

Decorating CD-ROMs and DVDs (Tutorial)

Abstract

After having burned a disk you sometimes need to add a label and, if the disk is stored in a jewel case, a booklet and an inlay for the jewel case. The following article describes how to create a label for the disk on a commercial label-sheet and a booklet and an inlay for the jewel case. The following solutions are based on ConTeXt's built-in layer capabilities.

Keywords

ConTeXt, CD-ROM, DVD, label, booklet, inlay, layer.

Label

The label's several elements can be switched on/off with the following three modes:

- enabling *draft* mode will draw an outline of the label. Beware, simultaneously enabling the *withBackground* mode will obscure the outline.
- enabling *withLogo* mode will show an image at the top of the label.
- enabling *withBackground* mode will place the background image on the label.

```
\enablemode[draft]
\enablemode[withLogo]
\enablemode[withBackground]
```

We begin by specifying the main language.

```
\mainlanguage[en]
```

Next, we choose the label font. Since we are using ConTeXt MkIV, we will choose the Iwona-medium font in its of variant.

```
\usetypscript[iwona-medium]
\setupbodyfont[iwona-medium, 10pt]
```

All texts in the different boxes on the various layers are of type `\framed`. As a visual aid, *draft* mode switches on all their frames.

```
\doifmodeelse{draft}
  {\def\Onoff{on}}
  {\def\Onoff{off}}
```

We will place two images in the background. As a convenience, we define two macros whose names are the names of their respective images. The *Labelbackground* image covers the complete label, while the *Logo* image will appear at the top of the label when the *withLogo* mode is enabled.

```
\def\Labelbackground{fish}
\def\Logo{eurotexlogo}
```

Texts can be placed at the top and bottom of the label, and to the left and right of the disk's center. The bottom area is divided into three sections, each wide enough to fit, depending on the position of the text, within the available space.

To keep things uncluttered, we place the texts in buffers here and use only the buffers' names in the later typesetting instructions. All the texts will be placed in the `\framed` environment.

```
\startbuffer[BottomtextA]
  \green Contains all files for decorating a CD or DVD:\crIf
  \em \tfx CD-label, CD-inlay-booklet, CD-inlay for the jewel case
\stopbuffer

\startbuffer[BottomtextB]
  \green \em Published in the Euro\TEX -proceedings
\stopbuffer

\startbuffer[BottomtextC]
  \green 2009
\stopbuffer
```

There is one text area at the top of the label.

```
\startbuffer[Toptext]
  {\tfc \red CD/DVD decoration}\blank \green{\tfc Tutorial}\vfill Over-
lays and layers
\stopbuffer
```

The text areas to the left and right of the disk's center are, like the area at the top of the label, based on a single block of text.

```
\startbuffer[Righttext]
  \green CD-ROM
  \blank
  \gray \currentdate
\stopbuffer

\startbuffer[Lefttext]
  \green \TEX -tutorial
  \blank
  Euro\TEX 2009
\stopbuffer
```

By separating the content elements above from the typesetting commands below, we can change the content without worrying about the code below this point. We will add a comment to emphasize this.

```
% -- Basically you do not need to edit the lines below
```

First, we tell ConTeXt how the label will look when typeset.

```
\setuppapersize[A4][A4]
\setuppagenumbering[state=stop]
\setupcolors[state=start]

\setuplayout
  [topspace=0pt,
  backspace=0pt,
  header=0pt,
  footer=0pt,
  margin=0pt,
  width=210mm,
  height=297mm,
  marking=on,
  location=middle]
```

As mentioned above, enabling *draft* mode will draw the label outline. The drawing itself consists of two concentric circles drawn with MetaPost.

```
\startreusableMPgraphic{CDShape}
  draw fullcircle scaled 117mm;
  draw fullcircle scaled 41mm;
\stopreusableMPgraphic
```

ConTeXt provides the `\doifmode[]{}{}` command, which we will use to set the label background to our predefined background image when the *withBackground* mode is enabled.

```
\doifmode{withBackground}
  {\defineoverlay
   [Lbackground]
   [{\externalfigure
    [\Labelbackground]
    [width=\overlaywidth,height=\overlayheight]]}]}
```

We use the same mechanism to place the optional logo image.

```
\doifmode{withLogo}
  {\defineoverlay
   [Logo]
   [{\externalfigure[\Logo]
    [width=\overlaywidth,height=\overlayheight]]}]}
```

Here we define a layer that will cover the entire page. To indicate that we do not want relative positioning of this layer, we set its position to `no` and its reference point to `bottom`. Lastly, we place the layer into the page area as a background.

```
\definelaye[PageLayer][position=no]
\setuplayer
 [PageLayer]
 [corner=bottom,location=c,height=\paperheight]
\setupbackgrounds[page][background=PageLayer]
```

We define a second layer to hold the label fields we have already defined. We set this layer's reference point to `top` and `left` and the location of the layer data to `bottom right`. We also define its width and height. Set `option=test` to see what ConTeXt does with these settings.

```
\definelaye
 [Label]
 [position=no,corner={top,left},location={bottom,right},
  width=117mm,height=117mm,option=test]
```

In the following lines we fill the Label layer with our predefined label fields, and then typeset it. To tell ConTeXt to flush the layer at the end of the page we enclose the filling commands and the typesetting command: `\placelayer[]` within a `\standardmakeup ... \stopstandardmakeup` block.

```
\starttext
\startstandardmakeup[page=no,doublesided=no]
\setlayerframed
 [Label]
 [x=\dimexpr(\textwidth-117mm)/2,y=21.43mm]
 [width=117mm,height=117mm,frame=\Onoff,background=Lbackground]
 {}
```

```

\setlayerframed
[Label]
[x=\dimexpr(\textwidth-13mm)/2,y=22.43mm]
[width=13mm,height=13mm,frame=\Onoff,align={top,middle},
background=Logo]
{}
\setlayerframed
[Label]
[x=\dimexpr(\textwidth-78mm)/2,y=35.43mm]
[width=78mm,height=24mm,frame=\Onoff,align={top,middle}]
{\getbuffer[Toptext]}
\setlayerframed
[Label]
[x=126mm,y=60mm]
[width=34mm,height=40mm,frame=\Onoff,align={middle,lohi}]
{\getbuffer[Righttext]}
\setlayerframed
[Label]
[x=50mm,y=60mm]
[width=34mm,height=40mm,frame=\Onoff,align={flushleft,lohi}]
{\getbuffer[Lefttext]}
\setlayerframed
[Label]
[x=\dimexpr(\textwidth-98mm)/2,y=100.43mm]
[width=98mm,height=\dimexpr(38mm/3),frame=\Onoff,align=middle]
{\getbuffer[BottomtextA]}
\setlayerframed
[Label]
[x=\dimexpr(\textwidth-72mm)/2,y=\dimexpr(100.43mm+38mm/3)]
[width=72mm,height=\dimexpr(38mm/3),frame=\Onoff,align=middle]
{\getbuffer[BottomtextB]}
\setlayerframed
[Label]
[x=\dimexpr(\textwidth-18mm)/2,y=\dimexpr(100.43mm+38mm/3*2)]
[width=18mm,height=\dimexpr(38mm/3),frame=\Onoff,align=middle]
{\getbuffer[BottomtextC]}
\doifmode{draft}
{\setlayer[PageLayer][x=.5\paperwidth,y=216.93mm]
{\useMPgraphic{CDSshape}}
\setlayer[PageLayer][x=.5\paperwidth,y=79.93mm]
{\useMPgraphic{CDSshape}}}
\placelayer[Label]
\stopstandardmakeup
\stoptext

```

As you can see in the code above, we move a piece of information to its correct position by adjusting its vertical and horizontal offsets.

Near the end of the code there is another conditional action that controls the placement of the label shape.

The preceding code yields the following result:

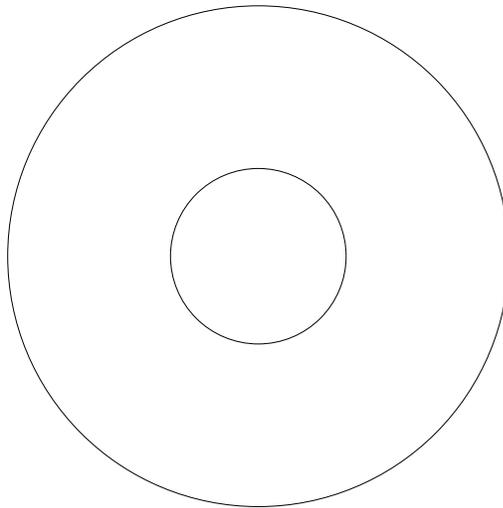
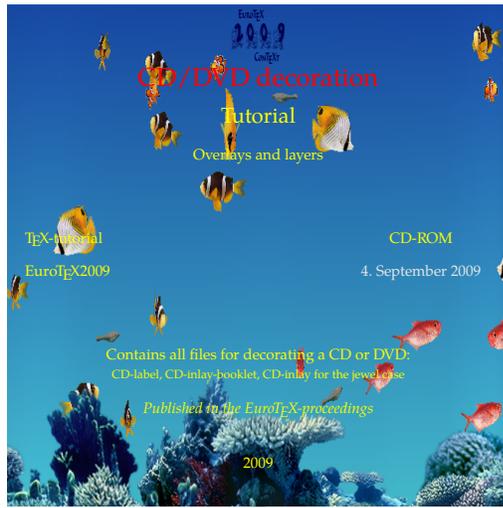


Figure 1. Example CD-label

Booklet

The CD-booklet is composed of a single section which we arrange with `\setuparranging[2UP]`. As before, we start with setting the main language and choosing the font for the booklet.

```
\mainlanguage[en]
\usetypescript[palatino]
\setupbodyfont[palatino,8pt]
```

If you want to place a background image on the front page, you can define a macro with a symbolic name which will be used later when you setup the page layer.

```
\def\Pageimage{fish}
```

We can add a title typeset along the spine of the cover page. The title is placed in a buffer:

```
\startbuffer[Sidetitle]
  {\tfc \yellow CD/DVD decoration}
\stopbuffer
```

In the next buffer we place the contents of the cover page.

```
\startbuffer[Covertext]
  \strut
  \blank[line]
  \startalignment[middle]
    \startlines
      {\tfc \red CD/DVD decoration}
      \blank \green{\tfb Tutorial}
      \blank Overlays and layers
    \stoplines
  \stopalignment
  \blank
\stopbuffer
```

The booklet contents is put into its own buffer.

```
\startbuffer[Bookletcontent]
  \input knuth\par
\stopbuffer
```

Now that all the content elements for the booklet have been defined, the positioning code that follows, once written, need not be changed.

`% -- Basically you do not need to edit the lines below`

The CD-booklet is typeset on a custom size of paper. We define this paper size and put it on landscape A4 sheets. We choose our layout parameters and, because we are using the standard-makeup-environment, we turn vertical centering off.

```
\definepapersize[CDbooklet][width=120mm,height=118mm]
\setuppapersize[CDbooklet][A4,landscape]
\setuppagenumbering[location={bottom,right},alternative=doublesided]
\setupcolors[state=start]
\setupnarrower[left=.5em]
\setuplayout
  [topspace=2mm,
  backspace=9mm,
  header=0pt,
```

```

    footer=5mm,
    margin=8.5mm,
    margindistance=.5mm,
    width=100mm,
    height=115mm,
    marking=on,
    location=middle]
\setupmakeup[standard][top=,bottom=]

```

The background image on the cover should be a bleeding image, i.e. it should be larger than the crop marks indicate. However, as soon as we use imposition, the image is cropped to the paper size and the bleed is gone. We define the bleed as follows:

```
\definemeasure[bleed][\dimexpr1mm\relax]
```

Now we define a layer which is filled with the background image.

```

\definelayer
  [Background]
  [position=no]
\setlayer
  [Background]
  {\externalfigure
   [\Pageimage]
   [height=\dimexpr118mm-2\measure{bleed},
    width=\dimexpr120mm+2\measure{bleed}]}

```

The filled layer is placed into the page background. Because we bled the image, we have to add a background offset equal to the bleed.

```
\setupbackgrounds[page][background=Background,
backgroundoffset=\measure{bleed}]
```

As mentioned earlier, the booklet is typeset with imposition.

```
\setuparranging[2UP]
```

Now that everything is in place, we can produce the booklet.

```

\starttext
\startstandardmakeup[doublesided=no,page=yes]
  \inleft{\rotate[rotation=90]{%
    \framed[frame=off,align={lohi,middle},width=\textheight]
    {\bfd \getbuffer[Sidetitle]}}}
  \getbuffer[Covertext]
\stopstandardmakeup
\setupbackgrounds[page][background=]
\getbuffer[Bookletcontent]
\stoptext

```

These parameters will produce the following (only the cover page is shown):

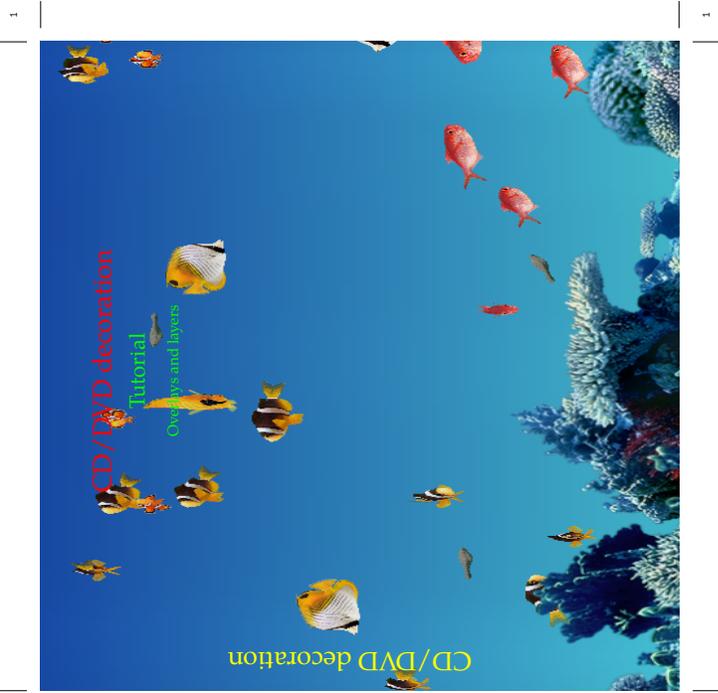


Figure 2. The cover of the booklet

Jewel case inlay

To complete the CD project, we want to prepare an inlay for the jewel case.

As in the previous sections, we start with setting the main language:

```
\mainlanguage[en]
```

We tell ConTeXt the font we want to use for the inlay

```
\usetyescript[palatino]
\setupbodyfont[palatino,10pt]
```

The inlay will be defined such that you can have as many as three images on the inlay. There is a page image, a text image and an image for the small strips to the left and right of the inlay. Again, we define macros in order to replace the actual filename in the code with a symbolic name.

```
\def\Pageimage{fish}
\def\Textimage{eurotexlogo}
\def\SideTitleimage{}
```

We have three text areas. At the left and right sides of the inlay there are small strips along the spine for the title information. Both strips are automatically filled with the same information. In between these is the main text area.

```
\startbuffer[Sidetitle]
  \tfa \yellow CD/DVD decoration\hfill2009
\stopbuffer
```

```
\startbuffer[Maintext]
  \startalignment[middle]
  \startlines
    {\tfc \red CD/DVD decoration}
    \blank \yellow{\tfb Tutorial}
    \blank Overlays and layers
  \stoplines
  \stopalignment
\stopbuffer
```

The following comment reminds us that the code below this point is usually left untouched.

```
% -- Basically you do not need to edit the lines below.
```

We must define the inlay paper size ourselves. Its dimensions are 150 × 118 mm and typeset on an A4.

```
\definepapersize[CDinlaycase][width=150mm,height=118mm]
\setuppapersize[CDinlaycase][A4]
```

We define a bleed measure to insure that the page image will cover the entire page.

```
\definemeasure[bleed][\dimexpr1mm\relax]
```

We specify the various layout settings. The width of the main text area is 14 mm smaller than the paper width. We use a backspace of 7 mm, which is filled with the margin + margin distance.

```
\setuplayout
  [topspace=0mm,
  backspace=7mm,
  margin=6.5mm,
  header=0pt,
```

```

    footer=0pt,
    margindistance=.5mm,
    width=136mm,
    height=118mm,
    location=middle,
    marking=on]

```

```
\setupcolors[state=start]
```

```
\setupmakeup[standard][bottom=,top=]
```

To fit the inlay into the jewel case we have to make two folds. It is important to make these folds accurately. To help, we add a layer with fold marks that extend out into the cut space.

```

\definelaye
[Foldmarks]
[position=no,
height=\dimexpr(\paperheight+10mm),
width=\dimexpr(\paperwidth+10mm),
x=0mm,
y=-8mm]

```

Next we define two layers, one each for the page and text images. We do not want the layers to be positioned relative to the text so we set each position to no.

```

\definelaye
[Pagebackground]
[position=no]

```

```

\definelaye
[Textbackground]
[position=no]

```

As mentioned earlier, we can use an overlay to add an image or a background along the spine. The image could be a previously defined external image, and the background could be a transparent color generated with MetaPost.

```

\defineoverl
[SideTitlebackground]
[{\externalfigure[\SideTitleimage][width=\overlaywidth,
height=\overlayheight]]]

```

or

```

\defineoverl
[SideTitlebackground]
[\useMPgraphic{TransparentBackground}]

```

We define two additional layers intended for the side titles. Again, there is no need for relative positioning.

```

\definelaye
[SideTitleL]
[position=no]

```

```

\definelaye
[SideTitleR]
[position=no]

```

We use MetaPost for the fold marks.

```

\startuniqueMPgraphic{Marks}
  path p;
  pair pt[];
  p := unitsquare xscaled 136mm yscaled 135mm;
  pt[1] := llcorner p;
  pt[2] := point 3.95 of p;
  pt[3] := point 3.05 of p;
  pt[4] := ulcorner p;
  pt[5] := lrcorner p;
  pt[6] := point 1.05 of p;
  pt[7] := point 1.95 of p;
  pt[8] := urcorner p;

  for i= 1 step 2 until 7 :
    draw pt[i]--pt[i+1];
  endfor;
\stopuniqueMPgraphic

```

It is easy to prepare a transparent colored background for an overlay with MetaPost.

```

\startreusableMPgraphic{TransparentBackground}
  path p;
  p:= unitsquare xscaled \overlaywidth yscaled \overlayheight;
  fill p withcolor transparent(1,0.3,yellow);
\stopreusableMPgraphic

```

Now we are ready to fill the layers with their respective content, and assign the layers as page or text backgrounds.

```

\setlayer
  [Foldmarks]{\useMPgraphic{Marks}}
\setlayer
  [Pagebackground]
  {\externalfigure
   [\Pageimage]
   [height=\dimexpr118mm+2\measure{bleed},
    width=\dimexpr150mm+2\measure{bleed}]}
\setlayer
  [Textbackground]
  {\externalfigure[\Textimage][height=\textheight,width=\textwidth]}
\setlayer
  [SideTitleL]
  [x=.5mm,y=.5mm]
  {\rotate[rotation=90]{%
   \framed
   [frame=off,align={right,lohi},
    width=\dimexpr\textheight-1mm,
    background=SideTitlebackground]
   {\bf \getbuffer[Sidetitle]}}}
\setlayer
  [SideTitleR]
  [x=-.5mm,y=.5mm]
  {\rotate[rotation=90]{%
   \framed
   [frame=off,align={right,lohi},
    width=\dimexpr\textheight-1mm,

```

```

background=SideTitlebackground]
{\bf \getbuffer[Sidestitle]}}
\setupbackgrounds
[page]
[background=Pagebackground,backgroundoffset=\measure{bleed}]
\setupbackgrounds[text][background={Foldmarks,Textbackground}]
\setupbackgrounds[text][leftmargin][background=SideTitleL]
\setupbackgrounds[text][rightmargin][background=SideTitleR]

```

What remains to be done is to start a document and add the information for the main text area.

```

\starttext
\strut
\framedtext
[frame=off,
rulethickness=3pt,
offset=10pt,
width=\textwidth,
height=\textheight,
align=middle]
{\getbuffer[Maintext]}
\stoptext

```

The result is as follows:



Figure 3. The inlay for the jewel case

Further reading

- ConT_EXt the manual. Hans Hagen. November 2001, available from <http://www.pragma-ade.com/>
- Metafun. Hans Hagen. January 2002, available from <http://www.pragma-ade.com/>
- It's in the details. Hans Hagen. Spring 2002, available from <http://www.pragma-ade.com/>
- <http://wiki.contextgarden.net/Layers>.

Conclusion

This small project demonstrates ConT_EXt's capability to place information and graphics at specific locations using layers. Once you become familiar with how they work, you will find many more situations where layers can be used in modern design. It is worthwhile having a look at the literature references given above since there is more to be said about layers than can be presented in this project.

Acknowledgements

I would like to thank Michael Guravage for proofreading this article and for all the improvements he has made to it.

Willi Egger
w.egger (at) boede.nl