

The `invoice2` package

Intelligent invoices with L^AT_EX3

Simon Dierl <simon.dierl@cs.tu-dortmund.de>
<https://github.com/no-preserve-root/invoice2>

Version 1.0
Saturday July 15, 2017

While there exists a vast amount of templates, unpublished packages and code snippets designed for creating invoices, the only solution available in the CTAN is the `invoice`¹ package, which is not actively maintained and lacks e.g. internationalization support while requiring use of a project structure that does not fit many use cases.

Thus, this package aims to reimplement the functionality provided by `invoice` using L^AT_EX3, while using the print-friendly layout provided by the `booktabs`² package.

Contents

I	User documentation	2
1	Package Configuration	2
2	Localization	3
3	Typesetting invoices	4
4	Examples	4
5	Version History	5
5.1	Version 1.0	5
6	License	6
II	<code>invoice2</code> implementation	6
1	Setup and Initialization	6

¹<https://www.ctan.org/pkg/invoice>

²<https://www.ctan.org/pkg/booktabs>

2	Variables and Constants	6
2.1	VAT Computation	7
2.2	Price Typesetting	7
2.3	Colorization	7
2.4	Internal State Tracking	8
3	Configuration	9
3.1	Package Parameters	9
3.2	<code>invoiceoptions</code> Command	9
4	The <code>invoice</code> Environment	10
4.1	Header Rendering	10
4.2	Content Rendering	11
4.3	Footer Rendering	15
4.4	The Environment itself	16
5	Messages and Errors	17
6	Dictionary Files	17
6.1	English Dictionary	17
6.2	German Dictionary	17
	Change History	17
	Index	18

Part I

User documentation

1 Package Configuration

`invoice2` offers extensive options for customization. Most of these options are not likely to change in the middle of a document; these should be passed as package options. However, for maximum flexibility, the global options can be changed at run time or overridden per `invoice` environment.

The options supported by this package are:

vat Controls the global default for VAT. VAT can be overridden per invoice item and is either automatically added or subtracted (see the next parameter for details). This is given as a floating point number, e.g. .19 for 19 %. Default is 0 %.

included-vat Controls if prices are provided with VAT included or excluded. If VAT is included, all prices are gross prices, the VAT is subtracted to show the net price. If VAT is excluded, all prices are net prices, the VAT is added to show the gross price. Default is excluded VAT.

currency-symbol The currency symbol to use, e.g. €. Default is \$.

currency-fraction-digits The fractional digits to use for the currency. For almost all currencies, this is two (100 of the smaller unit make 1 of the larger). Default is 2.

colorize Enables colorization. While the typographical results may be mixed, colorization can help read very long invoices. Default is off.

odd-color The color to use for odd invoice rows, if colorization is enabled. The color must be understood by `xcolor`. Default is white.

even-color The color to use for even invoice rows, if colorization is enabled. The color must be understood by `xcolor`. Default is light grey (lightgray).

title-color The color to use for the title row, if colorization is enabled. The color must be understood by `xcolor`. Default is white.

total-color The color to use for total rows, if colorization is enabled. The color must be understood by `xcolor`. Default is white.

2 Localization

Internationalization for this package is provided by the `translations`³ package. `translations` uses a key-to-translation mapping that can be overridden by the user. See the package documentation for further information.

At the moment, localizations for english and german are provided. See the documentation for the `translations` package for adding new localizations. Please consider contributing your translations to this project.

The following keys are defined:

invoice2-thousands-sep The separator between thousands, e.g. the space in 1 000. Default is a small space.

invoice2-decimal-point The separator between whole and fractional parts, e.g. the dot in 40.00. Default is a dot.

invoice2-amount The “Amount” column title.

invoice2-item The “Item” column title.

invoice2-vat The “VAT” column title.

invoice2-unit-price The “Unit Price” column title.

invoice2-price The “Price” column title.

invoice2-net-total The “Net Total” row label.

invoice2-vat-total The “VAT” row label.

invoice2-gross-total The “Gross Total” row label.

`\invoiceoptions`

`\invoiceoptions {<key=value...>}`

All of the package options can be changed in the document by using this command. The syntax is identical to the package options.

³<https://www.ctan.org/pkg/translations>

3 Typesetting invoices

An invoice is typeset as a table having between two and five columns. Each row will correspond to an item in the invoice. A row *always* has an item name and a price. If *any* item in the invoice has a VAT that is not 0%, *every* row will list its VAT. Also, if *any* item has an amount that is not one, *every* row will list both price per unit and the amount.

If an item has non-zero VAT, the net and gross totals will also differ. In this case, the package will compute the net total, the total VAT and the gross total and denote them below the invoice. If all items have 0%-VAT, only a gross total will be given.

<code>\begin{invoice}</code>	<code>\begin [⟨key=value...⟩] {invoice}</code>
<code>\end{invoice}</code>	<code>\end {invoice}</code>

Invoices are enclosed in `invoice` environments. These environments can not be nested. Each environment can override the package settings by means of its optional arguments. The overridden arguments only apply to the current invoice.

Inside the environment, *only* `\invoiceitem` and `\invoicesingleitem` commands may be safely used.

<code>\invoiceitem</code>	<code>\invoiceitem [⟨VAT⟩] {⟨amount⟩} {⟨item name⟩} {⟨unit price⟩}</code>
<code>\invoicesingleitem</code>	<code>\invoicesingleitem [⟨VAT⟩] {⟨item name⟩} {⟨unit price⟩}</code>

These commands add an item to the current invoice. The VAT argument is optional, if not given, VAT will default to the value set in the configuration. For singular items, the second form can be used to imply an amount of 1.

VAT must be given as a floating point value (see the configuration parameter list for details). The amount must be an integer, the unit price a floating point number. Do *not* add a currency symbol or formatting to the unit price.

4 Examples

Let us begin with a simple invoice with single items and no VAT. The invoice will only have two columns and a gross total. We do, however, change the currency symbol to €. Since we only have single copies, we will use the `\invoicesingleitem` command.

```
\begin{invoice}[currency-symbol={\euro{}}]
  \invoicesingleitem{Ignition!}{4087.99}
  \invoicesingleitem{The Art of Computer Programming 1--4}{162.99}
  \invoicesingleitem{The TeXbook}{55.69}
\end{invoice}
```

Item	Price
Ignition!	4 087.99 €
The Art of Computer Programming 1–4	162.99 €
The TeXbook	55.69 €
Gross Total	4 306.67 €

A more complex example involves amounts and a VAT. We will apply a 9% VAT

to our books and stock up on more copies of the TeXbook. This will enable all columns and an extended total.

```
\invoiceoptions{vat=.09}
\begin{invoice}[currency-symbol={\euro{}}]
  \invoicingleitem{Ignition!}{4087.99}
  \invoicingleitem{The Art of Computer Programming 1--4}{162.99}
  \invoiceitem{20}{The TeXbook}{55.69}
\end{invoice}
```

Amount	Item	VAT	Unit Price	Price
1	Ignition!	9 %	4 087.99 €	4 455.91 €
1	The Art of Computer Programming 1–4	9 %	162.99 €	177.66 €
20	The TeXbook	9 %	55.69 €	1 214.04 €
Net Total				5 364.78 €
VAT				482.83 €
Gross Total				5 847.61 €

Finally, let us toy with colorization. We wisely decide to leave most parts of the invoice white and only apply an extremely light grey to the even rows.

```
\begin{invoice}[currency-symbol={\euro{}}], colorize,
  even-color={lightgray!50}
  \invoicingleitem{Ignition!}{4087.99}
  \invoicingleitem{The Art of Computer Programming 1--4}{162.99}
  \invoicingleitem{The TeXbook}{55.69}
\end{invoice}
```

Item	Price
Ignition!	4 087.99 €
The Art of Computer Programming 1–4	162.99 €
The TeXbook	55.69 €
Gross Total	4 306.67 €

5 Version History

5.1 Version 1.0

- First public release.
- Support configuration via package options, `\invoiceoptions` command and environment options.
- Localization via translations.

6 License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Part II

invoice2 implementation

```
1 <*package>
2 <@@=invoicetwo>
```

1 Setup and Initialization

```
3 \RequirePackage{booktabs}
4 \RequirePackage{expl3}
5 \RequirePackage{l3keys2e}
6 \RequirePackage{longtable}
7 \RequirePackage{siunitx}
8 \RequirePackage{translations}
9 \RequirePackage[table]{xcolor}
10 \RequirePackage{xparse}
11 \ProvidesExplPackage{invoice2}{2017/07/15}{1.0}
12   {Next-generation invoice typesetting with booktabs}
```

Now, load our dictionary files and set fallback translations. We emit the dictionaries in section 6.

```
13 \LoadDictionaryFor{english}{invoice2}
14 \LoadDictionaryFor{german}{invoice2}
15 \DeclareTranslationFallback{invoice2-thousands-sep}{\,}
16 \DeclareTranslationFallback{invoice2-decimal-point}{.}
17 \DeclareTranslationFallback{invoice2-amount}{Amount}
18 \DeclareTranslationFallback{invoice2-item}{Item}
19 \DeclareTranslationFallback{invoice2-vat}{VAT}
20 \DeclareTranslationFallback{invoice2-unit-price}{Unit-Price}
21 \DeclareTranslationFallback{invoice2-price}{Price}
22 \DeclareTranslationFallback{invoice2-net-total}{Net~Total}
23 \DeclareTranslationFallback{invoice2-vat-total}{VAT}
24 \DeclareTranslationFallback{invoice2-gross-total}{Gross~Total}
```

2 Variables and Constants

All variables that control invoice typesetting can be set on

- a) the package level, as a package option
- b) the package level, using the `invoiceoptions` command

- c) per `invoice` environment, via the same syntax
- d) per invoice line, for some options (e. g. VAT).

2.1 VAT Computation

`\l__invoicetwo_vat_fp` The default value for VAT. If an invoice has only zero VAT for all entries, no VAT is added and the column is not rendered. VAT can be set per line.

25 `\fp_new:N \l__invoicetwo_vat_fp`

(End definition for \l__invoicetwo_vat_fp.)

`\l__invoicetwo_included_vat_bool` Controls if VAT is already included into the given prices. If true, the total will compute the net total and display it. If false, the total will compute the gross total. If the VAT is zero, this does nothing. We will render a gross total only.

26 `\bool_new:N \l__invoicetwo_included_vat_bool`

(End definition for \l__invoicetwo_included_vat_bool.)

`\l__invoicetwo_currency_symbol_tl` The currency symbol to use.

2.2 Price Typesetting

27 `\tl_new:N \l__invoicetwo_currency_symbol_tl`

(End definition for \l__invoicetwo_currency_symbol_tl.)

`\l__invoicetwo_currency_fraction_digits_int` The number of fractional digits for the currency. Contrary to popular opinion, this does not always equal two.

28 `\int_new:N \l__invoicetwo_currency_fraction_digits_int`

(End definition for \l__invoicetwo_currency_fraction_digits_int.)

2.3 Colorization

`\l__invoicetwo_colorize_bool` Controls if the invoice should be colorized at all. We only support alternating colorization for even and odd lines and a special color for the total line.

29 `\bool_new:N \l__invoicetwo_colorize_bool`

(End definition for \l__invoicetwo_colorize_bool.)

`\l__invoicetwo_odd_color_tl` The color for odd lines. Only used if colorization is enabled.

30 `\tl_new:N \l__invoicetwo_odd_color_tl`

31 `\tl_set:Nn \l__invoicetwo_odd_color_tl {white}`

(End definition for \l__invoicetwo_odd_color_tl.)

`\l__invoicetwo_even_color_tl` The color for even lines. Only used if colorization is enabled.

32 `\tl_new:N \l__invoicetwo_even_color_tl`

33 `\tl_set:Nn \l__invoicetwo_even_color_tl {lightgray}`

(End definition for \l__invoicetwo_even_color_tl.)

`\l__invoicetwo_title_color_tl` The color for the title line. Only used if colorization is enabled.

```
34 \tl_new:N \l__invoicetwo_title_color_tl
35 \tl_set:Nn \l__invoicetwo_title_color_tl {white}
```

(End definition for \l__invoicetwo_title_color_tl.)

`\l__invoicetwo_total_color_tl` The color for the total line. Only used if colorization is enabled.

```
36 \tl_new:N \l__invoicetwo_total_color_tl
37 \tl_set:Nn \l__invoicetwo_total_color_tl {white}
```

(End definition for \l__invoicetwo_total_color_tl.)

2.4 Internal State Tracking

`\l__invoicetwo_in_invoice_bool` Tracks if we are in an invoice environment. This allows us to crash when nesting invoice environments by accident.

```
38 \bool_new:N \l__invoicetwo_in_invoice_bool
```

(End definition for \l__invoicetwo_in_invoice_bool.)

`\l__invoicetwo_row_number_int` Counts the invoice rows. We use this to tell even from odd lines. Starts with 1; this must be reset at the beginning of each invoice.

```
39 \int_new:N \g__invoicetwo_row_number_int
```

(End definition for \l__invoicetwo_row_number_int.)

`\l__invoicetwo_vat_nonzero_bool` Tracks if we have already encountered a line with non-zero VAT. If so, we will render a VAT column and separate gross and net totals.

```
40 \bool_new:N \l__invoicetwo_vat_nonzero_bool
```

(End definition for \l__invoicetwo_vat_nonzero_bool.)

`\l__invoicetwo_amount_nonone_bool` Tracks if we have already encountered a line with non-one amount. If so, we will render an amount and an unit price column.

```
41 \bool_new:N \l__invoicetwo_amount_nonone_bool
```

(End definition for \l__invoicetwo_amount_nonone_bool.)

`\l__invoicetwo_net_total_fp` The running net total.

```
42 \fp_new:N \l__invoicetwo_net_total_fp
```

(End definition for \l__invoicetwo_net_total_fp.)

`\l__invoicetwo_vat_total_fp` The running VAT total.

```
43 \fp_new:N \l__invoicetwo_vat_total_fp
```

(End definition for \l__invoicetwo_vat_total_fp.)

`\l__invoicetwo_gross_total_fp` The running gross total.

```
44 \fp_new:N \l__invoicetwo_gross_total_fp
```

(End definition for \l__invoicetwo_gross_total_fp.)

`\l__invoicetwo_tabular_tl` The tabular under construction. We create the tabular contents on the fly, using a conditional for the VAT column that is expanded only when emitting the invoice.

```
45 \tl_new:N \l__invoicetwo_tabular_tl
```

(End definition for \l__invoicetwo_tabular_tl.)

3 Configuration

We define a `l3keys` key list for the package. We will reuse this for all configuration, except for item-specific VAT, which will override the global VAT setting. We handle this in the corresponding function.

```

46 \keys_define:nn {invoice2} {
47   vat .fp_set:N = \l__invoicetwo_vat_fp,
48   vat .value_required:n = true,
49   vat .initial:n = 0,
50   included-vat .bool_set:N = \l__invoicetwo_included_vat_bool,
51   included-vat .initial:n = false,
52   currency-symbol .tl_set:N = \l__invoicetwo_currency_symbol_tl,
53   currency-symbol .value_required:n = true,
54   currency-symbol .initial:n = {\$},
55   currency-fraction-digits .int_set:N = \l__invoicetwo_currency_fraction_digits_int,
56   currency-fraction-digits .value_required:n = true,
57   currency-fraction-digits .initial:n = 2,
58   colorize .bool_set:N = \l__invoicetwo_colorize_bool,
59   colorize .initial:n = false,
60   odd-color .initial:n = white,
61   odd-color .value_required:n = true,
62   odd-color .tl_set:N = \l__invoicetwo_odd_color_tl,
63   even-color .initial:n = lightgray,
64   even-color .value_required:n = true,
65   even-color .tl_set:N = \l__invoicetwo_even_color_tl,
66   title-color .initial:n = white,
67   title-color .value_required:n = true,
68   title-color .tl_set:N = \l__invoicetwo_title_color_tl,
69   total-color .initial:n = white,
70   total-color .value_required:n = true,
71   total-color .tl_set:N = \l__invoicetwo_total_color_tl
72 }

```

3.1 Package Parameters

Fortunately, `l3keys2e` handles the heavy lifting for us.

```

73 \ProcessKeysOptions{invoice2}

```

3.2 `invoiceoptions` Command

`\invoiceoptions` We also provide an interface to change the configuration at run time.

```

74 \NewDocumentCommand{\invoiceoptions}{m}{
75   \keys_set:nn {invoice2} {#1}
76 }

```

(End definition for `\invoiceoptions`. This function is documented on page 3.)

4 The invoice Environment

4.1 Header Rendering

`__invoicetwo_print_column_specification:` Emits the `\beginlongtable`. The amount column is centered or skipped, if all amounts are one. The item column is left-justified. The VAT column is centered or skipped, if all VATs are zero. The unit price and price columns are right-justified; unit price is skipped if amount is.

Since `longtable` refuses to expand the column specification, we force prior expansion of the booleans.

```

77 \cs_new:Nn {\__invoicetwo_print_begin_table:}{
78 \exp_args:Nx \longtable {
79 \bool_if:NT \l__invoicetwo_amount_nonone_bool { c }
80 l
81 \bool_if:NT \l__invoicetwo_vat_nonzero_bool { c }
82 r
83 \bool_if:NT \l__invoicetwo_amount_nonone_bool { r }
84 }
85 }

```

(End definition for `__invoicetwo_print_column_specification:.`)

`__invoicetwo_print_column_title:n` Emits a formatted column title.

```

86 \cs_new:Nn {\__invoicetwo_print_column_title:n}{
87 \multicolumn{1}{c}{
88 \bool_if:NT \l__invoicetwo_colorize_bool {
89 \cellcolor{ \l__invoicetwo_title_color_tl }
90 }
91 \begin{scriptsize}
92 \textbf{#1}
93 \end{scriptsize}}
94 }

```

(End definition for `__invoicetwo_print_column_title:n.`)

`__invoicetwo_print_amount_title:` Emits the “amount” column title. If all amounts are equal to 1, we emit nothing.

```

95 \cs_new:Nn {\__invoicetwo_print_amount_title:}{
96 \bool_if:NT \l__invoicetwo_amount_nonone_bool {
97 \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-amount}} &
98 }
99 }

```

(End definition for `__invoicetwo_print_amount_title:.`)

`__invoicetwo_print_item_title:` Emits the “item” column title.

```

100 \cs_new:Nn {\__invoicetwo_print_item_title:}{
101 \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-item}} &
102 }

```

(End definition for `__invoicetwo_print_item_title:.`)

`_invoicetwo_print_vat_title:` Emits the “VAT” column title. If all VATs are zero, we emit nothing.

```

103 \cs_new:Nn {\_invoicetwo_print_vat_title:}{
104     \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
105         \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-vat}} &
106     }
107 }
```

(End definition for _invoicetwo_print_vat_title:.)

`_invoicetwo_print_unit_price_title:` Emits the “unit price” column title. If all amounts are equal to 1, we emit nothing.

```

108 \cs_new:Nn {\_invoicetwo_print_unit_price_title:}{
109     \bool_if:NT \l__invoicetwo_amount_nonone_bool {
110         \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-unit-price}} &
111     }
112 }
```

(End definition for _invoicetwo_print_unit_price_title:.)

`_invoicetwo_print_price_title:` Emits the “price” column title.

```

113 \cs_new:Nn {\_invoicetwo_print_price_title:}{
114     \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-price}}
115 }
```

(End definition for _invoicetwo_print_price_title:.)

`_invoicetwo_print_header:` Emits the invoice header. If all amounts are equal to 1, we skip the amount column and the unit price column. If all VATs are 0, we skip the VAT column.

```

116 \cs_new:Nn {\_invoicetwo_print_header:}{
117     \__invoicetwo_print_begin_table:
118     \toprule
119     \__invoicetwo_print_amount_title:
120     \__invoicetwo_print_item_title:
121     \__invoicetwo_print_vat_title:
122     \__invoicetwo_print_unit_price_title:
123     \__invoicetwo_print_price_title:
124     \\
125     \midrule
126 }
```

(End definition for _invoicetwo_print_header:.)

4.2 Content Rendering

`_invoicetwo_print_currency_value:n` Emits a currency value, formatted with the current settings. The value must be a floating point number or variable.

`_invoicetwo_print_currency_value:N`

```

127 \cs_new:Nn {\_invoicetwo_print_currency_value:n}{
128     \num[round-precision={\int_use:N \l__invoicetwo_currency_fraction_digits_int},
129     round-mode=places,
130     round-integer-to-decimal,
131     group-minimum-digits=4,
132     group-separator={\GetTranslation{invoice2-thousands-sep}},
133     output-decimal-marker={\GetTranslation{invoice2-decimal-point}}]{
134     #1
```

```

135     }
136     \, \tl_use:N \l__invoicetwo_currency_symbol_tl
137   }
138   \cs_new:Nn {\__invoicetwo_print_currency_value:N}{
139     \__invoicetwo_print_currency_value:n {\fp_use:N #1}
140   }

```

(End definition for __invoicetwo_print_currency_value:n and __invoicetwo_print_currency_value:N.)

__invoicetwo_print_percentage:n Emits a percentage, formatted with the current settings. The value must be a floating point number.

```

141 \cs_new:Nn {\__invoicetwo_print_percentage:n}{
142   \num[round-integer-to-decimal,,
143     group-minimum-digits=4,
144     group-separator={\GetTranslation{invoice2-thousands-sep}},
145     output-decimal-marker={\GetTranslation{invoice2-decimal-point}}]{
146     \fp_eval:n {#1 * 100}
147   }
148   \, \,%
149 }

```

(End definition for __invoicetwo_print_percentage:n.)

__invoicetwo_update_trackers:nn Update the VAT-not-zero and amount-not-zero trackers for the given VAT and amount values.

```

150 \cs_new:Nn {\__invoicetwo_update_trackers:nn}{
151   \int_compare:nT {#1 != 1}{
152     \bool_set_true:N \l__invoicetwo_amount_nonone_bool
153   }
154   \fp_compare:nT {#2 != 0}{
155     \bool_set_true:N \l__invoicetwo_vat_nonzero_bool
156   }
157 }

```

(End definition for __invoicetwo_update_trackers:nn.)

__invoicetwo_update_totals:nmm Update the totals for the given amount, VAT and price per unit values. We increase the net total by \tilde{n} , the VAT total by \tilde{v} and the gross total by \tilde{g} . For unit price p , amount a , VAT v and non-included VATs, this is computed as

$$\begin{aligned}
\tilde{n} &= a \cdot p \\
\tilde{v} &= v \cdot \tilde{n} = a \cdot v \cdot p \\
\tilde{g} &= \tilde{n} + \tilde{v} = a \cdot (1 + v) \cdot p,
\end{aligned}$$

for included VATs, as

$$\begin{aligned}
\tilde{g} &= a \cdot p \\
\tilde{n} &= \frac{\tilde{g}}{1 + v} = \frac{a \cdot p}{1 + v} \\
\tilde{v} &= \tilde{g} - \tilde{n} = \frac{a \cdot v \cdot p}{1 + v}
\end{aligned}$$

```

158 \cs_new:Nn {\__invoicetwo_update_totals:nnn}{
159   \fp_add:Nn \l__invoicetwo_gross_total_fp {
160     #1 * #3 \bool_if:NF \l__invoicetwo_included_vat_bool { * (1 + #2) }
161   }
162   \fp_add:Nn \l__invoicetwo_vat_total_fp {
163     #1 * #2 * #3 \bool_if:NT \l__invoicetwo_included_vat_bool { / (1 + #2) }
164   }
165   \fp_add:Nn \l__invoicetwo_net_total_fp {
166     #1 * #3 \bool_if:NT \l__invoicetwo_included_vat_bool { / (1 + #2) }
167   }
168 }

```

(End definition for __invoicetwo_update_totals:nnn.)

__invoicetwo_print_amount:n Print the given amount value or nothing, if all amounts are one.

```

169 \cs_new:Nn {\__invoicetwo_print_amount:n}{
170   \bool_if:NT \l__invoicetwo_amount_nonone_bool {
171     #1 &
172   }
173 }

```

(End definition for __invoicetwo_print_amount:n.)

__invoicetwo_print_item:n Print the given item name.

```

174 \cs_new:Nn {\__invoicetwo_print_item:n}{
175   #1 &
176 }

```

(End definition for __invoicetwo_print_item:n.)

__invoicetwo_print_vat:n Print the given VAT or nothing, if all VATs are zero.

```

177 \cs_new:Nn {\__invoicetwo_print_vat:n}{
178   \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
179     \__invoicetwo_print_percentage:n {#1} &
180   }
181 }

```

(End definition for __invoicetwo_print_vat:n.)

__invoicetwo_print_unit_price:n Print the given unit price or nothing, if all amounts are one.

```

182 \cs_new:Nn {\__invoicetwo_print_unit_price:n}{
183   \bool_if:NT \l__invoicetwo_amount_nonone_bool {
184     \__invoicetwo_print_currency_value:n {#1} &
185   }
186 }

```

(End definition for __invoicetwo_print_unit_price:n.)

__invoicetwo_print_price:nnm Print the price \tilde{p} for the current item. Parameters are amount, VAT and price per unit. For unit price p , amount a , VAT v and non-included VATs, this is computed as

$$\tilde{p} = a \cdot (1 + v) \cdot p,$$

for included VATs, as

$$\tilde{p} = a \cdot p.$$

```

187 \cs_new:Nn {\__invoicetwo_print_price:nnn}{
188   \__invoicetwo_print_currency_value:n {
189     \fp_eval:n {
190       #1 \bool_if:NF \l__invoicetwo_included_vat_bool { * (1 + #2) } * #3
191     }
192   }
193 }

```

(End definition for __invoicetwo_print_price:nnn.)

__invoicetwo_colorize_row:nnn Colorize the current row according to even/odd colorization, if color was requested in the configuration.

```

194 \cs_new:Nn {\__invoicetwo_colorize_row:}{
195   \bool_if:NT \l__invoicetwo_colorize_bool {
196     \int_if_odd:nTF \g__invoicetwo_row_number_int {
197       \rowcolor{ \l__invoicetwo_odd_color_tl }
198     }
199     {
200       \rowcolor{ \l__invoicetwo_even_color_tl }
201     }
202   }
203   \int_gincr:N \g__invoicetwo_row_number_int
204 }

```

(End definition for __invoicetwo_colorize_row:nnn.)

__invoicetwo_add_row:nnnn Add a new entry with non-default VAT to the invoice. This will update the VAT-not-zero and amount-not-zero trackers. Arguments are expected to be (in this order): amount, item name, VAT, unit price.

```

205 \cs_new:Nn {\__invoicetwo_add_row:nnnn}{
206   \__invoicetwo_update_trackers:nn {#1} {#3}
207   \__invoicetwo_update_totals:nnn {#1} {#3} {#4}
208   \tl_put_right:Nn \l__invoicetwo_tabular_tl {
209     \__invoicetwo_colorize_row:
210     \__invoicetwo_print_amount:n {#1}
211     \__invoicetwo_print_item:n {#2}
212     \__invoicetwo_print_vat:n {#3}
213     \__invoicetwo_print_unit_price:n {#4}
214     \__invoicetwo_print_price:nnn {#1} {#3} {#4} \\
215   }
216 }

```

(End definition for __invoicetwo_add_row:nnnn.)

\invoiceitem This is the user interface to adding items. We read the VAT from the optional argument or the configuration, add an amount of one for single items and delegate to our implementation.

```

217 \NewDocumentCommand{\invoiceitem}{ommm}{
218   \__invoicetwo_add_row:nnnn {#2} {#3}
219   {\IfValueTF{#1}{#1}{\fp_use:N \l__invoicetwo_vat_fp}}
220   {#4}
221 }
222 \NewDocumentCommand{\invoicessingleitem}{omm}{
223   \__invoicetwo_add_row:nnnn {1} {#2}

```

```

224         {\IfValueTF{#1}{#1}{\fp_use:N \l__invoicetwo_vat_fp}}
225         {#3}
226     }

```

(End definition for `\invoiceitem` and `\invoicingleitem`. These functions are documented on page 4.)

4.3 Footer Rendering

`__invoicetwo_print_multicolumn_count:` Emits the column count to group in the final lines. This is the amount of columns minus one.

```

227 \cs_new:Nn {\__invoicetwo_print_multicolumn_count:}{
228     \int_eval:n {
229         1
230         \bool_if:NT \l__invoicetwo_vat_nonzero_bool {+1}
231         \bool_if:NT \l__invoicetwo_amount_nonone_bool {+2}
232     }
233 }

```

(End definition for `__invoicetwo_print_multicolumn_count:`.)

`__invoicetwo_print_footer_item:n` Emits a formatted footer item. We also handle colorization of the next cell here.

```

234 \cs_new:Nn {\__invoicetwo_print_footer_item:n}{
235     \multicolumn{\__invoicetwo_print_multicolumn_count:}{r}{
236         \bool_if:NT \l__invoicetwo_colorize_bool {
237             \cellcolor{ \l__invoicetwo_total_color_tl }
238         }
239         \textbf{#1}
240     } & \bool_if:NT \l__invoicetwo_colorize_bool {
241         \cellcolor{ \l__invoicetwo_total_color_tl }
242     }
243 }

```

(End definition for `__invoicetwo_print_footer_item:n`.)

`__invoicetwo_print_net_item:` Emits the “net total” item.

```

244 \cs_new:Nn {\__invoicetwo_print_net_item:}{
245     \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
246         \__invoicetwo_print_footer_item:n {\GetTranslation{invoice2-net-total}}
247         \__invoicetwo_print_currency_value:N \l__invoicetwo_net_total_fp \\
248     }
249 }

```

(End definition for `__invoicetwo_print_net_item:`.)

`__invoicetwo_print_vat_item:` Emits the “VAT total” item.

```

250 \cs_new:Nn {\__invoicetwo_print_vat_item:}{
251     \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
252         \__invoicetwo_print_footer_item:n {\GetTranslation{invoice2-vat-total}}
253         \__invoicetwo_print_currency_value:N \l__invoicetwo_vat_total_fp \\
254     }
255 }

```

(End definition for `__invoicetwo_print_vat_item:`.)

`_invoicetwo_print_gross_item:` Emits the “gross total” item.

```

256 \cs_new:Nn {\_invoicetwo_print_gross_item:}{
257   \_invoicetwo_print_footer_item:n {\GetTranslation{invoice2-gross-total}}
258   \_invoicetwo_print_currency_value:N \l\_invoicetwo_gross_total_fp \
259 }

```

(End definition for _invoicetwo_print_gross_item:.)

`_invoicetwo_print_footer:` Emits the invoice footer. If all VATs are 0, we skip the net and VAT rows.

```

260 \cs_new:Nn {\_invoicetwo_print_footer:}{
261   \midrule
262   \_invoicetwo_print_net_item:
263   \_invoicetwo_print_vat_item:
264   \_invoicetwo_print_gross_item:
265   \bottomrule
266   \endlongtable
267 }

```

(End definition for _invoicetwo_print_footer:.)

4.4 The Environment itself

`_invoicetwo_begin_invoice:n` Begins a new invoice environment. We check for nested environments and set up configuration overrides.

```

268 \cs_new:Nn {\_invoicetwo_begin_invoice:n}{
269   \bool_if:NT \l\_invoicetwo_in_invoice_bool {
270     \msg_error:nn {invoice2} {nested-invoice}
271   }
272   \bool_set_true:N \l\_invoicetwo_in_invoice_bool
273   \int_gset:Nn \g\_invoicetwo_row_number_int {1}
274   \keys_set:nn {invoice2} {#1}
275 }

```

(End definition for _invoicetwo_begin_invoice:n.)

`_invoicetwo_end_invoice:` End an invoice environment. Here, we can emit the `longtable` environment since we possess all required information.

```

276 \cs_new:Nn {\_invoicetwo_end_invoice:}{
277   \_invoicetwo_print_header:
278   \tl_use:N \l\_invoicetwo_tabular_tl
279   \_invoicetwo_print_footer:
280 }

```

(End definition for _invoicetwo_end_invoice:.)

`\begin{invoice}` The user interface is the `invoice` environment.

```

\end{invoice}
281 \NewDocumentEnvironment{invoice}{o}{
282   \IfValueTF{#1}{\_invoicetwo_begin_invoice:n {#1}}{\_invoicetwo_begin_invoice:n {}}
283 }
284 {
285   \_invoicetwo_end_invoice:
286 }

```

(End definition for \begin{invoice} and \end{invoice}. These functions are documented on page 4.)

5 Messages and Errors

```
287 \msg_new:nnnn {invoice2} {nested-invoice}
288   {\msg_error_text:n {invoice2}:-%
289     Nested-invoice-environments-are-not-supported.}
290   {Invoices-can-not-contain-invoices.\}%
291   Please-check-your-environment-delimiters.}
292 \end{package}
```

6 Dictionary Files

6.1 English Dictionary

```
293 \dictEnglish
294 \ProvideDictionaryFor{English}{invoice2}
295 \ProvideDictTranslation{invoice2-thousands-sep}{\,,}
296 \ProvideDictTranslation{invoice2-decimal-point}{.}
297 \ProvideDictTranslation{invoice2-amount}{Amount}
298 \ProvideDictTranslation{invoice2-item}{Item}
299 \ProvideDictTranslation{invoice2-vat}{VAT}
300 \ProvideDictTranslation{invoice2-unit-price}{Unit-Price}
301 \ProvideDictTranslation{invoice2-price}{Price}
302 \ProvideDictTranslation{invoice2-net-total}{Net-Total}
303 \ProvideDictTranslation{invoice2-vat-total}{VAT}
304 \ProvideDictTranslation{invoice2-gross-total}{Gross-Total}
305 \end{dictEnglish}
```

6.2 German Dictionary

```
306 \dictGerman
307 \ProvideDictionaryFor{German}{invoice2}
308 \ProvideDictTranslation{invoice2-thousands-sep}{\,,}
309 \ProvideDictTranslation{invoice2-decimal-point}{,}
310 \ProvideDictTranslation{invoice2-amount}{Anzahl}
311 \ProvideDictTranslation{invoice2-item}{Posten}
312 \ProvideDictTranslation{invoice2-vat}{MwSt.}
313 \ProvideDictTranslation{invoice2-unit-price}{St\ "uckpreis}
314 \ProvideDictTranslation{invoice2-price}{Preis}
315 \ProvideDictTranslation{invoice2-net-total}{Nettobetrag}
316 \ProvideDictTranslation{invoice2-vat-total}{MwSt.}
317 \ProvideDictTranslation{invoice2-gross-total}{Gesamtbetrag}
318 \end{dictGerman}
```

Change History

1.0
General: First public release 5

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	I
\ " 313	\IfValueTF 219, 224, 282
\\$ 54	int commands:
\% 148	\int_compare:nTF 151
\, 15, 136, 148, 295, 308	\int_eval:n 228
\\ 124, 214, 247, 253, 258, 290	\int_gincr:N 203
B	\int_gset:Nn 273
\begin 4, 91	\int_if_odd:nTF 196
\begin{invoice} 4, <u>281</u>	\int_new:N 28, 39
\beginlongtable 10	\int_use:N 128
bool commands:	invoice 2, 2, 4, 7, 10, 16
\bool_if:NTF 79, 81, 83, 88, 96, 104, 109,	\invoiceitem 4, 4, 4, <u>217</u>
160, 163, 166, 170, 178, 183, 190,	\invoiceoptions 3, 3, 5, <u>74</u>
195, 230, 231, 236, 240, 245, 251, 269	invoiceoptions 2, 6, 9
\bool_new:N 26, 29, 38, 40, 41	\invoicessingleitem 4, 4, 4, 4, <u>217</u>
\bool_set_true:N 152, 155, 272	invoicetwo internal commands:
\bottomrule 265	_invoicetwo_add_row:nmmn <u>205</u> , 218, 223
C	\l_invoicetwo_amount_nonone_bool <u>41</u> ,
\cellcolor 89, 237, 241	79, 83, 96, 109, 152, 170, 183, 231
cs commands:	_invoicetwo_begin_invoice:n <u>268</u> , 282
\cs_new:Nn 77, 86, 95, 100, 103, 108,	\l_invoicetwo_colorize_bool <u>29</u> , 58, 88, 195, 236, 240
113, 116, 127, 138, 141, 150, 158,	_invoicetwo_colorize_row: 194, 209
169, 174, 177, 182, 187, 194, 205,	_invoicetwo_colorize_row:mmm . <u>194</u>
227, 234, 244, 250, 256, 260, 268, 276	\l_invoicetwo_currency_fraction_digits_int <u>28</u> , 55, 128
D	\l_invoicetwo_currency_symbol_tl <u>27</u> , 52, 136
\DeclareTranslationFallback 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	_invoicetwo_end_invoice: . <u>276</u> , 285
E	\l_invoicetwo_even_color_tl <u>32</u> , 65, 200
\end 4, 93	\l_invoicetwo_gross_total_fp <u>44</u> , 159, 258
\endlongtable 266	\l_invoicetwo_in_invoice_bool <u>38</u> , 269, 272
\end{invoice} 4, <u>281</u>	\l_invoicetwo_included_vat_bool <u>26</u> , 50, 160, 163, 166, 190
exp commands:	\l_invoicetwo_net_total_fp <u>42</u> , 165, 247
\exp_args:Nx 78	\l_invoicetwo_odd_color_tl <u>30</u> , 62, 197
F	_invoicetwo_print_amount:n <u>169</u> , 210
fp commands:	_invoicetwo_print_amount_title: 95, 119
\fp_add:Nn 159, 162, 165	
\fp_compare:nTF 154	
\fp_eval:n 146, 189	
\fp_new:N 25, 42, 43, 44	
\fp_use:N 139, 219, 224	
G	
\GetTranslation ... 97, 101, 105, 110, 114, 132, 133, 144, 145, 246, 252, 257	

