

# A CD Cover Class

Sebastiano Vigna

Printed September 8, 2005

## 1 Introduction

The purpose of this class is to print CD covers. The main design line is allowing the creation of labels with minimum effort, without restraining the freedom to customise. There is also some support for multiple cover printing. Since version 1.2, slim CD boxes are supported.

Each CD cover is created by a number of commands which set the content of the front cover, back cover, track lists etc. After everything is ready, additional commands actually generate the covers. This is a very simple example:

```
\documentclass{cd}
\begin{document}

\covertext{
The Artist\\
\bfseries The Title
}

\leftspine{THE ARTIST}
\centerspine{THE TITLE}

\lefttracklist{
\track Song 1
\track Song 2
\track Song 3
}

\leftinfo{Words and Music by The Artist}

\makecover\par
\makeback\par
\end{document}
```

By compiling the file above, you will obtain your first CD cover. Using `\makeslimcover` instead of `\makecover` and `\makeback`, you will obtain a single cover for a slim CD box.

Equivalently, you can create a file `CD.dat` containing the lines between `\begin{document}` and `\makecover` and compile with  $\text{\LaTeX}$  the file `CD.tex` (or `slimCD.tex`). This is a better mechanism—each CD should have its own data (`.dat`) file, which is run through the driver file `CD.tex` or the more powerful list

mechanism described below. This also allows to set some parameters one for all (for instance, the font family) in the driver file. My driver file, for instance, is as follows (see below for the non-standard commands):

```

\documentclass[a4paper]{cd}
\usepackage[latin1]{inputenc}
\usepackage{avant}
\renewcommand\rmdefault{\sfdefault}
\onecorrection{.2}
\begin{document}
\makeCD
\end{document}

```

The CD class loads the `article` class, so commands like `\Large` or `\smallskip` are available. However, the CD class provides its own precise size-switching commands, and for greater accuracy it is advisable to use L<sup>A</sup>T<sub>E</sub>X's `\[\langle vspace \rangle]` mechanism in order to generate vertical spacing.

Note that the class uses heavily the `rotating` package, so you must convert the resulting dvi file into PostScript®, or use directly `pdflatex`.

## 2 The Text Commands

The content of a CD cover are set using the self-explaining `\covertext`, `\backtext`, `\insidetext`, `\leftspine`, `\centerspine`, `\rightspine`, `\lefttracklist`, `\righttracklist`, `\leftinfo` and `\rightinfo` commands (`\insidetext`, `\leftspine`, `\centerspine` and `\rightspine` are ignored for slim covers). Note that by default the material contained in `\covertext`, `\backtext` and `\insidetext` is bottom-aligned, and the arguments of the spine commands must not contain line breaks. The left and right track lists should use the `\track` command, which inserts a `\par` and an automatically numbered box with the track number. Should you need to set manually the track number, use `\setindex{\langle n \rangle}`. The text contained in `\leftinfo` and `\rightinfo` is bottom-aligned just under the respective track lists. Note that if the right information or track list box is empty, the left one will span across the whole cover. By default everything is typeset with no justification, and no paragraph indentation. One tenth of the current baseline skip is inserted between paragraphs.

In extreme cases you may want to create different spines (e.g., for R.E.M.'s *Fables Of The Reconstruction*); the `\leftspinebis`, `\centerspinebis` and `\rightspinebis` commands allows you to insert different content into the “back” spine.

## 3 The Font Commands

The CD class provides some simple commands for switching the font dimension and line spacing. The command `\fh{\langle height \rangle}` sets the font height to the given number of points (line spacing is not affected), while `\fhb{\langle height \rangle}{\langle baselineskip \rangle}` sets both the font height and the baseline skip (usually 6/5 of the font height will work). Note that you can just write `\fh7` in order to switch to a 7 point font,

and that the `\fhb` command always sets `\parskip` to 1/10 of the current baseline skip, so `\par` will always space a little more than `\.`

When you issue a `\newcd` command, all fonts are reset to their default values. But there are a number of self-explaining commands, i.e., `\coverfont`, `\backfont`, `\insidefont`, `\spinefont`, `\tracklistfont`, `\infofont` and `\indexfont`, that allow to change the font assigned to a part of the cover. In fact, they are just one-argument macros whose arguments are expanded just before the corresponding text commands, and can contain other formatting parameters.

## 4 The Cover Creation Commands

Before setting the content of the cover, the `\newcd` command takes care of resetting everything to default values. In particular, `\backtext` is the same as `\covertext` (unless you change it explicitly), so usually you do not need to set the former. Analogously, `\backfont` is the same as `\coverfont`.

Once everything is set up, the `\makecover` and `\makeback` commands will create a cover and a back cover using the data set so far, whereas the `\makeslimcover` will create a slim cover. Both command have an optional argument that can contain any of the letters `lrtb` (left, right, top, bottom), which create the respective crop marks. The default value is `lrtb`. The possibility of partially eliminating crop marks is particularly useful when stacking several covers in the same sheet.

It is possible to create a single  $\text{\LaTeX}$  document containing a CD cover, but it is usually more useful to create a data file containing all CD-specific command, and include it from a “driver” file, containing the `\makeCD` or the `\makeslimCD` command. With no argument, it checks for the existence of a `CD.dat` file. If it exists, it is input and then the (slim) CD cover is generated. Otherwise, the user is asked for a data file name (the CD class will try automatically to append the `.dat` extension to the name), which is read and processed. Of course, the optional argument can be used to specify a data file name.

Having a database of data files is particularly useful when using the `\makelist` or the `\makeslimlist` commands, which process an entire list of CDs, printing one cover (or two back covers) per page; the crop marks are suitably aligned so to minimise the cutting effort. The CD list must be contained in a list file, one data file name per line. With no argument, `\makelist` and `\makeslimlist` check for the existence of a `CD.lst` file. If it exists, it is input; otherwise, the user is asked for a list file name (the CD class will try automatically to append the `.lst` extension to the name), which is read and processed. Again, the optional argument can be used to specify a list file name.

## 5 The options

You can pass to the CD class all the options of the `article` class (e.g., paper size). Moreover, there are options `aligncovertop`, `aligninsidetop`, `alignbacktop` and `aligntop` (the last one resumes the first three ones), and analogously `aligncovercenter`, etc. that allow to change the default alignment behaviour. Finally, the `alignspine` option forces vertical centering of the spine text on the “real” height of the box involved, rather than on the height of a generic upper case character. This is not usually what you want, since, e.g., accents can lead to

ugly results. Experiment.

## 6 Getting Obsessed

PostScript fonts usually are set up in such a way that the metric of all digits is the same, regardless of the actual appearance. This (in particular with sans-serif fonts) can lead to a very ugly alignment of two-digit track numbers in which either the first or the last digit is a 1. The solution is to put in the preamble a `\onecorrection{fraction}` command: the positioning of two-digit numbers either starting or ending with 1 will be corrected by the given fraction of the width of a 1. For instance, `\onecorrection{.2}` works great for AvantGarde. The values for other fonts must be set by trial-and-error.

## 7 The Code

First of all we manage all options. This is done with a `\newif` for `alignspine`, and by defining suitably some macros representing the alignment option for the cover, inside and back text. Default is `b`. All options we do not process are passed to the `article` class.

```
1 (*class)
2 \newif\if@lignspine
3 \@lignspinefalse
4
5 \DeclareOption{alignspine}{\@lignspinettrue}
6
7 \def\@ligncover{b}
8 \def\@ligninside{b}
9 \def\@lignback{b}
10
11 \DeclareOption{aligncovertop}{\def\@ligncover{t}}
12 \DeclareOption{aligninsidetop}{\def\@ligninside{t}}
13 \DeclareOption{alignbacktop}{\def\@lignback{t}}
14
15 \DeclareOption{aligntop}{%
16   {\ExecuteOptions{aligncovertop,aligninsidetop,alignbacktop}}
17
18 \DeclareOption{aligncovercenter}{\def\@ligncover{c}}
19 \DeclareOption{aligninsidecenter}{\def\@ligninside{c}}
20 \DeclareOption{alignbackcenter}{\def\@lignback{c}}
21
22 \DeclareOption{aligncenter}{%
23   {\ExecuteOptions{aligncovercenter,aligninsidecenter,alignbackcenter}}
24
25 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
26
27 \ProcessOptions\relax
28 \LoadClass{article}
29 \RequirePackage{rotating}
```

The `\onecorrection` command defines a the fraction used for correcting the alignment of 1's. The default is 0.

```
30 \DeclareRobustCommand*\onecorrection[1]{\def\onec@rrfrac{#1}}
31 \onecorrection{0}
```

Now we have all the font and text declaration commands. They just define a certain macro to be their argument. This “double indirection” allows for default text definition, e.g., the back text and font is equal the cover ones unless otherwise specified.

```
32 \DeclareRobustCommand*\coverfont[1]{\def\coverf@nt{#1}}
33 \DeclareRobustCommand*\backfont[1]{\def\backf@nt{#1}}
34 \DeclareRobustCommand*\insidefont[1]{\def\insidef@nt{#1}}
35 \DeclareRobustCommand*\spinefont[1]{\def\spinef@nt{#1}}
36 \DeclareRobustCommand*\tracklistfont[1]{\def\tracklistf@nt{#1}}
37 \DeclareRobustCommand*\infofont[1]{\def\infof@nt{#1}}
38 \DeclareRobustCommand*\indexfont[1]{\def\indexf@nt{#1}}
39
40 \DeclareRobustCommand*\lefttracklist[1]{\def\lefttr@cklist{#1}}
41 \DeclareRobustCommand*\righttracklist[1]{\def\righttr@cklist{#1}}
42 \DeclareRobustCommand*\leftinfo[1]{\def\leftinf@{#1}}
43 \DeclareRobustCommand*\rightinfo[1]{\def\rightinf@{#1}}
44 \DeclareRobustCommand*\covertext[1]{\def\c@verttext{#1}}
45 \DeclareRobustCommand*\backtext[1]{\def\b@cktext{#1}}
46 \DeclareRobustCommand*\insidetext[1]{\def\insid@text{#1}}
47
48 \DeclareRobustCommand*\leftspine[1]{\def\leftspin@{#1}}
49 \DeclareRobustCommand*\centerspine[1]{\def\centerspin@{#1}}
50 \DeclareRobustCommand*\rightspine[1]{\def\rightspin@{#1}}
51 \DeclareRobustCommand*\leftspinebis[1]{\def\leftspin@bis{#1}}
52 \DeclareRobustCommand*\centerspinebis[1]{\def\centerspin@bis{#1}}
53 \DeclareRobustCommand*\rightspinebis[1]{\def\rightspin@bis{#1}}
```

We do not want any `lineskip`, as stacked covers should not be separated by any space. Analogously, we want no margins, no indentation and no hyphens. Offsets will be set command by each command.

```
54 \evensidemargin=0cm
55 \oddsidemargin=0cm
56 \topmargin=0cm
57 \headheight=0cm
58 \headsep=0cm
59 \footskip=0cm
60 \textwidth=\paperwidth
61 %\advance\textwidth by -3cm
62 \textheight=\paperheight
63 %\advance\textheight by -3cm
64
65 \lineskip=0pt
66 \lineskiplimit=0pt
67 \parskip=0pt
68 \parindent=0pt
69 \hyphenpenalty=10000
```

We set the unit for the `picture` environment to 1mm, and prepare a number of lengths which will be useful in aligning track numbers and spine text. `\square` holds the side length of the square framing the track numbers. `\h@nging` is its

hanging amount. `\@hstrip` and `\@wstrip` are used when aligning the spine. `\winf@` and `\wtr@cklist` are the width of the information and tracklist mini-pages.

```

70 \setlength{\unitlength}{1mm}
71 \newlength{\squ@re}
72 \newlength{\@temp}
73 \newlength{\h@nging}
74 \newlength{\@hstrip}
75 \newlength{\@wstrip}
76 \newlength{\winf@}
77 \newlength{\wtr@cklist}
78 \newlength{\onec@rrection}

```

The `\track` command typesets a hanging framed box with a small number inside. The number is given by a counter which is reset to 1 at each `\makeback`, and can be changed manually with the `\setindex` command. The alignment inside the small box will be corrected for numbers either starting or ending with a 1 by the fraction of the width of 1 specified with the `\onecorrection` command.

```

79 \newcounter{tr@ckindex}
80 \DeclareRobustCommand*\setindex}[1]{\setcounter{tr@ckindex}{#1}}
81
82 \DeclareRobustCommand*\track}{%
83   \par
84   \let\@firstdigit=\@empty
85   \setlength{\onec@rrection}{0pt}%
86   \settowidth{\@temp}{\indexf@nt1}
87   \expandafter\@tfor \expandafter\@digit
88     \expandafter:\expandafter=\number\value{tr@ckindex}\do {%
89     \ifx\@firstdigit\@empty
90       \let\@firstdigit=\@digit
91     \else
92       \if 1\@firstdigit
93         \if 1\@digit\else
94           \setlength{\onec@rrection}{-\onec@rrfrac\@temp}%
95         \fi
96       \else
97         \if 1\@digit
98           \setlength{\onec@rrection}{\onec@rrfrac\@temp}%
99         \fi
100      \fi
101    \fi
102  }%
103  \settoheight{\@temp}{M}%
104  \addtolength{\@temp}{-\squ@re}%
105  \raisebox{.5\@temp}{%
106    \setlength{\unitlength}{\squ@re}%
107    \hspace*{-\h@nging}%
108    \begin{picture}(1,1)
109      \put(0,0){%
110        \framebox(1,1){\hspace*\onec@rrection}\indexf@nt\thetr@ckindex}%
111      }
112    \end{picture}%
113  }%
114  \hspace*{6pt}%

```

```

115   \addtocounter{tr@ckindex}{1}%
116 }

We declare some utility commands which allow for easy font dimension switch.
The \newcd command resets to defaults all the fonts and the text defaults.

117 \AtBeginDocument{%
118   \pagestyle{empty}%
119   \thispagestyle{empty}%
120   \newcd
121 }
122
123 \DeclareRobustCommand*\fhb}[2]{%
124   \fontsize{#1pt}{#2pt}\selectfont
125   \parskip=.1\baselineskip
126 }
127
128 \DeclareRobustCommand*\fh}[1]{\fontsize{#1pt}{\baselineskip}\selectfont}
129
130 \DeclareRobustCommand*\newcd}{%
131   \lefttracklist{}%
132   \righttracklist{}%
133   \covertext{}%
134   \backtext{\c@vertext}%
135   \insidetext{}%
136   \leftspine{}%
137   \centerspine{}%
138   \rightspine{}%
139   \leftspinebis{\leftspin@}%
140   \centerspinebis{\centerspin@}%
141   \rightspinebis{\rightspin@}%
142   \leftinfo{}%
143   \rightinfo{}%
144   \coverfont{\fhb{16}{19}}%
145   \backfont{\coverfont}%
146   \insidefont{\fhb{10}{12}}%
147   \spinefont{\fhb{9}{11}\bfseries}%
148   \tracklistfont{\fhb{9}{10.5}}%
149   \infofont{\fhb{7}{8.3}}%
150   \indexfont{\fhb{5}{0}}%
151 }

```

The following two commands are useful in alignment. The first command decides the height and width of a given strip of text, to be inserted in the spine. The point is that unless the `alignspine` option has been requested, we do not set `\@hstrip`, which has been set previously to the maximum height of a capital letter. The `\align@baseline` command is used at the end of boxes which could be bottom aligned: it eliminates the additional height inserted when a box last line has a descendant.

```

152 \DeclareRobustCommand*\@sethwstrips}[1]{%
153   \settowidth{\@wstrip}{\spinefont #1}%
154   \if@lignspine
155     \settoheight{\@hstrip}{\spinefont #1}%
156   \fi
157 }

```

```

158
159 \DeclareRobustCommand*\align@baseline}{%
160   \settodepth{\@temp}{gjpgy}%
161   \vphantom{gjpgy}\par
162   \vspace*{-\@temp}\par
163 }

```

It is now easy to write down the `\makecover` command. It is just a matter of laying out the material, and print the requested crop marks.

```

164 \DeclareRobustCommand*\makecover}[1][lrbt]{%
165 \voffset=0in
166 \begin{picture}(120,240)
167 \end{picture}%
168 \begin{rotate}{90}%
169 \begin{picture}(240,120)
170   \@tfor\cr@pmark := #1 \do {
171     \if l\cr@pmark
172       \put(-1,0){\line(-1,0){5}}
173       \put(-1,120){\line(-1,0){5}}
174     \else\if r\cr@pmark
175       \put(241,0){\line(1,0){5}}
176       \put(241,120){\line(1,0){5}}
177     \else\if b\cr@pmark
178       \put(0,-1){\line(0,-1){5}}
179       \put(240,-1){\line(0,-1){5}}
180       \put(120,-1){\line(0,-1){1}}
181       \put(120,-3){\line(0,-1){1}}
182       \put(120,-5){\line(0,-1){1}}
183     \else\if t\cr@pmark
184       \put(0,121){\line(0,1){5}}
185       \put(240,121){\line(0,1){5}}
186       \put(120,121){\line(0,1){1}}
187       \put(120,123){\line(0,1){1}}
188       \put(120,125){\line(0,1){1}}
189     \else\if c\cr@pmark
190       \put(0,0){\line(1,0){240}}
191       \put(0,0){\line(0,1){120}}
192       \put(120,0){\line(0,1){120}}
193       \put(0,120){\line(1,0){240}}
194       \put(240,0){\line(0,1){120}}
195     \fi\fi\fi\fi\fi
196   }
197
198   \put(10,10){%
199     \makebox(100,100)[\@ligninside]{%
200       \parbox{10cm}{%
201         \raggedright\insidef@nt\insid@text\align@baseline
202       }%
203     }%
204   }
205   \put(130,10){%
206     \makebox(100,100)[\@ligncover]{%
207       \parbox{10cm}{%
208         \raggedright\coverf@nt\c@vert@text\align@baseline

```



```

209         }%
210     }%
211 }
212 \end{picture}%
213 \end{rotate}%
214 }

```

The `\makeback` command is slightly more complicated, as it must set up some values for the `\track` command to work. Moreover, it has to check for empty right information or tracklist minipages, as in this case the left ones must be enlarged.

```

215 \DeclareRobustCommand*\makeback}[1][lrb]{%
216 \voffset=-.5in
217 \setindex{1}%
218 \settowidth{\squ@re}{\indexf@nt0}%
219 \settoheight{@temp}{\indexf@nt0}%
220 \addtolength{\squ@re}{.4@temp}%
221 \setlength{\h@nging}{\squ@re}%
222 \addtolength{\h@nging}{6pt}%
223 \settoheight{@hstrip}{\spinef@nt ABCDEFGHIJKLMNOPQRSTUVWXYZ}%
224 %
225 \ifx\righttr@cklist\@empty
226     \setlength{\wtr@cklist}{12cm}%
227 \else
228     \setlength{\wtr@cklist}{5.5cm}%
229 \fi
230 %
231 \ifx\rightinf@\@empty
232     \setlength{\winf@}{12cm}%
233 \else
234     \setlength{\winf@}{5.5cm}%
235 \fi
236 %
237 \begin{picture}(151,118)
238     \@tfor\cr@pmark := #1 \do {
239         \if l\cr@pmark
240             \put(-1,0){\line(-1,0){5}}
241             \put(-1,118){\line(-1,0){5}}
242         \else\if r\cr@pmark
243             \put(152,0){\line(1,0){5}}
244             \put(152,118){\line(1,0){5}}
245         \else\if b\cr@pmark
246             \put(0,-1){\line(0,-1){5}}
247             \put(151,-1){\line(0,-1){5}}
248             \put(6.5,-1){\line(0,-1){1}}
249             \put(6.5,-3){\line(0,-1){1}}
250             \put(6.5,-5){\line(0,-1){1}}
251             \put(144.5,-1){\line(0,-1){1}}
252             \put(144.5,-3){\line(0,-1){1}}
253             \put(144.5,-5){\line(0,-1){1}}
254         \else\if t\cr@pmark
255             \put(0,119){\line(0,1){5}}
256             \put(151,119){\line(0,1){5}}
257             \put(6.5,119){\line(0,1){1}}

```

```

258     \put(6.5,121){\line(0,1){1}}
259     \put(6.5,123){\line(0,1){1}}
260     \put(144.5,119){\line(0,1){1}}
261     \put(144.5,121){\line(0,1){1}}
262     \put(144.5,123){\line(0,1){1}}
263 \else\if c\cr@pmark
264     \put(0,0){\line(1,0){151}}
265     \put(0,0){\line(0,1){118}}
266     \put(151,0){\line(0,1){118}}
267     \put(0,118){\line(1,0){151}}
268     \put(6.5,0){\line(0,1){118}}
269     \put(144.5,0){\line(0,1){118}}
270 \fi\fi\fi\fi\fi
271 }
272
273 \@sethwstrips{\leftspin@}
274
275 \put(0,4){%
276     \makebox(6.5,110)[b]{%
277         \makebox[\@hstrip][r]{%
278             \begin{rotate}{90}\spinef@nt\leftspin@end{rotate}%
279         }%
280     }%
281 }
282
283 \@sethwstrips{\centerspin@}
284
285 \put(0,4){%
286     \makebox(6.5,110){%
287         \raisebox{0pt}[\@wstrip]{\makebox[\@hstrip][r]{%
288             \begin{rotate}{90}\spinef@nt\centerspin@end{rotate}%
289         }%
290     }%
291 }
292
293 \@sethwstrips{\rightspin@}
294
295 \put(0,4){%
296     \makebox(6.5,110)[t]{%
297         \raisebox{0pt}[\@wstrip]{\makebox[\@hstrip][r]{%
298             \begin{rotate}{90}\spinef@nt\rightspin@end{rotate}%
299         }%
300     }%
301 }
302
303 \@sethwstrips{\leftspin@bis}
304
305 \put(144.5,4){%
306     \makebox(6.5,110)[t]{%
307         \makebox[\@hstrip][l]{%
308             \begin{rotate}{-90}\spinef@nt\leftspin@bisend{rotate}%
309         }%
310     }%
311 }

```

```

312
313 \sethwstrips{\centerspin@bis}
314
315 \put(144.5,4){%
316   \makebox(6.5,110){%
317     \raisebox{\@wstrip}{\@wstrip}{\makebox[\@hstrip][l]{%
318       \begin{rotate}{-90}\spinef@nt\centerspin@bis\end{rotate}%
319     }}%
320   }%
321 }
322
323 \sethwstrips{\rightspin@bis}
324
325 \put(144.5,4){%
326   \makebox(6.5,110)[b]{%
327     \raisebox{\@wstrip}{\@wstrip}{\makebox[\@hstrip][l]{%
328       \begin{rotate}{-90}\spinef@nt\rightspin@bis\end{rotate}%
329     }}%
330   }%
331 }
332
333 \put(17,0){%
334   \begin{picture}(121,118)
335     \put(0,82){%
336       \makebox(120,30)[@lignback]{%
337         \parbox{12.1cm}{%
338           \raggedright\backf@nt\b@cktext\align@baseline
339         }%
340       }%
341     }
342
343     \put(0,5){%
344       \makebox(55,70)[tl]{%
345         \begin{minipage}{\wtr@cklist}%
346           \lineskip=.5pt\lineskiplimit=1pt\raggedright
347           \tracklistf@nt\lefttr@cklist
348         \end{minipage}%
349       }%
350     }
351
352     \put(65,5){%
353       \makebox(55,70)[tl]{%
354         \begin{minipage}{\wtr@cklist}%
355           \lineskip=.5pt\lineskiplimit=1pt\raggedright
356           \tracklistf@nt\righttr@cklist
357         \end{minipage}%
358       }%
359     }
360
361     \put(0,5){%
362       \makebox(0,0)[bl]{%
363         \parbox{\winf@}{%
364           \raggedright\infof@nt\leftinf@\align@baseline
365         }%

```

```

366         }%
367     }
368
369     \put(65,5){%
370         \makebox(0,0)[bl]{%
371             \parbox{\winf@}{%
372                 \raggedright\infof@nt\rightinf@\align@baseline
373             }%
374         }%
375     }
376
377     \end{picture}%
378 }
379
380 \end{picture}%
381 }

```

The `\makeslimcover` command is essentially a mix of the previous two, as a single slim cover must contain the front matter and the track lists. Note that we have much less space.

```

382 \DeclareRobustCommand*\makeslimcover}[1][lrbt]{%
383 \voffset=0in
384 \setindex{1}%
385 \settowidth{\squ@re}{\indexf@nt00}%
386 \settoheight{\@temp}{\indexf@nt0}%
387 \addtolength{\squ@re}{.4\@temp}%
388 \setlength{\h@nging}{\squ@re}%
389 \addtolength{\h@nging}{6pt}%
390 \settoheight{\hstrip}{\spinef@nt ABCDEFGHIJKLMNOPQRSTUVWXYZ}%
391 %
392 \ifx\righttr@cklist\@empty
393     \setlength{\wtr@cklist}{10cm}%
394 \else
395     \setlength{\wtr@cklist}{4.7cm}%
396 \fi
397 %
398 \ifx\rightinf@\@empty
399     \setlength{\winf@}{10cm}%
400 \else
401     \setlength{\winf@}{4.7cm}%
402 \fi
403 %
404 \begin{picture}(120,240)
405 \end{picture}%
406 \begin{rotate}{90}%
407 \begin{picture}(240,120)
408     \@tfor\cr@pmark := #1 \do {
409         \if l\cr@pmark
410             \put(-1,0){\line(-1,0){5}}
411             \put(-1,120){\line(-1,0){5}}
412         \else\if r\cr@pmark
413             \put(241,0){\line(1,0){5}}
414             \put(241,120){\line(1,0){5}}
415         \else\if b\cr@pmark

```

```

416     \put(0,-1){\line(0,-1){5}}
417     \put(240,-1){\line(0,-1){5}}
418     \put(120,-1){\line(0,-1){1}}
419     \put(120,-3){\line(0,-1){1}}
420     \put(120,-5){\line(0,-1){1}}
421 \else\if t\cr@pmark
422     \put(0,121){\line(0,1){5}}
423     \put(240,121){\line(0,1){5}}
424     \put(120,121){\line(0,1){1}}
425     \put(120,123){\line(0,1){1}}
426     \put(120,125){\line(0,1){1}}
427 \else\if c\cr@pmark
428     \put(0,0){\line(1,0){240}}
429     \put(0,0){\line(0,1){120}}
430     \put(120,0){\line(0,1){120}}
431     \put(0,120){\line(1,0){240}}
432     \put(240,0){\line(0,1){120}}
433 \fi\fi\fi\fi\fi
434 }
435
436 \put(12,10){%
437     \begin{picture}(100,100)
438         \put(0,80){%
439             \makebox(100,30)[\@lignback]{%
440                 \parbox{10.1cm}{%
441                     \raggedright\backf@nt\b@cktext\align@baseline
442                 }%
443             }%
444         }
445
446         \put(0,15){%
447             \makebox(47,60)[tl]{%
448                 \begin{minipage}{\wtr@cklist}%
449                     \lineskip=.5pt\lineskiplimit=1pt\raggedright
450                     \tracklistf@nt\lefttr@cklist
451                 \end{minipage}%
452             }%
453         }
454
455         \put(55,15){%
456             \makebox(47,60)[tl]{%
457                 \begin{minipage}{\wtr@cklist}%
458                     \lineskip=.5pt\lineskiplimit=1pt\raggedright
459                     \tracklistf@nt\righttr@cklist
460                 \end{minipage}%
461             }%
462         }
463
464         \put(0,0){%
465             \makebox(0,0)[bl]{%
466                 \parbox{\winf@}{%
467                     \raggedright\infof@nt\leftinf@\align@baseline
468                 }%
469             }%

```

```

470     }
471
472     \put(55,0){%
473         \makebox(0,0)[bl]{%
474             \parbox{\winf@}{%
475                 \raggedright\infof@nt\rightinf@\align@baseline
476             }%
477         }%
478     }
479
480 \end{picture}%
481 }
482 \put(130,10){%
483     \makebox(100,100)[\@ligncover]{%
484         \parbox{10cm}{%
485             \raggedright\coverf@nt\c@vtext\align@baseline
486         }%
487     }%
488 }
489 \end{picture}%
490 \end{rotate}%
491 }

```

Finally, we have the high-level commands that allow to produce one or several CD from data files, `\makeCD`, `\makelist`, `\makeslimCD` and `\makeslimlist`. All have an additional argument for the file name, defaulting to `CD.dat` or `CD.lst`.

Two separate commands factor out the checks and the user interaction in case the file is not specified or does not exist.

A data file must contain only text declaration commands from the CD class. All  $\LaTeX$  stuff (preamble, etc.) and cover generation commands are handled automatically. A list file must contain a number of lines, each containing a data file name.

```

492 \DeclareRobustCommand*{\@skCDfile}[1][ ]{%
493 \def\CDname{#1}%
494 \ifx\CDname\@empty
495     \IfFileExists{CD.dat}{%
496         \def\CDname{CD.dat}%
497     }{%
498         \typein[\CDname]{Please insert CD data file name:}%
499     }%
500 \fi
501 \InputIfFileExists{\CDname.dat}{%
502 }{%
503     \InputIfFileExists{\CDname}{%
504     }{%
505         \ClassError{cd}{CD data file (\CDname.dat or \CDname) not found}{}%
506     }
507 }%
508 }
509
510 \DeclareRobustCommand*{\makeCD}[1][ ]{%
511 \@skCDfile{#1}\makecover\par\makeback\par
512 }
513

```

```

514 \DeclareRobustCommand*\makeslimCD}[1] [] {%
515 \@skCDfile{#1}\makeslimcover\par
516 }
517
518 \newread\CDlist
519
520 \newcounter{cd}
521 \setcounter{cd}{0}
522
523 \newif\ifne@f
524
525 \DeclareRobustCommand*\@sklistfile}[1] [] {%
526 \def\CDlistname{#1}%
527 \ifx\CDlistname\@empty
528   \IfFileExists{CD.lst}{%
529     \def\CDlistname{CD.lst}%
530   }{%
531     \typein[\CDlistname]{Please insert CD list file name:}
532   }%
533 \fi
534 \IfFileExists{\CDlistname.lst}{%
535   \immediate\openin\CDlist=\CDlistname.lst
536 }{%
537   \IfFileExists{\CDlistname}{%
538     \immediate\openin\CDlist=\CDlistname
539   }{%
540     \ClassError{cd}{CD list (\CDlistname.lst or \CDlistname) not found}{}%
541   }
542 }
543 \ne@ftrue
544 }
545
546 \DeclareRobustCommand*\makelist}[1] [] {%
547 \@sklistfile{#1}%
548 \advance\endlinechar\@M
549 \immediate\read\CDlist to \CDname
550 \advance\endlinechar-\@M
551 \ifeof\CDlist\ne@ffalse\fi
552 %
553 \@whilesw \ifne@f \fi {%
554   \newcd
555   \InputIfFileExists{\CDname.dat}{%
556   }{%
557     \InputIfFileExists{\CDname}{%
558     }{%
559       \ClassError{cd}{CD data file (\CDname.dat or \CDname) not found}{}%
560     }%
561   }%
562   \advance\endlinechar\@M
563   \immediate\read\CDlist to \CDname
564   \advance\endlinechar-\@M
565   \ifeof\CDlist\ne@ffalse\fi
566   \ifodd\value{cd}%
567     \makeback[lrb]\par\makecover\par

```

```

568 \else
569 \makecover\par\ifneof\makeback[lrt]\else\makeback\fi\par
570 \fi
571 \addtocounter{cd}{1}%
572 }
573 }
574
575 \DeclareRobustCommand*\makeslimlist}[1][ ]{%
576 \@sklistfile{#1}%
577 \advance\endlinechar\@M
578 \immediate\read\CDlist to \CDname
579 \advance\endlinechar-\@M
580 \ifeof\CDlist\ne@ffalse\fi
581 %
582 \@whilesw \ifneof \fi {%
583 \newcd
584 \InputIfFileExists{\CDname.dat}{%
585 }{%
586 \InputIfFileExists{\CDname}{%
587 }{%
588 \ClassError{cd}{CD data file (\CDname.dat or \CDname) not found}{%
589 }%
590 }%
591 \advance\endlinechar\@M
592 \immediate\read\CDlist to \CDname
593 \advance\endlinechar-\@M
594 \ifeof\CDlist\ne@ffalse\fi
595 \makeslimcover\par
596 }
597 }
598
599 </class>

```