bankstatement.cls

0.9.2

A LaTeX class for bank statements based on csv data

2015/11/14

Package author:
Josef Kleber

bankstatement@jklatex.de
http://bankstatement.jklatex.de/

Abstract

More and more banks allow their customers to download posting records in various formats. By using the `bankstatement` class, you can create bank statements – as long as a csv format is available! At the moment, the `csv-mt940` and `csv-camt` formats – used by many german Sparkassen – are supported. Moreover, it supports `csv-standard-bank-na`!

Furthermore, the following languages are supported: english, german, namibian

You can quite easily add support for other languages or csv formats. Simply define the order of the keys in the csv data file and how to use them.

The terminology in this class like BIC (Business Identifier Codes) or IBAN (International Bank Account Number) is based on SEPA (Single Euro Payments Area). But you can adjust the terminology to suit your needs.

1 Class options

format (`csv-mt940`) specifies the csv format of your data files

language (`english`) specifies the language of your document

left (`2cm`) specifies the left margin

right (`2cm`) specifies the right margin

top (`2cm`) specifies the top margin

bottom (`2cm`) specifies the bottom margin

2 The `bankstatement` command

`\bankstatement` `[`⟨*options*⟩`]{{`⟨*csv file*⟩`}}`

The `\bankstatement` command reads in `{`⟨*csv file*⟩`}` and outputs your bank statement. It supports the following options, which can also be used as class options with global scope:

title () specifies the title of your bank statement

logo () specifies the logo used in the bank statement. It may be the logo of your bank or the sports club, whose treasurer you are.

logowidth (`4cm`) defines the width of the logo

holder () specifies the account holder

bic () specisfies the BIC (Business Identifier Codes)

iban () specifies the IBAN (International Bank Account Number)

referencewidth (`11cm`) specifies the width of the second table column used for the reference. Depending on page size, margins and font you may need to adjust the width.

negativecolor (`red`) specifies the color used for negative amounts

sort (`PostingDate`) defines the sort key of the csv data base

openingbalance (`none`) specifies the opening balance. csv posting records do not contain an opening balance, only postings.

closingbalance (`none`) specifies the closing balance

Make sure to use the same encoding in your document than in the csv file(s)!

3 Customization

You can quite easily add support for other languages [2] and formats [1]. Please send a copy to the maintainer of `bankstatement`.

3.1 Languages

Supported languages:

- **`english`**
- `german`
- `namibian`

To support other languages, simply copy `stmenglish.def` to your local TeX tree, rename it and translate the definitions to your language! [2]

3.2 Formats

Supported formats:

- **`csv-mt940`**
- `csv-camt`
- `csv-standard-bank-na`

To support other formats [1], simply copy `csv-mt940.def` to your local TeX tree, rename it and adjust the definitions to your needs! That is, define the order of keys in your format and specify how to use them. Furthermore, you should specify the separator of the csv file and whether the csv file has a header row or not.

4 Example

```
\documentclass[a4paper,10pt,bic=BYLADEM1ERD,
               iban=DE70753574230240408336,
               holder={Josef Kleber},
               language=german,logo=stmlogo,
               format=csv-camt,
               logowidth=4cm,negativecolor=red]
              {bankstatement}
\usepackage[german]{babel}
```

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{bera}
\renewcommand\familydefault{\sfdefault}
\begin{document}
\bankstatement[title={Kontoauszug 12/2014},
               openingbalance={-12,34},
               closingbalance={82,13}]{201412.csv}
\end{document}
```

5 Implementation

```
⟨*class⟩
```

First, we provide the LaTeX class `bankstatement`.

```
\NeedsTeXFormat{LaTeX2e}%
\ProvidesClass{bankstatement}[2015/11/14 class for csv based bank statements v0.9.2]%
```

We need the `xkeyval` package and the `xkvlxp` package to allow curly braces and a bit more in global class options!

```
\RequirePackage{xkeyval}%
\RequirePackage{xkvltxp}%
```

We provide a macro `\STM@JK@define@key`, which defines class options with global scope and options for `\bankstatement` with local scope. It takes four arguments {⟨*prefix*⟩}, {⟨*package*⟩},{⟨*option*⟩} and {⟨*default*⟩}.

```
\newcommand*\STM@JK@define@key[4]%
{%
  \expandafter\gdef\csname#1@#3\endcsname{#4}%
  \define@key{#2.cls}{#3}[#4]%
  {%
    \expandafter\gdef\csname#1@#3\endcsname{##1}%
  }%
  \define@key{#2}{#3}%
  {%
    \expandafter\def\csname#1@#3\endcsname{##1}%
  }%
}%
```

Now, we can use this macro to define our options.

```
\STM@JK@define@key{STM@JK}{bankstatement}{format}{csv-mt940}%
\STM@JK@define@key{STM@JK}{bankstatement}{language}{english}%
\STM@JK@define@key{STM@JK}{bankstatement}{title}{}%
\STM@JK@define@key{STM@JK}{bankstatement}{logo}{}%
\STM@JK@define@key{STM@JK}{bankstatement}{logowidth}{4cm}%
```

```
\STM@JK@define@key{STM@JK}{bankstatement}{holder}{}%
\STM@JK@define@key{STM@JK}{bankstatement}{bic}{}%
\STM@JK@define@key{STM@JK}{bankstatement}{iban}{}%
\STM@JK@define@key{STM@JK}{bankstatement}{referencewidth}{11cm}%
\STM@JK@define@key{STM@JK}{bankstatement}{negativecolor}{red}%
\STM@JK@define@key{STM@JK}{bankstatement}{sort}{PostingDate}%
\STM@JK@define@key{STM@JK}{bankstatement}{openingbalance}{none}%
\STM@JK@define@key{STM@JK}{bankstatement}{closingbalance}{none}%
\STM@JK@define@key{STM@JK}{bankstatement}{left}{2cm}%
\STM@JK@define@key{STM@JK}{bankstatement}{right}{2cm}%
\STM@JK@define@key{STM@JK}{bankstatement}{top}{2cm}%
\STM@JK@define@key{STM@JK}{bankstatement}{bottom}{2cm}%
```

We execute the class options to define and set the option macros.

```
\DeclareOptionX*{\PassOptionsToClass{\CurrentOption}{article}}%
\ExecuteOptionsX{format,language,title,logo,logowidth,holder,bic,iban,%
                 referencewidth,negativecolor,sort,openingbalance,%
                 closingbalance,left,right,top,bottom}%
\ProcessOptionsX*\relax%
\LoadClass{article}%
```

We load some more needed packages.

```
\RequirePackage[left=\STM@JK@left,right=\STM@JK@right,top=\STM@JK@top,%
                bottom=\STM@JK@bottom]{geometry}%
\RequirePackage{longtable}%
\RequirePackage{tabularx}%
\RequirePackage{xcolor}%
\RequirePackage{graphicx}%
\RequirePackage{booktabs}%
\RequirePackage{datatool}%
\RequirePackage{calc}%
\RequirePackage{ifthen}%
\RequirePackage{siunitx}%
```

We define some macros, which will be redefined in language and format definition files!

```
\newcommand*\STM@JK@dbkeys{}%
\newcommand*\STM@JK@DTLforeach{}%
\newcommand*\STM@JK@holdername{}%
\newcommand*\STM@JK@bicname{}%
\newcommand*\STM@JK@ibanname{}%
\newcommand*\STM@JK@firstcolumnheading{}%
\newcommand*\STM@JK@secondcolumnheading{}%
\newcommand*\STM@JK@thirdcolumnheading{}%
\newcommand*\STM@JK@openingbalancename{}%
\newcommand*\STM@JK@closingbalancename{}%
```

We set some defaults and create a counter for unique data base names.

```
\newcommand*\STM@JK@headingsep{0.5cm}%
\newcommand*\STM@JK@noheader{false}%
```

```
%
\newcounter{STM@JK@count}%
```

We load the language and format definition files specified as class options.

```
\input{\STM@JK@format.def}%
\input{stm\STM@JK@language.def}%
```

This macro typesets a given logo at the right border.

```
\newcommand*\STM@JK@includelogo%
{%
  \ifthenelse{\equal{\STM@JK@logo}{}}%
  {}%
  {\hfill\includegraphics[width=\STM@JK@logowidth]{\STM@JK@logo}}%
}%
```

This macro typesets the header of the bank statement.

```
\newcommand*\STM@JK@header%
{%
  \noindent%
  \begin{tabularx}{\textwidth}{XXr}%
    \multicolumn{3}{l}{\Huge{}\hspace{.22em}\STM@JK@title\STM@JK@includelogo}\\%
    &  &\\%
    \begin{tabular}{l}\textbf{\STM@JK@holdername}\\\STM@JK@holder\end{tabular} &%
    \begin{tabular}{l}\textbf{\STM@JK@bicname}\\\STM@JK@bic\end{tabular} &%
    \begin{tabular}{l}\textbf{\STM@JK@ibanname}\\\STM@JK@iban\end{tabular}\\%
  \end{tabularx}%
  \vspace{\STM@JK@headingsep}%
}%
```

`\bankstatement` Here, we define the user command to typeset the bank statement.

`\bankstatement[`⟨*options*⟩`]{`⟨*csv file*⟩`}`

```
\newcommand\bankstatement[2][]%
{%
```

We start a group to keep the setting of options local. Then we step our unique counter and define a macro for the current data base name for multi command usage!

```
  \begingroup%
    \setkeys{bankstatement}{#1}%
    \stepcounter{STM@JK@count}%
    \xdef\STM@JK@dbname{stm\arabic{STM@JK@count}}%
```

Then we can load `{`⟨*csv file*⟩`}` into our data base depending on the `noheader` option! Finally, we sort our data base depending on the data base key specified with the `sort` option!

```
    \ifthenelse{\equal{\STM@JK@noheader}{false}}%
    {\DTLloadrawdb[keys={\STM@JK@dbkeys},noheader=false]{\STM@JK@dbname}{#2}}%
    {\DTLloadrawdb[keys={\STM@JK@dbkeys},noheader=true]{\STM@JK@dbname}{#2}}%
    \ifthenelse{\equal{\STM@JK@sort}{}}%
    {}%
    {\DTLsort{\STM@JK@sort}{\STM@JK@dbname}}%
```

Now we can typeset the header of the bank statement and start the `longtable`. Maybe, we still need to typeset an opening balance.

```
    \STM@JK@header%
    \begin{longtable}{llr}%
    \toprule%
    \STM@JK@firstcolumnheading &%
    \STM@JK@secondcolumnheading &%
    \STM@JK@thirdcolumnheading%
    \\\toprule%
    \endhead%
    \ifthenelse{\equal{\STM@JK@openingbalance}{none}}%
    {}%
    { & & \\ & \STM@JK@openingbalancename &%
      \DTLifStartsWith{\STM@JK@openingbalance}{-}%
      {\textcolor{\STM@JK@negativecolor}{\num{\STM@JK@openingbalance}}}%
      {\num{\STM@JK@openingbalance}} \\\midrule}%
```

Now, we can loop through our database and create a new row for each line in {⟨*csv file*⟩}. Finally, we can end the `longtable`.

```
    \STM@JK@DTLforeach%
    \end{longtable}%
  \endgroup%
}%
```

Finally, we disable the global class options `\AtBeginDocument`.

```
\AtBeginDocument{\disable@keys{bankstatement}%
                              {format,language,left,right,top,bottom}}%
```

```
⟨/class⟩
```

6 References

[1] Josef Kleber. HowTo support arbitrary CSV formats, 2015. http://bankstatement.jklatex.de/en/2015/11/howto-support-arbitrary-csv-formats-2/.

[2] Josef Kleber. HowTo support other languages, 2015. http://bankstatement.jklatex.de/en/2015/11/howto-support-other-languages/.

[3] Nicola L.C. Talbot. User Manual for datatool bundle version 2.22, 2014. http://mirrors.ctan.org/macros/latex/contrib/datatool/datatool-user.pdf.

[4] wikipedia.org. International Bank Account Number, 2014. https://en.wikipedia.org/wiki/International_Bank_Account_Number.

[5] wikipedia.org. ISO 9362, 2014. https://en.wikipedia.org/wiki/ISO_9362.

[6] wikipedia.org. Single Euro Payments Area, 2014. https://en.wikipedia.org/wiki/Single_Euro_Payments_Area.

[7] Joseph Wright. siunitx – A comprehensive (SI) units package, 2014. http://mirrors.ctan.org/macros/latex/contrib/siunitx/siunitx.pdf.

7 Change History

- v0.9.1
  - General: CTAN upload 6
- v0.9.2
  - General: added csv-standard-bank-na.def 6
    - added siunitx package to force output of exactly two digital places 7
    - added stmnamibian.def 6
    - updated stmenglish.def 6
  - `\bankstatement`: catch empty sort key → no sorting 8
    - changed `\DTLloaddb`→ `\DTLloadrawdb` 8