

# The `cd-cover` class\*

## Version v1.4 Beta

CVS Revision: 1.6

Christian Holm†

2002/01/28 14:30:44

CVS Date: 2002/01/28 14:30:44

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Usage</b>	<b>2</b>
2.1	Syntax . . . . .	2
2.2	User setup . . . . .	5
<b>3</b>	<b>Credits</b>	<b>5</b>
<b>4</b>	<b>Copyright</b>	<b>6</b>
<b>5</b>	<b>Implementation</b>	<b>6</b>
5.1	Initializing . . . . .	6
5.2	The various cover types . . . . .	10
5.2.1	The booklet type . . . . .	10
5.2.2	The back type . . . . .	13
5.2.3	The sleeve type . . . . .	15
5.2.4	The single type . . . . .	16
5.3	Modified output routines . . . . .	18
5.4	The final stuff . . . . .	19

---

\*Previously a package

†E-mail: [cholm@nbi.dk](mailto:cholm@nbi.dk)

# 1 Introduction

This is an  $\beta$  release of the `cd-cover` class. This was previously a package, but since a class is actually more appropriate for the commands and macros herein, this new version switches to a class implementation.

A number of improvements has been made in this release. The whole scheme of things has taken a drastic turn, providing many new advantages (see also section 5.2).

Bug reports, suggestions, praises, but not flames should be sent to `cholm@nbi.dk`. Please submit changes in form of patches to the most recent `cd-cover.dtx` file<sup>1</sup>. Patches are easily generated with

```
diff -c <original-file> <new-file> > <patch-file>
```

on any reasonable system. This makes merging changes into the class much easier, and the changes will more like be included in the normal distribution.

## 2 Usage

In this release four environments are provided:

- `bookletsheets` is for typesetting booklet pages for normal CD covers (plastic cover).
- `backsheet` is for typesetting the back side for normal CD covers.
- `backsheet*` Special version of the above, that rotates the back title the other way. Useful for typesetting double jewelcase covers, and for some nationalities (e.g., German) that like the title to run opposite to that of English.
- `sleevesheet` is for typesetting paper sleeves for CDs, for example if you lost the plastic one, or don't like the space they take up, or whatever.
- `singlesheet` is for typesetting covers for plastic single CD covers.

### 2.1 Syntax

The syntax of these environments is:

```
\begin{bookletsheets}<your text>\end{bookletsheets}  
\begin{backsheet}{<title>}<your text>\end{backsheet}  
\begin{backsheet*}{<title>}<your text>\end{backsheet*}
```

---

<sup>1</sup>Availible from CTAN in `macros/latex/contrib/other/cd-cover`.

```

\begin{sleevesheet}<your text>\end{sleevesheet}
\begin{singlesheet}<{title}><{slip text}><your text>\end{singlesheet}

```

Here  $\{<title>\}$  is the title on the back of the covers,  $\{<slip text>\}$  is the text that should go on the back slip of the single type cover.

In figure 1 to 4 there are some examples of these environments.

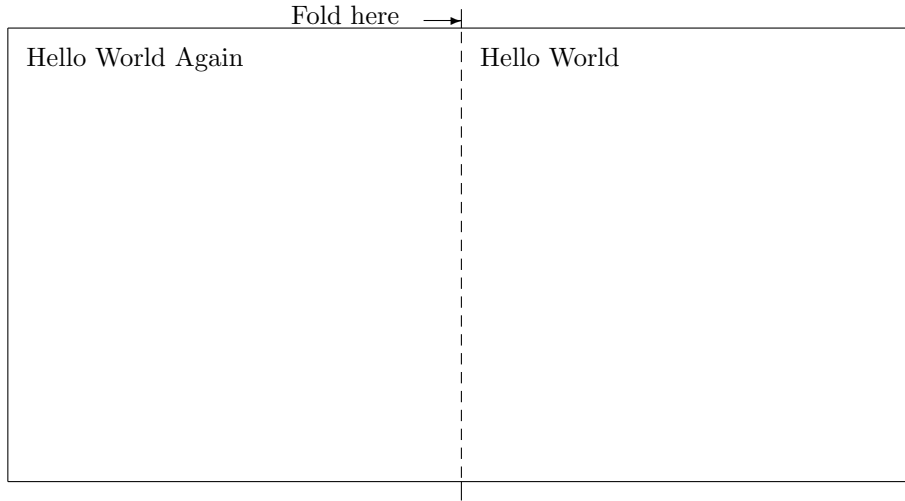


Figure 1: Example of bookletsheets output

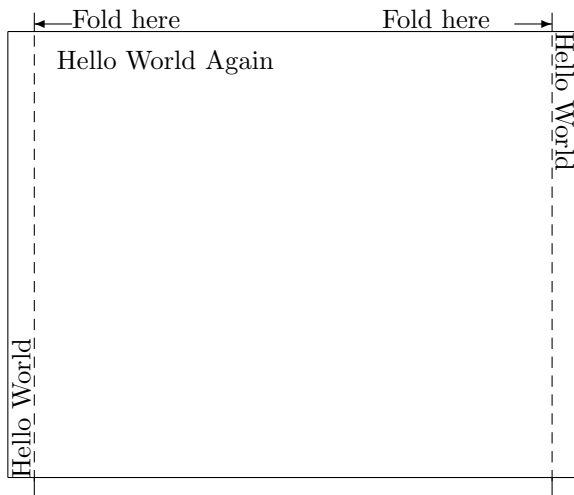


Figure 2: Example of backsheet output

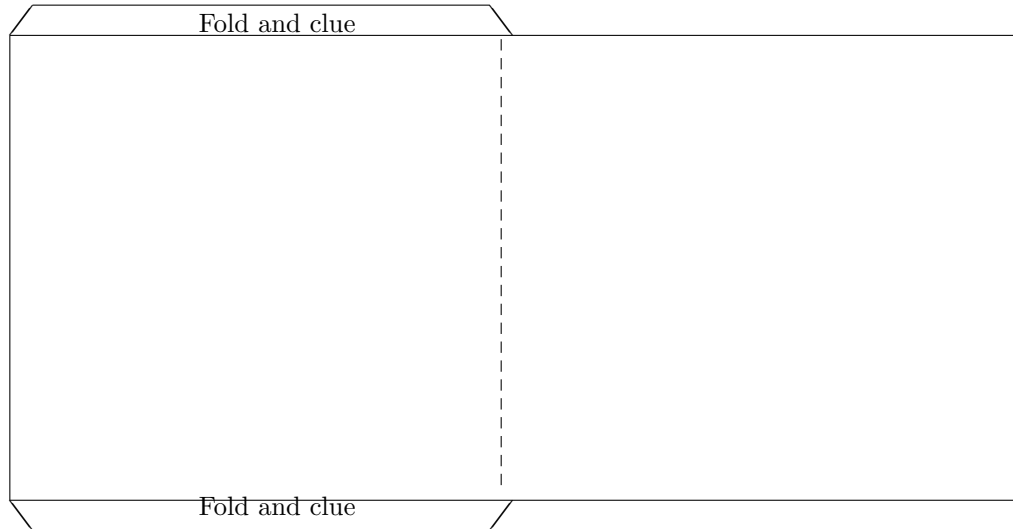


Figure 3: Example of sleevesheet output

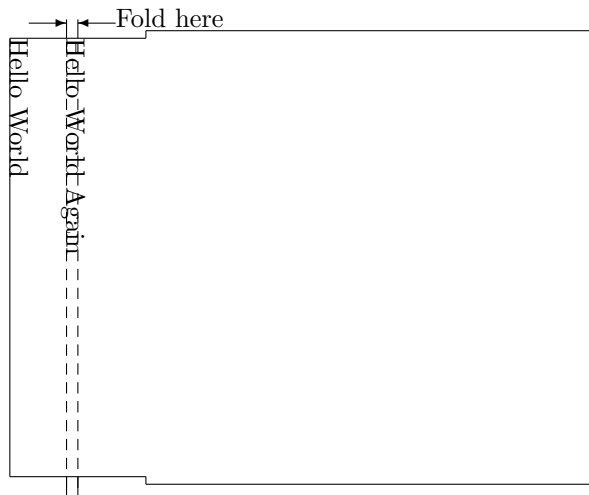


Figure 4: Example of singlesheet output

## 2.2 User setup

For each of the four environments, two user configurable lengths are defined. One is `\CD<type>TopMargin`, which determines the vertical space from the cover border to *<your text>*. The other is `\CD<type>Margin` which determines the horizontal space between the border and metayour text. The default values for these lengths are 5mm.

Here is the complete list of the lengths

<code>\CDbookletTopMargin</code>	<code>\CDbackTopMargin</code>
<code>\CDbookletMargin</code>	<code>\CDbackMargin</code>
<code>\CDsleeveTopMargin</code>	<code>\CDsingleTopMargin</code>
<code>\CDsleeveMargin</code>	<code>\CDsingleMargin</code>

For example, if you want to put a picture that fills out a complete page of the booklets, you can set `\CDbookletTopMargin` and `\CDbookletMargin` to `0pt`. For example

```
\documentclass{cd-cover}
\usepackage{epsfig}
\begin{document}
\parindent=0pt
\parskip=0pt
\CDbookletTopMargin=0pt
\CDbookletMargin=0pt
\begin{bookletsheets}
  \vspace{5mm} % These are specific for the file rh61img.ps
  \hspace{5mm}
  \begin{center}
    {\LARGE Debian GNU/Linux 2.2r5 --- potato}
  \end{center}
  \epsfig{file=rh61img.ps,width=11.99cm}
\end{bookletsheets}
\end{document}
```

Note that the graphics should not be 12cm exactly, but rather slightly less, say 11.99cm like above. This is to make sure that it only takes up one page, so that  $\LaTeX$  will not start a new page.

## 3 Credits

Many people have come up with suggestions, bugs, improvements to the first package version of `cd-cover`, and I'd like to thank them, especially for thier patience in me, for not getting this out sooner. So thanks to James A. Bednar, Steven Buehler, Patrik Carlsson, Jim Clark, Lionel Cons, Joerg Desch, Stefan A. Deutscher, Holger Dewes, Stephen Gildea, Anders Bruun Olsen, Rolf Niepraschk, Mark Peade,

Dominik Roettches, Carsten Schäfer, C. J. Walker, and Rainer Wiener. I may have forgotten some, for which I apologize.

And of course thanks to Donald E. Knuth for  $\text{\TeX}$ , Leslie Lamport for  $\text{\LaTeX}$ , Robin Fairbairns for CTAN and being very patient with me.

## 4 Copyright

$\text{\LaTeX} 2_{\epsilon}$  class cd-cover for typesetting a variety of cd covers Copyright ©1999 Christian Holm <cholm@nbi.dk>

This class is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the

Free Software Foundation, Inc.  
675 Mass Ave  
Cambridge  
MA 02139  
USA

## 5 Implementation

### 5.1 Initializing

#### Identification

First we must, as any good package or class should, identify ourselves, and tell what format we need. In this case it is ofcourse  $\text{\LaTeX} 2_{\epsilon}$ .

```
1 \NeedsTeXFormat{LaTeX2e}
2 \def\fileversion{v1.4}
3 \def\filedate{2002/01/28} % Format is YYYY/MM/DD
4 \ProvidesClass{cd-cover}[\filedate\space\fileversion
5 \space cd-cover class.]
```

## Lengths or dimensions

The next thing is to setup some lengths (dimensions) we need for various purposes. There is two temporary lengths `\temp@length` and `\temp@@length`, for use in variuos calculations; along width margin lengths, two for each type of environment, one for the odd pages `\booklet@odd@margins`, `\back@odd@margins`, `\sleeve@odd@margins`, and `\single@odd@margins`, and ofcourse the corresponding for the even pages `\booklet@even@margins`, `\back@even@margins`, `\sleeve@even@margins`, and `\single@even@margins`. These margin lengths is used so that we may do two-sided printing where the various covers align.

```
6 \newlength{\temp@length}
7 \newlength{\temp@@length}
8 \newlength{\booklet@even@margins}
9 \newlength{\booklet@odd@margins}
10 \newlength{\back@even@margins}
11 \newlength{\back@odd@margins}
12 \newlength{\sleeve@even@margins}
13 \newlength{\sleeve@odd@margins}
14 \newlength{\single@even@margins}
15 \newlength{\single@odd@margins}
```

## Options

`\cd@cover@setup@margins` This command sets up a number of dimensions, so that when doing two-side printing, the covers will be align on each side of the paper, and can neatly be cut out to provide a double-sided booklet, sleeve or cover. This is not implemeted fully yet. What is here is good for the booklet, but doesn't work for single other types of covers. It's mainly the horizontal alignment that's the problem.

```
16 \def\cd@cover@setup@margins{
17 % This is some old stuff, that I'm not sure is needed, but I leave it
18 % \global\paperheight=\temp@length
19 % \global\paperwidth=\temp@@length
20 \advance\temp@length-\booklet@height
21 \divide\temp@length2
22 \advance\temp@length-1in
23 \advance\temp@length-\headheight
24 \global\topmargin=\temp@length
25 \advance\temp@@length-\booklet@width
26 \divide\temp@@length2
27 \advance\temp@@length-1in
28 \oddsidemargin=\temp@@length}
29 \AtBeginDocument{\cd@cover@setup@margins}
```

We also provide a number of options, to specify the papersize. Notice, that since the article class will set the paper size, we use temporary local variables to hold the information, and then in `\cd@cover@setup@margins` we set the right dimensions.

```

30 \DeclareOption{a4paper}{
31   \global\temp@length=210mm
32   \global\temp@@length=297mm}
33 \DeclareOption{letterpaper}{
34   \global\temp@length=8.5in
35   \global\temp@@length=11in}
36 \DeclareOption{legalpaper}{
37   \setlength\temp@length{8.5in}
38   \setlength\temp@@length{14in}}
39 \DeclareOption{executivepaper}{
40   \setlength\temp@length{7.25in}
41   \setlength\temp@@length{10.5in}}

```

Notice, that some standard papersizes produces an error message, as they will be too small to hold the covers.

```

42 \DeclareOption{a5paper}{\cd@cover@paper@error{a5paper}}
43 \DeclareOption{b5paper}{\cd@cover@paper@error{b5paper}}

```

Also, all papersizes are already in ‘landscape’ version<sup>2</sup> so the landscape option is redundant with this class.

```

44 \DeclareOption{landscape}{
45   \ClassWarning{cd-cover}{Option ‘landscape’ redundant to this %
46     class.}}

```

Add an option to avoid foldlines

```

47 \newif\ifcd@cover@foldlines\cd@cover@foldlinestrue
48 \DeclareOption{nofoldlines}{\cd@cover@foldlinesfalse}
49 \DeclareOption{foldlines}{\cd@cover@foldlinestrue}

```

All other options are passed on to the standard article class, as expected. ‘letterpaper’ is then the default paper size, and we finally execute the options.

```

50 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
51 \ExecuteOptions{letterpaper}
52 \ProcessOptions\relax

```

`\cd@cover@paper@error` This is just an error message, so that the user doesn’t try to use a paper size that is too small to hold some of the covers.

```

53 \def\cd@cover@paper@error#1{
54   \ClassError{cd-cover}{^^J%
55     Paper format ‘#1’ is too small for the covers}{^^J%
56     You have given the option ‘#1’ to this class.^^J%
57     However, that paper size is too small to hold^^J%
58     the covers.^^J%
59     Please change the paper size option to another^^J%
60     format that CAN hold the covers (e.g., ‘a4paper’,^^J%
61     ‘letterpaper’, etc.)}}

```

---

<sup>2</sup>compared to the normal defintions in `article.cls`.



## Class and package loading

First we load the standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> class `article` with what ever options was passed from above. This provides the user with the standard environments and sectioning commands, and is therefore desired. The `article` class is chosen, since the sectioning commands of this class is appropriate for `cd-covers`. The package `rotating` is then loaded. This package facilitates rotation of text, using `\special` commands. This means, that this class may be sensible to the users PostScript driver. Also, as I myself suspected, and as others have confirmed, this package is flawed, meaning, that this class isn't as stable as it should be. However, the `rotating` package is the best available for the job, and hence used here<sup>3</sup>.

After loading the class, we set the paper height and width.

```
62 \LoadClass{article}
63 \paperheight=\temp@length
64 \paperwidth=\temp@length
65 \RequirePackage{rotating}
```

## Setting up fixed lengths

Next on the agenda is to define and setup some fixed lengths. These are the dimensions of the various cover styles. The really isn't much to say about these. I obtained these numbers by simple measuring some of my own (pre-printed) covers, with, as you can see, a metric meter stick.

```
66 \newlength{\booklet@width} \setlength{\booklet@width}{240mm}
67 \newlength{\booklet@height} \setlength{\booklet@height}{120mm}
68 \newlength{\back@width} \setlength{\back@width}{137mm}
69 \newlength{\back@height} \setlength{\back@height}{118mm}
70 \newlength{\back@slip@width} \setlength{\back@slip@width}{7mm}
71 \newlength{\sleeve@height} \setlength{\sleeve@height}{128mm}
72 \newlength{\sleeve@width} \setlength{\sleeve@width}{266mm}
73 \newlength{\single@height} \setlength{\single@height}{120mm}
74 \newlength{\single@width} \setlength{\single@width}{137mm}
75 \newlength{\single@slip@width} \setlength{\single@slip@width}{15mm}
76 \newlength{\single@back@width} \setlength{\single@back@width}{3mm}
```

## Setting up user lengths

The user also has some control of the page dimensions of the various cover types. These are provided via the below lengths. What the do should be obvious from the names<sup>4</sup>. We also set the default values of these lengths here.

```
77 \newlength{\CDbookletTopMargin} \setlength{\CDbookletTopMargin}{5mm}
78 \newlength{\CDbookletMargin} \setlength{\CDbookletMargin}{5mm}
```

---

<sup>3</sup>I'm sorry I don't have the time, or expertise, to write a new package, but that's life for you.

<sup>4</sup>Otherwise, refer to the documentation part of this document.

```

79 \newlength{\CDbackTopMargin}    \setlength{\CDbackTopMargin}{5mm}
80 \newlength{\CDbackMargin}      \setlength{\CDbackMargin}{5mm}
81 \newlength{\CDsleeveTopMargin}  \setlength{\CDsleeveTopMargin}{5mm}
82 \newlength{\CDsleeveMargin}    \setlength{\CDsleeveMargin}{5mm}
83 \newlength{\CDsingleTopMargin}  \setlength{\CDsingleTopMargin}{5mm}
84 \newlength{\CDsingleMargin}    \setlength{\CDsingleMargin}{5mm}

```

Finally we set the default page style to `empty`.

```
85 \pagestyle{empty}
```

## 5.2 The various cover types

The changing from one type of cover to the other is implemented using ‘page styles’. So each type of cover has its own page style. This has several advantages.

To those of you familiar with the first two (or was it three — I can’t remember) versions, this is where you really will note the difference. In version v1.0 from 1998/10/04, the interface was driven via commands. This limited a number of things, most noticeably the text length. You had to manually break each page up, which of course is very non- $\text{\TeX}$ -like. Secondly it limited the use of ‘floats’ such as tables and figures.

Well, in the *new* scheme, this is all changed. Since the `pagestyle` doesn’t effect the way pages are broken by  $\text{\TeX}$ , one no longer has to break up the pages manually! You can put as much text as you like into the pages,  $\text{\LaTeX}$  will just produce more pages as the text grows in length. Though page styles has *some* effect on the  $\text{\LaTeX}$   $2_\epsilon$  output routine and hence on floats, it is very minimal, and floats can be used extensively under this scheme.

Another advantage of the new scheme, is the use of user *configurable dimensions*. If all the boundary dimensions are set to zero, you can effectually fill out the whole cover with text. Though this doesn’t seem useful, consider the case where you want to fill out your cover with some marvelous picture you have created in Gimp, PhotoShop, or just downloaded from the internet!

### 5.2.1 The booklet type

Let me just say a few words on page styles. Every page style defines the commands `\@oddhead`, `\@evenhead`, `\@oddfoot`, and `\@evenfoot` to be some command containing some text<sup>5</sup>. There *must* be some text present, or otherwise  $\text{\TeX}$  will respond with a `Underfull \hbox...` while output is active, which is quite annoying. For this reason all the *border* macros contain a `\hfill` as the very last thing. Also,  $\text{\LaTeX}$  assumes, that the four macros above are no wider than `\textwidth` and no taller than `\headheight` or `\footskip`, whatever appropriate.

For the page styles at hand, I decided to `\let` the above two `\@...head` commands

---

<sup>5</sup>In  $\text{\TeX}$  terms this means *horizontal mode stuff*.

to the *border* commands below, and the two `\@...foot` to `\@empty`. Now these *border* commands are apparently *much* taller than `\headheight`, but if one takes a closer look, they actually aren't.

The first part of every one of them is one ore more `\kerns`. `\kerns` in T<sub>E</sub>X mind<sup>6</sup> doesn't take up space in anyway, so these commands are *invisible* in that sense. The next thing is a `picture` environment, and this is defined in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> so that it doesn't take up any space in the previous sense. Finishing of the macros is a `\hfill`, wich is there for reasons explained above. So the only command that is 'space visible' in the *border* commands, is the `\hfill!` Now if that isn't neat, I don't know what is. That said, I won't comment anymore on the page style definitons and move straight on to the rest of the macros and commands.

`\ps@bookletsheet`

```
86 \def\ps@bookletsheets{
87   \let\@oddhead\booklet@border\let\@evenhead\@oddhead
88   \let\@oddfont\@empty\let\@evenfont\@oddfont}
```

`\init@booklet` We need to do some manipulation of the dimensions `\textheight`, `\textwidth`, `\headsep` and since the booklet pages are set in double column mode, also on `\columnsep` and `\colhht`.

```
89 \def\init@booklet{
```

What we do is to set the text height limit to the proper value for the booklet — as defined in `\booklet@height` — then correct for the space wanted above and below the text — as defined in `\CDbookletTopMargin`. This ofcourse only sets the *lower* margin, and has absolutely nothing to do with the upper margin. To do that we fix the value of `\headsep` to `\CDbookletTopMargin`.

```
90   \temp@length=\booklet@height
91   \advance\temp@length-2\CDbookletTopMargin
92   \global\textheight=\temp@length
93   \global\headsep=\CDbookletTopMargin
```

The next thing is to set the text width. This limit is again set to it's appropriate value for this type of cover — defined in `\booklet@width` — and then corrected for for the wanted margin width — taken from `\CDbookletMargin`.

```
94   \temp@length=\booklet@width
95   \advance\temp@length-2\CDbookletMargin
96   \global\textwidth=\temp@length
```

A little extra care is needed since we type set in two column mode. We In the case, that the first page contains this sort of cover, we *have to* set `\colhht` equal to `\textheight` manually.

---

<sup>6</sup>Or mouth or stomach, I don't remember wich one it is, as Donald E. Knuth calls it in the T<sub>E</sub>Xbook.

The reason is a bit complicated, and I will not go into much detail about it. The point is, that L<sup>A</sup>T<sub>E</sub>X uses `\@colht` as the limit for the height of the columns. Now, when constructing the first column, `\@colht` still has the old value of `\textheight`, as set in the beginning of the document, and this is properly not what we want. *After* making the first column, it sets `\@colht` to the present value of `\textheight`, which is what we want, but then it is too late. The *only* way to get around this problem<sup>7</sup>, is to manually set `\@colht`. One should not be concerned on that this dimension is hand-set on subsequent pages.

The second thing to do, is to set `\columnsep` to whatever value the margins should have — as defined in `\CDbookletMargin`.

```
97 \global\@colht=\textheight
98 \global\columnsep=2\CDbookletMargin}
```

`bookletsheets` Now for the environment it self. We start off with initializing the dimensions as outlined above, and then we start the two column mode and set the page style to `bookletsheets`. To finish of the the environment, we make sure that the last page has a border by calling `\clearpage`, and revert to the default page style.

```
99 \newenvironment{bookletsheets}{
100 \init@booklet\twocolumn\pagestyle{bookletsheets}}{
101 \clearpage\pagestyle{empty}}
```

`\booklet@border` First we want to back up the margin width (`\CDbookletMargin`), so that the border will be properly spaced from the text. Secondly we want to draw the border.

This border isn't very complex. It's just a rectangular with two small lines indicating a folding line. For the users convenience, we put an arrow and some text next to the folding line.

As explained above this command is 'space invisible', except for the final `\hfill`, which is exactly what we want.

```
102 \def\booklet@border{%
103 \kern-\CDbookletMargin%
104 \setlength{\unitlength}{1mm}%
105 \begin{picture}(100,1)%
106 \put(0,0){\line(1,0){240}}% Top most horizontal line
107 \put(0,0){\line(0,-1){120}}% Left most vertical line
108 \put(0,-120){\line(1,0){240}}% Bottom most horizontal line
109 \put(240,0){\line(0,-1){120}}% Right most vertical line
110 \put(120,0){\line(0,1){5}}% Top folding mark
111 \put(120,-120){\line(0,-1){5}}% Bottom folding mark
112 \ifcd@cover@foldlines
113 \multiput(120,-1)(0,-5){24}{\line(0,-1){3}} % Fold line
114 \fi
115 \put(110,2){\vector(1,0){10}}% Arrow
```

<sup>7</sup>Or is ts a bug in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

```

116 \put(95,1){Fold here}%           Helping text
117 \end{picture}\hfill}

```

### 5.2.2 The back type

One could argue, that the page style scheme outlined on page 10 really isn't that advantages in this case, and that a `minipage`-like solution would be better, since the text in this environment shouldn't be more than *one* page, since there is only room for one back sheet on a normal cd-cover. While this is true, I decided, that it should be up to the user to restrain him/her-self from putting tom much text into these kinds of pages.

`\ps@backsheet` Page style for the backpage.

```

118 \def\ps@backsheet{
119 \let\@oddhead\back@border\let\@evenhead\@oddhead
120 \let\@oddfoot\@empty\let\@evenfoot\@empty}

```

`\init@back` This macro is similar to `\init@booklet` comment above, so I will suffice to say the instead of `\CDbooklet...` above read `\CDback...`, and the two-column notes are irrelevant.

```

121 \def\init@back{
122 \temp@length=\back@height
123 \advance\temp@length-2\CDbackTopMargin
124 \global\textheight=\temp@length
125 \temp@length=\back@width
126 \advance\temp@length-2\CDbackMargin
127 \global\textwidth=\temp@length
128 \global\headsep=\CDbackTopMargin}

```

`\back@title` This is a save box for the title on the back of the cover. See more below.

```

129 \newsavebox{\back@title}

```

`backsheet` This environment should take one argument, the title on the back of the cover.

`backsheet*` This will be type set two times; one on each end of the cover, rotated as to give the proper reading when your CD is put on the shelf. The argument is put into a `\parbox`, so that we may control the size of the text, and then put into a save box for later use (in `\back@border`).

The star'ed version `backsheet*` turns the back title opposite to the non-star'ed version. This is useful if one want's to typeset covers for double jewel cases. Also, in some countries, like Germany, they like having the text run the opposite direction, so for those users, that environment may be more appropriate.

`\ifback@anglereverse` To help distinguish between the two ways of rotation the back title, we have the macro `\ifback@anglereverse`, which is set to `\relax` if we're using the regular

non-star'ed version.

```
130 \newif\ifback@anglereverse\back@anglereversefalse
```

As in `bookletsheets` environment, we call the initializing macro, make sure we are in one-column mode, and set the page style. When finishing of the environment, we clear the page, to insure the last page style, and revert to the default page style.

```
131 \newenvironment{backsheet}[1]{
132   \back@anglereversefalse%
133   \init@back%
134   \savebox{\back@title}[\textheight]{%
135     \parbox[t][6mm]{\textheight}{#1}}
136   \onecolumn\pagestyle{backsheet}
137   \clearpage{\clearpage\pagestyle{empty}}
138 \newenvironment{backsheet*}[1]{
139   \back@anglereversetrue%
140   \init@back%
141   \savebox{\back@title}[\textheight]{%
142     \parbox[t][6mm]{\textheight}{#1}}
143   \onecolumn\pagestyle{backsheet}
144   \clearpage{\clearpage\pagestyle{empty}}
```

`\back@border` First we back up, just as in `\booklet@border`, but we also have to back up the length (height) of the slips, unlike above. Then the border is drawn with folding marks, and some help text and arrows.

```
145 \def\back@border{%
146   \kern-\CDbackMargin%
147   \kern-\back@slip@width%
148   \setlength{\unitlength}{1mm}%
149   \begin{picture}(0,0)%
150     \put(0,0){\line(1,0){151}}%      Top most horizontal line
151     \put(0,-118){\line(1,0){151}}%  Bottom most horizontal line
152     \put(0,0){\line(0,-1){118}}%    Left most vertical line
153     \put(151,0){\line(0,-1){118}}%  Right most vertical line
154     \put(7,0){\line(0,1){5}}%      Left top folding mark
155     \put(7,-118){\line(0,-1){5}}%  Left bottom folding mark
156     \ifcd@cover@foldlines
157       \multiput(7,-1)(0,-5){24}{\line(0,-1){3}} % Fold line
158     \fi
159     \put(144,0){\line(0,1){5}}%     Right top folding mark
160     \put(144,-118){\line(0,-1){5}}% Right bottom folding mark
161     \ifcd@cover@foldlines
162       \multiput(144,-1)(0,-5){24}{\line(0,-1){3}} % Fold line
163     \fi
164     \put(17,2){\vector(-1,0){10}}%  Left help arrow
165     \put(17,1){Fold here}%         Left help text
166     \put(134,2){\vector(1,0){10}}%  Right help arrow
```

```
167 \put(119,1){Fold here}%           Right help text
```

We finishes off with drawing the rotated titles. Please notice, that the left one, is the one closest to the opening of the CD cover. This should explain the somewhat strange rotating values.

Also, notice that we can rotate one or the other way, as explained above, conditionally on `\ifback@anglereverse{1}`.

```
168 \ifback@anglereverse%
169 \put(1,0){\begin{rotate}{-90}\usebox{\back@title}\end{rotate}}%
170 \put(150,-118){\begin{rotate}{90}\usebox{\back@title}\end{rotate}}%
171 \else%
172 \put(6,-118){\begin{rotate}{90}\usebox{\back@title}\end{rotate}}%
173 \put(145,0){\begin{rotate}{-90}\usebox{\back@title}\end{rotate}}%
174 \fi%
175 %\put(3,-118){\begin{rotate}{90}\usebox{\back@title}\end{rotate}}%
176 %\put(148,0){\begin{rotate}{-90}\usebox{\back@title}\end{rotate}}%
177 \end{picture}\hfill}
```

### 5.2.3 The sleeve type

This type shows how you can juggle around with  $\text{\LaTeX}$  and get it to do almost everything you want<sup>8</sup>. By altering a bit in the way  $\text{\LaTeXe}$  output two-column pages, I could get it to swap the columns, so that the last entered would come out first, and vice versa. This is exactly what we want for this page style, which is obvious when you see the finished output.

```
\ps@sleevesheet
```

```
178 \def\ps@sleevesheet{%
179 \let\@oddhead\sleeve@border\let\@evenhead\@oddhead
180 \let\@oddfoot\@empty\let\@evenfoot\@oddfoot}
```

```
\init@sleeve What I said about \init@booklet goes here with the exchange of \CDsleeve...
for \CDbooklet....
```

```
181 \def\init@sleeve{
182 \temp@length=\sleeve@height
183 \advance\temp@length-2\CDsleeveTopMargin
184 \global\textheight=\temp@length
185 \global\colht=\textheight
186 \temp@length=\sleeve@width
187 \advance\temp@length-2\CDsleeveMargin
188 \global\textwidth=\temp@length
189 \global\headsep=\CDsleeveTopMargin
190 \global\columnsep=2\CDsleeveMargin}
```

---

<sup>8</sup>I don't want to sound pretentious, but I was surprised how easily it all turned out to be.

`sleevesheet` Again we initialize, make sure we are in two column mode, and set the page style. The only thing to notice here, is we `\let \@outputdblcol` to `\sleeve@outputdblcol`, and when the environment ends, change back to the usual definition saved in `\ltx@outputdblcol`. It is important that the last change comes *after* the `\clearpage`, since otherwise, the last page will be output with the wrong output routine. Notice, that it doesn't make sense to make two-side printing for this kind of cover, so the scheme works on *any* page. Ofcourse, you should make the following page blank, by putting `\vfill\clearpage` after the environment.

```

191 \newenvironment{sleevesheet}{%
192   \init@sleeve
193   \twocolumn\pagestyle{sleevesheet}
194   \global\let\@outputdblcol\sleeve@outputdblcol}{
195   \clearpage\pagestyle{empty}
196   \global\let\@outputdblcol\ltx@outputdblcol}

```

`\sleeve@border` Two flaps is outlined for clueing.

```

197 \def\sleeve@border{%
198   \kern-\CDsleeveMargin%
199   \setlength{\unitlength}{1mm}%
200   \begin{picture}(100,1)%
201     \put(0,0){\line(3,4){6}}%           Top flap
202     \put(6,8){\line(1,0){121}}%
203     \put(133,0){\line(-3,4){6}}%
204     \ifcd@cover@foldlines
205       \multiput(133,-1)(0,-5){25}{\line(0,-1){3}} % Fold line
206     \fi
207     \put(50,1){Fold and clue}%         Help text in top flap
208     \put(0,-123){\line(3,-4){6}}%     Bottom flap
209     \put(6,-131){\line(1,0){121}}%
210     \put(133,-123){\line(-3,-4){6}}%
211     \put(50,-131){Fold and clue}%     Help text in bottom flap
212     \put(0,0){\line(1,0){266}}%       Top most horizontal line
213     \put(0,0){\line(0,-1){123}}%     Left most vertical line
214     \put(0,-123){\line(1,0){266}}%   Bottom most horizontal line
215     \put(266,0){\line(0,-1){123}}%   Right most vertical line
216   \end{picture}\hfill}

```

## 5.2.4 The single type

`\ps@singlesheet` To facilitate two side printing, we have to have two kinds of borders, since the border outline is asymmetric along a vertical line, hence `\@oddhead` and `\@evenhead` has different values.

```

217 \def\ps@singlesheet{
218   \let\@oddhead\odd@single@border
219   \let\@evenhead\even@single@border

```



```
220 \let\@oddfoot\@empty\let\@evenfoot\@oddfoot}
```

`\single@title` These two save boxes are used below.

```
\single@slip
221 \newsavebox{\single@title}
222 \newsavebox{\single@slip}
```

`\init@single` Again exchange `\CDsingle...` with `\CDback...` in the description of `\init@back`, and there you have it.

```
223 \def\init@single{
224   \temp@length=\single@height
225   \advance\temp@length-2\CDsingleTopMargin
226   \advance\temp@length-4mm
227   \global\textheight=\temp@length
228   \temp@length=\single@width
229   \advance\temp@length-2\CDsingleMargin
230   \global\textwidth=\temp@length
231   \global\headsep=\CDsingleTopMargin}
```

`singlesheet` As in the backsheet environment, we save the arguments in `\parbox`'es to control the size, and put all that into save boxes, which are used in `\odd@single@border`. Everything else is as above.

```
232 \newenvironment{singlesheet}[2]{
233   \init@single
234   \savebox{\single@title}[\textwidth]{%
235     \parbox[t][3mm]{\textwidth}{#1}}
236   \savebox{\single@slip}[\textwidth]{%
237     \parbox[t][\single@slip@width]{\textwidth}{#2}}
238   \onecolumn\pagestyle{singlesheet}
239   \clearpage{\clearpage\pagestyle{empty}}
```

`\odd@single@border` We define two border for the reasons stated above. They differ in that they are mirror reflections of each other, and the back title and slip text isn't put on even sides.

```
240 \def\odd@single@border{
241   \kern-\CDsingleMargin%
242   \kern-\single@slip@width%
243   \kern-\single@back@width%
244   \setlength{\unitlength}{1mm}%
245   \begin{picture}(0,0)%
246     \put(0,0){\line(0,-1){116}}%           Left most vertical line
247     \put(0,0){\line(1,0){36}}%           First top horizontal line
248     \put(36,0){\line(0,1){2}}%           Jump
249     \put(15,0){\line(0,1){5}}%           Fold mark
250     \ifcd@cover@foldlines
251     \multiput(15,-1)(0,-5){24}{\line(0,-1){3}} % Fold line
252   \fi}
```

```

253 \put(18,0){\line(0,1){5}}% Fold mark
254 \ifcd@cover@foldlines
255 \multiput(18,-1)(0,-5){24}{\line(0,-1){3}} % Fold line
256 \fi
257 \put(36,2){\line(1,0){119}}% Second top line
258 \put(155,2){\line(0,-1){120}}% Right most vertical line
259 \put(0,-116){\line(1,0){36}}% First bottom line
260 \put(36,-116){\line(0,-1){2}}% Jump
261 \put(15,-116){\line(0,-1){5}}% Fold Mark
262 \put(18,-116){\line(0,-1){5}}% Fold Mark
263 \put(36,-118){\line(1,0){119}}% Second bottom line
264 \put(5,4){\vector(1,0){10}}% Arrows
265 \put(28,4){\vector(-1,0){10}}% Arrows
266 \put(28,3){Fold here}% Help text
267 \put(0,0){\begin{rotate}{-90}\usebox{\single@title}\end{rotate}}%
268 \put(15,0){\begin{rotate}{-90}\usebox{\single@slip}\end{rotate}}%
269 \end{picture}\hfill}

```

`\even@single@border`

```

270 \def\even@single@border{
271 \kern-\CDsingleMargin%
272 \setlength{\unitlength}{1mm}%
273 \begin{picture}(0,0)%
274 \put(0,2){\line(0,-1){120}}% 1st left vert line
275 \put(119,0){\line(1,0){36}}% 2nd top horiz line
276 \put(119,0){\line(0,1){2}}% 2nd top right vert line
277 \put(137,0){\line(0,1){5}}% 1st top fold mark
278 \put(140,0){\line(0,1){5}}% 2nd top fold mark
279 \put(0,2){\line(1,0){119}}% 1st top horiz line
280 \put(155,0){\line(0,-1){116}}% 1st right vert line
281 \put(119,-116){\line(1,0){36}}% 2nd bototm horiz line
282 \put(119,-116){\line(0,-1){2}}% 2nd bottom right vert line
283 \put(137,-116){\line(0,-1){5}}% 1st bottom fold mark
284 \put(140,-116){\line(0,-1){5}}% 2nd bottom fold mark
285 \put(0,-118){\line(1,0){119}}% 1st bottom horiz line
286 \put(127,4){\vector(1,0){10}}% 1st help arrow
287 \put(150,4){\vector(-1,0){10}}% 2nd help arrow
288 \put(112,3){Fold here}% Help text
289 \end{picture}\hfill}

```

### 5.3 Modified output routines

Here are some modifications to the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> output routine to *invert* the page orders in some of the environments, such as `sleeve`. First we save the old definition in `\ltx@outputdblcol`.

```

290 \let\ltx@outputdblcol\@outputdblcol

```

`\sleeve@outputdblcol` Then we go on to define `\sleeve@outputdblcol`, changing the order of column output.

The first part is the same as in `ltoutput.dtx`, so no need to comment on that here.

```
291 \def\sleeve@outputdblcol{%
292   \if@firstcolumn
293     \global\@firstcolumnfalse
294     \global\setbox\@leftcolumn\box\@outputbox
295   \else
296     \global\@firstcolumntrue
```

Now for the building of the actual output box, we *interchange* the order of `\@leftcolumn` and `\@outputbox` in the body definition of `\@outputbox`, tricky isn't it!

```
297   \setbox\@outputbox\vbox{%
298     \hb@xt@\textwidth{%
299       \hb@xt@\columnwidth{%
300         \box\@outputbox\hss}% \@leftcolumn changed to \@outputbox
301         \hfil\vrule\@width\columnseprule\hfil
302         \hb@xt@\columnwidth{%
303           \box\@leftcolumn\hss}}}% \@outputbox changed to \@leftcolumn
304   \@combinedblfloats
305   \@outputpage
306   \begingroup
307     \@dblfloatplacement
308     \@startdblcolumn
309     \@whiles\if@colmade\fi
310       {\@outputpage
311         \@startdblcolumn}%
312   \endgroup
313   \fi
314 }
```

## 5.4 The final stuff

If we can figure out the graphics driver automatically, then we emit a `\special` or equivalent command to the output file. If we can not find the driver, we warn the user that the printing should be done in landscape mode.

```
315 \def\cdcover@endmessage{cd-cover class: Remember that this document is
316   type set in landscape mode, and therefore dvips should be
317   passed the '-t landscape' option.}
318 \@ifundefined{Gin@driver}{%
319   \AtEndDocument{\typeout{\cdcover@endmessage}}}%
320 \filename@parse{\Gin@driver}%
321 \def\reserved@a{dvips}%
322 \ifx\filename@base\reserved@a\relax%
```

```

323 \message{DVIPS driver found}
324 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
325 \AtBeginDvi{\special{papersize=\the\paperheight,\the\paperwidth}}%
326 \else
327 \def\reserved@a{pdfTeX}\relax%
328 \ifx\filename@base\reserved@a
329 \message{PDF driver found}
330 \pdfpagewidth=\the\paperwidth\pdfpageheight=\the\paperheight%
331 \else
332 \def\reserved@a{VTeX}\relax%
333 \ifx\filename@base\reserved@a
334 \message{VTeX driver found}
335 \mediawidth=\the\paperwidth\mediaheight=\the\paperheight%
336 \else
337 \AtEndDocument{\typeout{\cdcover@endmessage}}
338 \fi
339 \fi
340 \fi}

```

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<b>Symbols</b>	<code>\@outputpage</code> .. 305, 310	<code>\back@odd@margin</code> s . 11
<code>\@colht</code> ..... 97, 185	<code>\@startdblcolumn</code> ..	<code>\back@slip@width</code> 70, 147
<code>\@combinedblfloats</code> . 304	..... 308, 311	<code>\back@title</code> 129, 134,
<code>\@dblfloatplacement</code> 307	<code>\@whiles</code> w ..... 309	141, 169, 170,
<code>\@empty</code> 88, 120, 180, 220	<code>\@width</code> ..... 301	172, 173, 175, 176
<code>\@evenfoot</code> .....		<code>\back@width</code> .... 68, 125
. 88, 120, 180, 220	<b>A</b>	<code>backsheet</code> (environ-
<code>\@evenhead</code> .....	<code>\advance</code> . 20, 22, 23,	ment) ..... <u>130</u>
. 87, 119, 179, 219	25, 27, 91, 95,	<code>backsheet*</code> (environ-
<code>\@firstcolumnfalse</code> . 293	123, 126, 183,	ment) ..... <u>130</u>
<code>\@firstcolumntrue</code> . 296	187, 225, 226, 229	<code>\begin</code> 105, 149, 169,
<code>\@ifundefined</code> ..... 318	<code>\AtBeginDocument</code> .. 29	170, 172, 173,
<code>\@leftcolumn</code> .....	<code>\AtBeginDvi</code> ... 324, 325	175, 176, 200,
.... 294, 300, 303	<code>\AtEndDocument</code> 319, 337	245, 267, 268, 273
<code>\@oddf</code> oot .....	<b>B</b>	<code>\begingroup</code> ..... 306
. 88, 120, 180, 220	<code>\back@anglereversefalse</code>	<code>\booklet@border</code> 87, <u>102</u>
<code>\@od</code> head .....	..... 130, 132	<code>\booklet@even@margin</code> s
. 87, 119, 179, 218	<code>\back@anglereverset</code> true	..... 8
<code>\@outputbox</code> .....	..... 139	<code>\booklet@height</code> ...
. 294, 297, 300, 303	<code>\back@border</code> .. 119, <u>145</u>	..... 20, 67, 90
<code>\@outputdblcol</code> ....	<code>\back@even@margin</code> s . 10	<code>\booklet@odd@margin</code> s 9
.... 194, 196, 290	<code>\back@height</code> ... 69, 122	<code>\booklet@width</code> 25, 66, 94

bookletsheets (environment) . . . .	99	240, 270, 291, 315, 321, 327, 332	\hss . . . . .	300, 303
\box . . . . .	294, 300, 303	\divide . . . . .	21, 26	
<b>C</b>				
\cd@cover@foldlinesfalse . . . . .	48	\else . . . . .	171, 295, 326, 331, 336	
\cd@cover@foldlinestrue . . . . .	47, 49	\end . . . . .	117, 169, 170, 172, 173, 175–177, 216, 267–269, 289	
\cd@cover@paper@error . . . . .	42, 43, 53	\endgroup . . . . .	312	
\cd@cover@setup@margins . . . . .	16	environments:		
\CDbackMargin . . . . .	80, 126, 146	backsheet . . . . .	130	
\CDbackTopMargin . . . . .	79, 123, 128	backsheet* . . . . .	130	
\CDbookletMargin . . . . .	78, 95, 98, 103	bookletsheets . . . . .	99	
\CDbookletTopMargin . . . . .	77, 91, 93	singlesheet . . . . .	232	
\cdcover@endmessage . . . . .	315, 319, 337	sleevesheet . . . . .	191	
\CDsingleMargin . . . . .	84, 229, 241, 271	\even@single@border . . . . .	219, 270	
\CDsingleTopMargin . . . . .	83, 225, 231	\ExecuteOptions . . . . .	51	
\CDsleeveMargin . . . . .	82, 187, 190, 198	<b>F</b>		
\CDsleeveTopMargin . . . . .	81, 183, 189	\fi 114, 158, 163, 174, 206, 252, 256, 309, 313, 338–340		
\ClassError . . . . .	54	\filedate . . . . .	3, 4	
\ClassWarning . . . . .	45	\filename@base . . . . .	322, 328, 333	
\clearpage . . . . .	101, 137, 144, 195, 239	\filename@parse . . . . .	320	
\columnsep . . . . .	98, 190	\fileversion . . . . .	2, 4	
\columnseprule . . . . .	301	<b>G</b>		
\columnwidth . . . . .	299, 302	\Gin@driver . . . . .	320	
\CurrentOption . . . . .	50	\global . . . . .	18, 19, 24, 31, 32, 34, 35, 92, 93, 96–98, 124, 127, 128, 184, 185, 188–190, 194, 196, 227, 230, 231, 293, 294, 296	
<b>D</b>				
\DeclareOption . . . . .	30, 33, 36, 39, 42–44, 48–50	<b>H</b>		
\def . . . . .	2, 3, 16, 53, 86, 89, 102, 118, 121, 145, 178, 181, 197, 217, 223,	\hb@xt@ . . . . .	298, 299, 302	
<b>E</b>				
<b>I</b>				
<b>K</b>				
<b>L</b>				
<b>M</b>				
<b>N</b>				
\else . . . . .	171, 295, 326, 331, 336	\headheight . . . . .	23	
\end . . . . .	117, 169, 170, 172, 173, 175–177, 216, 267–269, 289	\headsep 93, 128, 189, 231		
\endgroup . . . . .	312	\hfil . . . . .	301	
environments:		\hfill . . . . .	117, 177, 216, 269, 289	
backsheet . . . . .	130			
backsheet* . . . . .	130	\if@fcolmade . . . . .	309	
bookletsheets . . . . .	99	\if@firstcolumn . . . . .	292	
singlesheet . . . . .	232	\ifback@anglereverse . . . . .	130, 168	
sleevesheet . . . . .	191	\ifcd@cover@foldlines . . . . .	47, 112, 156, 161, 204, 250, 254	
\even@single@border . . . . .	219, 270	\ifx . . . . .	322, 328, 333	
\ExecuteOptions . . . . .	51	\init@back 121, 133, 140		
<b>F</b>				
<b>I</b>				
<b>K</b>				
<b>L</b>				
<b>M</b>				
<b>N</b>				
\fi 114, 158, 163, 174, 206, 252, 256, 309, 313, 338–340		\init@booklet . . . . .	89, 100	
\filedate . . . . .	3, 4	\init@single . . . . .	223, 233	
\filename@base . . . . .	322, 328, 333	\init@sleeve . . . . .	181, 192	
\filename@parse . . . . .	320	<b>K</b>		
\fileversion . . . . .	2, 4	\kern . . . . .	103, 146, 147, 198, 241–243, 271	
<b>G</b>				
<b>L</b>				
<b>M</b>				
<b>N</b>				
\Gin@driver . . . . .	320	\let . . . . .	87, 88, 119, 120, 179, 180, 194, 196, 218–220, 290	
\global . . . . .	18, 19, 24, 31, 32, 34, 35, 92, 93, 96–98, 124, 127, 128, 184, 185, 188–190, 194, 196, 227, 230, 231, 293, 294, 296	\line . . . . .	106–111, 113, 150–155, 157, 159, 160, 162, 201–203, 205, 208–210, 212–215, 246–249, 251, 253, 255, 257–263, 274–285	
\hb@xt@ . . . . .	298, 299, 302	\LoadClass . . . . .	62	
\headheight . . . . .	23	\ltx@outputdblcol . . . . .	196, 290	
\headsep 93, 128, 189, 231		<b>M</b>		
\hfil . . . . .	301	\mediaheight . . . . .	335	
\hfill . . . . .	117, 177, 216, 269, 289	\mediawidth . . . . .	335	
<b>H</b>				
<b>I</b>				
<b>K</b>				
<b>L</b>				
<b>M</b>				
<b>N</b>				
\if@fcolmade . . . . .	309	\message . . . . .	323, 329, 334	
\if@firstcolumn . . . . .	292	\multiput . . . . .	113, 157, 162, 205, 251, 255	
\ifback@anglereverse . . . . .	130, 168	<b>N</b>		
\ifcd@cover@foldlines . . . . .	47, 112, 156, 161, 204, 250, 254	\NeedsTeXFormat . . . . .	1	
\ifx . . . . .	322, 328, 333	\newenvironment 99, 131, 138, 191, 232		
\init@back 121, 133, 140		\newif . . . . .	47, 130	
\init@booklet . . . . .	89, 100	\newlength . . . . .	6–15, 66–84	
\init@single . . . . .	223, 233	\newsavebox 129, 221, 222		
\init@sleeve . . . . .	181, 192			

<b>O</b>		<b>R</b>		<code>\space</code> . . . . . 4, 5
<code>\odd@single@border</code> .	218, <a href="#">240</a>	<code>\relax</code> . 52, 322, 327, 332	<code>\RequirePackage</code> . . . 65	<code>\special</code> . . . . . 324, 325
. . . . . 218, <a href="#">240</a>		<code>\reserved@a</code> 321, 322,		<b>T</b>
<code>\oddsidemargin</code> . . . . 28		327, 328, 332, 333		<code>\temp@length</code> . . . . .
<code>\onecolumn</code> 136, 143, 238				. . 7, 19, 25–28,
<b>P</b>		<b>S</b>		32, 35, 38, 41, 64
<code>\pagestyle</code> . . . . 85,		<code>\savebox</code> . . . . .	<code>\temp@length</code> . 6, 18,	
100, 101, 136,		. 134, 141, 234, 236	20–24, 31, 34,	
137, 143, 144,		<code>\setbox</code> . . . . . 294, 297	37, 40, 63, 90–	
193, 195, 238, 239		<code>\setlength</code> 37, 38, 40,	92, 94–96, 122–	
<code>\paperheight</code> . 18, 63,		41, 66–84, 104,	127, 182–184,	
324, 325, 330, 335		148, 199, 244, 272	186–188, 224–230	
<code>\paperwidth</code> . . 19, 64,		<code>\single@back@width</code> .	<code>\textheight</code> . . . . .	
324, 325, 330, 335		. . . . . 76, 243	. . . 92, 97, 124,	
<code>\parbox</code> 135, 142, 235, 237		<code>\single@even@margins</code> 14	134, 135, 141,	
<code>\PassOptionsToClass</code> 50		<code>\single@height</code> . 73, 224	142, 184, 185, 227	
<code>\pdfpageheight</code> . . . . 330		<code>\single@odd@margins</code> 15	<code>\textwidth</code> . . . . .	
<code>\pdfpagewidth</code> . . . . 330		<code>\single@slip</code> . . . . .	. . 96, 127, 188,	
<code>\ProcessOptions</code> . . . 52		. . . . . <a href="#">221</a> , 236, 268	230, 234–237, 298	
<code>\ProvidesClass</code> . . . . . 4		<code>\single@slip@width</code> .	<code>\the</code> . . 324, 325, 330, 335	
<code>\ps@backsheet</code> . . . . . <a href="#">118</a>		. . . . . 75, 237, 242	<code>\topmargin</code> . . . . . 24	
<code>\ps@bookletsheet</code> . . . <a href="#">86</a>		<code>\single@title</code> . . . . .	<code>\twocolumn</code> . . . 100, 193	
<code>\ps@bookletsheets</code> . . . 86		. . . . . <a href="#">221</a> , 234, 267	<code>\typeout</code> . . . . . 319, 337	
<code>\ps@singlesheet</code> . . . . <a href="#">217</a>		<code>\single@width</code> . . 74, 228	<b>U</b>	
<code>\ps@sleevesheet</code> . . . . <a href="#">178</a>		<code>singlesheet</code> (environ-	<code>\unitlength</code> . . . 104,	
<code>\put</code> . . . . . 106–111,		ment) . . . . . <a href="#">232</a>	148, 199, 244, 272	
115, 116, 150–		<code>\sleeve@border</code> 179, <a href="#">197</a>	<code>\usebox</code> . . . . . 169,	
155, 159, 160,		<code>\sleeve@even@margins</code> 12	170, 172, 173,	
164–167, 169,		<code>\sleeve@height</code> . 71, 182	175, 176, 267, 268	
170, 172, 173,		<code>\sleeve@odd@margins</code> 13		
175, 176, 201–		<code>\sleeve@outputdblcol</code>	<b>V</b>	
203, 207–215,		. . . . . 194, <a href="#">291</a>	<code>\vbox</code> . . . . . 297	
246–249, 253,		<code>\sleeve@width</code> . . 72, 186	<code>\vector</code> 115, 164, 166,	
257–268, 274–288		<code>sleevesheet</code> (environ-	264, 265, 286, 287	
		ment) . . . . . <a href="#">191</a>	<code>\vrule</code> . . . . . 301	