

A CD Cover Class

Sebastiano Vigna

Printed June 13, 2014

Contents

1	Introduction	1
2	The Text Commands	2
3	The Graphic Commands	3
4	The Font Commands	3
5	The Cover Creation Commands	4
6	The Options	4
7	Getting Obsessed	5
8	The Code	5
	Change History	21

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 1.1, slim CD boxes are supported.

Each CD cover is created by a number of commands that 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}
\centerspines{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 L^AT_EX the file `CD.tex` (or `slimCD.tex`). This is a better mechanism—each CD should have its own data (`.dat`) file that 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^AT_EX'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 following self-explaining commands.

```

\covertext, \backtext, \insidetext,
\leftspine, \centerspine, \rightspine,
\lefttracklist, \righttracklist,
\leftinfo, \rightinfo

```

The 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. There is also a `\trackrange` command that takes a parameter and adds a range indication (two numbered boxes separated by a dash). Should you need to set manually the track number, use `\setindex{<n>}`. 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.

The height of the track list (in millimeters) can be set with `\tracklistheight`, which expects a dimensionless, positive integer value. If not specified, the height is set to 70. The sum of the track list height and the height of the area used to typeset `\covertext` on the back cover equals 100.

3 The Graphic Commands

In the case you want to fill the cover or the inside of your CD with a picture, the commands `\covergraphics` and `\insidegraphics` work like `\covertext` and `\insidetext`, but they create no border (as opposed to the standard 1 cm border for text).

4 The Font Commands

The CD class provides some simple commands for switching the font dimension and line spacing. The command `\fh{<height>}` sets the font height to the given number of points (line spacing is not affected), while `\fhb{<height>}{<baselineskip>}` sets both the font height and the baseline skip (usually 6/5th 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 1/10th 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.

5 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 (note that, of course, this does *not* happen with `\covergraphics`). 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 commands have an optional argument that can contain any of the letters `lrbt` (left, right, top, bottom), which create the respective crop marks (note that the argument must be enclosed in brackets). The default value is `lrbt`. 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 \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 `jobname.dat` file (where `jobname` is the root of the \LaTeX file under compilation—e.g., `CD.dat` when compiling `CD.tex`). If such a file 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 (which, note again, must be enclosed in brackets) 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.

6 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. The `covergraphics` option lets you use the entire cover area (instead of a centered 10 cm×10 cm square). Finally, the `alignspine` option forces vertical centring 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.

7 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.

8 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{covergraphics}{\def\@lignback{t}}
16
17 \DeclareOption{aligntop}{%
18   {\ExecuteOptions{aligncovertop,
19                   aligninsidetop,
20                   alignbacktop}}
21
22 \DeclareOption{aligncovercenter}{\def\@ligncover{c}}
```

```

23 \DeclareOption{aligninsidecenter}{\def\@ligninside{c}}
24 \DeclareOption{alignbackcenter}{\def\@lignback{c}}
25
26 \DeclareOption{aligncenter}%
27   {\ExecuteOptions{aligncovercenter,
28                   aligninsidecenter,
29                   alignbackcenter}}
30
31 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
32
33 \ProcessOptions\relax

```

Now we load the `article` class and the `rotating` package, which is fundamental in typesetting the spine text.

```

34 \LoadClass{article}
35 \RequirePackage{rotating}

```

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

```

36 \DeclareRobustCommand*\onecorrection[1]{\def\onec@rrfrac{#1}}
37 \onecorrection{0}

```

Now we have all the font and text declaration commands. They just define a certain macro to be their argument.

```

38 \DeclareRobustCommand*\coverfont}[1]{\def\coverf@nt{#1}}
39 \DeclareRobustCommand*\backfont}[1]{\def\backf@nt{#1}}
40 \DeclareRobustCommand*\insidefont}[1]{\def\insid@nt{#1}}
41 \DeclareRobustCommand*\spinefont}[1]{\def\spinef@nt{#1}}
42 \DeclareRobustCommand*\tracklistfont}[1]{\def\tracklistf@nt{#1}}
43 \DeclareRobustCommand*\infofont}[1]{\def\infof@nt{#1}}
44 \DeclareRobustCommand*\indexfont}[1]{\def\indexf@nt{#1}}
45
46 \DeclareRobustCommand{\lefttracklist}[1]{\def\lefttr@cklist{#1}}
47 \DeclareRobustCommand{\righttracklist}[1]{\def\righttr@cklist{#1}}
48 \DeclareRobustCommand{\leftinfo}[1]{\def\leftinf@{#1}}
49 \DeclareRobustCommand{\rightinfo}[1]{\def\rightinf@{#1}}
50 \DeclareRobustCommand{\covertext}[1]{\def\c@vertext{#1}}
51 \DeclareRobustCommand{\backtext}[1]{\def\b@cktext{#1}}
52 \DeclareRobustCommand{\insidetext}[1]{\def\insid@text{#1}}
53
54 \DeclareRobustCommand{\covergraphics}[1]{%
55   \def\c@vertext{#1}\def\c@vergraphics{}}
56 \DeclareRobustCommand{\insidegraphics}[1]{%
57   \def\insid@text{#1}\def\insid@graphics{}}
58
59 \DeclareRobustCommand*\leftspine}[1]{\def\leftspin@{#1}}
60 \DeclareRobustCommand*\centerspine}[1]{\def\centerspin@{#1}}
61 \DeclareRobustCommand*\rightspine}[1]{\def\rightspin@{#1}}
62 \DeclareRobustCommand*\leftspinebis}[1]{\def\leftspin@bis{#1}}
63 \DeclareRobustCommand*\centerspinebis}[1]{\def\centerspin@bis{#1}}

```

```

64 \DeclareRobustCommand*\rightspinebis}[1]{\def\righspin@bis{#1}}
65
66 \DeclareRobustCommand*\tracklistheight}[1]{%
67   \begingroup
68     \xdef\tr@cklistheight{#1}%
69     \@tempcnta=100
70     \advance\@tempcnta by -\tr@cklistheight
71     \xdef\c@vertextheight{\the\@tempcnta}%
72     \@tempcnta=12
73     \advance\@tempcnta by \tr@cklistheight
74     \xdef\c@verpos{\the\@tempcnta}%
75   \endgroup
76 }

```

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 hyphenation. Offsets will be set command by each command.

```

77 \evensidemargin=0cm
78 \oddsidemargin=0cm
79 \topmargin=0cm
80 \headheight=0cm
81 \headsep=0cm
82 \footskip=0cm
83 \textwidth=\paperwidth
84 %\advance\textwidth by -3cm
85 \textheight=\paperheight
86 %\advance\textheight by -3cm
87
88 \lineskip=0pt
89 \lineskiplimit=0pt
90 \parskip=0pt
91 \parindent=0pt
92 \hyphenpenalty=10000

```

We set the unit for the `picture` environment to 1 mm and prepare a number of lengths that 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. `\h@ngingg` is the hanging amount of a track range. `\@hstrip` and `\@wstrip` are used when aligning the spine. `\winf@` and `\wtr@cklist` are the width of the information and tracklist minipages.

```

93 \setlength{\unitlength}{1mm}
94 \newlength{\square}
95 \newlength{\@temp}
96 \newlength{\@temppp}
97 \newlength{\d@shwidth}
98 \newlength{\h@nging}
99 \newlength{\h@ngingg}
100 \newlength{\@hstrip}
101 \newlength{\@wstrip}

```

```

102 \newlength{\winf@}
103 \newlength{\wtr@cklist}
104 \newlength{\onec@rrection}

```

The `\tr@ck` command typesets a hanging framed box with a small number inside. The number is given by a counter that is reset to 1 at each `\makeback`, and can be changed manually with the `\setindex` command. Two parameters makes it possible to set the hanging amount and the amount of space that follows the box. 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. The associated user commands are `\track` and `\trackrange`; the latter makes it possible to typeset a range given its length.

```

105 \newcounter{tr@ckindex}
106 \DeclareRobustCommand*{\setindex}[1]{\setcounter{tr@ckindex}{#1}}
107
108 \DeclareRobustCommand*{\tr@ck}[2]{%
109   \let\@firstdigit=\@empty
110   \setlength{\onec@rrection}{0pt}%
111   \settowidth{\@temp}{\indexf@nt1}%
112   \expandafter\@tfor \expandafter\@digit
113     \expandafter:\expandafter=\number\value{tr@ckindex}\do {%
114     \ifx\@firstdigit\@empty
115       \let\@firstdigit=\@digit
116     \else
117       \if 1\@firstdigit
118         \if 1\@digit\else
119           \setlength{\onec@rrection}{-\onec@rrfrac\@temp}%
120         \fi
121       \else
122         \if 1\@digit
123           \setlength{\onec@rrection}{\onec@rrfrac\@temp}%
124         \fi
125       \fi
126     \fi
127   }%
128   \setlength{\@temp}{\squ@re}%
129   \settoheight{\@temp}{\tracklistf@nt M}%
130   \addtolength{\@temp}{-\@temp}%
131   \raisebox{-.5\@temp}{%
132     \setlength{\unitlength}{\squ@re}%
133     \hspace*{-#1}%
134     \begin{picture}(1,1)
135       \put(0,0){%
136         \framebox(1,1){%
137           \hspace*{\onec@rrection}\indexf@nt\thetr@ckindex}%
138         }
139     \end{picture}%
140   }%
141   \hspace*{#2}%

```



```

142 \addtocounter{tr@ckindex}{1}%
143 }
144
145 \DeclareRobustCommand*\track{%
146 \par\tr@ck{\h@nging}{6pt}%
147 }
148
149 \DeclareRobustCommand*\trackrange}[1]{%
150 \par
151 \tr@ck{\h@nging}{1pt}%
152 \addtocounter{tr@ckindex}{#1}%
153 \addtocounter{tr@ckindex}{-2}%
154 \settoheight{\@temp}{\indexf@nt M}%
155 \settoheight{\@temp}{\tracklistf@nt M}%
156 \addtolength{\@temp}{-\@temp}%
157 \raisebox{-.5\@temp}{\indexf@nt-}%
158 \hspace*{1pt}%
159 \tr@ck{0pt}{6pt}%
160 }

```

We declare some utility commands that allow for easy font dimension switch.
The `\newcd` command resets font and text values to its default values.

```

161 \AtBeginDocument{%
162 \pagestyle{empty}%
163 \thispagestyle{empty}%
164 \newcd
165 }
166
167 \DeclareRobustCommand*\fhb}[2]{%
168 \fontsize{#1pt}{#2pt}\selectfont
169 \parskip=.1\baselineskip
170 }
171
172 \DeclareRobustCommand*\fh}[1]{%
173 \fontsize{#1pt}{\baselineskip}\selectfont
174 }
175
176 \DeclareRobustCommand*\newcd{%
177 \lefttracklist{}%
178 \righttracklist{}%
179 \coverttext{}%
180 \insidetext{}%
181 \leftspine{}%
182 \centerspine{}%
183 \rightspine{}%
184 \leftspinebis{\leftspin@}%
185 \centerspinebis{\centerspin@}%
186 \rightspinebis{\rightspin@}%
187 \leftinfo{}%
188 \rightinfo{}%

```

```

189 \coverfont{\fhb{16}{19}}%
190 \backfont{\coverfont}%
191 \insidefont{\fhb{10}{12}}%
192 \spinefont{\fhb{9}{11}\bfseries}%
193 \tracklistfont{\fhb{9}{10.5}}%
194 \infofont{\fhb{7}{8.3}}%
195 \indexfont{\fhb{5}{0}}%
196 \tracklistheight{70}%
197 }

```

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 that could be bottom aligned: it eliminates the additional height inserted when a box's last line has a descender.

```

198 \DeclareRobustCommand*\@sethwstrips}[1]{%
199 \settowidth{\@wstrip}{\spinefont #1}%
200 \if@lignspine
201   \settoheight{\@hstrip}{\spinefont #1}%
202 \fi
203 }
204
205 \DeclareRobustCommand*\align@baseline){%
206 \settodepth{\@temp}{gjjpqy}%
207 \vphantom{gjjpqy}\par
208 \vspace*{-\@temp}\par
209 }

```

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

```

210 \DeclareRobustCommand*\makecover}[1][lrbt]{%
211 \voffset=0in
212 \begin{picture}(120,240)
213 \end{picture}%
214 \begin{rotate}{90}%
215   \begin{picture}(240,120)
216     \@tfor\cr@pmark := #1 \do {
217       \if l\cr@pmark
218         \put(-1,0){\line(-1,0){5}}
219         \put(-1,120){\line(-1,0){5}}
220       \else\if r\cr@pmark
221         \put(241,0){\line(1,0){5}}
222         \put(241,120){\line(1,0){5}}
223       \else\if b\cr@pmark
224         \put(0,-1){\line(0,-1){5}}
225         \put(240,-1){\line(0,-1){5}}
226         \put(120,-1){\line(0,-1){1}}

```

```

227         \put(120,-3){\line(0,-1){1}}
228         \put(120,-5){\line(0,-1){1}}
229     \else\if t\cr@pmark
230         \put(0,121){\line(0,1){5}}
231         \put(240,121){\line(0,1){5}}
232         \put(120,121){\line(0,1){1}}
233         \put(120,123){\line(0,1){1}}
234         \put(120,125){\line(0,1){1}}
235     \else\if c\cr@pmark
236         \put(0,0){\line(1,0){240}}
237         \put(0,0){\line(0,1){120}}
238         \put(120,0){\line(0,1){120}}
239         \put(0,120){\line(1,0){240}}
240         \put(240,0){\line(0,1){120}}
241     \fi\fi\fi\fi\fi
242 }
243
244 \ifx\insid@graphics\@empty
245     \put(0,0){%
246         \makebox(120,120)[\@ligninside]{%
247             \parbox{12cm}{%
248                 \raggedright
249                 \inside@nt
250                 \insid@text
251                 \align@baseline}}}}
252 \else
253     \put(10,10){%
254         \makebox(100,100)[\@ligninside]{%
255             \parbox{10cm}{%
256                 \raggedright
257                 \inside@nt
258                 \insid@text
259                 \align@baseline}}}}
260 \fi
261 \ifx\c@vergraphics\@empty
262     \put(120,0){%
263         \makebox(120,120)[\@ligncover]{%
264             \parbox{12cm}{%
265                 \raggedright
266                 \coverf@nt
267                 \c@vertext
268                 \align@baseline}}}}
269 \else
270     \put(130,10){%
271         \makebox(100,100)[\@ligncover]{%
272             \parbox{10cm}{%
273                 \raggedright
274                 \coverf@nt
275                 \c@vertext
276                 \align@baseline}}}}

```

```

277     \fi
278     \end{picture}%
279 \end{rotate}%
280 }

```

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, and it must try to use the text from the cover page if no back text has been specified.

```

281 \DeclareRobustCommand*\makeback}[1][lrb]{%
282   \voffset=-.5in
283   \setindex{1}%
284   \settowidth{\d@shwidth}{\indexfont-}%
285   \settowidth{\squ@re}{\indexfont0}%
286   \settoheight{\@temp}{\indexfont0}%
287   \addtolength{\squ@re}{.4\@temp}%
288   \setlength{\h@nging}{\squ@re}%
289   \setlength{\h@ngingg}{\squ@re}%
290   \addtolength{\h@ngingg}{\squ@re}%
291   \addtolength{\h@ngingg}{\d@shwidth}%
292   \addtolength{\h@nging}{6pt}%
293   \addtolength{\h@ngingg}{8pt}%
294   \settoheight{\@hstrip}{\spinefont ABCDEFGHIJKLMNOPQRSTUVWXYZ}%
295 %
296   \ifx\righttracklist\@empty
297     \setlength{\wtr@cklist}{12cm}%
298   \else
299     \setlength{\wtr@cklist}{5.5cm}%
300   \fi
301 %
302   \ifx\rightinf@\@empty
303     \setlength{\winf@}{12cm}%
304   \else
305     \setlength{\winf@}{5.5cm}%
306   \fi
307 %
308 \begin{picture}(151,118)
309   \@tfor\cr@pmark := #1 \do {
310     \if l\cr@pmark
311       \put(-1,0){\line(-1,0){5}}
312       \put(-1,118){\line(-1,0){5}}
313     \else\if r\cr@pmark
314       \put(152,0){\line(1,0){5}}
315       \put(152,118){\line(1,0){5}}
316     \else\if b\cr@pmark
317       \put(0,-1){\line(0,-1){5}}
318       \put(151,-1){\line(0,-1){5}}
319       \put(6.5,-1){\line(0,-1){1}}
320       \put(6.5,-3){\line(0,-1){1}}

```

```

321     \put(6.5,-5){\line(0,-1){1}}
322     \put(144.5,-1){\line(0,-1){1}}
323     \put(144.5,-3){\line(0,-1){1}}
324     \put(144.5,-5){\line(0,-1){1}}
325     \else\if t\cr@pmark
326     \put(0,119){\line(0,1){5}}
327     \put(151,119){\line(0,1){5}}
328     \put(6.5,119){\line(0,1){1}}
329     \put(6.5,121){\line(0,1){1}}
330     \put(6.5,123){\line(0,1){1}}
331     \put(144.5,119){\line(0,1){1}}
332     \put(144.5,121){\line(0,1){1}}
333     \put(144.5,123){\line(0,1){1}}
334     \else\if c\cr@pmark
335     \put(0,0){\line(1,0){151}}
336     \put(0,0){\line(0,1){118}}
337     \put(151,0){\line(0,1){118}}
338     \put(0,118){\line(1,0){151}}
339     \put(6.5,0){\line(0,1){118}}
340     \put(144.5,0){\line(0,1){118}}
341     \fi\fi\fi\fi\fi
342 }
343
344 \@sethwstrips{\leftspin@}
345
346 \put(0,4){%
347   \makebox(6.5,110)[b]{%
348     \makebox[\@hstrip][r]{%
349       \begin{rotate}{90}%
350         \spinef@nt
351         \leftspin@
352       \end{rotate}}}}
353
354 \@sethwstrips{\centerspin@}
355
356 \put(0,4){%
357   \makebox(6.5,110){%
358     \raisebox{0pt}[\@wstrip]{%
359       \makebox[\@hstrip][r]{%
360         \begin{rotate}{90}%
361           \spinef@nt
362           \centerspin@
363         \end{rotate}}}}}}
364
365 \@sethwstrips{\rightspin@}
366
367 \put(0,4){%
368   \makebox(6.5,110)[t]{%
369     \raisebox{0pt}[\@wstrip]{%
370       \makebox[\@hstrip][r]{%

```

```

371         \begin{rotate}{90}%
372         \spinef@nt
373         \rightspin@
374         \end{rotate}}}}
375
376 \sethwstrips{\leftspin@bis}
377
378 \put(144.5,4){%
379   \makebox(6.5,110)[t]{%
380     \makebox[\@hstrip][l]{%
381       \begin{rotate}{-90}%
382       \spinef@nt
383       \leftspin@bis
384       \end{rotate}}}}
385
386 \sethwstrips{\centerspin@bis}
387
388 \put(144.5,4){%
389   \makebox(6.5,110){%
390     \raisebox{\@wstrip}{\@wstrip}{%
391       \makebox[\@hstrip][l]{%
392         \begin{rotate}{-90}%
393         \spinef@nt
394         \centerspin@bis
395         \end{rotate}}}}
396
397 \sethwstrips{\rightspin@bis}
398
399 \put(144.5,4){%
400   \makebox(6.5,110)[b]{%
401     \raisebox{\@wstrip}{\@wstrip}{%
402       \makebox[\@hstrip][l]{%
403         \begin{rotate}{-90}%
404         \spinef@nt
405         \rightspin@bis
406         \end{rotate}}}}
407
408 \put(17,0){%
409   \begin{picture}(121,118)
410     \put(0,\c@verpos){%
411       \makebox(120,\c@vertexheight)[\@lignback]{%
412         \parbox{12.1cm}{%
413           \raggedright
414           \backf@nt
415           \ifx\b@cktext\undefined
416             \ifx\c@vergraphics\@empty\else\c@vertext\fi
417           \else
418             \b@cktext
419           \fi
420           \align@baseline}}

```

```

421
422     \put(0,5){%
423         \makebox(55,\tr@cklistheight)[t1]{%
424             \begin{minipage}{\wtr@cklist}%
425                 \lineskip=.5pt
426                 \lineskiplimit=1pt
427                 \raggedright
428                 \tracklistf@nt
429                 \lefttr@cklist
430             \end{minipage}}}}
431
432     \put(65,5){%
433         \makebox(55,\tr@cklistheight)[t1]{%
434             \begin{minipage}{\wtr@cklist}%
435                 \lineskip=.5pt
436                 \lineskiplimit=1pt
437                 \raggedright
438                 \tracklistf@nt
439                 \righttr@cklist
440             \end{minipage}}}}
441
442     \put(0,5){%
443         \makebox(0,0)[bl]{%
444             \parbox{\winf@}{%
445                 \raggedright
446                 \infof@nt
447                 \leftinf@
448                 \align@baseline}}}}
449
450     \put(65,5){%
451         \makebox(0,0)[bl]{%
452             \parbox{\winf@}{%
453                 \raggedright
454                 \infof@nt
455                 \rightinf@
456                 \align@baseline}}}}
457     \end{picture}%
458 }
459 \end{picture}%
460 }

```

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.

```

461 \DeclareRobustCommand*\makeslimcover}[1][lrb]{%
462     \voffset=0in
463     \setindex{1}%
464     \settowidth{\d@shwidth}{\indexf@nt-}%
465     \settowidth{\squ@re}{\indexf@nt00}%
466     \settoheight{\@temp}{\indexf@nt0}%

```

```

467 \addtolength{\squ@re}{.4\@temp}%
468 \setlength{\h@nging}{\squ@re}%
469 \setlength{\h@ngingg}{\squ@re}%
470 \addtolength{\h@ngingg}{\squ@re}%
471 \addtolength{\h@ngingg}{\d@shwidth}%
472 \addtolength{\h@nging}{6pt}%
473 \addtolength{\h@ngingg}{8pt}%
474 \settoheight{\@hstrip}{\spinef@nt ABCDEFGHIJKLMNOPQRSTUVWXYZ}%
475 %
476 \ifx\righttr@cklist\@empty
477 \setlength{\wtr@cklist}{10cm}%
478 \else
479 \setlength{\wtr@cklist}{4.7cm}%
480 \fi
481 %
482 \ifx\rightinf@\@empty
483 \setlength{\winf@}{10cm}%
484 \else
485 \setlength{\winf@}{4.7cm}%
486 \fi
487 %
488 \begin{picture}(120,240)
489 \end{picture}%
490 \begin{rotate}{90}%
491 \begin{picture}(240,120)
492 \@tfor\cr@pmark := #1 \do {
493 \if l\cr@pmark
494 \put(-1,0){\line(-1,0){5}}
495 \put(-1,120){\line(-1,0){5}}
496 \else\if r\cr@pmark
497 \put(241,0){\line(1,0){5}}
498 \put(241,120){\line(1,0){5}}
499 \else\if b\cr@pmark
500 \put(0,-1){\line(0,-1){5}}
501 \put(240,-1){\line(0,-1){5}}
502 \put(120,-1){\line(0,-1){1}}
503 \put(120,-3){\line(0,-1){1}}
504 \put(120,-5){\line(0,-1){1}}
505 \else\if t\cr@pmark
506 \put(0,121){\line(0,1){5}}
507 \put(240,121){\line(0,1){5}}
508 \put(120,121){\line(0,1){1}}
509 \put(120,123){\line(0,1){1}}
510 \put(120,125){\line(0,1){1}}
511 \else\if c\cr@pmark
512 \put(0,0){\line(1,0){240}}
513 \put(0,0){\line(0,1){120}}
514 \put(120,0){\line(0,1){120}}
515 \put(0,120){\line(1,0){240}}
516 \put(240,0){\line(0,1){120}}

```



```

517     \fi\fi\fi\fi\fi
518   }
519
520   \put(11,0){%
521     \begin{picture}(100,120)
522       \put(0,\c@verpos){%
523         \makebox(100,\c@vertextheight)[\@lignback]{%
524           \parbox{10.1cm}{%
525             \raggedright
526             \backf@nt
527             \ifx\b@cktext\undefined
528               \ifx\c@vergraphics\@empty\else\c@vertext\fi
529             \else
530               \b@cktext
531             \fi
532             \align@baseline}}%
533
534     \put(0,5){%
535       \makebox(47,\tr@cklistheight)[t1]{%
536         \begin{minipage}{\wtr@cklist}%
537           \lineskip=.5pt
538           \lineskiplimit=1pt
539           \raggedright
540           \tracklistf@nt
541           \lefttr@cklist
542         \end{minipage}}%
543
544     \put(55,5){%
545       \makebox(47,\tr@cklistheight)[t1]{%
546         \begin{minipage}{\wtr@cklist}%
547           \lineskip=.5pt
548           \lineskiplimit=1pt
549           \raggedright
550           \tracklistf@nt
551           \righttr@cklist
552         \end{minipage}}%
553
554     \put(0,5){%
555       \makebox(0,0)[bl]{%
556         \parbox{\winf@}{%
557           \raggedright
558           \infof@nt
559           \leftinf@
560           \align@baseline}}%
561
562     \put(55,5){%
563       \makebox(0,0)[bl]{%
564         \parbox{\winf@}{%
565           \raggedright
566           \infof@nt

```

```

567         \rightinf@
568         \align@baseline}}
569     \end{picture}%
570 }
571 \ifx\c@vergraphics\@empty
572     \put(120,0){%
573         \makebox(120,120)[\@ligncover]{%
574             \parbox{12cm}{%
575                 \raggedright
576                 \coverf@nt
577                 \c@vertext
578                 \align@baseline}}}
579 \else
580     \put(130,10){%
581         \makebox(100,100)[\@ligncover]{%
582             \parbox{10cm}{%
583                 \raggedright
584                 \coverf@nt
585                 \c@vertext
586                 \align@baseline}}}
587 \fi
588 \end{picture}%
589 \end{rotate}%
590 }

```

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 `\jobname.dat` or `\jobname.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 L^AT_EX 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.

```

591 \DeclareRobustCommand*\@skCDfile}[1]{%
592   \def\CDname{#1}%
593   \ifx\CDname\@empty
594     \IfFileExists{\jobname.dat}{%
595       \def\CDname{\jobname.dat}%
596     }{%
597       \typein[\CDname]{Please insert CD data file name:}%
598     }%
599 \fi
600 \InputIfFileExists{\CDname.dat}{%
601 }{%
602   \InputIfFileExists{\CDname}{%
603 }{%
604   \ClassError{cd}

```

```

605             {CD data file (\CDname.dat or \CDname) not found}
606             }%
607     }%
608 }%
609 }
610
611 \DeclareRobustCommand*\makeCD}[1] [] {%
612   \@skCDfile{#1}%
613   \makecover\par
614   \makeback\par
615 }
616
617 \DeclareRobustCommand*\makeslimCD}[1] [] {%
618   \@skCDfile{#1}%
619   \makeslimcover\par
620 }
621
622 \newread\CDlist
623
624 \newcounter{@cd}
625 \setcounter{@cd}{0}
626
627 \newif\ifne@f
628
629 \DeclareRobustCommand*\@sklistfile}[1] {%
630   \def\CDlistname{#1}%
631   \ifx\CDlistname\@empty
632     \IfFileExists{\jobname.lst}{%
633       \def\CDlistname{\jobname.lst}%
634     }{%
635       \typein[\CDlistname]{Please insert CD list file name:}
636     }%
637   \fi
638   \IfFileExists{\CDlistname.lst}{%
639     \immediate\openin\CDlist=\CDlistname.lst
640   }{%
641     \IfFileExists{\CDlistname}{%
642       \immediate\openin\CDlist=\CDlistname
643     }{%
644       \ClassError{cd}
645         {CD list (\CDlistname.lst or \CDlistname) not found}
646         }%
647     }%
648   }%
649   \ne@ftrue
650 }
651
652 \DeclareRobustCommand*\makelist}[1] [] {%
653   \@sklistfile{#1}%
654   \advance\endlinechar\@M

```

```

655 \immediate\read\CDlist to \CDname
656 \advance\endlinechar-\@M
657 \ifeof\CDlist\ne@ffalse\fi
658 %
659 \@whilesw \ifne@f \fi {%
660   \newcd
661   \InputIfFileExists{\CDname.dat}{%
662   }{%
663     \InputIfFileExists{\CDname}{%
664     }{%
665       \ClassError{cd}
666         {CD data file (\CDname.dat or \CDname) not found}
667         }{%
668     }%
669   }%
670   \advance\endlinechar\@M
671   \immediate\read\CDlist to \CDname
672   \advance\endlinechar-\@M
673   \ifeof\CDlist\ne@ffalse\fi
674   \ifodd\value{@cd}%
675     \makeback[lrb]\par
676     \makecover\par
677   \else
678     \makecover\par
679     \ifne@f\makeback[lrt]\else\makeback\fi\par
680   \fi
681   \addtocounter{@cd}{1}%
682 }%
683 }
684
685 \DeclareRobustCommand*\makeslimlist}[1][ ]{%
686   \@sklistfile{#1}%
687   \advance\endlinechar\@M
688   \immediate\read\CDlist to \CDname
689   \advance\endlinechar-\@M
690   \ifeof\CDlist\ne@ffalse\fi
691 %
692 \@whilesw \ifne@f \fi {%
693   \newcd
694   \InputIfFileExists{\CDname.dat}{%
695   }{%
696     \InputIfFileExists{\CDname}{%
697     }{%
698       \ClassError{cd}
699         {CD data file (\CDname.dat or \CDname) not found}
700         }{%
701     }%
702   }%
703   \advance\endlinechar\@M
704   \immediate\read\CDlist to \CDname

```

```

705 \advance\endlinechar-\@M
706 \ifeof\CDlist\ne@ffalse\fi
707 \makeslimcover\par
708 }%
709 }
710 </class>

```

Change History

v0.0		v1.3	
General: Created file.	1	General: Implement .dat/.lst file-	
v0.1		name from job name.	1
General: New macros for data file		Implement covergraphics, etc. . .	1
handling.	1	v1.4	
v1.0		General: Add <code>\trackrange</code> com-	
General: Bump revision number. . .	1	mand.	1
v1.1		Fix typo in <code>\track</code> command. . .	1
General: Add support for slim		Make slim and normal back cov-	
cases.	1	ers use the same height.	1
Better positioning, too.	1	Make track list height config-	
		urable.	1