju.com

2007/02/26

Abstract

The twoup assists in two-up or booklet printing using the 2up or booklet packages or using the psnup program and friends.

Contents

---

[*] This file describes version 1.3, 2007/02/26.

[†] The twoup package is distributed under the LaTeX Project Public License; please see Section 4.

I am familiar with three ways of producing two-up documents[1] and booklets:[2] the 2up package, the booklet package, and the postscript tool psnup. The 2up and booklet packages are both available from CTAN, and psnup is included with many TeX distributions including MiKTeX. All three will by default accomplish this task by scaling the page down. However, a one-up page layout does not necessarily work well when scaled. The twoup package provides a page layout designed for two-up printing and sets the relevant 2up or booklet parameters or gives advise on how to use the postscript tools.

1 Usage

Give the paper size you intend to print on as an option to `\documentclass`.[3] The twoup package re-calculates the paper size and margins for two-up printing.[4] Then choose the method:

1.1 Plain two-up with the 2up package

Load the 2up package *before* the twoup package.

```
\documentclass[a4paper,12pt]{article}
\usepackage{2up}
\usepackage{twoup}
```

Unfortunately cross references may not work out under the 2up package. The solution is, as explained in the 2up documentation, to comment out the `\usepackage{2up}` line, run your .tex file through LaTeX until all references work out, then load 2up again and run your document through LaTeX once (an only once).

1.2 Booklets with the booklet package

Load the booklet package *before* the twoup package.

```
\documentclass[letterpaper,12pt]{article}
\usepackage[print]{booklet}
\usepackage{twoup}
```

Without the `print` option, the booklet package does nothing.

The booklet package has miraculously solved the cross-reference problem of the 2up package but does not provide a plain two-up option.

---

[1] That is, printing two logical pages side by side on one side of one sheet of paper.

[2] That is, rearranging the logical pages before printing them two-up on both sides of the paper in such a manner that the resulting sheets of paper can be folded down the middle into a small book.

[3] If you intend to print on your default paper size, you do not have to give any paper size option.

[4] If you print on `a4paper`, twoup calculates the same paper size as the `a5paper` option but with different (smaller) margins. You may consider this discrepancy either a bug or a feature.

1.3 Plain two-up with the postscript tools

Simply load the `twoup` package.

```
\documentclass[letterpaper,12pt]{article}
\usepackage{twoup}
```

When you run your .tex file through LaTeX, the `twoup` package will tell you which options to use with the psnup program; look for the psnup command line near the end of the transcript.[5] If you give a paper size option, the `twoup` package will also tell you which lines you will have to add to your postscript file by hand.

Run the .dvi file through dvips[6] to make a postscript file.

```
dvips ⟨file⟩
```

Then run psnup with the options that the `twoup` package gave you above. Even Win95 allows you to cut-and-paste the command line which could look something like this:

```
psnup -pletter -W396.0pt -H612.0pt -2 ⟨file⟩.ps ⟨twoup⟩.ps
```

The postscript file, ⟨*twoup*⟩`.ps`, has by now lost all consciousness of which paper size it is to be printed on. If you are about to print on your printer's default paper size, that is no problem. To print on any other paper size you most likely have to place a paper size command in ⟨*twoup*⟩`.ps`. With a plain text editor add the lines `twoup` gave you in the LaTeX run to ⟨*twoup*⟩`.ps`. The lines to add could look like this:

```
%%BeginPaperSize: letter
letter
%%EndPaperSize
```

Place the new lines just after the `%%EndComments` line near the top of ⟨*twoup*⟩`.ps`.

1.4 Booklets with the postscript tools

The LaTeX side of making booklets is much like simple two-up. You may want to use the `twoside` option to `\documentclass` to make the inner margins a bit larger etc.

```
\documentclass[a4paper,11pt,twoside]{article}
\usepackage{twoup}
```

As above, the `twoup` package will advise you on which options to use with the psnup program. Run your .dvi file through dvips:

```
dvips ⟨file⟩
```

---

[5] Both on screen and in the .log file.

[6] The current version of the `twoup` package requires that you use dvips. If you want to use some other .dvi-to-postscript program, please see code line 37 and the commentary around it.

Now the fun begins: Before using psnup you must use the psbook program to fold your postscript file into a signature:

```
psbook ⟨file⟩.ps ⟨book⟩.ps
```

The number of pages in a signature must be divisible by four, so psbook may add a few blank pages to the end of your booklet. Run the result through psnup with the options the twoup package calculated in the LaTeX run. The command line could look something like this:[7]

```
psnup -pa4 -Pa5 -2 ⟨book⟩.ps ⟨twoup⟩.ps
```

Unless you are about to print on your printer's default paper size, add the paper size command to ⟨*twoup*⟩`.ps` as above.

You can print your postscript file, ⟨*twoup*⟩`.ps`, now, but if you want to print double sided, you may have to split the file in two with the psselect program:

```
psselect -o ⟨twoup⟩.ps ⟨odd⟩.ps
psselect -e ⟨twoup⟩.ps ⟨even⟩.ps
```

Then print ⟨*odd*⟩`.ps`, turn the paper over, and print ⟨*even*⟩`.ps`.[8]

1.5 Options

By default the twoup package leaves room for running headers and footers. Use the `noheadfoot`, `nofoot`, or `nohead` option to reclaim that space. As a side efect, these options set the page style: `noheadfoot` implies the `empty` page style, `nofoot` implies the `headings` page style, and `nohead` implies the `plain` page style. Moreover, the `nofoot` and `noheadfoot` options disable the `plain` page style[9] to prevent the `\chapter` command from inserting a `\thispagestyle{plain}` command.

The twoup package knows the standard paper size options: `letterpaper`, `legalpaper`, `executivepaper`, `a4paper`, `a5paper`, and `b5paper`. You should use these options as global options to `\documentclass`. If you print two-up on `legalpaper`, the text width is larger than what is considered comfortable. You should use the `twocolumn` option to `\documentclass` or, better, the multicol package.

If you use twoup without 2up or booklet, then the twoup package will insert a paper-size `\special` in your .dvi file. Use the `nospecial` option to suppress this.

1.6 Additional commands

`\cleartolastpage` If you are making a booklet, the number of pages must be divisible by four; indeed, the booklet package or the psbook program will add blank pages as necessary. If,

---

[7] Under Unix, you can pipe the psbook and psnup commands together; Windows, however, does not like long command lines.

[8] Depending on your printer, you may want to use psselect's `-r` option to print the ⟨*even*⟩ pages in reverse order.

[9] By making `plain` synonymous with the `empty` style.

however, you want to print on the last page, the back of your booklet, you can use the `\cleartolastpage` command in a `twosided` context to (print all pending tables, figures, and footnotes and) insert as many blank pages as necessary to reach a page number divisible by four. In detail,

```
\cleartolastpage[⟨div⟩][⟨rem⟩]
\cleartolastpage*[⟨div⟩][⟨rem⟩]
```

both print all pending floats and insert blank pages until the page number leaves remainder ⟨*rem*⟩ when divided by ⟨*div*⟩. The default value of ⟨*rem*⟩ is 0, and the default value of ⟨*div*⟩ is 4 under the `twosided` class option and 2 otherwise. The starred version suppresses the printing of page numbers on the inserted blank pages.

Thus `\cleartolastpage[1]` is equivalent to LaTeX's `\clearpage` command and `\cleartolastpage[2][1]` is equivalent to `\cleardoublepage`.

The `\cleartolastpage` command may fail to put you one the last page if page 1 is not the very first page (maybe because you have differently numbered front matter); use the optional ⟨*rem*⟩ argument to overcome this.

2 Installation

As you must have figured, you generate the documentation for the twoup package by running the file `twoup.dtx` through LaTeX—thrice to resolve cross references.

To extract the package itself from the .dtx file, run `twoup.ins` through LaTeX:

```
latex twoup.ins
```

You now have to decide what to do with several files.

- You may have to move the file `twoup.sty` to some directory where LaTeX can find it; `(local)texmf/tex/latex/misc` would be the natural choice [2].
- Move the documentation, `twoup.dvi` or `twoup.pdf`, to `(local)texmf/doc/latex/misc`.
- You may discard the source files, `twoup.dtx` and `twoup.ins`, or store them in `(local)texmf/source/latex/misc`.
- Discard all remaining `twoup.*` files.

3 Limitations

The twoup package is designed for and tested on `letterpaper`, `legalpaper`, and `a4paper`. Printing two-up on smaller paper is hardly feasible; printing on larger paper is beyond the normal desktop printer.

The margins calculated by twoup are too small for `\marginpars`.[10]

---

[10] But just large enough for the line numbers inserted by the lineno package.

If you print two-up on `legalpaper`, twoup leaves a text width larger than what is considered comfortable. You should use the `twocolumn` option to `\documentclass` or, better, the multicol package.

The twoup package is not compatible with the geometry package. Both packages re-calculate the margins and the disagree on the interpretation of the paper size options.

I know very little about postscript, yet I tell you to go add lines to .ps files. The advise twoup gives has always worked for me; if it doesn't work for you, please teach me the correct way.

The current implementation of twoup requires dvips (unless you use the 2up or booklet packages). This limitation is easily overcome with your help: please see code line 37 and the commentary around it.

4 License

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in `http://www.latex-project.org/lppl.txt` and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

This work has the LPPL maintenance status 'maintained.'

The Current Maintainer of this work is Mogens Lemvig Hansen.

This work consists of the files `twoup.dtx` and `twoup.ins` and the derived file `twoup.sty`.

5 Implementation

The twoup package doesn't have to do much—just re-calculate the paper size and the margins and put together the advise on how to use the psnup program. The twoup package does its thing based on the plain LaTeX parameters `\paperwidth`, `\paperheight`, `\headheight`, `\headsep`, `\footskip`, `\topskip`, and `\baselineskip`.

5.1 Identification

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}%
3 \ProvidesPackage{twoup}[2007/02/26 version 1.3]
```

5.2 Declaration of Options

The `noheadfoot` option sets the relevant lenghts to zero. Since there is no room for headers or footers, the only sensible page style is `empty`. The `\chapter` command issues a `\thispagestyle{plain}` command, so re-define the `plain` style to be a synonym for the `empty` style.[11]

```
\DeclareOption{noheadfoot}{%
   \setlength{\headheight}{0pt}%
   \setlength{\headsep}{0pt}%
   \setlength{\footskip}{0pt}%
   \let\ps@plain\ps@empty
   \pagestyle{empty}}
```

With room for a header but without a footer the most sensible page style would be `headings`. Again the `plain` page style must be disabled.

```
\DeclareOption{nofoot}{%
   \setlength{\footskip}{0pt}%
   \let\ps@plain\ps@empty
   \pagestyle{headings}}
```

With room for a footer but no header the most sensible page style is `plain`—so don't change `plain`.

```
\DeclareOption{nohead}{%
   \setlength{\headheight}{0pt}%
   \setlength{\headsep}{0pt}%
   \pagestyle{plain}}
```

`\TwoUp@info`
`\TwoUp@paper`

The paper size options record part of the advise on how to use the psnup program in `\TwoUp@info` and part of the advise on how to edit the final postscript file in `\TwoUp@paper`.

```
\DeclareOption{letterpaper}{%
   \def\TwoUp@info{-pletter -W5.5in -H8.5in}%
   \def\TwoUp@paper{letter}}%
\DeclareOption{legalpaper}{%
   \def\TwoUp@info{-plegal -W7in -H8.5in}%
   \def\TwoUp@paper{legal}}%
\DeclareOption{executivepaper}{%
   \def\TwoUp@info{-pexecutive -W5.25in -H7.25in}%
   \def\TwoUp@paper{executive}}%
\DeclareOption{a4paper}{%
   \def\TwoUp@info{-pa4 -Pa5}%
   \def\TwoUp@paper{a4}}%
\DeclareOption{a5paper}{%
   \def\TwoUp@info{-pa5 -W105mm -H148mm}%
   \def\TwoUp@paper{a5}}%
\DeclareOption{b5paper}{%
   \def\TwoUp@info{-pb5 -W125mm -H176mm}%
   \def\TwoUp@paper{b5}}%
```

`\TwoUp@special`

The twoup package needs to record the (smaller) paper size in the postscript file. That means placing a suitable `\special` command in the .dvi file. Unfortunately,

---

[11] I borrowed the idea of brutally making `plain` a synonym for `empty` from David Carlisle's nopageno package.

"suitable" depends on which .dvi-to-postscript program you use, so we need an option for each such program. I have implemented the option for dvips but was too lazy to figure out the appropriate command for other .dvi-to-postscript programs.

```
\DeclareOption{dvips}{%
   \def\TwoUp@special{%
      \special{papersize=\the\paperwidth,\the\paperheight}}}
```

If you need to use the twoup package with another postscript driver, please let me know so I can incorporate more options in future releases of twoup. While you wait for that, use the `nospecial` option to suppress the `\special`.

```
\DeclareOption{nospecial}{%
   \def\TwoUp@special{\relax}}
```

5.3 Execution of Options

Default to dvips leaving space for headers and footer as the class sees fit.

```
\ExecuteOptions{dvips}
\ProcessOptions
```

5.4 Main Code

5.4.1 Paper Size and psnup Advise

`\TwoUp@info` Before re-calculating the paper size, we must record part of the psnup advise. If a paper size option did that already, all is fine. Otherwise we must be printing on the default paper size which we must then determine. We will try to recognize the most common default paper sizes: a4, letter, and legal. Since this is the default paper size, we may assume that the printer knows how to handle it. We therefore suppress the advise on editing the final postscript file by *not* defining `\TwoUp@paper`.

```
\ifx\TwoUp@info\undefined
   \setlength{\@tempdima}{8.5in}
   \ifdim\paperwidth=\@tempdima
      \setlength{\@tempdima}{11in}
      \ifdim\paperheight=\@tempdima
         \def\TwoUp@info{-pletter -W5.5in -H8.5in}
      \fi
      \setlength{\@tempdima}{14in}
      \ifdim\paperheight=\@tempdima
         \def\TwoUp@info{-plegal -W7in -H8.5in}
      \fi
   \fi
   \setlength{\@tempdima}{210mm}
   \ifdim\paperwidth=\@tempdima
      \setlength{\@tempdima}{297mm}
      \ifdim\paperheight=\@tempdima
         \def\TwoUp@info{-pa4 -Pa5}
      \fi
```

```
  \fi
\fi
```

If `\TwoUp@info` is still undefined, we give up and use the `-w` and `-h` options to psnup.[12]

```
\ifx\TwoUp@info\undefined
  \PackageWarningNoLine{twoup}{Unknown paper size.
    You may want to use a \MessageBreak
    paper size option like letterpaper}
```

TeX displays `\the\paperwidth` as however many *pt*, by which TeX means traditional points; one inch is 72.27 pt. However, psnup expects the paper width as so-and-so many *pt*, by which psnup means postscript points, the unit TeX calls *bp*, or big points; one inch is 72 bp. To overcome this confusion, we must multiply by $\frac{72}{72.27}$ or 0.99627.

```
  \setlength{\@tempdima}{0.99627\paperwidth}
```

The `\@settopoint` command rounds (down) to an integer. Rounding causes the psnup command line to be slightly wrong but less ugly.

```
  \@settopoint\@tempdima
```

Then record the information in `\TwoUp@info`. The psnup syntax for the output paper size is `-w`⟨*width*⟩ `-h`⟨*height*⟩.

```
  \edef\TwoUp@info{-w\the\@tempdima}
```

Record the paper height in the same manner.

```
  \setlength{\@tempdima}{0.99627\paperheight}
  \@settopoint\@tempdima
  \edef\TwoUp@info{\TwoUp@info\space -h\the\@tempdima}
```

Then add the input paper size to the list of psnup options. The syntax for the input paper size is `-W`⟨*width*⟩ `-H`⟨*height*⟩. As before, we must correct for the confusion between traditional and postscript points.

```
  \setlength{\@tempdima}{0.5\paperheight}
  \setlength{\@tempdima}{0.99627\@tempdima}
  \@settopoint\@tempdima
  \edef\TwoUp@info{\TwoUp@info\space -W\the\@tempdima}
  \setlength{\@tempdima}{0.99627\paperwidth}
  \@settopoint\@tempdima
  \edef\TwoUp@info{\TwoUp@info\space -H\the\@tempdima}
\fi
```

The advise on psnup options is ready; now we must decide whether to use it. But first change the paper size: swap the width and the height; then divide the new width by two.

```
\setlength{\@tempdima}{\paperwidth}
\setlength{\paperwidth}{0.5\paperheight}
\setlength{\paperheight}{\@tempdima}
```

---

[12] We could of course have used the `-w` and `-h` options also for a4, letter, and legal sized paper, but the psnup command line looks a lot prettier with a `-pa4`, `-pletter`, or `-plegal` option.

If either the 2up or booklet package is loaded, ignore the psnup advise and issue suitable `\source` and `\target` commands—both 2up and booklet define `\source` and `\target`.

```
\@ifpackageloaded{2up}{%
   \source{\magstep0}{\paperwidth}{\paperheight}%
   \target{\magstep0}{2\paperwidth}{\paperheight}%
}{%
\@ifpackageloaded{booklet}{%
   \source{\magstep0}{\paperwidth}{\paperheight}%
   \target{\magstep0}{2\paperwidth}{\paperheight}%
}{%
```

Otherwise we are going the postscript route. Record the new smaller paper size in the postscript file by placing a suitable `\special` command in the .dvi file.

```
   \TwoUp@special
```

Then have the psnup advise presented at the end of the LaTeX run. Using `\PackageInfo` would have been natural, but I want the advise to appear on the screen (as well as in the .log file); also the psnup command line is rather long, so we need *ad hoc* formatting.

```
   \AtEndDocument{\typeout{^^J%
      Package twoup Info: Use psnup with these options:^^J%
      psnup \TwoUp@info\space -2 \jobname.ps twoup.ps^^J}}
```

If we had an explicit paper size option (indicating that we are to print on a paper size other than the default), `\TwoUp@paper` is defined and we can give advise on editing the postscript file.

```
   \ifx\TwoUp@paper\undefined\else
      \AtEndDocument{\typeout{^^J%
         Package twoup Info: You may have to add these
         three lines to your .ps file:^^J%
         \@percentchar\@percentchar BeginPaperSize: \TwoUp@paper^^J%
         \TwoUp@paper^^J%
         \@percentchar\@percentchar EndPaperSize^^J}}
   \fi%
}}%
```

5.4.2 Text Width and Horizontal Margins

The standard LaTeX classes [1] ensure that the line length is not too large for comfortable reading. For two-up printing on `letterpaper`, `a4paper`, or smaller, that is a non-issue. However, `legalpaper` does leave room for oversized lines.[13] If I leave the line length "small," the layout of the two-up printed page looks odd. I therefore allow the oversized lines. The user should consider using `twocolumn` mode or—better—the multicol package.

---

[13] Up to 25% "too large."

I think that a total horizontal margin of one inch looks fine.[14] The `\textwidth` is then what is left of the `\paperwidth`.

```
\setlength{\@tempdima}{1in}
\setlength{\textwidth}{\paperwidth}
\addtolength{\textwidth}{-\@tempdima}
```

If we are printing in `twoside` mode, the inner margin should be larger than the outer margin. A ratio of about $1 : 2$ looks fine. The outer margin is then what is left of the total horizontal margin.

```
\if@twoside
   \setlength{\evensidemargin}{0.33\@tempdima}
   \setlength{\oddsidemargin}{\@tempdima}
   \addtolength{\oddsidemargin}{-\evensidemargin}
\else
```

In `oneside` mode, simply split the margins even.

```
   \setlength{\oddsidemargin}{0.5\@tempdima}
   \setlength{\evensidemargin}{\oddsidemargin}
\fi
```

So far we have ignored that the "real" left margin is `1 inch` + `\hoffset` + `\oddsidemargin` (or `\evensidemargin`). We must subtract that inch.

```
\addtolength{\oddsidemargin}{-1in}
\addtolength{\evensidemargin}{-1in}
```

5.4.3 Text Height and Vertical Margins

A top margin of half an inch looks good to me. As a first approximation for the text height, simply subtract all the header and footer stuff from the paper height. (Subtract `\topmargin` twice for the sake of the bottom margin.)

```
\setlength{\topmargin}{0.5in}
\setlength{\textheight}{\paperheight}
\addtolength{\textheight}{-2\topmargin}
\addtolength{\textheight}{-\headheight}
\addtolength{\textheight}{-\headsep}
\addtolength{\textheight}{-\footskip}
```

The text height must then be corrected to fit an integer number of lines. The first line has height `\topskip` while all other lines have height `\baselineskip`. Thus subtract `\topskip` and divide by `\baselineskip`.

```
\addtolength{\textheight}{-\topskip}
\divide\textheight\baselineskip
```

The `\divide` command performs integer division, so `\textheight` is now ⟨*number of lines*⟩ sp. Store the result in a counter to get a *bona fide* number.

```
\@tempcnta\textheight
```

---

[14] Fine enough to just leave room (in `oneside` mode) for the numbers added by the lineno package.

Now we can calculate the text height: the correct number of `\baselineskip`s plus a `\topskip` for the first line.

```
\setlength{\textheight}{\@tempcnta\baselineskip}
\addtolength{\textheight}{\topskip}
```

Again, the “real” top margin is `1 inch` + `\voffset` + `\topmargin`, so subtract one inch from `\topmargin`.

```
\addtolength{\topmargin}{-1in}
```

`\cleartolastpage`

The `\cleartolastpage` command clears the page, checks for a star, prepares to issue a suitable `\thispagestyle` command, and passes control to `\cleartolastp@ge`.

```
\newcommand{\cleartolastpage}{%
   \clearpage
   \@ifstar
      {\def\TwoUp@blankpagestyle{\thispagestyle{empty}}\cleartolastp@ge}
      {\let\TwoUp@blankpagestyle\relax\cleartolastp@ge}}
```

The `\cleartolastp@ge` command stores the first optional argument, ⟨*div*⟩, in the temporary counter `\@tempcnta` with a default that depends on the `twoside` class option.

```
\if@twoside
   \newcommand{\cleartolastp@ge}[1][4]{%
      \@tempcnta#1\relax
      \cleartol@stp@ge}%
\else
   \newcommand{\cleartolastp@ge}[1][2]{%
      \@tempcnta#1\relax
      \cleartol@stp@ge}%
\fi
```

The `\cleartol@stp@ge` command does the real work: store a copy of the page number, `\c@page`, in a temporary counter, divide and multiply by ⟨*div*⟩ to round down to a multiple of ⟨*div*⟩, and add ⟨*rem*⟩.

```
\newcommand{\cleartol@stp@ge}[1][0]{%
   \@tempcntb\c@page
   \divide\@tempcntb\@tempcnta
   \multiply\@tempcntb\@tempcnta
   \advance\@tempcntb#1\relax
```

Now `\@tempcntb` leaves remainder ⟨*rem*⟩ when divided by ⟨*div*⟩, but it may still be smaller than the page number. If so, add ⟨*div*⟩ to get the next number that leaves the same remainder.

```
   \ifnum\@tempcntb<\c@page
      \advance\@tempcntb\@tempcnta
   \fi
```

Now `\@\tempcntb` is the page number we want to reach.

```
   \loop\ifnum\c@page<\@tempcntb
      \hbox{}\TwoUp@blankpagestyle\newpage
```

```
    \repeat
}
⟨/package⟩
```

References

[1] Leslie Lamport, Frank Mittelbach, and Johannes Braams. *Standard Document Classes for LaTeX version 2e.* CTAN, 1999/09/10 edition, 1999.

[2] TUG Working Group on a TeX Directory Structure (TWG-TDS). *A Directory Structure for TeX Files.* CTAN, version 0.9995, January 26, 1998 edition, 1998.