

ticket.sty v0.4c September 6, 2016

Making labels, visiting cards, pins and flash-cards with  
 $\text{\LaTeX}$

A short introduction

Who? Thomas Emmel [thomas@family-emmel.de](mailto:thomas@family-emmel.de)

When? September 7, 2016

## A short introduction

`ticket.sty`

provides a simple interface to produce visiting (or calling) cards, labels for your files, stickers, pins and other stuff for your office, conferences etc.

All you need is a definition of your 'ticket' included in a ticket definition file (`.tdf`),  
and the two commands `\ticketdefault` and `\ticket`.  
See these examples to get an idea what you can get:

`ex_file`

back labels for your files...

`ex_pin`

pins for a conference...

`ex_vcard`

visiting-cards (also business cards, name cards...)

`ex_marks`

different markers used...

`ex_flashcard`

flash-cards, try it...

`ex_flashcard_dup`

in duplex mode

`ex_flashcard_rm`

in rows

## How to use it

Just add this line to the header of your latex-file:

```
\usepackage[tdf-file, other options]{ticket}
```

Use the letter or article documentclass and set up the paper-size that you need.

## The ticket definition file

A sample ticket definition (1z1680.tdf) for a 'LEITZ'-product is:

```
%%  
%% ticket for "Leitz 1680"  
%% pre-cutted label for wide files  
%%  
\unitlength=1mm  
%% tested for an HP5SiMX, adjust it for your printer  
\hoffset=-16.2mm  
\voffset=-6.6mm  
\ticketNumbers{1}{4}  
\ticketSize{189}{56.5}      % in unitlength  
\ticketDistance{0}{9.8}    % in unitlength
```

The meaning of `\unitlength` is obvious, `\hoffset` and `\voffset` are used to adjust the printed output. `\ticketNumbers{ $n_x$ }{ $n_y$ }` are the number of the tickets on the sheet in horizontal and vertical direction. `\ticketSize{width}{height}` is the size of one ticket and `\ticketDistance{distx}{disty}` are spaces between the tickets.

## Missing the ticket definition file

If you like to test a new ticket definition or you like to create only a single output without reusing those definitions anymore, `ticket.sty` assumes default values which can be redefined in your file.

Simply call `ticket.sty` without any ticket definition:

```
\usepackage[other options]{ticket}
```

and add your definitions directly to the head of the file after loading `ticket.sty`.

## How to fill your ticket

`ticket.sty` enables a picture-environment for all tickets. You can put things into this picture with `\ticketdefault` and `\ticket`. Both commands offer one argument.

`\ticketdefault`

is used to put a default and unique background to your ticket, that will be – of course – not changed for all tickets unless you change it with:

```
\renewcommand{\ticketdefault}{%  
  \put(100, 5){\includegraphics[width=35mm]{logo}}%  
  \put(100,50){\begin{rotate}{180}\Huge ... \end{rotate}}%  
}
```

`\ticket`

is used to provide the content which will be changed for the individual tickets. The simplest way to do that is to define a new command like this:

```
\newcommand{\myticket}[3]{\ticket{%  
  \put( 0, 5){#1}%  
  \put( 20,20){\bfseries\large#2\\#3}%  
}}
```

## Additional Options

Most of the options other than the tdf-file are used to create marks or decorations around the ticket:

crossmark  

Add a cross at all four corners of the ticket

circlemark  

Add a circle at all four corners

emptycrossmark  

A crossmark where the part of the cross inside the ticket is not visible

cutmark  

Just add cutmarks at the outer region of the page

boxed 

Add a frame around each ticket

rowmode

This option creates the tickets in rows instead of columns, which might be useful in some cases.

## Special commands

### reset tickets

You can add additional page-breaks with `\newpage\ticketreset`. This clears the ticket-buffers and sets the corresponding values to initial values. This can be used if you create a handful of tickets multiple times.

### `\backside`

is useful for flash-cards, see the use-cases below.

### skip tickets

Once you printed a series of tickets you mostly end up with some empty stickers on your sheet. In this case use `\ticketskip{5}` to skip 5 tickets (New feature)



## Organising and reading input

In general it is a good idea to create a file that contains all definitions and the whole latex-setup and a second file that contains only the data that you like to print which can be referenced by `\input`.

For example, if you reference a huge database of names of people which are used for the pins of a conference, you can easily write a script (perl, python...) that extracts the data to a file that is itself latex-formated with a `\myticket{ }{ }{ }` on each line.

Sometimes – of course – you only have to print only a small amount of data and all data can easily be written to a single file.

Sometimes a small generator-function can help to create numbers on tickets:

```
\newcounter{numcards}  
...  
\begin{document}  
...  
\whiledo{\thenumcards<25}{\stepcounter{numcards}\acard{\thenumcards}}%
```

## Use-cases

Some examples were referenced at the beginning of this manual. I will finish this short introduction with an overview of special examples and their implementations...

## Files and pins

This is a simple example for your files.

The same applies to pins. Start with a `\ticketdefault`, with this background:

```
\renewcommand{\ticketdefault}{%  
  \put(155,5){\begin{sideways}%  
    \includegraphics[width=35mm]{logo}\end{sideways}}%  
  \put(150,50){\begin{rotate}{180}\Huge C O M P U T E R\end{rotate}}%  
}
```

Now, create the content of the `\ticket` which differs:

```
\newcommand{\filelabel}[3]{\ticket{%  
  \put(33,48){\begin{sideways}\bfseries\Large#1\end{sideways}}%  
  \put(43,5){\begin{sideways}%  
    \begin{minipage}[t]{50mm}%  
      {\bfseries\large#2}\[2ex]\large#3%  
    \end{minipage}%  
  \end{sideways}}%  
}}
```

And fill the tickets:

```
\begin{document}  
...  
\filelabel{1}{\LaTeX{} - styles}{Standard styles\Own styles\%  
  \hspace*{5mm}\bfseries ...\\hspace*{5mm}\bfseries ticket}%  
...
```

## Visiting-cards

In this example, the `\ticketdefault` is empty, since we need a front and and background that differs:

```
\renewcommand{\ticketdefault}{}%

\newcommand{\vcardfront}[5]{\ticket{%
  \put( 63, 5){\includegraphics[width=40mm,angle=90,origin=bl]{logo}}
  \put( 63.2, 2){\color{uni}your logo}
  \put( 5, 13){\color{uni}\line(1,0){56}}
  \put( 7, 10){\small your company}
  \put( 7,6.5){\small street 1}\put( 7, 3){\small xxxxx city and state}
  \put( 7, 24){\small phone:}\put( 7, 20){\small fax:}
  \put(6,40){\bfseries\LARGE #1}
  \put(7,36){#2}\put(15,24){\small #3}
  \put(15,20){\small #4}\put( 7,16){\small #5}
}}

\newcommand{\vcardback}[1]{\ticket{%
  \put( 7, 26){\small private:}
  \put( 5, 23){\color{uni}\line(1,0){56}}
  \put( 7,19){\small%
    \begin{minipage}[t]{50mm}{#1}\hss\end{minipage}}%
}}
```

And fill the tickets:

```
% a counter, which makes life easier...
\newcounter{numcards}

\begin{document}
...
% ten cards for the front
\whiledo{\thenumcards<10}{%
  \stepcounter{numcards}%
  \vcardfront{Mr. X}{Degree/Profession}
               {+YY/XXXX/ZZZ ZZ ZZ}{+YY/XXXX/ZZZ ZZ ZZ}
               {yourmail@your-provider.xxx}}
\setcounter{numcards}{0}
% and ten for the backside
\whiledo{\thenumcards<10}{%
  \stepcounter{numcards}%
  \vcardback{phone: ---/HH HH HHH\\fax: ---/HH HH HHH\\%
             homestreet 7\\everywhere}}
...
```

## Flashcards

In this case we work with two different `\ticket`-definitions since we have a file with two different commands. However, this can be changed according to your taste:

```
% now what do you like to put in your ticket
\newcommand{\card}[2]{\ticket{%
    \put(32.5,15){\makebox[0mm]{\centering{\huge{\textbf{#1}}}}}
}}
\newcommand{\raiz}[3]{\ticket{%
    \put(2,31){{$\sqrt{\enspace}$}}
    \put(32.5,15){\makebox[0mm]{ {\huge{\textit{{ #1 }}}}}}
}}
```

All data is in a file:

```
...
\begin{document}
\input{words}
```

Now change the command for the backside:

```
\renewcommand{\card}[2]{\ticket{%  
  \put(3,30){\parbox{58mm}{\large{\textbf{ #1:}}}}%  
  \put(4,23){\parbox[t]{58mm}{\small {\blt #2}}}%  
}}  
\renewcommand{\raiz}[3]{\ticket{%  
  \put(3,30){\parbox{58mm}{\textbf{#1:} #2}}%  
  \put(3,13){\parbox{58mm}{#3}}%  
}}
```

Use the special command `\backside`, do some further adjustment for your printer and reread the file:

```
\backside%  
\oddsidemargin=8pt  
\input{words}  
...
```

If you want to print this on a duplex-printer rearrange the pages with this command for example:

```
psselect -p1,3,2,4 ex_flashcard.ps ex_flashcard_dup.ps ...
```



## Acknowledgements...

Many thanks

to those who used ticket.sty for many years and helped to find some small bugs and those who provided ticket-definition-files (tdf) and suggestions for new features.

Many special thanks to Santiago Mejia who created an initial version of the flash-card use-case.