The invoice2 package
Intelligent invoices with LaTeX3

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

Version 1.2
Monday January 15, 2018

While there exists a vast amount of templates, unpublished packages and code snippets designed for creating invoices, the only solution available in the CTAN is the invoice[1] package, which is not actively maintained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 LaTeX3, while using the print-friendly layout provided by the booktabs[2] package.

Contents

---

[1] `https://www.ctan.org/pkg/invoice`
[2] `https://www.ctan.org/pkg/booktabs`

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 shold 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.

currency-in-header Controls if the currency symbol should be printed in the invoice header instead of printing it after each amount. Default is printing after each amount.

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, german and swiss 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.

---

[3] `https://www.ctan.org/pkg/translations`

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 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 hat 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 hat 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.

`\begin{invoice}`
`\end{invoice}`

```
\begin [⟨key=value...⟩] {invoice}
\end {invoice}
```

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.

`\invoiceitem`
`\invoicesingleitem`

```
\invoiceitem [⟨VAT⟩] {⟨amount⟩} {⟨item name⟩} {⟨unit price⟩}
\invoicesingleitem [⟨VAT⟩] {⟨item name⟩} {⟨unit price⟩}
```

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 and the unit price can be an integer or 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 € (provided by the `eurosym` package). Since we only have single copies, we will use the `\invoicesingleitem` command.

```
\usepackage{eurosym}
\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 €

For some currencies, the symbol ist fairly long. In this case, it is advisable to move the currency symbol to the invoice header.

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

Item	**Price (CHF)**
Ignition!	4 087.99
The Art of Computer Programming 1–4	162.99
The TeXbook	55.69
Gross Total (CHF)	4 306.67

Now, let us apply a VAT of 9 %. This will show the VAT column, the item price column and the extended total.

```
\usepackage{eurosym}
\invoiceoptions{vat=.09}
\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	**VAT**	**Unit Price**	**Price**
Ignition!	9 %	4 087.99 €	4 455.91 €
The Art of Computer Programming 1–4	9 %	162.99 €	177.66 €
The TeXbook	9 %	55.69 €	60.70 €
		Net Total	4 306.67 €
		VAT	387.60 €
		Gross Total	4 694.27 €

Note that if we specify included VAT in the above example, the output is different. Additionally, we specify our options as environment options.

```
\usepackage{eurosym}
```

```
\begin{invoice}[vat=.09, included-vat=true, 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	**VAT**	**Unit Price**	**Price**
Ignition!	9 %	3 750.45 €	4 087.99 €
The Art of Computer Programming 1–4	9 %	149.53 €	162.99 €
The TeXbook	9 %	51.09 €	55.69 €
		Net Total	3 951.07 €
		VAT	355.60 €
		Gross Total	4 306.67 €

A more complex example involves amounts and a VAT. We keep the 9 % VAT for our books, buy duplicate editions of The Art of Computer Programming 1 and 2 (simulated by buying 1.5 copies) and stock up on more copies of the TeXbook. This will enable all columns and an extended total.

```
\usepackage{eurosym}
\invoiceoptions{vat=.09}
\begin{invoice}[currency-symbol={\euro{}}]
    \invoicesingleitem{Ignition!}{4087.99}
    \invoiceitem{1.5}{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 €
1.5	The Art of Computer Programming 1–4	9 %	162.99 €	266.49 €
20	The TeXbook	9 %	55.69 €	1 214.04 €
			Net Total	5 446.28 €
			VAT	490.16 €
			Gross Total	5 936.44 €

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

```
\usepackage{eurosym}
\begin{invoice}[currency-symbol={\euro{}}, colorize,
    even-color={lightgray!50}]
    \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 €

5 Version History

5.1 Version 1.2

- Fix option loading for the xcolor package. Thanks to Alfred H. Gitter for reporting.
- Corrected german translation of “VAT” to “USt”. Thanks to Alfred H. Gitter for reporting.
- Clarified that eurosym is required to use the `euro` symbol. Thanks to Alfred H. Gitter for the suggestion.
- Split the `README.md` into a Github version and one for CTAN.
- Add support for Swiss German. Thanks to `@foreachthing` for the translation.
- Permit moving the currency symbol to the header. Thanks to `@foreachthing` for the report.

5.2 Version 1.1

- Spelling fix in `README.md`. Thanks to Ina Dau for noticing.
- Clean up unused `README` generation in the `.ins` file.
- Report in with the correct motto.
- Support non-integer amounts. Thanks to Gijs Hillenius for the suggestion.
- Print the unit price column if an item has either VAT or amount $\neq 1$. Thanks to Gijs Hillenius for the suggestion.

5.3 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

```
⟨*package⟩
⟨@@=invoicetwo⟩
```

1 Setup and Initialization

```
\RequirePackage{booktabs}
\RequirePackage{expl3}
\RequirePackage{l3keys2e}
\RequirePackage{longtable}
\RequirePackage{siunitx}
\RequirePackage{translations}
\PassOptionsToPackage{table}{xcolor}
\RequirePackage{xcolor}
\RequirePackage{xparse}
\ProvidesExplPackage{invoice2}{2018/01/15}{1.2}
    {Intelligent invoices with LaTeX3}
```

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

```
\LoadDictionaryFor{english}{invoice2}
\LoadDictionaryFor{german}{invoice2}
\LoadDictionaryFor{swissgerman}{invoice2}
\DeclareTranslationFallback{invoice2-thousands-sep}{\,}
\DeclareTranslationFallback{invoice2-decimal-point}{.}
\DeclareTranslationFallback{invoice2-amount}{Amount}
\DeclareTranslationFallback{invoice2-item}{Item}
\DeclareTranslationFallback{invoice2-vat}{VAT}
\DeclareTranslationFallback{invoice2-unit-price}{Unit~Price}
\DeclareTranslationFallback{invoice2-price}{Price}
\DeclareTranslationFallback{invoice2-net-total}{Net~Total}
\DeclareTranslationFallback{invoice2-vat-total}{VAT}
\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.

```
\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.

```
\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

```
\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.

```
\int_new:N \l__invoicetwo_currency_fraction_digits_int
```

(*End definition for* `\l__invoicetwo_currency_fraction_digits_int`*.*)

`\l__invoicetwo_currency_in_header_bool`

Controls if the invoice currency symbol should be moved intro the table header (“Unit Price ($)”) instead of printing it after each amount (“123.45 $”). Can save space for long currencies.

```
\bool_new:N \l__invoicetwo_currency_in_header_bool
```

(*End definition for* `\l__invoicetwo_currency_in_header_bool`*.*)

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.

```
\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.

```
\tl_new:N \l__invoicetwo_odd_color_tl
\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.

```
\tl_new:N \l__invoicetwo_even_color_tl
\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.

```
\tl_new:N \l__invoicetwo_title_color_tl
\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.

```
\tl_new:N \l__invoicetwo_total_color_tl
\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.

```
\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.

```
\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.

```
\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.

```
\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.

```
\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.

```
\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.

```
\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.

```
\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.

```
\keys_define:nn {invoice2} {
    vat .fp_set:N = \l__invoicetwo_vat_fp,
    vat .value_required:n = true,
    vat .initial:n = 0,
    included-vat .bool_set:N = \l__invoicetwo_included_vat_bool,
    included-vat .initial:n = false,
    currency-symbol .tl_set:N = \l__invoicetwo_currency_symbol_tl,
    currency-symbol .value_required:n = true,
    currency-symbol .initial:n = {\$},
    currency-fraction-digits .int_set:N = \l__invoicetwo_currency_fraction_digits_int,
    currency-fraction-digits .value_required:n = true,
    currency-fraction-digits .initial:n = 2,
    currency-in-header .bool_set:N = \l__invoicetwo_currency_in_header_bool,
    currency-in-header .initial:n = false,
    colorize .bool_set:N = \l__invoicetwo_colorize_bool,
    colorize .initial:n = false,
    odd-color .initial:n = white,
    odd-color .value_required:n = true,
    odd-color .tl_set:N = \l__invoicetwo_odd_color_tl,
    even-color .initial:n = lightgray,
    even-color .value_required:n = true,
    even-color .tl_set:N = \l__invoicetwo_even_color_tl,
    title-color .initial:n = white,
    title-color .value_required:n = true,
    title-color .tl_set:N = \l__invoicetwo_title_color_tl,
    total-color .initial:n = white,
    total-color .value_required:n = true,
    total-color .tl_set:N = \l__invoicetwo_total_color_tl
}
```

3.1 Package Parameters

Fortunately, l3keys2e handles the heavy lifting for us.

```
\ProcessKeysOptions{invoice2}
```

3.2 `invoiceoptions` Command

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

```
\NewDocumentCommand{\invoiceoptions}{m}{
    \keys_set:nn {invoice2} {#1}
}
```

(*End definition for* `\invoiceoptions`. *This function is documented on page 4.*)

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.

```
\cs_new:Nn {\__invoicetwo_print_begin_table:}{
    \exp_args:Nx \longtable {
        \bool_if:NT \l__invoicetwo_amount_nonone_bool { c }
        l
        \bool_if:NT \l__invoicetwo_vat_nonzero_bool { c }
        r
        \bool_if:nT { \l__invoicetwo_amount_nonone_bool || \l__invoicetwo_vat_nonzero_bool }
    }
}
```

(*End definition for* `\__invoicetwo_print_column_specification:`.)

`\__invoicetwo_print_column_title:n`

Emits a formatted column title.

```
\cs_new:Nn {\__invoicetwo_print_column_title:n}{
    \multicolumn{1}{c}{
        \bool_if:NT \l__invoicetwo_colorize_bool {
            \cellcolor{ \l__invoicetwo_title_color_tl }
        }
        \begin{scriptsize}
            \textbf{#1}
        \end{scriptsize}}
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_print_amount_title:}{
    \bool_if:NT \l__invoicetwo_amount_nonone_bool {
        \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-amount}} &
    }
}
```

(*End definition for* `\__invoicetwo_print_amount_title:`.)

`\__invoicetwo_print_item_title:`

Emits the “item” column title.

```
\cs_new:Nn {\__invoicetwo_print_item_title:}{
    \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-item}} &
}
```

(*End definition for* `\__invoicetwo_print_item_title:`.)

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

```
\cs_new:Nn {\__invoicetwo_print_vat_title:}{
    \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
        \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-vat}} &
    }
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_print_unit_price_title:}{
    \bool_if:nT { \l__invoicetwo_amount_nonone_bool || \l__invoicetwo_vat_nonzero_bool } {
        \__invoicetwo_print_column_title:n {
            \GetTranslation{invoice2-unit-price}
            \bool_if:NT \l__invoicetwo_currency_in_header_bool {
                \ (\tl_use:N \l__invoicetwo_currency_symbol_tl)
            }
        } &
    }
}
```

(*End definition for* `\__invoicetwo_print_unit_price_title:`.)

__invoicetwo_print_price_title: Emits the “price” column title.

```
\cs_new:Nn {\__invoicetwo_print_price_title:}{
    \__invoicetwo_print_column_title:n {\GetTranslation{invoice2-price}
        \bool_if:NT \l__invoicetwo_currency_in_header_bool {
            \ (\tl_use:N \l__invoicetwo_currency_symbol_tl)
        }
    }
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_print_header:}{
    \__invoicetwo_print_begin_table:
    \toprule
    \__invoicetwo_print_amount_title:
    \__invoicetwo_print_item_title:
    \__invoicetwo_print_vat_title:
    \__invoicetwo_print_unit_price_title:
    \__invoicetwo_print_price_title:
    \\
    \midrule
}
```

(*End definition for* `\__invoicetwo_print_header:`.)

4.2 Content Rendering

__invoicetwo_print_real_value:n

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

```
\cs_new:Nn {\__invoicetwo_print_real_value:n}{
    \num[round-integer-to-decimal,
    group-minimum-digits=4,
    group-separator={\GetTranslation{invoice2-thousands-sep}},
    output-decimal-marker={\GetTranslation{invoice2-decimal-point}}]{
        #1
    }
}
```

(*End definition for* `\__invoicetwo_print_real_value:n`.)

__invoicetwo_print_currency_value:n
__invoicetwo_print_currency_value:N

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

```
\cs_new:Nn {\__invoicetwo_print_currency_value:n}{
    \num[round-precision={\int_use:N \l__invoicetwo_currency_fraction_digits_int},
    round-mode=places,
    round-integer-to-decimal,
    group-minimum-digits=4,
    group-separator={\GetTranslation{invoice2-thousands-sep}},
    output-decimal-marker={\GetTranslation{invoice2-decimal-point}}]{
        #1
    }
    \bool_if:NF \l__invoicetwo_currency_in_header_bool {
        \, \tl_use:N \l__invoicetwo_currency_symbol_tl
    }
}
\cs_new:Nn {\__invoicetwo_print_currency_value:N}{
    \__invoicetwo_print_currency_value:n {\fp_use:N #1}
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_print_percentage:n}{
    \num[round-integer-to-decimal,
    group-minimum-digits=4,
    group-separator={\GetTranslation{invoice2-thousands-sep}},
    output-decimal-marker={\GetTranslation{invoice2-decimal-point}}]{
        \fp_eval:n {#1 * 100}
    }
    \, \%
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_update_trackers:nn}{
    \fp_compare:nT {#1 != 1}{
        \bool_set_true:N \l__invoicetwo_amount_nonone_bool
    }
    \fp_compare:nT {#2 != 0}{
        \bool_set_true:N \l__invoicetwo_vat_nonzero_bool
    }
}
```

(*End definition for* `\__invoicetwo_update_trackers:nn`.)

`\__invoicetwo_update_totals:nnn`

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

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

```
\cs_new:Nn {\__invoicetwo_update_totals:nnn}{
    \fp_add:Nn \l__invoicetwo_gross_total_fp {
        #1 * #3 \bool_if:NF \l__invoicetwo_included_vat_bool { * (1 + #2) }
    }
    \fp_add:Nn \l__invoicetwo_vat_total_fp {
        #1 * #2 * #3 \bool_if:NT \l__invoicetwo_included_vat_bool { / (1 + #2) }
    }
    \fp_add:Nn \l__invoicetwo_net_total_fp {
        #1 * #3 \bool_if:NT \l__invoicetwo_included_vat_bool { / (1 + #2) }
    }
}
```

(*End definition for* `\__invoicetwo_update_totals:nnn`.)

`\__invoicetwo_print_amount:n`

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

```
\cs_new:Nn {\__invoicetwo_print_amount:n}{
    \bool_if:NT \l__invoicetwo_amount_nonone_bool {
        \__invoicetwo_print_real_value:n {#1} &
    }
}
```

(*End definition for* `\__invoicetwo_print_amount:n`.)

`\__invoicetwo_print_item:n`

Print the given item name.

```
\cs_new:Nn {\__invoicetwo_print_item:n}{
    #1 &
}
```

(*End definition for* `\__invoicetwo_print_item:n`.)

`\__invoicetwo_print_vat:n`

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

```
\cs_new:Nn {\__invoicetwo_print_vat:n}{
    \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
        \__invoicetwo_print_percentage:n {#1} &
    }
}
```

(*End definition for* `\__invoicetwo_print_vat:n`.)

`\__invoicetwo_print_unit_price:n`

Print the given unit price or nothing, if all amounts are one. Parameters are VAT and price per unit. For unit price p, VAT v and non-included VATs, this is computed as

$$\tilde{p} = \frac{p}{1+v},$$

for included VATs, as

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

```
\cs_new:Nn {\__invoicetwo_print_unit_price:nn}{
    \bool_if:nT { \l__invoicetwo_amount_nonone_bool || \l__invoicetwo_vat_nonzero_bool } {
        \__invoicetwo_print_currency_value:n {
            \fp_eval:n {
                #2 \bool_if:NT \l__invoicetwo_included_vat_bool { / (1 + #1) }
            }
        } &
    }
}
```

(*End definition for* `\__invoicetwo_print_unit_price:n`.)

`\__invoicetwo_print_price:nnn`

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

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

```
\cs_new:Nn {\__invoicetwo_print_price:nnn}{
    \__invoicetwo_print_currency_value:n {
        \fp_eval:n {
            #1 \bool_if:NF \l__invoicetwo_included_vat_bool { * (1 + #2) } * #3
        }
    }
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_colorize_row:}{
    \bool_if:NT \l__invoicetwo_colorize_bool {
        \int_if_odd:nTF \g__invoicetwo_row_number_int {
            \rowcolor{ \l__invoicetwo_odd_color_tl }
        }
        {
            \rowcolor{ \l__invoicetwo_even_color_tl }
        }
    }
    \int_gincr:N \g__invoicetwo_row_number_int
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_add_row:nnnn}{
    \__invoicetwo_update_trackers:nn {#1} {#3}
    \__invoicetwo_update_totals:nnn {#1} {#3} {#4}
    \tl_put_right:Nn \l__invoicetwo_tabular_tl {
        \__invoicetwo_colorize_row:
        \__invoicetwo_print_amount:n {#1}
        \__invoicetwo_print_item:n {#2}
        \__invoicetwo_print_vat:n {#3}
        \__invoicetwo_print_unit_price:nn {#3} {#4}
        \__invoicetwo_print_price:nnn {#1} {#3} {#4} \\
    }
}
```

(*End definition for* `\__invoicetwo_add_row:nnnn`*.*)

`\invoiceitem`
`\invoicesingleitem`

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.

```
\NewDocumentCommand{\invoiceitem}{ommm}{
    \__invoicetwo_add_row:nnnn {#2} {#3}
        {\IfValueTF{#1}{#1}{\fp_use:N \l__invoicetwo_vat_fp}}
        {#4}
}
\NewDocumentCommand{\invoicesingleitem}{omm}{
    \__invoicetwo_add_row:nnnn {1} {#2}
        {\IfValueTF{#1}{#1}{\fp_use:N \l__invoicetwo_vat_fp}}
        {#3}
}
```

(*End definition for* `\invoiceitem` *and* `\invoicesingleitem`*. 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.

```
\cs_new:Nn {\__invoicetwo_print_multicolumn_count:}{
    \int_eval:n {
        1
        \bool_if:NT \l__invoicetwo_vat_nonzero_bool {+1}
        \bool_if:NT \l__invoicetwo_amount_nonone_bool {+1}
        \bool_if:nT { \l__invoicetwo_amount_nonone_bool || \l__invoicetwo_vat_nonzero_bool }
    }
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_print_footer_item:n}{
    \multicolumn{\__invoicetwo_print_multicolumn_count:}{r}{
        \bool_if:NT \l__invoicetwo_colorize_bool {
            \cellcolor{ \l__invoicetwo_total_color_tl }
        }
        \textbf{#1}
    } & \bool_if:NT \l__invoicetwo_colorize_bool {
        \cellcolor{ \l__invoicetwo_total_color_tl }
    }
}
```

(*End definition for* `\__invoicetwo_print_footer_item:n`.)

__invoicetwo_print_net_item:

Emits the “net total” item.

```
\cs_new:Nn {\__invoicetwo_print_net_item:}{
    \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
        \__invoicetwo_print_footer_item:n {
            \GetTranslation{invoice2-net-total}
            \bool_if:NT \l__invoicetwo_currency_in_header_bool {
                \ (\tl_use:N \l__invoicetwo_currency_symbol_tl)
            }
        }
        \__invoicetwo_print_currency_value:N \l__invoicetwo_net_total_fp \\
    }
}
```

(*End definition for* `\__invoicetwo_print_net_item:`.)

__invoicetwo_print_vat_item:

Emits the “VAT total” item.

```
\cs_new:Nn {\__invoicetwo_print_vat_item:}{
    \bool_if:NT \l__invoicetwo_vat_nonzero_bool {
        \__invoicetwo_print_footer_item:n {\GetTranslation{invoice2-vat-total}}
        \__invoicetwo_print_currency_value:N \l__invoicetwo_vat_total_fp \\
    }
}
```

(*End definition for* `\__invoicetwo_print_vat_item:`.)

__invoicetwo_print_gross_item:

Emits the “gross total” item.

```
\cs_new:Nn {\__invoicetwo_print_gross_item:}{
    \__invoicetwo_print_footer_item:n {
        \GetTranslation{invoice2-gross-total}
        \bool_if:NT \l__invoicetwo_currency_in_header_bool {
            \ (\tl_use:N \l__invoicetwo_currency_symbol_tl)
        }
    }
    \__invoicetwo_print_currency_value:N \l__invoicetwo_gross_total_fp \\
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_print_footer:}{
    \midrule
    \__invoicetwo_print_net_item:
    \__invoicetwo_print_vat_item:
    \__invoicetwo_print_gross_item:
    \bottomrule
    \endlongtable
}
```

(*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.

```
\cs_new:Nn {\__invoicetwo_begin_invoice:n}{
    \bool_if:NT \l__invoicetwo_in_invoice_bool {
        \msg_error:nn {invoice2} {nested-invoice}
    }
    \bool_set_true:N \l__invoicetwo_in_invoice_bool
    \int_gset:Nn \g__invoicetwo_row_number_int {1}
    \keys_set:nn {invoice2} {#1}
}
```

(*End definition for* `\__invoicetwo_begin_invoice:n`.)

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

```
\cs_new:Nn {\__invoicetwo_end_invoice:}{
    \__invoicetwo_print_header:
    \tl_use:N \l__invoicetwo_tabular_tl
    \__invoicetwo_print_footer:
}
```

(*End definition for* `\__invoicetwo_end_invoice:`.)

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

```
\NewDocumentEnvironment{invoice}{o}{
    \IfValueTF{#1}{\__invoicetwo_begin_invoice:n {#1}}{\__invoicetwo_begin_invoice:n {}}
}
{
    \__invoicetwo_end_invoice:
}
```

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

5 Messages and Errors

```
\msg_new:nnnn {invoice2} {nested-invoice}
    {\msg_error_text:n {invoice2}:~%
    Nested~invoice~environments~are~not~supported.}
    {Invoices~can~not~contain~invoices.\\%
    Please~check~your~environment~delimiters.}
⟨/package⟩
```

6 Dictionary Files

6.1 English Dictionary

```
⟨*dictEnglish⟩
\ProvideDictionaryFor{English}{invoice2}
\ProvideDictTranslation{invoice2-thousands-sep}{\,}
\ProvideDictTranslation{invoice2-decimal-point}{.}
\ProvideDictTranslation{invoice2-amount}{Amount}
\ProvideDictTranslation{invoice2-item}{Item}
\ProvideDictTranslation{invoice2-vat}{VAT}
\ProvideDictTranslation{invoice2-unit-price}{Unit~Price}
\ProvideDictTranslation{invoice2-price}{Price}
\ProvideDictTranslation{invoice2-net-total}{Net~Total}
\ProvideDictTranslation{invoice2-vat-total}{VAT}
\ProvideDictTranslation{invoice2-gross-total}{Gross~Total}
⟨/dictEnglish⟩
```

6.2 German Dictionary

```
⟨*dictGerman⟩
\ProvideDictionaryFor{German}{invoice2}
\ProvideDictTranslation{invoice2-thousands-sep}{\,}
\ProvideDictTranslation{invoice2-decimal-point}{,}
\ProvideDictTranslation{invoice2-amount}{Anzahl}
\ProvideDictTranslation{invoice2-item}{Posten}
\ProvideDictTranslation{invoice2-vat}{USt}
\ProvideDictTranslation{invoice2-unit-price}{St\"uckpreis}
\ProvideDictTranslation{invoice2-price}{Preis}
\ProvideDictTranslation{invoice2-net-total}{Nettobetrag}
\ProvideDictTranslation{invoice2-vat-total}{USt}
\ProvideDictTranslation{invoice2-gross-total}{Gesamtbetrag}
⟨/dictGerman⟩
```

6.3 Swiss German Dictionary

```
⟨*dictSwissGerman⟩
\ProvideDictionaryFor{Swiss German}{invoice2}
\ProvideDictTranslation{invoice2-thousands-sep}{’}
\ProvideDictTranslation{invoice2-decimal-point}{.}
⟨/dictSwissGerman⟩
```

Change History

1.0
    General: First public release 7
1.1
    General: Bugfix release 7
1.2
    General: Bugfix release 7

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.