The flabels package*

Volker Börchers†
Universität Bremen

1999/05/19

Abstract

This package provides macros for typesetting pretty labels (optionally colored) for the back of files or binders. So far they are only applicable for the special format of the (for a4 paper) widely used "Leitz-Ordner" (ring binder). We use 2 macros, the first one for a number of empty labels (for handwriting) while the second contains a text field.

Contents

1 Introduction

Thanks to photo-copiers a lot of paper is produced that has to be kept somewhere and (perhaps) even in some order. If you want to keep an overview over it you will punch it and put it into a file/binder (otherwise one or more stacks will do). This package have been invented to make life with this binders a little bit nicer, prettier and more colored.

*This file has version number v1.0, dated 1999/05/19.

†email: boercher@physik.uni-bremen.de

The `flabels` ('filelabels') package provides two macros for producing labels for the back of a binder in the wide-spread format (for a4 paper!) 28.3 × 31.8 cm (11.1 × 12.5 inch) with the back height 31.8 cm (12.5 inch) and different available widths.

I hope that further versions of the `flabels` package will support more than one binder format. (But this will depend on *You*, the friendly reader who want to contribute to `flabels` by adding support for *Your* favourite binder `:-)`.

2 Using the flabels package

Invoke the `flabels` package by requesting it in the preamble (Note that this package requires LaTeX 2_ε!):

```
\usepackage[...]{flabels}
```

2.1 Quick start

- Use option `a4paper` if you need it.
- Check the format of the binder you have: If the label should be 37mm wide choose option `narrow`; the *default* width is 60mm.
- Decide if you want large labels (full height of the binder back). If you do include option `fullheight` in the `usepackage` command.
- Use option `color` if you have a color printer or if your printer at least can translate colors into greyscale-'colors'.
- Neither call any package that sets page layout parameters (like `\textheight`) *after* `flabels` nor set them by yourself.
- If you are using colors: Create a file `color.cfg` including the line `\ExecuteOptions{<driver>}`, where `<driver>` stands for your color graphics driver, e.g. `dvips` or `xdvi` (see description of the `color` package).
- If you are using colors: Set the fore- and background colors for the "company label" and the whole label using `\setbgcompany`, `\setfgcompany`, `\setbglabel`, `\setfglabel`, each one of this accepts a color name as argument.
- Choose the text for the "company label" (the command for getting the default would be `\company{\huge LEITZ}`).
- In the document:
  - If you want to create 4 "empty" labels for handwriting use command `\emptylabel{4}`
  - If you want to print a label with text use the macro `\labeltext{text}`.
  - Avoid empty lines (paragraphs) between the label because this may lead to a wrong vertical and horizontal placement of the labels.
- Check for overfull `\hboxes` and (if necessary for the printer) shift the labels on the page up- or downwards using the macro `\extratopmargin<dim>`.

2.2 Options

Options

The package recognizes the following options: `leitz`, `fullheight`, `narrow`, `color`, `a4paper`, `nice` and `nohole`.

`leitz` This option is more-or-less a "dummy option" since `leitz` is the default and, even more, the user has no other choice. Nevertheless it is an option because this should change in the future. To cut a long story short: You can ignore this option for now!

`fullheight` See the pictures for the difference between `fullheight` and labels of normal height in sec. 2.4. You may switch from large labels to the smaller ones (and vice versa) whenever you want from inside the document using the macros `\fullheight` and `\normalheight`.

`narrow` Change the default width of the labels from 60mm to 37mm. You may also change the width from inside the document with the commands `\narrowlabels` and `\widelabels`.

`color` Use this option if you have a color printer[1] to include the standard LaTeX 2_ε package `color`. This package (see `color`-documentation `grfguide.tex`) needs to know the driver for which the color informations should be generated. So you have to create a file `color.cfg` including the line `\ExecuteOptions{<driver>}`, where `<driver>` stands for your color graphics driver, e.g. `dvips` or `xdvi`. The colors to be used may be set in the preamble or in the document (see sec. 2.3).

`a4paper` So far only "Leitz-Ordner" are supported by this package which are used for a4 paper. So normally you will include this option (or use the equivalent option for the document classes.

`nice` This option should be used *only* if you want to include labels in a normal document! Normally this package uses the whole page, changing nearly every layout parameter. Since this makes `flabels` very un-co-operative and forbids the use of this package inside a normal text document (unlikely that there is any need for this this but the documentation you read now uses the option `nice`). The option `nice` makes `flabels` nicer to other packages. Warning: If you use this option you have to take care for correct placement on the page for yourself!

`nohole` Do not include a marker for the hole in the back of a "Leitz-Ordner" (to be used for an exact alignment of the label).

2.3 User commands

It follows a complete list of all user accessible macros, lengths and counters.

2.3.1 Label properties

Some of these macros have been described before in the options section.

`\narrowlabels`

- To switch to narrow labels use the macro `\narrowlabels`.

\widelabels
- To switch to wide labels use the macro `\widelabels`.

\fullheight
- To switch to large labels use the macro `\fullheight`.

\normalheight
- To switch to small labels use the macro `\normalheight`

\company
- To change the text of the "company label" use the macro `\company{<name>}`

numberauxlines
- Change the number of auxiliary lines in `\emptylabel` this way: `\setcounter{numberauxlines}{<number>}` (default is `4`). Usually you will also change the distance between the lines ...

\auxlinedistance
- ... using the macro `\auxlinedistance<dim>` where `<dim>` can be any valid TEX-dimension.

\companylabelheight
\ylowercompany
\yuppercompany
- You may also change the appearance of the "company label" via the redefinition of this three *macros*: `\companylabelheight`, `\ylowercompany` and `\yuppercompany`. Each of them are set to a *number*, that is a lenght measured in the unit `mm`. You can get the default height of the company label e.g. by `\renewcommand{\companylabelheight}{17}`, since –by default– this height is `17mmm`. The other two macros determine the *y-coordinate of the bottom* of the upper and the lower label. Their default (for `leitz`) are `160` resp. `-68`.

2.3.2 Changing colors

\setbgcompany
\setfgcompany
\setbglabel
\setfglabel
Set the fore- and background colors for the "company label" and the whole label using `\setbgcompany`, `\setfgcompany`, `\setbglabel`and `\setfglabel` (`fg` stands for 'foreground' and `bg` for 'background'), e.g. `\setbglabel{white}`.

You have to use (predefined) color-*names*, like `black` or `red`. Which names are already defined depends on you graphics driver. You may define *new* colors following the `color`-package documentation `grfguide.tex`.

2.3.3 Layout parameters

As mentioned in the options section (see discussion of option `nice`, this package leaves only few layout parameters untouched to make the labels fit onto a page. Vertically there is no room for a user to adjust anything except the distance between the border of the paper and the labels. This is accessible via the macro

\extratopmargin
`\extratopmargin<dim>`. The default for this length is `0mm` – then the large labels will be vertically centered on the page. With a positive `\extratopmargin` the labels will be shifted down (negative values are allowed).

The horizontal layout is not that restricted, for example you can change the text width (as usual using the macro `\textwidth`) and all margins. The distance between two labels is generally handled by TEX like any space between two words. (There must be space between the labels in order to give TEX the chance to break "lines" of labels!) You may want to specify a *minimum* distance (default is `0mm`) using `\setlength{\hspaceinterlabel}{<dim>}`.

\hspaceinterlabel

\labeltextmargin
The distance between the text box of `\labeltext` and the label border is controlled by the length `\labeltextmargin`. It is preset to `3mm`. This value is at the same time top-, left- and right margin of the text (the lower boundary is free). It can be changed directly this way: `\setlength{\labeltextmargin}{<dim>}`.

[1] You may also use colors if your printer can translate real colors into greyscale-'colors'!

2.3.4 Creating labels

\emptylabel
\labeltext

To create labels you have the choice between 2 macros: `\emptylabel{<number>}` and `\labeltext{<text>}`. Instead of a longish discussion look at the examples in sec. 2.4. Note that the argument of `\labeltext` is typeset as plain text – there is no predefined font change etc. The only special is the switch to the text color `\c@lorl@bel@fg` (if option `color` is chosen). If you want all your labels to be typeset in a special font perhaps with a title in a larger font size, you should define a macro, e.g. like this:

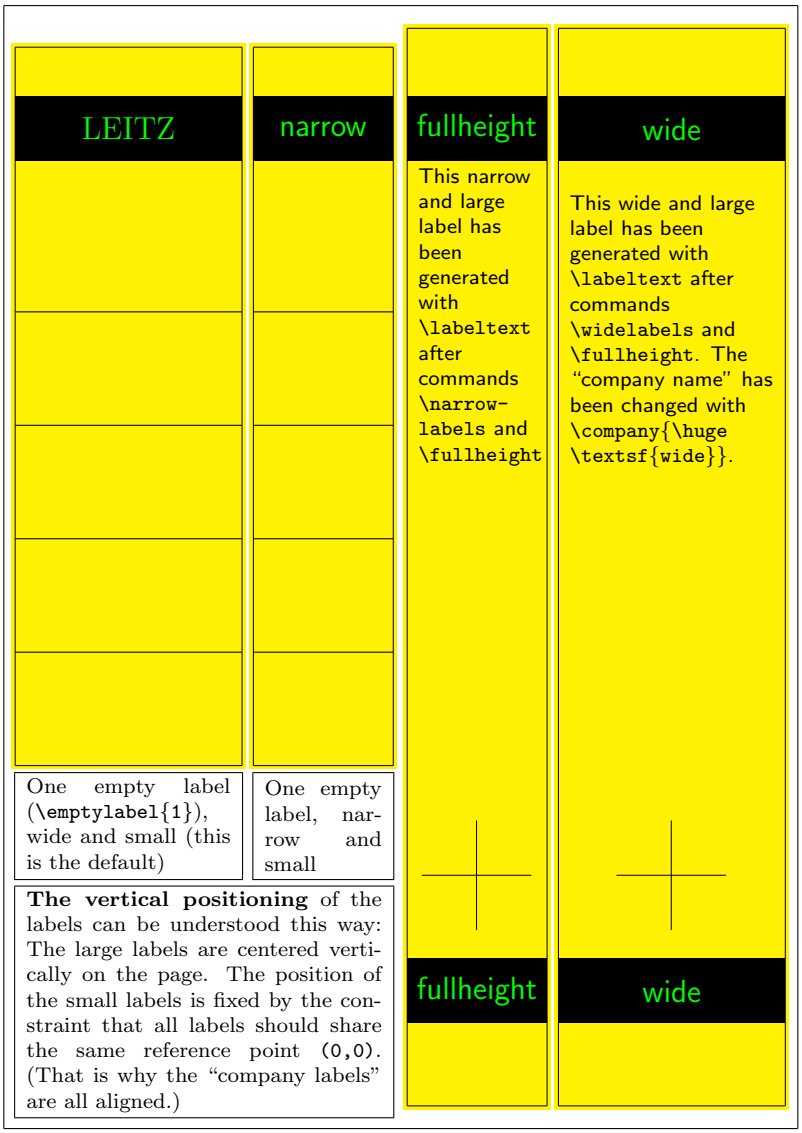
```
\newcommand{\mylabel}[2]{\labeltext{\vspace{1ex}\sffamily%
 \begin{flushleft}%
  \textbf{{\huge #1}}\\
  \textsl{\Large\flushleft{\large #2}}%
 \end{flushleft}}}
```

To get a good positioning of the labels may take a while. The best advice I can give is: Avoid 'paragraphs' between the labels, perhaps you have to remove also any 'space' by ending the lines with a `%`. Then insert pagebreaks, `vspace` and `hspace` as needed. Having both large and small labels (`\fullheight` and `\normalheight`) on one page results in a wrong vertical placement of the smaller labels (the *bottoms* of the labels instead of their reference points are vertically aligned)[2]

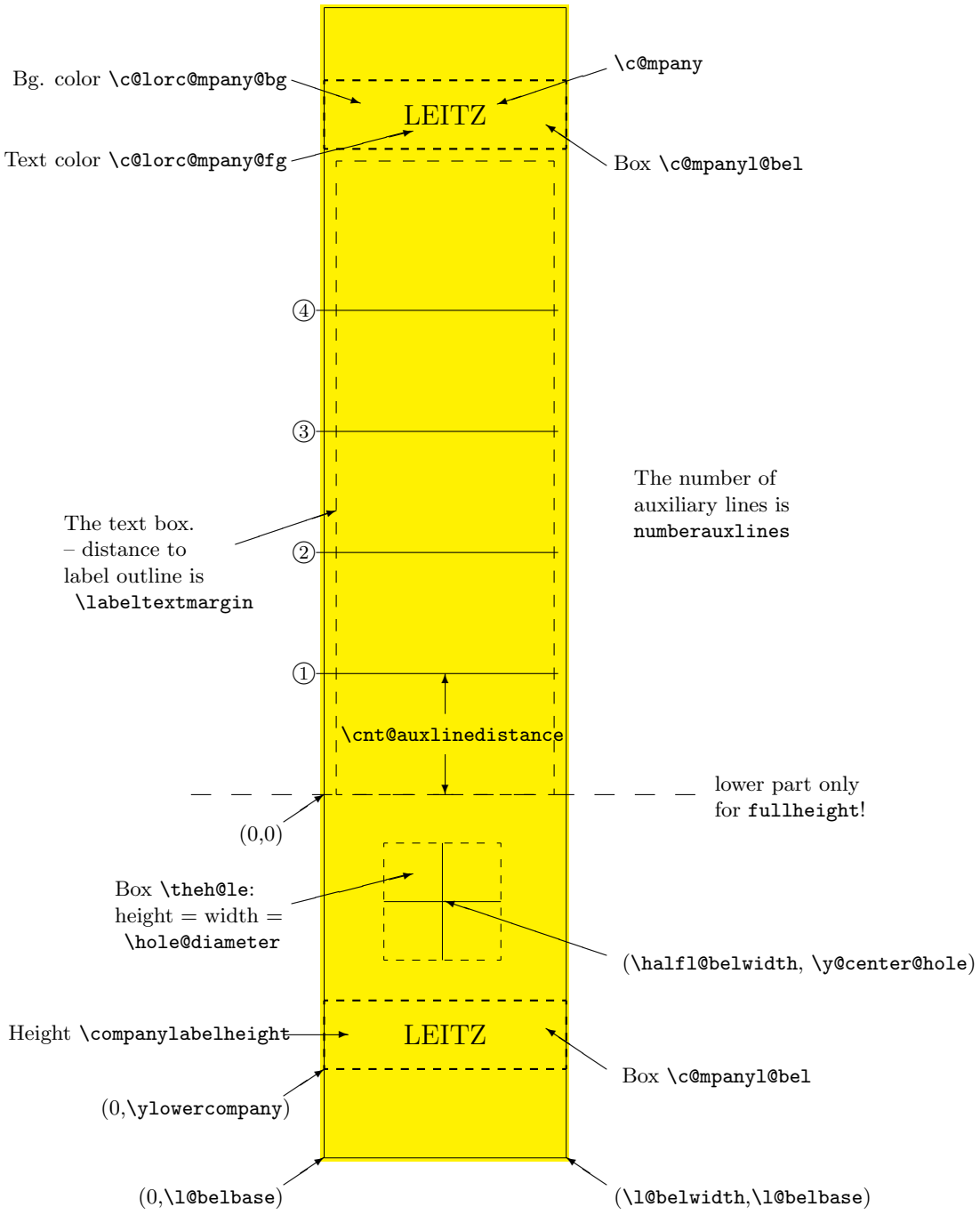
2.4 Examples

The following labels are drawn with a `\unitlength` of `0.5mm` instead of `1mm` (that is why they fit onto this page!). The *vertical* label position on the paper (paper contour is drawn too) is the original one, while the *horizontal* positioning shown here is very unusual (therefore you had to vary `\textwidth`, `\odd-` and `\evensidemargin`).

[2]The reason for that I currently do not unterstand, perhaps *You* do?!



The next picture is not an example, but gives a listing of nearly all dimensions used by this package and a description of 'empty' and text labels. Note that all variables containing an `@` are not directly accessible in the document (but some are via macros).



2.5 To do

- The package should support further binder formats!
- Perhaps the package should provide further special labels e.g. for a table of contents on the first page etc.

- Perhaps there should be an option for a landscape mode (using the `Lscape` package)?
- Perhaps the package should care for horizontal spacing and should help to avoid `Overfull \hbox` warnings.
- Perhaps even more lengths should be user-accessible (most of the macro names contain an `@` so you can not use them in a normal document).
- To be solved: Placing full- and normal-height labels on one page currently causes problems — I have no idea why.

If You have any problems, suggestions, critical remarks – or whatever according this package write to Volker Börchers (email-address see title).

3 The Macros

1 ⟨∗package⟩

`Options` At the start of the package all Options are declared but for most options the implied actions are executed later (controlled by a corresponding `\newif`).

`leitz` As mentioned in section 2.2 the option `leitz` is so far a "dummy option" since there is no alternative. If you want to add support for another label format you should make it choose-able as an option. The declarations below are only a part of the format-dependent settings (especially all widths are not set here). Some (not all!) other of these settings are in a `\ifl@itz ... \fi` construction (see macros `\f@llheight` and `\n@rrow`).

```
2 \newif\ifl@itz \l@itztrue
3 \DeclareOption{leitz}{\l@itztrue
4  \newcount\cnt@auxlinedistance \cnt@auxlinedistance=30
5  \newcounter{numberauxlines}\setcounter{numberauxlines}{4}
6  \def\hole@diameter{29}
7  \def\hole@radius{14.5}
8  \def\y@center@hole{-26.5}
9  \def\ylowercompany{-68}
10 \def\yuppercompany{160}
11 \def\companylabelheight{17}
12 \def\f@lll@belheight{285}}
```

`fullheight` If this option is chosen, the macro `\fullheight` (respectively `\f@llheight1`) is invoked later. The label will then extend (nearly) over the full height of the binder back (this is not the default).

```
13 \newif\iff@llheight \f@llheightfalse
14 \DeclareOption{fullheight}{\f@llheighttrue}
```

`narrow` Invoke the macro `\narrowlabels` (resp. `\n@rrow1`) later to create narrower labels (default is wide labels).

```
15 \newif\ifn@rrow \n@rrowfalse
16 \DeclareOption{narrow}{\n@rrowtrue}
```

`color` Load the package `color` later. This is not the default because not everyone has a color printer.

```
\newif\ifcolorl@bel \colorl@belfalse
\DeclareOption{color}{\colorl@beltrue}
```

`a4paper` The same thing that the `\documentclass` option `a4paper` does. Perhaps this option should imply further actions?.

```
\DeclareOption{a4paper}{\paperheight 297mm\paperwidth 210mm}
```

`nice` This option turns off the special, extreme, page layout of package `flabels`.

```
\newif\ifbe@nice \be@nicefalse
\DeclareOption{nice}{\be@nicetrue}
```

`nohole` Do not include a marker for the hole in the back of a "Leitz-Ordner" (to be used for an exact alignment of the label). A new option for a binder that do not have such a hole should include the command `\ExecuteOptions{nohole}`.

```
\newif\ifno@hole \no@holefalse
\DeclareOption{nohole}{\no@holetrue}
```

The end of the option section of the code: Produce an error message when unknown options (type mistakes...) are given; make `leitz` the default and force evaluation of the given option list.

```
\DeclareOption*{\PackageWarning{flabels}{%
    Unknown option `\CurrentOption' (Known option:\MessageBreak
    `fullheight', `narrow', `color', `nohole', `a4paper', `nice')}}
\ExecuteOptions{leitz}
\ProcessOptions\relax
```

`\unitlength` `\C@nvertToCount` `\cnt@paperheight` The unitlength of the pictures used to create the labels is 1 millimeter. Since TEX-dimensions are internally represented by integer variables (unit `sp`: scaled point) they are also 'counters'. This macro converts `sp` into `mm` (code taken from `layout` package). `\cnt@paperheight` keeps the paper height in `mm` as a counter.

```
\setlength{\unitlength}{1mm}
\def\C@nvertToCount#1#2{#1=#2 \divide #1 by 186468}
\newcount\cnt@paperheight
\C@nvertToCount\cnt@paperheight\paperheight
```

`\cnt@pictvoffset` `\tmp@count` `\tmp@count` The counter `\cnt@pictvoffset` (to be computed later) is used to place the label vertically centered on the page. `\tmp@count` and `\tmp@dim` are used as temporary variables.

```
\newcount\cnt@pictvoffset
\newcount\tmp@count
\newdimen\tmp@dim
```

`pagelayout` Since the labels can be very large (compared to the paper height) we use the full page height if option `nice` is *not* chosen. (This is not the same as the `fullpage` package does. Here we use really the whole page!) Horizontally this package leaves only a margin of `1in` on the left side but leaves `\textwidth` untouched.

On the other hand this makes `flabels` very sensible to the sequence of `\usepackage` commands. If one loads a package after `flabels` that overwrites this settings again (like `a4`) the layout will be wrong.

```
\ifbe@nice\relax
\else
 \textheight\paperheight
 \topmargin -1in
 \oddsidemargin 0mm
 \evensidemargin 0mm
 \marginparwidth 0mm
 \marginparsep 0mm
 \headheight0mm
 \headsep0mm
\fi
```

`\extr@topmargin` `\extratopmargin` Definition of a dimension for extra vertical space at the top of a page and the macro `\extratopmargin` as an user interface to it. Note: A `\topmargin` of `-1in` means that the text/labels can start right at the top of the page. The way the macro is defined hides this offset of `-1in` from the user.

```
\newdimen\extr@topmargin \extr@topmargin=0mm
\def\extratopmargin#1{\extr@topmargin=#1
  \advance\extr@topmargin by -1in
  \topmargin\extr@topmargin}
```

`\cnt@auxlinedistance` `\auxlinedistance` The counter `\cnt@auxlinedistance` contains the distance between two auxiliary lines (for `\emptylabels`) in the unit `1mm`. It is preset by the option `leitz` to the value 30. To allow the user to use other dimensions than `mm` the macro `\auxlinedistance` is used.

```
\def\auxlinedistance#1{\tmp@dim=#1
 \C@nvertToCount\cnt@auxlinedistance\tmp@dim}
```

`\labeltextmargin` `\l@beltextwidth` `\hspaceinterlabel` `\l@beltextwidth` is the width of the text in the label text field – it is a computed value (see macro `\labeltext`) since the label width is variable. The user can set `\labeltextmargin`, the margin between text and label outline (left and right side, top) instead to adjust the text width. `\hspaceinterlabel` (also user-accessible) is the *minimal* horizontal space between two labels (it is more usually).

```
\newdimen\labeltextmargin \labeltextmargin=3mm
\newdimen\l@beltextwidth
\newdimen\hspaceinterlabel \hspaceinterlabel=0mm
```

`\label@textheight` The text box extends from `y=0` to the "company label". We compute the text height in macro `\labeltext` as `\label@textheight` = `\yuppercompany` - `\labeltextmargin`.
Note: making the box smaller moves the text down.

```
\newcount\label@textheight
```

Now the options `fullheight`, `narrow` and `color` have to be processed. (Till now only a corresponding `\newif` has been set.)

`->fullheight` `\full@height` The first of this three options is `fullheight`. Of course we have to set here a big part of the label dimensions. Firstly we define a macro for internal use (the user interface for it is provided by the macros `\fullheight` and `\normalheight`). (For the explanation of `\l@belheight` and –`width` see figure on page 6.)

```
\def\full@height#1{%
 \ifl@itz
```

```
  \ifnum #1=0
    \f@llheightfalse
    \def\l@belheight{190}
    \def\l@belbase{0}
  \else
    \f@llheighttrue
    \def\l@belheight{\f@lll@belheight}
    \def\l@belbase{-90}
  \fi
 \fi
```

`\cnt@pictvoffset` The small labels (`normalheight`) should have the same y-coordinate `y=0` as the big labels *on the page*. We could achieve this by setting the y-offset (argument of the label picture environment!) to `\l@belbase`. Then the top of the big labels would start right the beginning of the paper.

If we additionally want the big labels (`fullheight`) be vertically centered on the whole page we have to divide the remaining vertical space between top and bottom (this ends the macro `\full@height`):

```
 \cnt@pictvoffset=\cnt@paperheight
 \advance\cnt@pictvoffset by -\f@lll@belheight
 \divide\cnt@pictvoffset by 2
 \advance\cnt@pictvoffset by \l@belbase
}
```

After having defined `fullheight` and `normalheight` now we can call `\full@height` with the appropriate argument:

```
\iff@llheight
 \full@height1
\else
 \full@height0
\fi
```

`->narrow` `\n@rrow` Now option `narrow`! Here *all horizontal* dimensions have to be set. As for `fullheight` we define a macro for internal use and call it then according to the chosen option.

```
\def\n@rrow#1{%
 \ifl@itz
  \ifnum #1=0
    \n@rrowfalse
    \def\l@belwidth{60}
    \def\halfl@belwidth{30}
  \else
    \n@rrowtrue
    \def\l@belwidth{37}
    \def\halfl@belwidth{18.5}
  \fi
 \fi
}
```

Process option `narrow` now:

```
\ifn@rrow
 \n@rrow1
\else
```

```
\n@rrow0
\fi
```

->color The option `color` first loads the package `color`. — *Note:* The package `color` needs to know for which driver color informations have to be generated (`dvips`, `xdvi`, ...). While we can not select the driver here (as an option: `\RequirePackage[<driver>]{color}`), we require the `color` macros *now*. So the user must specify the driver in the file `color.cfg` with an `\ExecuteOptions{<driver>}` command.

\c@lorc@mpany@bg
\c@lorc@mpany@fg
\c@lorl@bel@fg
\c@lorl@bel@bg

The colors of the labels will depend on 4 colors: `\c@lorc@mpany@bg` (background of the "company label"), `\c@lorc@mpany@fg` (foreground of the "company label"), `\c@lorl@bel@fg` (color of the labeltext), `\c@lorl@bel@bg` (back ground of the whole label). We define 4 macros as an user interface to these colors and use them to set the default colors.

```
\ifcolorl@bel
 \RequirePackage{color}
 \def\setbgcompany#1{\def\c@lorc@mpany@bg{#1}}
 \def\setfgcompany#1{\def\c@lorc@mpany@fg{#1}}
 \def\setbglabel#1{\def\c@lorl@bel@bg{#1}}
 \def\setfglabel#1{\def\c@lorl@bel@fg{#1}}
 \ifl@itz
  \setbgcompany{black}
  \setfgcompany{green}
  \setbglabel{yellow}
  \setfglabel{black}
 \fi
\fi
```

\c@mpanyl@bel
\rem@kecompanyl@bel

Small labels have a "company label" at the top of the label and large labels in addition another near the bottom. It is a small box that may contain the logo of the company or institute or perhaps the user's name. We save it in the box `\c@mpanyl@bel`. The macro `\rem@kecompanyl@bel` is used for each label by `\emptylabel` and `\labeltext`.

The text in this box is kept in the macro `\c@mpany` and is user-accessible with the macro `\company` (defined below).

```
\ifl@itz
 \def\c@mpany{\huge LEITZ}
\fi
\newsavebox{\c@mpanyl@bel}
\def\rem@kecompanyl@bel{%
 \ifcolorl@bel
  \protect\savebox{\c@mpanyl@bel}{%
   \fboxsep0pt\protect\colorbox{\c@lorc@mpany@bg}{%
    \protect\makebox(\l@belwidth,\companylabelheight){%
     \color{\c@lorc@mpany@fg}\c@mpany}}}%
 \else
  \thicklines\protect\savebox{\c@mpanyl@bel}{%
   \protect\framebox(\l@belwidth,\companylabelheight){\c@mpany}}%
 \fi
}
```

\theh@le

This is specific to the label format. A "Leitz-Ordner" has a hole in the back of the binder to make the large binder handier (diameter 29mm). Since TEX is not

capable to typeset circles larger then 15mm, we mark this hole with a cross. The reference point of this box is its center.

```
\newsavebox{\theh@le}
\savebox{\theh@le}{%
 \ifl@itz
  \begin{picture}(\hole@diameter,\hole@diameter)(\hole@radius,\hole@radius)
   \linethickness{0.01pt}
   \put(0,\hole@radius){\line(1,0){\hole@diameter}}
   \put(\hole@radius,0){\line(0,1){\hole@diameter}}
  \end{picture}%
 \fi
}
```

`\l@beloutline` This macro draws the common label outline for text and empty labels. It has to be called from inside a picture environment. The background color of the label is realised by putting a box that is slightly larger than the label (2× 0.5mm more) on the appropriate place.

```
\def\l@beloutline{
 \fboxsep0.5mm
 \ifcolorl@bel
  \put(-1,\l@belbase){%
   \colorbox{\c@lorl@bel@bg}{\makebox(\l@belwidth,\l@belheight){\relax}}}
 \fi
 \linethickness{0.01pt}
 \multiput(0,\l@belbase)(\l@belwidth,0){2}{\line(0,1){\l@belheight}}
 \multiput(0,\l@belbase)(0,\l@belheight){2}{\line(1,0){\l@belwidth}}
 \thinlines
 \put(0,\yuppercompany){\usebox{\c@mpanyl@bel}}
 \iff@llheight
  \ifno@hole\relax\else
   \put(\halfl@belwidth,\y@center@hole){\usebox{\theh@le}}
  \fi
  \put(0,\ylowercompany){\usebox{\c@mpanyl@bel}}
 \fi
}
```

`User commands` To switch between `fullheight`/`normalheight` and `narrow`/`wide` from inside the document we have these macros:

`\narrowlabels` `\widelabels` `\fullheight` `\normalheight`

```
\def\narrowlabels{\n@rrow1}
\def\widelabels{\n@rrow0}
\def\fullheight{\full@height1}
\def\normalheight{\full@height0}
```

To redefine the company name:

```
\def\company#1{\def\c@mpany{#1}}
```

`\emptylabel` The first macro that actually creates label is `\emptylabel`. It has as argument the number of identical labels to draw. Every 'empty' label consists of the common label outline and a number (`numberauxlines`, user accessible) of auxiliary lines.

```
\def\emptylabel#1{%
 \rem@kecompanyl@bel
 \tmp@count=#1\loop
 \begin{picture}(\l@belwidth,\l@belheight)(0,\cnt@pictvoffset)
```

```
  \l@beloutline
  \multiput(0,\cnt@auxlinedistance)(0,\cnt@auxlinedistance){%
   \value{numberauxlines}}{\line(1,0){\l@belwidth}}
 \end{picture}
 \hspace{\hspaceinterlabel}
 \advance\tmp@count by -1
 \ifnum\tmp@count>0\repeat
}
```

`\labeltext` Here the second macro for creating labels: `\labeltext`. Perhaps it would have been nicer to declare this as an environment but this should do. The text-width and -height is computed here.

Note that the text is plain text – there is no predefined font change etc. The only special is the switch to the text color `\c@lorl@bel@fg` (if option `color` is chosen).

```
\def\labeltext#1{%
 \rem@kecompanyl@bel
 \l@beltextwidth=\l@belwidth mm
 \advance\l@beltextwidth by -2\labeltextmargin
 \C@nvertToCount{\label@textheight}{-\labeltextmargin}
 \advance\label@textheight by \yuppercompany
 \begin{picture}(\l@belwidth,\l@belheight)(0,\cnt@pictvoffset)
  \l@beloutline
  \put(0,0){\protect\makebox(\l@belwidth,\label@textheight)[t]{%
     \parbox{\l@beltextwidth}{%
        \ifcolorl@bel\textcolor{\c@lorl@bel@fg}{#1}\else #1\fi}}}
 \end{picture}
 \hspace{\hspaceinterlabel}%
}
```

⟨/package⟩

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.