

qrbill v1.02

L^AT_EX-package to create QR-bills based on the Swiss payments standards

Marei Peischl <marei@peitex.de>

2020/08/25

A collaborative project of peiT_EX and foobar LLC

1. Introduction

qrbill.sty, the L^AT_EX-package, is the Free Software and Open Source answer to the launch of the swiss payments conversion, introduced mid 2020, replacing the former payment order slip by a restructured, QR enhanced slip. The published source code is based on the payment standard guidelines issued for Switzerland and intentionally has a modular structure to enable further development and adaptation for international use.

By means of the L^AT_EX framework, the qrbill can simply be integrated into existing accounting environments and automated billing systems.

2. Example of use

The easiest way to use the qrbill package is to create a bill with the standalone documentclass:

```
\documentclass{standalone}
\usepackage[ngerman]{babel}

\usepackage{xcolor}
\usepackage{qrbill}

\begin{document}

\QRbill[
creditor={foobar LLC\\
Postfach 404\\
2342 Zurich\\
```

```

CH},
Account=CH1280808005649899718,
% insert additional data here
]

```

```

\end{document}

```

`\QRbill` [`\Data Setup`] The macro `\QRbill` creates a box of 210 mm × 105 mm size. It will include a qrbill as described in the design guide [2] and insert all available data.

There is also an example in the appendix showing all data fields (see appendix A on page 10).

The headings are provided in German, English, Italian and French and will be translated using the document’s language setup. The terms have been taken from the official guidelines.

3. Technical Requirements

qrbill has been built for T_EXLive 2020. Currently it is not possible to ensure it’s functionality with older Releases of T_EX Distributions.

The packages loaded by qrbill are:

- expl3
- fontspec (except one is using a custom font setup)
- graphicx
- scrbase (which is part of the koma-script bundle)
- qrcode
- iftex
- l3keys2e
- numprint

As a default font “Liberation Sans” will be used. This will be loaded using fontspec, which would require the use of Lua_TE_X or Xe_LA_TE_X as a compiler. To have a workaround for this, see the `font` option on page 3 for a custom font setup.

4. Package options

billinginfo (true/false) (default: **true**)
Activate/deactivate the automatic fill of the BillingInformation. If this is set to **false** one still can add data to this field using the **BillingInformation** key, as Described in section 5.

creditorprefix (String) (default: CR-/UD-)
debtorprefix For custom setups the predefined prefixes for the debtor and creditoraddress can be changes. The initial configuration is a requirement of the Swiss standards.

font (Frutiger/Arial/Helvetica/Liberation Sans/custom) (default: **Liberation Sans**)
The official guideline for the Swiss qrbill limits the choice of the font to the first four values of this option. The font selection and sizes currently hardcoded to fit to the size. Custom setups might be provided in the future.

In case one wants to use a custom font, you can set the option **font=custom**. This will disable all font setup and not prevent the fontspec package from being loaded. You can redefine the macro **qrbillfont** to use any font available on your system, for example:

```
\usepackage{fontspec}
% The Laconic font has to be installed
\renewcommand*{\qrbillfont}{\fontspec{Laconic}}
```

frame (true/false/top/bottom) (default: **true**)
Switch to disable the frame around the created QRbill. The top/bottom options should be preferred when the bill is embedded into a document of a4 papersize.

ibanseparator (tokenlist) (default: \,)
Set the tokenlist to separate the account numbers. See option **sep-iban** for further information.

icon (swiss-cross/filename) (default: <empty>)
The swiss standard describes the placement of a swiss-cross icon in the center of the QRcode. This can be achieved using this option. The default is set empty, so no icon will be placed. If this option holds another string than “swiss-cross” it will be interpreted as filename and try to load a custom image.

iconwidth (length) (default: 7mm)
This option allows custom scaling for custom icons.

qrscheme (Name of a custom QRbill scheme) (default: **swiss**)
Loads the definitions for the QRcode and the BillingInformation. Currently only the

swiss qrbill scheme is available, but one can define own variants based on this file. Users can copy the file `swiss.qrbill-cfg.tex` as an example.

<code>referenceseparator</code>	(tokenlist)	(default: <code>\,</code>)
	Set the tokenlist to separate the reference codes. See option <code>sep-reference</code> for further information.	
<code>sep-iban</code>	(integer)	(default: 0)
<code>sep-reference</code>	Set the size of character groups to separate these. Positive values will be counted left to right. Negative ones the other way round. The most variants of the Swiss QRbill use values of <code>sep-iban=4</code> and <code>sep-reference=-5</code> e.g.:	
	Iban: CH12 8080 8005 6498 9971 8	
	Reference: 00 00000 00000 00000 00000 00152	
<code>separate</code>	(false/text/symbol)	(default: <code>text</code>)
	Setup the separation rule between the document and the qrbill. If the value is <code>text</code> or <code>symbol</code> the Info “Separate before paying in” or the corresponding translation will be printed on top of the frame. This option will be ignored if the frame has no top rule.	

5. Data setup

<code>\qrbillsetdata</code>	The data fields can be filled either using the optional argument of <code>\QRbill</code> or using the macro <code>\qrbillsetdata</code> . Section 6.7 shows a list of all available fields and appendix A provides an example of use. For further information on the construction of the QRcode see the specification [1].
-----------------------------	--

To simplify the usage qrbill provides an interface to use the different types of address data and is able to create a “billing information” string as described in [1] using the single data elements.

6. Setup for the preconfigured Swiss QR-bill

6.1. Preconfigured data

The initial configuration which uses the Swiss standard already adds the following data:

```
\SetupQrBill{
  QRType=SPC,
  Version=0210,
  CodingType=1,
```

```
Trailer=EPD,
}
```

These fields are required to use these values for Version 2.1 of the QRbill standard. If any future changes on the standard would require changes, they can be overwritten or redefined by using a custom implementation file, similar to `swiss.qrbill-cfg.tex`.

Be aware that we use the versioncode 0210, as this is the current, implemented standard from the official guidelines, which replaced version 2.0 in 09/2019. As the mentioned example in v2.1 of the official guidelines use an outdated version code (e.g. 0200), some banking apps may refuse the processing of the QR code. Hence, this is not a bug of the qrbill package, but of these apps. Feel free to change the version back to 0200 to override and support the replaced standard. The corresponding issue and discussion can be found at <https://github.com/peiTeX/qrbill/issues/3>.

Additionally the Currency has been initialized to use “CHF”. This also can be overwritten. Currently the Swiss standards only allow the values `CHF` or `EUR`.

6.2. Address data

The Swiss payment standards for QR-bills require the following data for the address of creditor (Prefix `CR-`) and debtor (Prefix `UD-`):

AddressType	K	S
Name	Name	
Address1	Street	Street with number
Address2	Number	Postal code with City
ZIP	Postal code	Enforced Empty
City	City	Enforced Empty
Country	Country Code	

To set an address of type “K” one can use the option key `debtor` or `creditor`. Type “S” can be achieved using the starred variant (`debtor*`/`creditor*`).

```
creditor={Name\\
Street Nr.\\
PostalCode City\\
CountryCode},
creditor*={Name\\
Street\\
Nr.\\
PostalCode\\
City\\
CountryCode},
```

6.3. Billing information

The billing information is a string which can consist of the following data elements:

<code>invoicenum</code>	invoice number	
<code>invoicedate</code>	invoice data	
<code>customerref</code>	customer reference	
<code>vat</code>	VAT number	
<code>vatdate</code>	VAT date	The data has to be in the required format.
<code>vatdetails</code>	VAT percentage / details	
<code>importvat</code>	import VAT	
<code>conditions</code>	payment conditions	

`qrbill` might add features to automatically validate this in the future.

Beside setting the data fields all on their own, `qrbill` can also handle a complete billing information string. One could simply set it by

```
BillingInfo=//S1/10/10201409/11/190512/20/1400.000-53/30/106017086  
/31/180508/32/7.7/40/2:10;0:30
```

6.4. AV-Parameters

The QRcode can use 2 AV Parameters. They have to carry the Prefix of “Name AV1:”/“Name AV2:”. This will be automatically added if the keys `AV1`/`AV2` are used instead of `AV1-Parameters`/`AV2-Parameters`.

6.5. Remaining data elements

The remaining data elements should be set directly. For restrictions on the content see the specification. Currently there is no validation taking place. The remaining for the preconfigures Swiss standard are: Account, Amount, ReferenceType, Reference, Message,

6.6. The Swiss cross icon

[2] requires a swiss cross of 7 mm size to be placed in the center of the QRcode. This package supports this by using the package option for the icon:

```
\usepackage[icon=swiss-cross, ...]{qrbill}
```

This setting is not activated by default.

6.7. Overview over all data fields

Data fields which are used in the QRcode:

QRType
Version
CodingType
Account
CR-AddressType
CR-Name
CR-Address1
CR-Address2
CR-PostalCode
CR-City
CR-Country
UCR-AddressType
UCR-Name
UCR-Address1
UCR-Address2
UCR-ZIP
UCR-City
UCR-Country
Amount
Currency
UD-AddressType
UD-Name
UD-Address1
UD-Address2
UD-PostalCode
UD-City
UD-Country
ReferenceType
Reference
Message
Trailer
BillingInfo
AV1-Parameters
AV2-Parameters

Data fields, which are custom created by the qrbill package to simplify the use:

creditor and creditor*
debtor and debtor*
AV1 and AV2

Data fields to automatically create the BillingInfo string. The key in parentheses indicates the prefix to be used to construct the string.

```
//S1 (prefix)
invoicenum (10)
invoicedate (11)
customerref (20)
vat (30)
vatdate (31)
vatdetails (32)
importvat (33)
conditions (40)
```

```
\qrbillsetup{
    creditorprefix =CR-,
    debtorprefix =UD-,
}
```

```
\SetQrScheme{
    QRType,
    Version,
    CodingType,
    Account,
    CR-AddressType,
    CR-Name,
    CR-Address1,
    CR-Address2,
    CR-PostalCode,
    CR-City,
    CR-Country,
    UCR-AddressType,
    UCR-Name,
    UCR-Address1,
    UCR-Address2,
    UCR-ZIP,
    UCR-City,
    UCR-Country,
    Amount,
    Currency,
    UD-AddressType,
    UD-Name,
    UD-Address1,
    UD-Address2,
    UD-PostalCode,
    UD-City,
    UD-Country,
    ReferenceType,
```



```

Reference,
Message,
Trailer,
BillingInfo,
AV1-Parameters,
AV2-Parameters
}

\SetBillingInfoScheme{
  prefix=//S1,
  10=invoiceenum,
  11=invoicedate,
  20=customerref,
  30=vat,
  31=vatdate,
  32=vatdetails,
  33=importvat,
  40=conditions
}

\SetupQrBill{
  QRType=SPC,
  Version=0210,
  CodingType=1,
  Trailer=EPD,
  Currency=CHF,
  ReferenceType=NON,
}

```

Change History

v1.00		add referenceseparator option	4
General: First official version	1	add sep-iban/sep-reference option	4
v1.01		Fix placement issue with separate	
General: Add separate option to modify		option	4
the top rule of the QR bill	4	Note the version number	
v1.02		inconsistency in the official	
General: add ibanseparator option	3	guidelines	5

References

- [1] Swiss Implementation Guidelines QR-bill: Technical and professional specifications of the payment part with Swiss QR Code and of the receipt. Version 2.1, with effect from 30 September 2019. <https://www.paymentstandards.ch/dam/downloads/ig-qr-bill-en.pdf>

- [2] Style Guide QR-bill Style Guide QR-bill: The right layout pays off. Layout rules and recommendations for the payment part with Swiss QR Code and for the receipt.
<https://www.paymentstandards.ch/dam/downloads/style-guide-en.pdf>

A. Example of standalone qrbill

```
% !TeX program=lualatex
\documentclass{standalone}
\usepackage[nswissgerman]{babel}

\RequirePackage{xcolor}
\usepackage[icon=swiss-cross,separate=false]{qrbill}

\begin{document}

\QRbill[
  creditor*={foobar LLC\\
    Postfach\\
    404\\
    2342\\
    Zurich\\
    CH},
  Account=CH1280808005649899718,
  vat=123123123,% VAT number with stripped CH and periods
  debtor*={peiTeX\\
    TeXnikerweg\\
    78\\
    23420\\
    Hamburg\\
    DE},
  Amount=1337.42,
  Message=Bestellung vom 27.06.2020,
  invoicenum=100-4242,
  invoicedate=200701,%ymmdd
  vatdetails=0,% 0% VAT
  vatdate=200701,%ymmdd
  AV1=LX;FOOBAR;2342,
]

\end{document}
```

B. qrbill scrletter example

```
% !TeX Program=lualatex
\documentclass[foldmarks=b]{scrletter}
\usepackage[frame=top]{qrbill}
```

```

% Setup layer
\DeclareNewLayer[
align=bl,
voffset=\paperheight,
hoffset=0pt,
contents={\QRbill},
width=\paperwidth,
height=105mm,
]{qrbill}

\begin{document}

\qrbillsetdata{
  creditor*={foobar LLC\
    Postfach\
    404\
    2342\
    Zurich\
    CH},
  Account=CH1280808005649899718,
  vat=123123123,% VAT number with stripped CH and periods
  debtor*={peiTeX\
    TeXnikerweg\
    78\
    23420\
    Hamburg\
    DE},
  Amount=1337.42,
  Message=Bestellung vom 27.06.2020,
  invoicenum=100-4242,
  invoicedate=200701,%ymmdd
  vatdetails=0,% 0% VAT
  vatdate=200701,%ymmdd
  AV1=LX;FOOBAR;2342,
}

\setkomavar{fromaddress}{\insertcreditor}

\begin{letter}{\insertdebtor}

\opening{opening}

Text

\closing{closing}

% Add qbill on last page

```

```
% For more details
\AddLayersToPageStyle{@everystyle@}{qrbill}

\end{letter}
\end{document}
```