

The zugferd package*

Creating electronic and hybrid invoices using L^AT_EX

Marei Peischl
marei@peitex.de

December 8, 2024

Abstract

Invoicing is getting more and more automated. Starting with public sector, within Germany there already is a requirement to stick to the Faktur-X Standard. First Invoices based on this implementation here have been created back in 2021. And this is now the trial to create a more universal and public package to support the current Version of ZUGFeRD and therefore also X-Rechnung and Faktur-X. The fundamental idea of this package was to use the calculation within L^AT_EX as well. So it also creates the XML file for the attachment on the fly. To match typical setups there is a wrapper package which usually would also hold the personal Invoicing layout configuration.

Sponsors & Supporters

Most of this package has been created within my free time and for my personal use. At start, it was not a paid project at all. Since it is addressing business users it would be great if we could keep this actively maintained. If you are able to support this either financially for the maintenance effort, a custom extension, I'd love to hear from you.

This project was financially supported by:

- Pengutronix e.K., <https://pengutronix.de>
Special thanks to them, as they also sponsored the minimal portable T_EX Live setup.
- Frederik Schwan ([Schwan.IT](https://www.schwan.it))
- byte physics e.K., <https://www.byte-physics.de>

*This document corresponds to zugferd v0.9c, dated 2024-12-08.

Contents

1	Quick start	3
1.1	Disclaimer concerning the zugferd-invoice Package	3
2	Package Options	3
3	User Commands	4
4	Commands for template authors	4
4.1	Interfaces to write the XML contents	5
4.2	Commands to temporary disable/re-enable the XML writing interfaces	7
4.3	Escaping macros inside XML data	7
4.4	Rounding Interface	7
5	Adding data to the XML	8
5.1	General Invoicing Data	8
5.1.1	Invoice number/document ID	8
5.1.2	Currency	9
5.1.3	Dates	9
5.1.4	Payment terms	9
5.1.5	Notes: Adding additional information	9
5.2	Trade parties	10
5.2.1	Buyer Reference	11
5.2.2	Document references	11
5.2.3	Payment Means	12
5.3	Variables which may be changed per invoice item	12
5.3.1	Units	13
5.3.2	Tax category and rate	13
6	Implementation	14
6.1	Preparation to write the .xml file	16
6.2	Number rounding	22
6.3	XML indentation	23
6.4	Invoice Items	38
6.4.1	SpecifiedLineTradeDelivery	39
	Change History	44
	Index	45

1 Quick start

This package is still in development and does not provide any validation. To ensure your invoice is created correctly you should also validate the output files. There are tools like the *Mustang Project* [7] providing an easy-to-use interface for the validation. In the appendix I will add some notes on my setup and how I use it within pipelines.

The bundle provides an example file called `DEMO-rechnung-zugferd.tex`. This includes a basic setup for a valid X-Rechnung currently matching Version 3.0.2 of the standard. Details on the requirements can be found in the documentation at [1].

1.1 Disclaimer concerning the zugferd-invoice Package

The included package `zugferd-invoice` is an example project which might match your own invoicing structure. It holds all the layout information which is static across all the invoices. This package is an example implementation and should not be used in production. It is published as a part of the documentation.

The idea is to create your own version of this package to use your own layout and internally load the `zugferd` package that way. Of course, it's possible to use a copy of this package within your personal setup. But the syntax used in the DEMO file may change, so you have to ensure yourself to be compatible with updates.

The interfaces for `zugferd` will hopefully stay the same. At least changes will be announced and build compatible during a deprecation period.

2 Package Options

The package supports a few fundamental settings. These have to be set when the package is loaded as they are used internally to setup the scheme or activate the XML mechanism.

`format=` `(xrechnung/xrechnung3.0/xrechnung2.3/basic/minimum)` (default: `xrechnung`)

`format` selects the scheme to be used for the `zugferd` invoice. Currently `xrechnung3.0`, `xrechnung2.3`, and the `basic` and `minimum` schemes are supported.

The value `xrechnung` is set as an alias to `xrechnung3.0` and will always use the latest version supported by `zugferd`.

This option also adjusts the file name which is used for embedding the XML. It will be called `factur-x.xml` for all formats but the `xrechnung` ones. In that case it will be `xrechnung.xml`.

`zugferd=` `(boolean)` (default: `true`)

This option can be used to deactivate the XML embedding. It would also disable the `write-xml` option. This can be used to create a package which can use the same structure to also create invoices without XML attachment. It can also be used with older \LaTeX releases than this package requires. There will be a warning, but the visible part should be okay.

`write-xml=` `(boolean)` (default: `true`)

Disable the XML output. This can be used if you want to create the XML attachment with different software than this package.

In that case you can either rename your file to `\jobname_zugferd.xml` or also adjust the `xml-file` option.

`xml-file=` `(filename)` (default: `\jobname_zugferd.xml`)

Adjust the file name of the created or loaded XML file.

The option `xrechnung` is only used internally to set the global parameters for all `xrechnung` variants.

`auto-exemption=` (*boolean*) (default: `true`)

`zugferd` tries to automatically add an exemption-reason for the most common VAT categories. In case a more specific reason is required this setting can be disabled and everything should be configured manually. See [subsection 5.3.2](#) for more explanation of this feature and the categories this applies to.

3 User Commands

The end user is only asked to set or access the data to be used by `zugferd`.

`\SetZUGFeRDData*{<key value list>}`

`\SetZUGFeRDData` The two modes of `\SetZUGFeRDData` control if the argument is expanded before the fields are set. Depending on the source of the data this might be necessary. Fields which are involved in the calculation will be expanded anyway, but the text fields will not, to support special characters.

`\InsertZUGFeRDData`

`\InsertZugferdData`

`\InsertZugferdData[<special mode option>]{<data-selection>}`

`ZUGFeRD` uses the same data as the XML file inside the PDF. To simplify the reuse of data this command is designed to simplify the access to data fields, for example:

```
\InsertZUGFeRDData{id}
{\InsertZUGFeRDData[set-today]{date}\today}
\InsertZUGFeRDData[AddressData]{seller}
```

As special modes the command currently supports the following:

By default `zugferd` tries to find the variable holding the data itself. First a token list is tried, afterwards a string. Global variables are preferred over local ones.

As the variable names may contain underscores and the option usually prefers dashes, dashes are converted to underscores for the detection.

`AddressData` Allows `seller` or `buyer` for the data selection. Will print the address, to be used in letters.

`set-today` For dates there also exists the variant which will not print the variable but parse the variable to be used as `\today`. Using this the date format can be controlled easier using the language setting of the document. Here you should take care to use it within a group to restore the real value of `\today` afterwards.

4 Commands for template authors

`ZUGFeRD` (*env.*) To simplify the structure of the wrapper package, `zugferd` provides an environment for the XML mechanism and does the attachment to the PDF file (of course only, if enabled, see [section 2](#)). This provides the public interface bundling some steps together to reduce maintenance effort for any template maintainer using this package. It also avoids the use of internal commands.

This environment opens the XML file using `\startWritingZUGFeRDxml` and afterwards writes the XML header including the File and Scheme information, the `ExchangedDocumentContext` and information of the `ExchangedDocument`. Notes will also be written within this step. Afterwards the environment should include all the mechanisms to write the invoice positions as well as summation.

At the end of the environment the footer is inserted, before the output stream is closed using `\stopWritingZUGFeRDxml`. Which also attaches the XML file to the PDF.

`\startWritingZUGFeRDxml` `\startWritingZUGFeRDxml` is opening the output stream for the XML file. It also adjusts the indentation. If `write-xml` is false, this option only opens a group to achieve the same structure in both modes.

`\stopWritingZUGFeRDxml` Here the output stream is closed and the XML file is attached. In case `write-xml` is not active, the attachment will be made if that's not deactivated separately using `zugferd`. It also ends the group started by `\startWritingZUGFeRDxml`.

4.1 Interfaces to write the XML contents

In case you are using `write-xml=true` (which is the default) You need to ensure to call the XML writing functions in the correct order. For example after setting the global invoice data, like it's done in the example file. The minimal example below would create a valid XML. The interface commands are described afterwards.

```
\begin{ZUGFeRD}
  \zugferd_write_Item:nnnnnn {1} {} {Plushie~\TeX\ lion} {31.89} {2}
  ↪ {63.78}
  \zugferd_startInvoiceSums:
  \zugferd_write_TaxEntry:nnnn {S} {19} {63.78} {12.12}
  \zugferd_write_Summation:nnnnnnnn {63.78} {0} {0} {63.78} {12.12}
  ↪ {75.90} {0} {75.90}
  \zugferd_stopInvoiceSums:
\end{ZUGFeRD}
```

`\zugferd_write_Item:nnnnnn` This command is the interface to write invoice items to the XML file. If the XML interface is enabled this is a reference to the internal command `__zugferd_insert_TradeLineItem:nnnnnn`.

```
\zugferd_write_Item:nnnnn
  {\LineID}{\optional: item id ("SellerAssignedID")}{\item name}
  {\NetPriceProductTradePrice}
  {\BilledQuantity}
  {\LineTotalAmount}
```

Within the product name macros are disabled using `\zugferd_disable_macros:`, see [subsection 4.3](#).

This command is using the local values of tax information as well as the unit code. If you want to overwrite them, adjust them locally using the corresponding options, e.g.:

```
\group_begin:
  \keys_set:nn {zugferd/item}{tax/rate=19, tax/category=S}
  \zugferd_write_Item:nnnnnn {1} {} {Plushie \TeX\ lion} {31.89} {2}
  ↪ {63.78}
  % Code using the data for visual representation
\group_end:
```

This will set the tax rate to 19% unregarding the global setting.

This structure might look a bit overcomplicated as one might think the options could also be set as an additional argument. This works as long as the Code for the visual part of the invoice is not referencing the internal data. In case you don't do this it's also possible to use the following variant:

`\zugferd_write_Item:nnnnnnn` This is grouping the command and adding an argument in front to add additional options.
`\zugferd_write_Item:ennnnnn`

```
\zugferd_write_Item:nnnnn
  {\additional local options}
  {\LineID}{\optional: item id ("SellerAssignedID")}{\item name}
  {\NetPriceProductTradePrice}
  {\BilledQuantity}
  {\LineTotalAmount}
```

The example above could then be replaced by

```
\zugferd_write_Item:nnnnnnn {tax/rate=19,tax/category=S} {1} {Plushie-01}
→ {Plushie \TeX\ lion} {31.89} {2} {63.78}
% visual representation now may not refer to the data
```

`\zugferd_startInvoiceSums:` There is some global data which is placed in the XML file after the invoice items have been placed. Typically, in L^AT_EX this block is started after the items have been printed and will enclose the summation block.

The starting includes the so called “ApplicableHeaderTradeAgreement” which contains the address data of both trade parties, see [subsection 5.2](#) And this will also print the “SpecifiedTradeSettlementPaymentMeans”, see [subsubsection 5.2.3](#).

`\zugferd_write_TaxEntry:nnnn` This command is writing the sum over a tax rate. This command has to be used `\zugferd_write_TaxEntry:ennnnonce` per rate applied to the items.

```
\zugferd_write_TaxEntry:nnnn {\tax category code} {\tax rate in %} {\basis
amount the tax applies to} {\tax amount}
```

The tax amount could of course be calculated internally. In the example package this is done automatically, but the interface needs to support manual input as a lot of use cases for L^AT_EX invoicing use it only to create the output file.

`\zugferd_write_Summation:nnnnnnnn`

The total values are all collected with a single macro.

```
\zugferd_write_Summation:nnnnnnnn
  {\LineTotalAmount}{\ChargeTotalAmount}{\AllowanceTotalAmount}
  {\TaxBasisTotalAmount}{\TaxTotalAmount}
  {\GrandTotalAmount}{\TotalPrepaidAmount}{\DuePayableAmount}
```

This commnd is also writing the payment terms to the XML file. Please be aware that it's in general not possible to calculate the tax values in here, as there might be multiple tax rates applied. This is only taking the sums over all tax entries.

In case you are using some specials like category “E” the exemption reason will also be written at that point. For that it is referencing the current value of the setting.

4.2 Commands to temporary disable/re-enable the XML writing interfaces

`\zugferd_enable_XML_interfaces:`
`\zugferd_disable_XML_interfaces:`

As there are a lot of usecases where code is processed multiple times, it's necessary to provide an interface to temporary disable the XML writing mechanism. A lot of these situations appear within table structures whereas a local adjustment would not be helpful. Therefore these adjustments have to be done globally.

The example package `zugferd-invoice` provides an example for this to ensure the XML data is not written multiple times. The ZUGFeRD environment has been constructed that way, that it would automatically enable the interface when it begins and also when it ends, to write the data. So you should ensure this environment is only processed once or use the lower level interfaces directly. Setting up the catcodes to simplify the XML indentation.

4.3 Escaping macros inside XML data

`\zugferd_disable_macros:` Since we allow the use of \LaTeX code in some fields there has to be a mechanism to disable macros inside the XML output. The mechanism is created similar to the one by `hyperref`, and we also use some definitions from there to use those as a starting point. To have a detailed list of the redefinition, please have a look at the implementation of this command.

There exists a hook to extend or overwrite these definitions `zugferd/disable-macros`. You can add own redefinitions using this. For example if you want to overwrite the setting mapping a `\newline` to a new line char instead of space, you could add the following to your setup:

```
\hook_gput_code:nnn {zugferd/disable-macros}
  {newline-to-LF}
  {\def\n newline{\iow_newline:}}
```

`\zugferd_tl_set_escaped_xml:Nn` As described before and also in [#9](#) there are characters which are special in XML but
`\zugferd_tl_gset_escaped_xml:Nn` not within \LaTeX . Additionally there also are characters which aren't special in \LaTeX but within XML. In version v0.9c `zugferd` added some auxiliary commands to support users escaping those. This mechanism involves turning the category codes of characters like "`<>'&`" to active and might have side effects. It is an experimental feature and should be handled with care!

4.4 Rounding Interface

The demo package's implementation is also doing VAT calculations. This rounding mechanism might have side effects if only the final values are rounded as reported in [#17](#). It is only an issue if fractions of units are used, but was fixed in v0.9a. To avoid this generally there now exists an interface to be used to do the rounding before the summation.

`\zugferd_fp_set_rounded:Nn` This command can be used to access the `siunitx`'s rounding mechanism within `zugferd`.
`\zugferd_fp_gset_rounded:Nn` The command is based on `\fp_set:Nn` just is doing the rounding before the assignment.

```
\zugferd_fp_gset_rounded:Nn \g_tmpa_fp {(\amount)* ((VAT rate)/100) * \unit
\to price}}
```

5 Adding data to the XML

All data which does not directly depend on amounts or specific items is provided using a key-value interface. For some fields there is the option to define a global preset but locally overwrite it for a specific item. This only applies to data fields used by the writing interfaces described in [subsection 4.1](#).

This package is using the UN/CEFACT Cross Industry Invoice Syntax for the data. Currently it is not planned to implement the UBL syntax as well, but generally this would be possible.

Please be aware that the `zugferd` package does currently not handle any replacements concerning the content. Therefore it might be necessary to escape special characters, like `&` to `&`. This also applies to `<`, `>`, `"` and `'`. It's technically possible to do this either via active characters or string replacements. But since it's adjusting the content this feature would never be enabled by default (Issue [#9](#)).

In most cases this functionality will be used to change the tax setting or unit for a single item. [subsection 4.1](#) also provided an example for this.

This section will now take all data which can be set using `\SetZUGFeRDData`.

5.1 General Invoicing Data

Some of the general data currently supports only one value, which is already selected by default. The interface already exists and may be extended later.

`document-type=` (*<type>*) (default: `commercial-invoice`)

Select the document type. Since v0.9c `zugferd` supports the following types:

`commercial-invoice` 380

`partial-invoice` 326

`corrected-invoice` 384

`self-billed-invoice` 389

`credit-note` 381

`partial-construction-invoice` 875

`partial-final-construction-invoice` 876

`final-construction-invoice` 877

5.1.1 Invoice number/document ID

`id=` (`komavar`/*<document ID/invoice number>*) (*<initially unset>*)

This has to be set. Leaving it empty will lead to an invalid XML file.

The value `komavar` would reference the data provided the KOMA-Script letter variable `invoice`. In case you don't use `scrletter` you should not use this setting. More information can be found in the documentation [\[4\]](#).

5.1.2 Currency

`currency= (EUR/USD/CHF/€)` (default: EUR)

Currently zugferd only supports one currency for an invoice. This might be extended later. The currency is pre-configured to use Euro.

5.1.3 Dates

`date= (auto/<date formatted as YYYYMMDD>)` (default: auto)
`delivery-date= (auto/<date formatted as YYYYMMDD>)` (default: auto)
`due-date= <date formatted as YYYYMMDD>` (initially unset)

Currently there are three kinds of dates implemented. The XML-Standard requires them to use the structure `<YYYYMMDD>`. For the day this document was compiled this would be: “20241208” (December 8, 2024).

Instead of providing a date value directly it’s also possible to use `\today`. This is done using the `which` which is the default setting for `date` and `delivery-date`. Please be aware, that this would change if you rebuild the document later. So you might want to use an actual value here.

5.1.4 Payment terms

`payment-terms= (<string>)` (initially unset)

One option to set payment terms is the `due-date` mentioned before. If this is not set or the setting is more complex one can use `payment-terms` to add more information.

This setting is a string. In case there is expansion required this has to be done before.

5.1.5 Notes: Adding additional information

`subject= (komavar/<Tokenlist>)` (initially unset)
`fromaddress= (komavar/<Tokenlist>)` (initially unset)
`add-note= <Tokenlist>` (initially unset)

The ZUGFeRD example files^[11] use all visible data to add them to the XML as a note. `subject` and `fromaddress` are used to support this. The data should not be too relevant but zugferd want’s to support adding additional data to the XML using the note element. So these fields can be left out but in case they are not empty, they will also be written to the XML.

The `which` corresponds to the mechanism provided by `scletter`. It accesses the variable expands it to be used directly. If you don’t use this package, you can ignore this setting or add content manually.

5.2 Trade parties

The XML scheme knows 6 different Trade Parties:

- Seller
- Buyer
- Payee
- ShipTo
- SellerTaxRepresentative

Currently zugferd supports only Buyer, Seller and ShipTo, but can be easily extended to support the others as well. The data for each party follows the same structure, except the “BuyerReference” which is described later in this section.

Some of the data is optional for specific parties. As this also depends on the selected scheme and version we will not list the details. All fields for a trade party can be set using the “group” named by the party. For example setting all the seller data is done in the following listing:

```
\SetZUGFeRDData{
  seller/id = {ID - usually internal ID provided by buyer},
  seller/name = {peiTeX (Marei Peischl)},
  seller/legal-description = {Additional legal information},
  seller/legal-id = {legal registration ID},
  seller/trading-name = {trading name},
  seller/email = {invoicing@peitex.de},
  seller/vatid = {DE123456789},
  seller/contact= {Marei\\+4900000000\\marei@peitex.de},
  seller/address = {Address Line 1\\Address Line 2\\Address Line 3},
  seller/postcode = {20253},
  seller/city = {Hamburg},
  seller/country = {DE},
}
```

All this data is saved within a property list, which is internally called `\g__zugferd_<seller/buyer/shipto>_prop`. By default this property list is empty. The users themselves have to ensure to add the required data.

The outer braces are not required, if the data does not contain an equal sign or a comma. In case the final data is unknown, it’s recommended to use them anyway.

```
<party>/name= <name>                                <initially unset>
<party>/email= <email address>                       <initially unset>
<party>/vatid= <VAT ID>                               <initially unset>
<party>/taxid= <VAT ID>                               <initially unset>
<party>/address= <address>                           <initially unset>
```

As shown in the example `address` can use three lines separated by `\\`. It’s possible to set all fields for all trade contacts, but e. g. for the `shipto`-party email and vatid will not be used in the XML.

Alternatively it’s also possible to use `<party>/lineone`, `<party>/linetwo` and `<party>/linethree` separately. This may be helpful if you use a custom input format. In any way you should

ensure that all macros used within the data either are expandable or disabled using `\zugferd_disable_macros:`.

```

<party>/postcode= <postal code> <initially unset>
  <party>/city= <city> <initially unset>
<party>/subdivision= <subdivision> <initially unset>
  <party>/country= <country code> <initially unset>

```

The two letter country codes allowed here can be found in [2].

```

<party>/contact= <Combined contact data> <initially unset>

```

The contact person can either be set using the combined structure similar to `<party>/address`. It either consists of 3 or 4 entries, depending on if a department should be used or not.

```

\SetZUGFeRDData{
  seller/contact = {
    <contact-name>\\
    <contact-phone>\\
    <contact-email>
  },
  seller/contact = {
    <contact-name>\\
    <contact-department>\\
    <contact-phone>\\
    <contact-email>
  }
}

```

As for `seller/address` it's also possible to set the keys directly:

```

\SetZUGFeRDData{
  seller/contact-name= {<contact-name>},
  seller/contact-department = {<contact-department>},
  seller/contact-phone = {<contact-phone>},
  seller/contact-email= {<contact-email>}
}

```

5.2.1 Buyer Reference

```

buyer/reference= (komavar/<Reference>) <initially unset>

```

The reference field only exists for the `buyer` trade party. Depending on the process it's required to use some unique identifier referring to the `buyer`. Within Germany these numbers are called "Leitweg-ID"[6].

In any way the `buyer` may choose what is used here. Also may be some PO number or similar reference.

As defined for other variables the `reference` can also use the `value` to refer to the value of `komavar yourref`[4].

5.2.2 Document references

Additionally to the general buyer reference there may be additional data used by the buyer a reference. These fields are technically optional, but the buyer may enforce them to be used. These were not supported before v0.9c.

5.2.3 Payment Means

The payment means are selected by numeric codes. Currently we support:

- 1 = Instrument not defined
- 10 = In cash
- 30 = Credit Transfer
- 31 = Debit Transfer
- 42 = Payment to bank account
- 48 = Bank card
- 49 = Direct Debit
- 57 = Standing agreement
- 58 = SEPA credit transfer
- 59 = SEPA direct debit
- 97 = Clearing between partners

Others may be added in the future but it's not planned to include a full list.

The codes will automatically add the corresponding string inside the “Information” field. The initial version only included German strings, but currently they are also included in English. It's possible to overwrite them using the same structure:

```
\setupZUGFeRDStrings{payment-means}{  
  10 = Bargeld,  
  58 = Zahlung per SEPA Überweisung.,  
}
```

The language selection is done using at hook executed at `\begin{document}` and will try to use the document's language. If this is not defined English will be used.

Internally the commands are predefined as a key-value list like the argument in the example above. They macros are called `\zugferd@paymentMeans@language`. Currently `zugferd` defines these for `english` and `german` (also `ngerman` as a compatibility alias).

5.3 Variables which may be changed per invoice item

Some settings may have the same value for all invoice items. These are defined to take some preset but are set locally. So it's possible to adjust them for a single invoice item if necessary. An example is shown in [subsection 4.1](#).

5.3.1 Units

`unit= (hour/day/one/piece/(unit code))` *(initially unset)*

The Faktur-X standard requires the unit to be selected. These are called “/UN/CEFACT Common Codes” and can be found within [9].

Currently `zugferd` supports `hour` (HUR), `day` (DAY), `one` (C62) and `piece` (H87). For these the corresponding codes have been implemented within the package. Other units can be selected using the codes listed in [9].

This option is not case sensitive. The value is automatically converted to uppercase. If the selected option is different from the predefined ones, there will be a warning, as `zugferd` does not know if the selection is valid or not.

5.3.2 Tax category and rate

`tax/category= (category code/alias)` (default: `standard`)

The Tax data requires a category code. For details have a look at the Specification [e.g. at 5]. `zugferd` implements all of those, but the user has to take care to select the correct one for each invoice item. The example file includes 2 different VAT values using the same category.

The labels have been chosen to simplify the usage. It's also possible to enter the codes directly. This option is not case sensitive.

`standard` Standard rate and reduced rate item, `category=S`

`zero` Zero rated sale, `category=Z`

`exempt` Exempted from VAT. This requires a reason via `exemption-reason,category=E`

`reverse-charge` Reverse Charge, `category=AE`

`intra-community` Intra-Community Supply, `category=K`
or `EEA`

`export` Free export item, tax not charged, `category=G`

`0` Services outside scope of tax

`canary-islands` Canary Islands general indirect tax, `category=L`

`ceuta` Ceuta and Melilla, `category=M`
or `melilla`

`tax/exemption-reason= (Text)` *(initially unset)*

`tax/exemption-reason-code= (exemption reason code)` *(initially unset)*

Add Reasons for a tax exempt, as required by `category=E,K,AE,G,O`. This can either be added using a text (`exemption-reason`) or a predefined code (`exemption-reason-code`). The codes are listed at [10].

In most common cases `zugferd` tries to automatically match them if the package option `auto-exemption` is enabled, which is the default. In that case the following settings would apply:

`s` Exemption reason: *(empty)*; Exemption reason code: *(empty)*

- Z Not configured.
- E Not configured, as there are too many options.
- AE Exemption reason: Reverse Charge; Exemption reason code: vatex-eu-ae
- K Exemption reason: Intra-Community Supply; Exemption reason code: vatex-eu-ic
- G Exemption reason: Export outside the EU; Exemption reason code: vatex-eu-g
- O Exemption reason: No subject to VAT; Exemption reason code: vatex-eu-o

In case there is no pre-configured selection `zugferd` will create a warning to remind the user to add a selection themselves.

`tax/rate=` *<floating point>* (default: 19)

The value given will be used for tax calculation. By default it's configured to 19 to match the German standard VAT rate.

`item/start-date=` *<date formatted as YYYYMMDD>* *<initially unset>*

`item/end-date=` *<date formatted as YYYYMMDD>* *<initially unset>*

With version v0.9 support for `BillingSpecifiedPeriod` was added. This supports setting `start-date` and `end-date` per item. As this is optional, there is no default. The element will be printed if both dates are set, as setting a single one will enforce the element to be invalid. This element should be set as all the other dates (see [subsection 5.1.3](#)).

6 Implementation

```

\l__zugferd_tmp_tl
\g__zugferd_format_str      1 \tl_new:N \l__zugferd_tmp_tl
\g__zugferd_businessProcessId_str 2 \str_new:N \g__zugferd_format_str
\g__zugferd_writeTradeContact_bool 3 \str_new:N \g__zugferd_businessProcessId_str
\g__zugferd_writePaymentMeans_bool 4 \bool_new:N \g__zugferd_writeTradeContact_bool
\g__zugferd_minimum_bool     5 \bool_new:N \g__zugferd_writePaymentMeans_bool
\g__zugferd_conformance_level_str 6 \bool_new:N \g__zugferd_minimum_bool
\g__zugferd_xml_embedded_file_tl 7 \str_new:N \g__zugferd_conformance_level_str
                               8 \tl_new:N \g__zugferd_xml_embedded_file_tl

```

(End of definition for \l__zugferd_tmp_tl and others.)

```

format
xrechnung      9 \char_set_catcode_other:N \#%
write-xml     10 \keys_define:nn {zugferd} {
  zugferd    11 xrechnung .code:n = {
    xml-file 12 \bool_gset_true:N \g__zugferd_writeTradeContact_bool
  auto-exemption 13 \bool_gset_true:N \g__zugferd_writePaymentMeans_bool
                14 \str_gset:Nn \g__zugferd_conformance_level_str {XRECHNUNG}
                15 \tl_gset:Nn \g__zugferd_xml_embedded_file_tl {xrechnung.xml}
                16 },
                17 format .choice:,
                18 format / xrechnung3.0 .code:n = {
                19 \str_gset:Nx \g__zugferd_format_str {
                20 urn:cen.eu:en16931:2017#compliant#urn:xeinkauf.de:kosit:xrechnung_3.0
                21 }

```

```

22  \str_gset:Nx \g__zugferd_businessProcessId_str {
23    urn:fdc:peppol.eu:2017:poacc:billing:01:1.0
24  }
25  \keys_set:nn {zugferd}{xrechnung}
26 },
27 format / xrechnung2.3 .code:n = {
28   \str_gset:Nx \g__zugferd_format_str {
29     urn:cen.eu:en16931:2017#compliant#urn:xoev-de:kosit:standard:xrechnung_2.3
30   }
31   \keys_set:nn {zugferd}{xrechnung}
32 },
33 format / basic .code:n = {
34   \str_gset:Nx \g__zugferd_format_str {
35     urn:cen.eu:en16931:2017#compliant#urn:factur-x.eu:1p0:basic
36   }
37   \bool_gset_false:N \g__zugferd_writeTradeContact_bool
38   \bool_gset_false:N \g__zugferd_writePaymentMeans_bool
39   \str_gset:Nn \g__zugferd_conformance_level_str {BASIC}
40   \tl_gset:Nn \g__zugferd_xml_embedded_file_tl {factur-x.xml}
41 },
42 format / minimum .code:n = {
43   \str_gset:Nx \g__zugferd_format_str {
44     urn:factur-x.eu:1p0:minimum
45   }
46   \bool_gset_true:N \g__zugferd_minimum_bool
47   \bool_gset_false:N \g__zugferd_writeTradeContact_bool
48   \bool_gset_false:N \g__zugferd_writePaymentMeans_bool
49   \str_gset:Nn \g__zugferd_conformance_level_str {MINIMUM}
50   \tl_gset:Nn \g__zugferd_xml_embedded_file_tl {factur-x.xml}
51 },
52 format / xrechnung .meta:n = { format = xrechnung3.0 },
53 format .initial:n = xrechnung,
54 format .usage:n = load,
55 write-xml .bool_gset:N = \g__zugferd_write_xml_bool,
56 write-xml .initial:n = true,
57 write-xml .usage:n = load,
58 zugferd .bool_gset:N = \g__zugferd_active_bool,
59 zugferd .initial:n = true,
60 zugferd .default:n = true,
61 zugferd .usage:n = load,
62 ZUGFerD .meta:n = {zugferd = #1},
63 xml-file .tl_gset:N = \g__zugferd_xml_file_tl,
64 xml-file .initial:n = \jobname_zugferd.xml,
65 xml-file .usage:n = load,
66 auto-exemption .bool_gset:N = \g__zugferd_auto_exemption_bool,
67 auto-exemption .initial:n = true,
68 auto-exemption .default:n = true,
69 auto-exemption .usage:n =load,
70 }
71 \char_set_catcode_parameter:N \##%
72
73 \ProcessKeyOptions[zugferd]

```

6.1 Preparation to write the .xml file

_zugferd_xml_writer_iow

```
74 \iow_new:N \_zugferd_xml_writer_iow
```

(End of definition for _zugferd_xml_writer_iow.)

To adjust the metadata it is necessary to use the `pdfmanagement-testphase` by `pdfmanagement-testphase`. She had prepared some experiment files for the PDF attachment in the experiments of the repository. We use these to embed the XML file.

This part prepares the XMP metadata according to the required scheme.

```
75 \bool_if:NT \g__zugferd_active_bool {
76 \cs_if_exist:NF \pdfmeta_xmp_xmlns_new:nn {
77 \msg_new:nnnn {zugferd} {PDFmanagement-not-active} {
78 The~\LaTeX-PDF-management-is-not-active.\\
79 Activate-it-using~\string\DocumentMetadata.
80 } {
81 See~ZUGFeRD-or-PDFmanagement-documentation-for-more-information.
82 }
83 \msg_error:nn {zugferd} {PDFmanagement-not-active}
84 }

85 \IfPackageAtLeastTF{pdfmanagement-testphase}{2024/09/13}{}{
86 \msg_new:nnn {zugferd} {PDFmanagement-too-old} {
87 Your~version~of~\LaTeX's~PDF~management~is~too~old.
88 You~need~to~update~your~LaTeX~distribution~to~be~able~to~use~the~zugferd~package~correctl
89 }
90 \msg_error:nn{zugferd} {PDFmanagement-too-old}
91 }
92 %% based on experiments for l3pdfmeta by Ulrike Fischer
93 \pdfmeta_xmp_xmlns_new:nn {fx}{
94 urn:factur-x:pdfa:CrossIndustryDocument:invoice:1p0\c_hash_str
95 }
96
97 \pdfmeta_xmp_schema_new:nnn
98 {Factur-X-PDFA-Extension-Schema}
99 {fx}
100 {urn:factur-x:pdfa:CrossIndustryDocument:invoice:1p0\c_hash_str}
101
102 \pdfmeta_xmp_property_new:nnnnn
103 {fx}
104 {DocumentFileName}
105 {Text}
106 {external}
107 {name-of-the-embedded-XML-invoice-file}
108
109 \pdfmeta_xmp_property_new:nnnnn
110 {fx}
111 {DocumentType}
112 {Text}
113 {external}
114 {INVOICE}
115
116 \pdfmeta_xmp_property_new:nnnnn
117 {fx}
```



```

118     {Version}
119     {Text}
120     {external}
121     {The~actual~version~of~the~factur~x~schema}
122
123 \pdfmeta_xmp_property_new:nnnnn
124     {fx}
125     {ConformanceLevel}
126     {Text}
127     {external}
128     {The~conformance~level~of~the~factur~x~data}
129
130 \exp_args:Ne \pdfmeta_xmp_add:n {
131     % fix INVOICE
132     <fx:DocumentType>INVOICE</fx:DocumentType>\iow_newline:
133     % fix factur-x.xml
134     <fx:DocumentFileName>\g__zugferd_xml_embedded_file_tl</fx:DocumentFileName>\iow_newline:
135     % fix schema version
136     <fx:Version>1.0</fx:Version>\iow_newline:
137     % zulässige Werte MINIMUM, BASIC WL, BASIC, EN 16931, EXTENDED, XRECHNUNG
138     <fx:ConformanceLevel>\g__zugferd_conformance_level_str</fx:ConformanceLevel>%
139     \iow_newline:
140     %
141     }
142 }

```

\SetZUGFeRDData

```

143 \NewDocumentCommand{\SetZUGFeRDData}{sm}{
144     \IfBooleanTF{#1}
145     {\keys_set:ne}
146     {\keys_set:nn}
147     {zugferd} {#2}
148     }
149 \let\SetZugferdData\SetZUGFeRDData

```

`\InsertZUGFeRDData` To simplify the usage in letters we also add fields to be able to use the zugferd data
`\InsertZugferdData` within L^AT_EX output. Country is still missing.

```

150 \NewDocumentCommand{\InsertZUGFeRDData}{om}{
151   \str_case:nnF {#1} {
152     {AddressData} {
153       \clist_map_inline:nn {name, lineone, linetwo} {
154         \prop_if_in:cnT {g__zugferd_#2_AddressData_prop } {##1} {
155           \prop_item:cn {g__zugferd_#2_AddressData_prop } {##1}\
156         }
157       }
158       \prop_item:cn {g__zugferd_#2_AddressData_prop } {postcode}
159       \space
160       \prop_item:cn {g__zugferd_#2_AddressData_prop } {city}
161     }
162     {set-today} {
163       \__zugferd_set_today:v {g__zugferd_#2_tl}
164     }
165   } {

```

Try to find the variable automatically.

- replace dashes by underscores
- try if a tokenlist or a string
- prefer global over local
- take the first existing variable and break the loop

```

166   \str_set:Ne \l_tmpa_str {#2}
167   \str_replace_all:Nnn \l_tmpa_str {-} {_}
168   \str_replace_all:Nnn \l_tmpa_str {/} {_}
169   \bool_set_true:N \g_tmpa_bool
170   \clist_map_inline:nn {tl, str} {
171     \clist_map_inline:nn {g, l} {
172       \use:c {##1_if_exist:cT} {####1__zugferd_ \l_tmpa_str _##1}
173       {
174         \use:c {####1__zugferd_ \l_tmpa_str _##1}
175         \bool_gset_false:N \g_tmpa_bool
176       }
177       \bool_if:NF \g_tmpa_bool {\clist_map_break:}
178     }
179   \bool_if:NF \g_tmpa_bool {\clist_map_break:}
180 }
181 }
182 }
183 \providecommand{\InsertZugferdData}{\InsertZUGFeRDData}
184 \providecommand{\insertZugferdData}{\InsertZUGFeRDData}

```

_zugferd_set_today:c Auxiliary function to use a date variable within the current group to be used as \today.

```

185 \cs_new:Nn \_zugferd_set_today:n {
186 \_zugferd_set_today_aux:w #1 \q_stop
187 }
188 \cs_generate_variant:Nn \_zugferd_set_today:n {v}
189 \cs_new:Npn \_zugferd_set_today_aux:w #1 #2 #3 #4 #5 #6 #7 #8 \q_stop{
190 \int_set:Nn \year {#1#2#3#4}
191 \int_set:Nn \month {#5#6}
192 \int_set:Nn \day {#7#8}
193 }

194 \NewDocumentEnvironment{ZUGFeRD}{o}{
195 \IfNoValueF{#1}{
196 \SetZUGFeRDData{#1}
197 }
198 \zugferd_enable_XML_interfaces:
199 \startWritingZUGFeRDxml
200 \zugferd_write_Header:
201 \ignorespaces
202 }{
203 \zugferd_enable_XML_interfaces:
204 \zugferd_write_Footer:
205 \stopWritingZUGFeRDxml
206 \zugferd_disable_XML_interfaces:
207 }

208 \newcommand*{\startWritingZUGFeRDxml}{
209 \begingroup
210 \bool_if:NTF \g__zugferd_write_xml_bool {
211 \char_set_active_eq:nN {13} \_zugferd_xml_newline_indent:
212 \iow_open:Nn \_zugferd_xml_writer_iow {\g__zugferd_xml_file_tl}
213 }{
214 \msg_info:nn {zugferd} {no-xml-write}
215 }
216 }
217 \msg_new:nnn {zugferd} {no-xml-write} {
218 The-option-write-xml=false-was-set.\
219 Writing-of-XML-file-is-deactivated.
220 }

```

The PDF attachment is done after the writing stream is closed.

```

221 %% The metadata elements are taken by Ulrike Fischer's faktur-x experiments
222 %% https://github.com/latex3/pdfresources/tree/main/experiments/faktur-x-bills
223 \newcommand*{\stopWritingZUGFeRDxml}{%
224 \bool_if:NT \g__zugferd_write_xml_bool
225 {\iow_close:N \_zugferd_xml_writer_iow}%
226 \endgroup
227 \bool_if:NT \g__zugferd_active_bool {
228 \group_begin:
229 \pdfdict_put:nnn {l_pdffile/Filespec} {AFRelationship}{/Alternative}
230 %or /Source in some cases
231 \pdfdict_put:nnn {l_pdffile/Filespec} {Desc}{(Faktur-X/ZUGFeRD-Rechnung)}
232 \exp_args:NnV \pdffile_embed_file:nnn {\g__zugferd_xml_file_tl} \g__zugferd_xml_embedded
233 {zugferd/rechnung}

```

```
234 \group_end:
```

Since some tools require the Catalog entry to match the file name of the embedded file this this has to be set separately (#26).

```
235 \pdfmanagement_add:nee
236   {Catalog/Names/EmbeddedFiles}
237   {\g__zugferd_xml_embedded_file_t1}
238   {\pdf_object_ref:n{zugferd/rechnung}}
239   % steht in der doku ist aber pdf 2.0 ....
240 \pdfmanagement_add:nx{Catalog}{AF}{\pdf_object_ref:n{zugferd/rechnung}}
241 }
242 }
```

Provide public interfaces and the ZUGFeRD environment.

\zugferd_enable_XML_interfaces:
\zugferd_disable_XML_interfaces:

```
243 \cs_new:Nn \zugferd_enable_XML_interfaces: {
244 \bool_if:NT \g__zugferd_write_xml_bool {
245 \cs_gset:Nn \zugferd_write_Header: {
246 \__zugferd_insert_Header:
247 \__zugferd_insert_FrontMatter:
248 }
249 \cs_gset:Nn \zugferd_write_Footer: {
250 \__zugferd_insert_Footer:
251 }
252 \bool_if:NTF \g__zugferd_minimum_bool {
253 \cs_gset_eq:NN \zugferd_write_Item:nnnnnn \use_none:nnnnnn
254 }{
255 \cs_gset_eq:NN \zugferd_write_Item:nnnnnn
256 \__zugferd_insert_TradeLineItem:nnnnnn
257 }
258 \cs_gset:Nn \zugferd_startInvoiceSums: {
259 \__zugferd_ApplicableHeaderTradeAgreement:
260 \__zugferd_ApplicableHeaderTradeSettlement_start:
261 \__zugferd_SpecifiedTradeSettlementPaymentMeans:
262 }
263 \cs_gset:Nn \zugferd_stopInvoiceSums: {
264 \__zugferd_ApplicableHeaderTradeSettlement_stop:
265 }
266 \bool_if:NTF \g__zugferd_minimum_bool {
267 \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \use_none:nnnn
268 } {
269 \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \__zugferd_ApplicableTradeTax:nnnn
270 }
271 \cs_gset:Nn \zugferd_write_Summation:nnnnnnnn {
272 \__zugferd_SpecifiedTradePaymentTerms:
273 \__zugferd_SpecifiedTradeSettlementHeaderMonetarySummation:nnnnnnnn
274 {##1} {##2} {##3} {##4} {##5} {##6} {##7} {##8}
275 }
276 }
277 }
278 \cs_new:Nn \zugferd_disable_XML_interfaces: {
279 \cs_gset_eq:NN \zugferd_write_Header: \prg_do_nothing:
280 \cs_gset_eq:NN \zugferd_write_Footer: \prg_do_nothing:
281 \cs_gset_eq:NN \zugferd_write_Item:nnnnnn \use_none:nnnnnn
282 \cs_gset_eq:NN \zugferd_startInvoiceSums: \prg_do_nothing:
283 \cs_gset_eq:NN \zugferd_stopInvoiceSums: \prg_do_nothing:
284 \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \use_none:nnnn
285 \cs_gset_eq:NN \zugferd_write_Summation:nnnnnnnn \use_none:nnnnnnnn
286 }
287 \bool_if:NTF \g__zugferd_write_xml_bool {
288 \zugferd_enable_XML_interfaces:
289 }{
290 \zugferd_disable_XML_interfaces:
291 }
```

`\zugferd_write_Item:nnnnnnn`

```
292 \cs_new:Nn \zugferd_write_Item:nnnnnnn {
293   \group_begin:
294   \keys_set:nn {zugferd/item} {#1}
295   \zugferd_write_Item:nnnnnnn {#2} {#3} {#4} {#5} {#6} {#7}
296   \group_end:
297 }
298 \cs_generate_variant:Nn \zugferd_write_Item:nnnnnnn {ennnnnn}
```

`\zugferd_write_TaxEntry:ennn`

```
299 \cs_generate_variant:Nn \zugferd_write_TaxEntry:nnnn {ennn}
```

`__zugferd_write_xml:n`

`__zugferd_define_xml_writer:Nn`

`__zugferd_define_xml_content:Nn`

These commands are used to toggle the writing of the XML file. This corresponds to the option `write-xml`.

```
300 \bool_if:NTF \g__zugferd_write_xml_bool {
301   \cs_new:Nn \__zugferd_write_xml:n {
302     \iow_now:Ne \__zugferd_xml_writer_iow {\__zugferd_xml_auto_indent: #1}
303   }
304   \cs_new_eq:NN \__zugferd_define_xml_writer:Nn \cs_new:Nn
305   \cs_new_eq:NN \__zugferd_define_xml_content:Nn \cs_new:Nn
306 } {
307   \cs_set_eq:NN \__zugferd_write_xml:n \use_none:n
308   \cs_set:Nn \__zugferd_define_xml_writer:Nn {\cs_new:Nn #1 {}}
309   \cs_set:Nn \__zugferd_define_xml_content:Nn {\cs_new:Nn #1 {}}
310 }
311
312 \cs_generate_variant:Nn \__zugferd_write_xml:n {e}
```

6.2 Number rounding

As `siunitx` is implementing this, `zugferd` uses it instead of building our own mechanism. In version v0.9a a public interface was added, as described in section 4.4.

```
313 \RequirePackage{siunitx}
```

```
\__zugferd_number_format:nNn
\__zugferd_number_format:nNe
```

```
314 \cs_new:Nn \__zugferd_number_format:nNn {
315   \sisetup{
316     parse-numbers=true,
317     round-mode=places,
318     round-precision=#1,
319     round-pad = false,
320     group-digits=none,
321     minimum-decimal-digits=#1,
322     output-decimal-marker=.
323   }
324   \siunitx_number_format:nN {#3} #2
325 }
326 \cs_generate_variant:Nn \__zugferd_number_format:nNn {nNe}
```

```
\__zugferd_write_rounded:nnnn
\__zugferd_write_rounded:nnn
```

```
327 \cs_new:Nn \__zugferd_write_rounded:nnnn {
328   \__zugferd_number_format:nNe {#1} \l__zugferd_tmp_tl {#4}
329   \__zugferd_write_xml:e {<ram:#2#3>\l__zugferd_tmp_tl</ram:#2>}
330 }
331 \cs_new:Nn \__zugferd_write_rounded:nnn {
332   \__zugferd_write_rounded:nnnn {#1} {#2} {} {#3}
333 }
```

6.3 XML indentation

The indentation of the XML does not really matter. For debugging, it's a lot simpler to have it included and this also helped to maintain the structure of the code during development, so I decided to keep it. The indentation is created using a bunch of auxiliary commands and variables which are defined here.

```
\g__zugferd_indent_int
```

```
334 \int_new:N \g__zugferd_indent_int
(End of definition for \g__zugferd_indent_int.)
```

```
\__zugferd_indent:
\__zugferd_xml_auto_indent:
\__zugferd_xml_newline_indent:
```

```
335 \cs_new:Nn \__zugferd_indent: {
336   \space\space
337 }
338 \cs_new:Nn \__zugferd_indent:n {
339   \prg_replicate:nn {#1} {\__zugferd_indent:}
340 }
341 \cs_new:Nn \__zugferd_xml_auto_indent: {
342   \__zugferd_indent:n {\g__zugferd_indent_int}
343 }
```

The idea was to redefine the `\newlinechar` to automatically indent the following line.

```

344 \cs_new:Nn \__zugferd_xml_newline_indent: {
345 \iow_newline: \__zugferd_xml_auto_indent:
346 }

```

Setting up the catcodes to simplify the XML indentation.

\zugferd_disable_macros:

The definition was mostly taken from `hyperref` [[hyperref](#)]. Most likely not all of these are required, but it's probably easier to take this as a reasonable choice instead of creating an own collection.

```

347 \cs_new:Nn \zugferd_disable_macros: {
348 \let\{\c_left_brace_str
349 \let\textbraceleft\c_left_brace_str
350 \let\}\c_right_brace_str
351 \let\textbraceright\c_right_brace_str
352 \let\#\c_hash_str

```

This only is a part of the list. There is no real use of printing the whole list, it's inside `zugferd.sty` anyway.

```

353 \tl_new:N \g__zugferd_tmp_tl
354 \cs_new:Nn \zugferd_tl_gset_escaped_xml:Nn {
355 \group_begin:
356 \zugferd_disable_macros:
357 \__zugferd_escape_xml_aux:
358 \tl_gset_rescan:Nnn #1 {
359 \cctab_select:N \c_str_cctab
360 \__zugferd_escape_xml_activate:
361 } {#2}
362 \tl_gset:Ne #1 {#1}
363 \group_end:
364 }
365 \cs_new:Nn \zugferd_tl_set_escaped_xml:Nn {
366 \zugferd_tl_gset_escaped_xml:Nn \g__zugferd_tmp_tl {#2}
367 \tl_set_eq:NN #1 \g__zugferd_tmp_tl
368 }
369 \cs_new:Nn \__zugferd_escape_xml_aux: {
370 \edef\&\tl_to_str:n {\&}}
371 \edef\textgreater{\tl_to_str:n {\>}}
372 \edef\textless{\tl_to_str:n {\<}}
373 \edef\textquotedbl{\tl_to_str:n {\"}}
374 \edef\textquotesingle{\tl_to_str:n {\&apos;}}
375 \char_set_active_eq:NN > \textgreater
376 \char_set_active_eq:NN < \textless
377 \char_set_active_eq:NN " \textquotedbl
378 \char_set_active_eq:NN ' \textquotesingle
379 \char_set_active_eq:NN & \&
380 }
381 \cs_new:Nn \__zugferd_escape_xml_activate: {
382 \char_set_catcode_active:N >
383 \char_set_catcode_active:N <
384 \char_set_catcode_active:N "
385 \char_set_catcode_active:N '

```



```

386 \char_set_catcode_active:N &
387 }

```

```

\uzugferd_fp_set_rounded:Nn
\uzugferd_fp_gset_rounded:Nn

```

```

388 % \begin{function}{\zugferd_fp_set_rounded:Nn, \zugferd_fp_gset_rounded:Nn}
389 \cs_set:Nn \zugferd_fp_set_rounded:Nn {
390 \__zugferd_number_format:nNn {2} \l__zugferd_tmp_tl {\fp_eval:n {#2}}
391 \fp_set:Nn #1 {\l__zugferd_tmp_tl }
392 }
393 \cs_set:Nn \zugferd_fp_gset_rounded:Nn {
394 \__zugferd_number_format:nNn {2} \l__zugferd_tmp_tl {\fp_eval:n {#2}}
395 \fp_gset:Nn #1 {\l__zugferd_tmp_tl }
396 }

```

```

\g__zugferd_notes_seq
\g__zugferd_id_tl
\g__zugferd_date_tl
\g__zugferd_subject_tl
\g__zugferd_fromaddress_tl
\g__zugferd_DocumentTypeCode_tl

```

```

397 \seq_new:N \g__zugferd_notes_seq
398 \tl_new:N \g__zugferd_id_tl
399 \tl_new:N \g__zugferd_date_tl
400 \tl_new:N \g__zugferd_delivery_date_tl
401 \tl_new:N \g__zugferd_subject_tl
402 \tl_new:N \g__zugferd_fromaddress_tl
403 \tl_new:N \g__zugferd_DocumentTypeCode_tl
404 \tl_new:N \l__zugferd_currency_tl

```

(End of definition for \g__zugferd_notes_seq and others.)

```

405 \keys_define:nn { zugferd } {
406 %TODO define others

```

```

id
document-type

```

```

407 document-type .choice:,
408 document-type / commercial-invoice .code:n =
409 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {380},
410 document-type / partial-invoice .code:n =
411 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {326},
412 document-type / corrected-invoice .code:n =
413 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {384},
414 document-type / self-billed-invoice .code:n =
415 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {389},
416 document-type / credit-note .code:n =
417 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {381},
418 document-type / partial-construction-invoice .code:n =
419 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {875},
420 document-type / partial-final-construction-invoice .code:n =
421 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {876},
422 document-type / final-construction-invoice .code:n =
423 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {877},
424 document-type .initial:n = commercial-invoice,
425 id .choice:,
426 id / komavar .code:n = \tl_gset:Nf \g__zugferd_id_tl {\scr@invoice@var},
427 id / unknown .code:n = \tl_gset:Nn \g__zugferd_id_tl {#1},

```

```

428 currency .choices:nn = {EUR, USD, CHF} {
429   \tl_set_eq:NN \l__zugferd_currency_tl \l_keys_choice_tl
430 },
431 currency / € .meta:n = {currency = EUR},
432 currency / unknown .code:n = {
433   \exp_args:Nnnx \keys_if_choice_exist:nnnTF {zugferd} {unit} {
434     \str_uppercase:f {#1}
435   }
436   {\keys_set:nx { zugferd } {unit= {\str_uppercase:f {#1}}}}
437   {
438     \msg_warning:nnn { zugferd } {unknown-value} {currency} {#1}
439     \tl_set:Ne \l__zugferd_currency_tl {\str_uppercase:f {#1}}
440   }
441 },
442 currency .initial:n = EUR,

date
delivery-date 443 date .choice:,
444 date / auto .code:n = \tl_gset:Ne \g__zugferd_date_tl {
445   \the\year
446   \int_compare:nNnT {\month} < {10} {0} \the\month
447   \int_compare:nNnT {\day} < {10} {0}\the\day
448 },
449 date / unknown .code:n = \tl_gset:Nn \g__zugferd_date_tl {#1},
450 date .initial:n = auto,
451 delivery-date .choice:,
452 delivery-date / auto .code:n = \tl_gset:Ne \g__zugferd_delivery_date_tl {
453   \the\year
454   \int_compare:nNnT {\month} < {10} {0} \the\month
455   \int_compare:nNnT {\day} < {10} {0}\the\day
456 },
457 delivery-date / unknown .code:n = {
458   \tl_gset:Nn \g__zugferd_delivery_date_tl {#1}
459 },
460 delivery-date .initial:n = auto,
461 due-date .tl_gset:N = \g__zugferd_due_date_tl,
462 due-date .initial:n =,

payment-terms
463 payment-terms .str_gset:N = \g__zugferd_payment_terms_str,
464 payment-terms .initial:n =,

subject
fromaddress
add-note 465 subject .choice:,
466 subject / komavar .code:n = {
467   \tl_gset:Nf \g__zugferd_subject_tl {\scr@subject@var}
468 },
469 subject / unknown .code:n = \tl_gset:Nn \g__zugferd_subject_tl {#1},
470 fromaddress .choice:,
471 fromaddress / komavar .code:n = \tl_gset:Nf \g__zugferd_fromaddress_tl
472   {\scr@fromaddress@var},
473 fromaddress / unknown .code:n = \tl_gset:Nn \g__zugferd_fromaddress_tl {#1},
474 add-note .code:n = \seq_gput_right:Nn \g__zugferd_notes_seq {#1},
475 }

```

```

476 \msg_new:nnnn {zugferd} {unknown-value} {
477 You-selected-a-#1-which-was-not-predefined.\\
478 I-will-directly-use-your-selection-'#1=#2'.
479 Please-ensure-the-selection-is-valid!
480 } {
481 For-more-information-see-the-zugferd-documentation.
482 }

```

Macro to write notes

```

483 \cs_new:Nn \__zugferd_note_if_not_empty:N {%
484 \tl_if_empty:NF #1 {%
485 \__zugferd_write_note:n {#1}%
486 }%
487 }%

```

ApplicableHeaderTradeAgreement

Contains information on seller and buyer trade party:

- BuyerReference
- SellerTradeParty
- BuyerTradeParty

`\g__zugferd__metaparty__address` Seller and Buyer are specified the same way.

```

\g__zugferd__metaparty__address 488 \clist_map_inline:nn {seller,buyer,shipto} {
\g__zugferd__buyer__AddressData__prop 489 \prop_new:c {g__zugferd__#1__AddressData__prop}
\g__zugferd__shipto__AddressData__prop 490 \keys_define:nn {zugferd / #1} {
491 address .code:n = {
492 \seq_set_split:Nnn \l_tmpa_seq {\} {##1}
493 %TODO error if more than 2 lines or only 1
494 \keys_set:nx {zugferd / #1} {
495 lineone=\seq_item:Nn \l_tmpa_seq {1},
496 linetwo=\seq_item:Nn \l_tmpa_seq {2},
497 linethree=\seq_item:Nn \l_tmpa_seq {3}
498 }
499 },
500 contact .code:n = {
501 \seq_set_split:Nnn \l_tmpa_seq {\} {##1}
502 \int_compare:nNnTF {\seq_count:N \l_tmpa_seq} > {3} {
503 \keys_set:ne {zugferd/#1} {
504 contact-name = \seq_item:Nn \l_tmpa_seq {1},
505 contact-department = \seq_item:Nn \l_tmpa_seq {2},
506 contact-phone = \seq_item:Nn \l_tmpa_seq {3},
507 contact-email= \seq_item:Nn \l_tmpa_seq {4}
508 }
509 }{
510 \keys_set:ne {zugferd/#1} {
511 contact-name = \seq_item:Nn \l_tmpa_seq {1},
512 contact-phone = \seq_item:Nn \l_tmpa_seq {2},
513 contact-email= \seq_item:Nn \l_tmpa_seq {3}
514 }
515 }
516 },
517 unknown .code:n = \tl_if_blank:nF {##1} {
518 \prop_gput:cVe {g__zugferd__#1__AddressData__prop}

```

```

519     \l_keys_key_tl {\tl_trim_spaces:n {##1}}
520   }
521 }
522 }

```

(End of definition for \g__zugferd_seller_AddressData_prop, \g__zugferd_buyer_AddressData_prop, and \g__zugferd_shipto_AddressData_prop.)

```

\__zugferd_TradePartyIdentity:N \__zugferd_PostalTradeAddress:N
\__zugferd_DefinedTradeContact:N

```

Wrappers to map the property list items to the writing macro.

```

523 \cs_new:Nn \__zugferd_TradePartyIdentity:N {
524   \__zugferd_TradePartyIdentity:eeeeee
525   {\prop_item:Nn #1 {id}}
526   {\prop_item:Nn #1 {name}}
527   {\prop_item:Nn #1 {legal-description}}
528   {\prop_item:Nn #1 {legal-id}}
529   {\prop_item:Nn #1 {trading-name}}
530 }
531 \cs_new:Nn \__zugferd_PostalTradeAddress_short:N {
532   \exp_args:Ne \tl_if_blank:nF {\prop_item:Nn #1 {name}}
533   {<ram:Name>\prop_item:Nn #1 {name}</ram:Name>\iow_newline:\__zugferd_xml_auto_indent:}
534   \__zugferd_PostalTradeAddress:eeeeeee
535   {\prop_item:Nn #1 {postcode}}
536   {\prop_item:Nn #1 {lineone}}
537   {\prop_item:Nn #1 {linetwo}}
538   {\prop_item:Nn #1 {linethree}}
539   {\prop_item:Nn #1 {city}}
540   {\prop_item:Nn #1 {country}}
541   {\prop_item:Nn #1 {subdivision}}
542 }
543 \cs_new:Nn \__zugferd_PostalTradeAddress:N {%
544   \__zugferd_PostalTradeAddress:eeeeeee
545   {\prop_item:Nn #1 {postcode}}
546   {\prop_item:Nn #1 {lineone}}
547   {\prop_item:Nn #1 {linetwo}}
548   {\prop_item:Nn #1 {linethree}}
549   {\prop_item:Nn #1 {city}}
550   {\prop_item:Nn #1 {country}}
551   {\prop_item:Nn #1 {subdivision}}
552   \__zugferd_UniversalCommunication:n {\prop_item:Nn #1 {email}}
553   \__zugferd_TaxRegistration:nn {\prop_item:Nn #1 {vaid}} {\prop_item:Nn #1 {taxid}}
554 }%
555 \cs_new:Nn \__zugferd_DefinedTradeContact:N {%
556   \__zugferd_DefinedTradeContact:eeee
557   {\prop_item:Nn #1 {contact-name}}
558   {\prop_item:Nn #1 {contact-department}}
559   {\prop_item:Nn #1 {contact-phone}}
560   {\prop_item:Nn #1 {contact-email}}
561 }%
562 \cctab_begin:N \g__zugferd_xml_cctab%
563 \__zugferd_define_xml_content:Nn \__zugferd_TradePartyIdentity:nmnnn {%

```

```

564 %%<!-- BT-29 -->
565 \__zugferd_write_inline:nn {ram:ID} {#1}
566 %%<!-- BT-27 -->
567 \__zugferd_write_inline:nn {ram:Name} {#2}
568 %%<!-- BT-33 -->
569 \__zugferd_write_inline:nn {ram:Description} {#3}
570 \tl_if_blank:eF {#4#5} {%
571 <ram:SpecifiedLegalOrganization>
572 %%<!-- BT-30 -->
573 \__zugferd_write_inline_i:nnn {ram:ID} {~schemeID="0088"} {#4}
574 %%<!-- BT-28 -->
575 \__zugferd_write_inline_i:nn {ram:TradingBusinessName} {#5}
576 </ram:SpecifiedLegalOrganization>
577 }%
578 }%

579 \__zugferd_define_xml_content:Nn \__zugferd_PostalTradeAddress:nnnnnnn {%
580 <ram:PostalTradeAddress>
581 \bool_if:NF \g__zugferd_minimum_bool {%
582 \__zugferd_indent: <ram:PostcodeCode>#1</ram:PostcodeCode>
583 \tl_if_blank:nF {#2} {%
584 \__zugferd_indent: <ram:LineOne>#2</ram:LineOne>
585 }%
586 \tl_if_blank:nF {#3} {%
587 \__zugferd_indent: <ram:LineTwo>#3</ram:LineTwo>
588 }%
589 \tl_if_blank:nF {#4} {%
590 \__zugferd_indent: <ram:LineThree>#4</ram:LineThree>
591 }%
592 \__zugferd_indent: <ram:CityName>#5</ram:CityName>
593 }%
594 \__zugferd_indent: <ram:CountryID>#6</ram:CountryID>
595 \bool_if:NF \g__zugferd_minimum_bool {%
596 \tl_if_blank:nF {#7} {%
597 \__zugferd_indent: <ram:CountrySubDivisionName>#7</ram:CountrySubDivisionName>
598 }%
599 }%
600 </ram:PostalTradeAddress>%
601 }%

602 \bool_if:NTF \g__zugferd_minimum_bool {%
603 \cs_set_eq:NN \__zugferd_UniversalCommunication:n \use_none:n%
604 } {%
605 \__zugferd_define_xml_content:Nn \__zugferd_UniversalCommunication:n {%
606 \tl_if_blank:eF {#1} {
607 <ram:URIUniversalCommunication>
608 \__zugferd_indent: <ram:URIID~schemeID="EM">#1</ram:URIID>
609 </ram:URIUniversalCommunication>%
610 }%
611 }%
612 }%

613 \__zugferd_define_xml_content:Nn \__zugferd_TaxRegistration:nn {%
614 %%<!-- BT-34 -->
615 \tl_if_blank:eF {#1} {
616 <ram:SpecifiedTaxRegistration>
617 \__zugferd_indent: <ram:ID~schemeID="VA">#1</ram:ID>

```

```

618 </ram:SpecifiedTaxRegistration>%
619 }%
620 %%%<!-- BT-32 -->
621 \tl_if_blank:eF {#2} {
622 <ram:SpecifiedTaxRegistration>
623 \__zugferd_indent: <ram:ID~schemeID="FC">#2</ram:ID>
624 </ram:SpecifiedTaxRegistration>%
625 }%
626 }%
627 %
628 % Contact data phone/email to a specific contact person
629 \__zugferd_define_xml_content:Nn \__zugferd_DefinedTradeContact:nnnn {%
630 % Do not print if name is empty
631 \tl_if_blank:nT {#1} {use_none:nnn} %
632 \bool_if:NT \g__zugferd_writeTradeContact_bool {%
633 <ram:DefinedTradeContact>
634 \__zugferd_indent: <ram:PersonName>#1</ram:PersonName>
635 \tl_if_blank:nF {#2} {%
636 \__zugferd_indent: <ram:DepartmentName>#2</ram:DepartmentName>
637 }%
638 \tl_if_blank:nF {#3} {%
639 \__zugferd_indent: <ram:TelephoneUniversalCommunication>
640 \__zugferd_indent:n {2} <ram:CompleteNumber>#3</ram:CompleteNumber>
641 \__zugferd_indent: </ram:TelephoneUniversalCommunication>
642 }%
643 \tl_if_blank:nF {#4} {%
644 \__zugferd_indent: <ram:EmailURIUniversalCommunication>
645 \__zugferd_indent:n {2} <ram:URIID>#4</ram:URIID>
646 \__zugferd_indent: </ram:EmailURIUniversalCommunication>
647 }%
648 </ram:DefinedTradeContact>%
649 }%
650 }%
651 %
652 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeAgreement: {%
653 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeAgreement>}%
654 \int_gincr:N \g__zugferd_indent_int%
655 \bool_lazy_and:nnF%
656 {\tl_if_blank_p:V \g__zugferd_buyer_reference_tl}%
657 {\g__zugferd_minimum_bool}%
658 {%
659 \__zugferd_write_xml:e {%
660 <ram:BuyerReference>\g__zugferd_buyer_reference_tl</ram:BuyerReference>
661 }%
662 }%
663 \__zugferd_write_xml:n {<ram:SellerTradeParty>}%
664 \int_gincr:N \g__zugferd_indent_int%
665 \__zugferd_write_xml:e {%
666 \__zugferd_TradePartyIdentity:N \g__zugferd_seller_AddressData_prop
667 \__zugferd_DefinedTradeContact:N \g__zugferd_seller_AddressData_prop%
668 \__zugferd_PostalTradeAddress:N \g__zugferd_seller_AddressData_prop%
669 }%
670 \int_gdecr:N \g__zugferd_indent_int%
671 \__zugferd_write_xml:e {%

```

```

672 </ram:SellerTradeParty>
673 <ram:BuyerTradeParty>%
674 }%
675 \int_gincr:N \g__zugferd_indent_int%
676 \__zugferd_write_xml:e {%
677 \__zugferd_TradePartyIdentity:N \g__zugferd_buyer_AddressData_prop
678 \bool_if:NF \g__zugferd_minimum_bool {
679 \__zugferd_DefinedTradeContact:N \g__zugferd_buyer_AddressData_prop%
680 \__zugferd_PostalTradeAddress:N \g__zugferd_buyer_AddressData_prop%
681 }%
682 }%
683 \int_gdecr:N \g__zugferd_indent_int%
684 \__zugferd_write_xml:n {</ram:BuyerTradeParty>}%
685 \clist_map_inline:nn {SellerOrder,BuyerOrder,Contract} {%
686 \__zugferd_write_ReferencedDocument:n {##1}%
687 }%
688 \int_gdecr:N \g__zugferd_indent_int%
689 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeAgreement>}%
690 \bool_if:NNTF \g__zugferd_minimum_bool%
691 { \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeDelivery/>} }%
692 { \__zugferd_ApplicableHeaderTradeDelivery:V \g__zugferd_delivery_date_tl}%
693 }%
694 %
695 %
696 % delivery date
697 \__zugferd_define_xml_content:Nn \__zugferd_ApplicableHeaderTradeDelivery:n {%
698 \bool_lazy_and:nnF {\prop_if_empty_p:N \g__zugferd_shipto_AddressData_prop}%
699 {\tl_if_blank_p:n {#1}}%
700 {%
701 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeDelivery>}%
702 \prop_if_empty:NF \g__zugferd_shipto_AddressData_prop {%
703 \int_gincr:N \g__zugferd_indent_int%
704 \__zugferd_write_xml:n {<ram:ShipToTradeParty>}%
705 \int_gincr:N \g__zugferd_indent_int%
706 \__zugferd_write_xml:e {%
707 \__zugferd_PostalTradeAddress_short:N \g__zugferd_shipto_AddressData_prop%
708 }%
709 \int_gdecr:N \g__zugferd_indent_int%
710 \__zugferd_write_xml:n {</ram:ShipToTradeParty>}%
711 \int_gdecr:N \g__zugferd_indent_int%
712 }%
713 \tl_if_blank:nF {#1} {%
714 \__zugferd_write_xml:n {%
715 \__zugferd_indent: <ram:ActualDeliverySupplyChainEvent>
716 \__zugferd_indent:n {2} <ram:OccurrenceDateTime>
717 \__zugferd_indent:n {3}<udt:DateTimeString~format="102">#1</udt:DateTimeString>
718 \__zugferd_indent:n {2} </ram:OccurrenceDateTime>
719 \__zugferd_indent: </ram:ActualDeliverySupplyChainEvent>%
720 }%
721 }%
722 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeDelivery>}%
723 }%
724 }%
725 %

```

```

726 \cctab_end:
727 \cs_generate_variant:Nn \__zugferd_TradePartyIdentity:nnnnn {eeee}
728 \cs_generate_variant:Nn \__zugferd_DefinedTradeContact:nnnn {eeee}
729 \cs_generate_variant:Nn \__zugferd_PostalTradeAddress:nnnnnn {eeeeee}
730 \cs_generate_variant:Nn \__zugferd_ApplicableHeaderTradeDelivery:n {V}

```

buyer/reference

```

731 \tl_new:N \g__zugferd_buyer_reference_tl
732 \keys_define:nn {zugferd/buyer} {
733   reference .choice:,
734   reference / komavar .code:n = {
735     \tl_gset:Nf \g__zugferd_buyer_reference_tl {\scr@yourref@var}
736   },
737   reference / unknown .code:n = {
738     \tl_gset:Nn \g__zugferd_buyer_reference_tl {#1}
739   }
740 }

741 \prop_new:N \g__zugferd_document_references_prop
742 \clist_new:N \g__zugferd_buyer_document_references_clist
743 \keys_define:nn {zugferd} {
744   contract-reference .prop_gput:N = \g__zugferd_document_references_prop,
745   purchaseorder-reference .prop_gput:N = \g__zugferd_document_references_prop,
746   purchase-order-reference .meta:n = {purchaseorder-reference=#1},
747   salesorder-reference .prop_gput:N = \g__zugferd_document_references_prop,
748   sales-order-reference .meta:n = {salesorder-reference=#1},
749 }
750 \cctab_begin:N \g__zugferd_xml_cctab%
751 \__zugferd_define_xml_writer:Nn \__zugferd_write_ReferencedDocument:n {%
752   \prop_if_in:NeT \g__zugferd_document_references_prop {\str_lowercase:n {#1}-reference} {%
753     \__zugferd_write_xml:e {%
754       <ram:#1ReferencedDocument>
755         \__zugferd_indent:<ram:IssuerAssignedID>\prop_item:Ne \g__zugferd_document_references_prop
756         reference}</ram:IssuerAssignedID>
757       }%
758     }%
759   }%
760 \cctab_end:

```

payment-means

```

761 \ExplSyntaxOff
762 \providecommand*{\zugferd@paymentMeans@german}{
763   1 = Keine Zahlungsart definiert,
764   10 = Bargeld,
765   30 = Überweisung,
766   42 = Zahlung an Bankkonto,
767   48 = Kartenzahlung,
768   49 = Lastschriftverfahren,
769   57 = Dauerauftrag,
770   58 = Zahlung per SEPA Überweisung.,
771   59 = SEPA Lastschrift,
772   97 = Ausgleich zwischen Partnern
773 }

```



```

774 \def\zugferd@paymentMeans@ngerman{\zugferd@paymentMeans@german}
775 \providecommand*\zugferd@paymentMeans@english{
776   1 = Instrument not defined,
777   10 = In cash,
778   30 = Credit Transfer,
779   31 = Debit Transfer,
780   42 = Payment to bank account,
781   48 = Bank card,
782   49 = Direct Debit,
783   57 = Standing agreement,
784   58 = SEPA credit transfer,
785   59 = SEPA direct debit,
786   97 = Clearing between partners
787 }
788 \ExplSyntaxOn
789 \tl_new:N \g__zugferd_payment_means_tl
790 \keys_define:nn {zugferd} {
791   payment-means / type .choices:nn = {1,10,30,42,48,49,57,58,59,97} {
792     \tl_gset_eq:NN \g__zugferd_payment_means_tl \l_keys_choice_tl
793   },
794   payment-means / unknown .code:n = {
795     \msg_warning:nnn { zugferd } {unknown-value} {payment-means} {#1}
796     \tl_gset:Nn \g__zugferd_payment_means_tl {\int_eval:n {#1}}
797   }
798 }
799
800 \clist_map_inline:nn {iban,account-holder,bic} {
801   \keys_define:nn {zugferd/payment-means} {
802     #1 .tl_gset:c = {g__zugferd_payment_#1_tl}
803   }
804 }
805
806 \prop_new:c {g__zugferd_payment-means_names_prop}
807 \newcommand*\setupZUGFeRDStrings}[2]{
808   \prop_gset_from_keyval:cn {g__zugferd_#1_names_prop} {
809     #2
810   }
811 }
812
813 \hook_gput_code:nnn {begindocument/end}{zugferd/payment-means}{
814   \prop_if_empty:cT {g__zugferd_payment-means_names_prop} {
815     \exp_args:Nne \setupZUGFeRDStrings{payment-means}{
816       \use:c {zugferd@paymentMeans@
817         \cs_if_exist:cTF {zugferd@paymentMeans@\languagenname} {\languagenname} {english}
818     }
819   }
820 }
821 }

```

unit

```

822 \keys_define:nn { zugferd } {
823   unit .choices:nn = {HUR,DAY,C62,H87} {
824     \tl_set_eq:NN \l__zugferd_unit_code_tl \l_keys_choice_tl
825   },

```

```

826 unit / hour .meta:n = {unit=HUR},
827 unit / day .meta:n = {unit=DAY},
828 unit / one .meta:n= {unit= C62},
829 unit / piece .meta:n = {unit=H87},

```

If unknown, the value is converted to uppercase and we use the selection directly. There is a warning in that case

```

830 unit / unknown .code:n = {
831   \tl_set:Nx \l_tmpa_tl {\str_uppercase:f {#1}}
832   \exp_args:NnnV \keys_if_choice_exist:nnnTF {zugferd} {unit}
833   \l_tmpa_tl
834   {
835     \keys_set:ne { zugferd } {unit= {\l_tmpa_tl}}
836   } {
837     \msg_warning:nnn { zugferd } {unknown-unit} {#1}
838     \tl_set_eq:NN \l__zugferd_unit_code_tl \l_tmpa_tl
839   }
840 },
841 unit .usage:n = general,
842 }
843
844 \msg_new:nnnn {zugferd} {unknown-unit} {
845   You~selected~a~unit~which~was~not~predefined.\\
846   I~will~directly~use~your~selection~'unit=#1'~as~Common~Code.\\
847   Please~ensure~the~selection~is~valid!
848 } {
849   For~more~information~see~the~zugferd~documentation\\
850   and~the~/UN/CEFACT~Common~Code~list.
851 }

```

category
exemption-reason
rate

```

852 \msg_new:nnnn {zugferd} {unknown-tax-category} {
853   You~selected~an~unknown~tax~category.\\
854   I~will~directly~use~your~selection~'category=#1'~as~Code.\\
855   Please~ensure~the~selection~is~valid!
856 } {
857   For~more~information~see~the~zugferd~documentation\\
858   and~the~corresponding~code~list.
859 }
860 \msg_new:nnnn {zugferd} {no-auto-exemption} {
861   You~selected~tax/category=#1~together~with~the~auto~exemption~option.\\
862   I~don't~have~any~pre~configured~exemption~setting~for~category=#1.\\
863   Please~make~sure~you~add~a~valid~setting~yourself.
864 } {
865   For~more~information~see~the~zugferd~documentation.
866 }
867 \keys_define:nn { zugferd / tax } {
868   category .choices:nn = {S,Z,E,AE,K,G,O,L,M} {
869     \tl_set_eq:NN \l__zugferd_tax_category_code_tl \l_keys_choice_tl
870     \bool_if:NT \g__zugferd_auto_exemption_bool {
871       \keys_if_choice_exist:nnnTF {zugferd/tax} {exemption-reason-auto} {#1} {
872         \keys_set:nn {zugferd/tax} {exemption-reason-auto=#1}
873       } {
874         \msg_warning:nnn { zugferd } {no-auto-exemption} {#1}

```

```

875 }
876 }
877 },
878 exemption-reason .tl_set:N = \l__zugferd_tax_exemption_reason_tl,
879 exemption-reason .initial:V = \c_empty_tl,
880 exemption-reason .usage:n = general,
881 exemption-reason-code .tl_set:N = \l__zugferd_tax_exemption_code_tl,
882 exemption-reason-code .initial:V = \c_empty_tl,
883 exemption-reason .usage:n = general,
884 exemption-reason-auto .choice:,
885 exemption-reason-auto / S .code:n = {
886   \keys_set:nn {zugferd/tax} {exemption-reason=,exemption-reason-code=}
887 },
888 exemption-reason-auto / K .code:n = {
889   \keys_set:nn {zugferd/tax} {
890     exemption-reason= Intra-Community-Supply,
891     exemption-reason-code={vatex-eu-ic}
892   }
893 },
894 exemption-reason-auto / AE .code:n = {
895   \keys_set:nn {zugferd/tax}{
896     exemption-reason=Reverse-Charge,
897     exemption-reason-code={vatex-eu-ae}
898   }
899 },
900 exemption-reason-auto / G .code:n = {
901   \keys_set:nn {zugferd/tax}{
902     exemption-reason=Export-outside-the-EU,
903     exemption-reason-code={vatex-eu-g}
904   }
905 },
906 exemption-reason-auto / O .code:n = {
907   \keys_set:nn {zugferd/tax}{
908     exemption-reason=No-subject-to-VAT,
909     exemption-reason-code={vatex-eu-o}
910   }
911 },
912 standard .meta:n = {category=S},
913 zero .meta:n = {category=Z},
914 exempt .meta:n = {category=E},
915 reverse-charge .meta:n = {category=AE},
916 intra-community .meta:n = {category=K},
917 EEA .meta:n = {category=K},
918 export .meta:n = {category=G},
919 canary-islands .meta:n = {category=L},
920 ceuta .meta:n = {category=M},
921 melilla .meta:n = {category=M},
922 category / unknown .code:n = {
923   \exp_args:Nnnx \keys_if_choice_exist:nmnTF {zugferd} {category}
924   {\str_uppercase:f {#1}}
925   {
926     \keys_set:nx { zugferd } {category= {\str_uppercase:f {#1}}}
927   } {

```

```

928   \msg_warning:nnn { zugferd } {unknown-tax-category} {#1}
929   }
930 },
931 category .initial:n = S,
932 category .usage:n = general,

933 rate .fp_set:N = \l__zugferd_tax_rate_fp,
934 rate .initial:n = 19,
935 rate .usage:n = general
936 }%
937 \keys_define:nn {zugferd/item} {
938   tax .choice:,
939   tax / unknown .code:n = \keys_set:ne {zugferd/tax} { \l_keys_key_str = \exp_not:V \l_keys_v
940   unknown .code:n = \keys_set:ne {zugferd} { \l_keys_key_str = \exp_not:V \l_keys_value_tl}
941 }

start-date
end-date
942 \keys_define:nn { zugferd / item } {
943   start-date .tl_gset:N = \l__zugferd_start_date_tl,
944   start-date .initial:n =,
945   end-date .tl_gset:N = \l__zugferd_end_date_tl,
946   end-date .initial:n =,
947 }%
948 \__zugferd_define_xml_content:Nn \__zugferd_write_inline:nn {
949   \__zugferd_write_inline:nnn {#1} {} {#2}
950 }
951 \__zugferd_define_xml_content:Nn \__zugferd_write_inline:nnn {
952   \tl_if_blank:nF {#3} {
953     <#1#2>#3</#1>
954   }
955 }
956 \cs_generate_variant:Nn \__zugferd_write_inline:nn {ne}
957 \__zugferd_define_xml_content:Nn \__zugferd_write_inline_i:nnn {
958   \__zugferd_indent: \__zugferd_write_inline:nnn {#1} {#2} {#3}
959 }
960 \__zugferd_define_xml_content:Nn \__zugferd_write_inline_i:nn {
961   \__zugferd_write_inline_i:nnn {#1} {} {#2}
962 }
963 \cctab_begin:N \g__zugferd_xml_cctab%
964 %
965 \bool_if:NTF \g__zugferd_minimum_bool {%
966   \cs_set_eq:NN \__zugferd_write_note:n \use_none:n%
967 } {%
968   \__zugferd_define_xml_writer:Nn \__zugferd_write_note:n {%
969     \begingroup%
970     \let\\\iow_newline:%
971     \__zugferd_write_xml:e {%
972       <ram:IncludedNote>
973       \__zugferd_indent: <ram:Content>
974       #1
975       \__zugferd_indent: </ram:Content>
976       </ram:IncludedNote>%
977     }%
978   \endgroup%

```

```

979 }%
980 }%
981 %
982 \beginigroup%
983 \char_set_catcode_other:N \#%
984 \char_set_catcode:nn {32}{10}%
985 \__zugferd_define_xml_writer:Nn \__zugferd_insert_Header: {%
986 \__zugferd_write_xml:e {%
987 <?xml version='1.0' encoding='UTF-8' ?>
988 <rsm:CrossIndustryInvoice %
989 xmlns:rsm="urn:un:unece:unefact:data:standard:CrossIndustryInvoice:100" %
990 xmlns:qdt="urn:un:unece:unefact:data:standard:QualifiedDataType:100" %
991 xmlns:ram="urn:un:unece:unefact:data:standard:ReusableAggregateBusinessInformationEntity:
992 xmlns:xs="http://www.w3.org/2001/XMLSchema" %
993 xmlns:udt="urn:un:unece:unefact:data:standard:UnqualifiedDataType:100">%
994 }%
995 \int_gincr:N \g__zugferd_indent_int%
996 \__zugferd_write_xml:n {<rsm:ExchangedDocumentContext>}%
997 \int_gincr:N \g__zugferd_indent_int%
998 \__zugferd_write_xml:e {%
999 \str_if_empty:NF \g__zugferd_businessProcessId_str {%
1000 <ram:BusinessProcessSpecifiedDocumentContextParameter>
1001 \__zugferd_indent: <ram:ID>\g__zugferd_businessProcessId_str</ram:ID>
1002 </ram:BusinessProcessSpecifiedDocumentContextParameter>
1003 }%
1004 <ram:GuidelineSpecifiedDocumentContextParameter>
1005 \__zugferd_indent: <ram:ID>\g__zugferd_format_str</ram:ID>
1006 </ram:GuidelineSpecifiedDocumentContextParameter>%
1007 }%
1008 \int_gdecr:N \g__zugferd_indent_int%
1009 \__zugferd_write_xml:n {</rsm:ExchangedDocumentContext>}%
1010 }%
1011 \endigroup%
1012 \__zugferd_define_xml_writer:Nn \__zugferd_insert_FrontMatter: {%
1013 \__zugferd_write_xml:n {<rsm:ExchangedDocument>}%
1014 \int_gincr:N \g__zugferd_indent_int%
1015 \__zugferd_write_xml:e {%
1016 <ram:ID>\g__zugferd_id_tl</ram:ID>
1017 <ram:TypeCode>\g__zugferd_DocumentTypeCode_tl</ram:TypeCode>
1018 <ram:IssueDateTime>
1019 % space required!
1020 \__zugferd_indent:<udt:DateTimeString~format="102">\g__zugferd_date_tl</udt:DateTimeString>
1021 </ram:IssueDateTime>%
1022 }%
1023 \__zugferd_note_if_not_empty:N \g__zugferd_subject_tl%
1024 \__zugferd_note_if_not_empty:N \g__zugferd_fromaddress_tl%
1025 \seq_map_inline:Nn \g__zugferd_notes_seq {%
1026 \__zugferd_write_note:n {##1}%
1027 }%
1028 \int_gdecr:N \g__zugferd_indent_int%
1029 \__zugferd_write_xml:e {%
1030 </rsm:ExchangedDocument>
1031 <rsm:SupplyChainTradeTransaction>%
1032 }%

```

```

1033 \int_gincr:N \g__zugferd_indent_int%
1034 }%
1035 %
1036 % footer
1037 \__zugferd_define_xml_writer:Nn \__zugferd_insert_Footer: {%
1038 \int_gdecr:N \g__zugferd_indent_int%
1039 \__zugferd_write_xml:n {</rsm:SupplyChainTradeTransaction>}%
1040 \int_gdecr:N \g__zugferd_indent_int%
1041 \__zugferd_write_xml:n {</rsm:CrossIndustryInvoice>}%
1042 }%
1043 \cctab_end:

```

6.4 Invoice Items

Each item consists of 5 parts:

- AssociatedDocumentLineDocument
- SpecifiedTradeProduct
- SpecifiedLineTradeAgreement
- SpecifiedLineTradeDelivery
- SpecifiedLineTradeSettlement

These are implemented as separate commands to be more flexible. The wrapper command is called `__zugferd_insert_TradeLineItem:nnnnn` and is created to be used in your own invoicing package-

```

1044 \cctab_begin:N \g__zugferd_xml_cctab%

```

```

\underline{\__zugferd_AssociatedDocumentLineDocument:n}

```

```

1045 \__zugferd_define_xml_writer:Nn \__zugferd_AssociatedDocumentLineDocument:n {%
1046 <ram:AssociatedDocumentLineDocument>
1047 \__zugferd_indent: <ram:LineID>#1</ram:LineID>
1048 </ram:AssociatedDocumentLineDocument>%
1049 }%

```

```

\underline{\__zugferd_SpecifiedTradeProduct:nn}

```

```

1050 \__zugferd_define_xml_content:Nn \__zugferd_SpecifiedTradeProduct:nn {%
1051 <ram:SpecifiedTradeProduct>
1052 \bool_lazy_or:nnF%
1053 {\tl_if_blank_p:n {#1}}%
1054 {\str_if_eq_p:Vn \g__zugferd_conformance_level_str {BASIC}} {%
1055 \__zugferd_indent: <ram:SellerAssignedID>#1</ram:SellerAssignedID>
1056 }%
1057 \__zugferd_indent: <ram:Name>#2</ram:Name>
1058 </ram:SpecifiedTradeProduct>%
1059 }%

```

_zugferd_ProductTradePrice:nn

```
1060 \\_zugferd_define_xml_writer:Nn \\_zugferd_ProductTradePrice:nn {%
1061 <ram:\str_uppercase:n #1PriceProductTradePrice>
1062 \\_zugferd_indent: <ram:ChargeAmount>#2</ram:ChargeAmount>
1063 </ram:\str_uppercase:n #1PriceProductTradePrice>%
1064 }%
```

_zugferd_SpecifiedLineTradeAgreement:nn

```
1065 \\_zugferd_define_xml_writer:Nn \\_zugferd_SpecifiedLineTradeAgreement:nn {%
1066 <ram:SpecifiedLineTradeAgreement>
1067 \\_zugferd_indent:<ram:GrossPriceProductTradePrice>
1068 \\_zugferd_indent:n {2} <ram:ChargeAmount>#1</ram:ChargeAmount>
1069 \\_zugferd_indent:</ram:GrossPriceProductTradePrice>
1070 \\_zugferd_indent:<ram:NetPriceProductTradePrice>
1071 \\_zugferd_indent:n {2} <ram:ChargeAmount>#2</ram:ChargeAmount>
1072 \\_zugferd_indent:</ram:NetPriceProductTradePrice>
1073 </ram:SpecifiedLineTradeAgreement>
1074 }%
```

6.4.1 SpecifiedLineTradeDelivery

:nn <unit code > <number>

_zugferd_SpecifiedLineTradeDelivery:nn

```
1075 \\_zugferd_define_xml_content:Nn \\_zugferd_SpecifiedLineTradeDelivery:nn {%
1076 <ram:SpecifiedLineTradeDelivery>
1077 % SPACE!
1078 \\_zugferd_indent: <ram:BilledQuantity~unitCode="#1">#2</ram:BilledQuantity>
1079 </ram:SpecifiedLineTradeDelivery>
1080 }%
```

_zugferd_SpecifiedLineTradeSettlement:nnn
_zugferd_SpecifiedLineTradeSettlement:Vnn

```
1081 \_zugferd_define_xml_writer:Nn \_zugferd_write_SpecifiedLineTradeSettlement:nnn {%
1082 \_zugferd_write_xml:n {<ram:SpecifiedLineTradeSettlement>}%
1083 \int_gincr:N \g_zugferd_indent_int%
1084 \_zugferd_write_xml:e {\_zugferd_Line_ApplicableTradeTax:nn {#1} {#2}}%
1085 \_zugferd_write_xml:e {\_zugferd_BillingSpecifiedPeriod:VV \l_zugferd_start_date_tl \l
1086 \_zugferd_write_xml:e {\_zugferd_SpecifiedTradeSettlementLineMonetarySummation:n {#3}}%
1087 \int_gdecr:N \g_zugferd_indent_int%
1088 \_zugferd_write_xml:n {</ram:SpecifiedLineTradeSettlement>}%
1089 }%
1090 \cs_generate_variant:Nn \_zugferd_write_SpecifiedLineTradeSettlement:nnn {Vnn}%
1091 \_zugferd_define_xml_content:Nn \_zugferd_Line_ApplicableTradeTax:nn {%
1092 <ram:ApplicableTradeTax>
1093 %BT-151
1094 \_zugferd_indent: <ram:TypeCode>VAT</ram:TypeCode>
1095 %BT-151
1096 \_zugferd_indent: <ram:CategoryCode>#1</ram:CategoryCode>
1097 %BT-152
1098 \_zugferd_indent: <ram:RateApplicablePercent>#2</ram:RateApplicablePercent>
1099 </ram:ApplicableTradeTax>
1100 }%
1101 \_zugferd_define_xml_content:Nn \_zugferd_BillingSpecifiedPeriod:nn {%
1102 \bool_lazy_or:nnF {\tl_if_blank_p:n {#1}} {\tl_if_blank_p:n {#2}} {%
1103 <ram:BillingSpecifiedPeriod>
1104 %<!-- BT-134 -->
1105 \_zugferd_indent:<ram:StartDateTime>
1106 \_zugferd_indent:n {2} <udt:DateTimeString~format="102">#1</udt:DateTimeString>
1107 \_zugferd_indent:</ram:StartDateTime>
1108 %<!-- BT-135 -->
1109 \_zugferd_indent:<ram:EndDateTime>
1110 \_zugferd_indent:n {2} <udt:DateTimeString~format="102">#2</udt:DateTimeString>
1111 \_zugferd_indent:</ram:EndDateTime>
1112 </ram:BillingSpecifiedPeriod>%
1113 }%
1114 }%
1115 \_zugferd_define_xml_content:Nn \_zugferd_SpecifiedTradeSettlementLineMonetarySummation:n
1116 <ram:SpecifiedTradeSettlementLineMonetarySummation>
1117 % BT-131
1118 \_zugferd_indent: <ram:LineTotalAmount>#1</ram:LineTotalAmount>
1119 </ram:SpecifiedTradeSettlementLineMonetarySummation>%
1120 }
1121
1122 \cctab_end:
1123 \cs_generate_variant:Nn \_zugferd_SpecifiedLineTradeSettlement:nnn {Vnn}
1124 \cs_generate_variant:Nn \_zugferd_BillingSpecifiedPeriod:nn {VV}
```

The exemption reason was placed wrong with the pre-CTAN release. This was fixed in August 2024. The old macro will be kept for a bit longer but will be removed soon. Please update your own implementations accordingly.

```
1125 \cs_new:Nn \_zugferd_SpecifiedLineTradeSettlement:nnnn {
1126 \msg_warning:nnnn {zugferd} {macro-deprecated}
```



```

1127 { \__zugferd_SpecifiedLineTradeSettlement:nnnn }
1128 { \__zugferd_SpecifiedLineTradeSettlement:nnn }
1129 \__zugferd_SpecifiedLineTradeSettlement:nnn {#2} {#3} {#4}
1130 }
1131 \cs_generate_variant:Nn \__zugferd_SpecifiedLineTradeSettlement:nnnn {VVnn}

```

```

\__zugferd_insert_TradeLineItem:nnnnnn
\__zugferd_write_TradeLineItem:nnnnnn

```

Writing an invoice item using the helper commands defined before, to the XML file.

```

1132 \__zugferd_define_xml_writer:Nn \__zugferd_insert_TradeLineItem:nnnnnn {
1133 \__zugferd_write_xml:n {<ram:IncludedSupplyChainTradeLineItem>
1134 \int_gincr:N \g__zugferd_indent_int
1135 \__zugferd_write_xml:e {\__zugferd_AssociatedDocumentLineDocument:n {#1}}%
1136
1137 \begingroup
1138 \zugferd_disable_macros:
1139 \__zugferd_write_xml:e {\__zugferd_SpecifiedTradeProduct:nn {#2} {#3}}
1140 \endgroup
1141
1142 \__zugferd_write_xml:n {<ram:SpecifiedLineTradeAgreement>
1143 \int_gincr:N \g__zugferd_indent_int%
1144 \__zugferd_write_xml:e {\__zugferd_ProductTradePrice:nn {net} {#4}}
1145 \int_gdecr:N \g__zugferd_indent_int%
1146 \__zugferd_write_xml:n {</ram:SpecifiedLineTradeAgreement>
1147 \__zugferd_write_xml:e {%
1148 \__zugferd_SpecifiedLineTradeDelivery:nn { \l__zugferd_unit_code_tl } {#5}
1149 }%
1150 \__zugferd_write_SpecifiedLineTradeSettlement:Vnn \l__zugferd_tax_category_code_tl %
1151 { \fp_use:N \l__zugferd_tax_rate_fp } {#6}%
1152 \int_gdecr:N \g__zugferd_indent_int%
1153 \__zugferd_write_xml:n {</ram:IncludedSupplyChainTradeLineItem>
1154 }

```

ApplicableHeaderTradeSettlement

```

1155 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeSettlementHeaderMonetarySummation:nnnnnn
1156 \__zugferd_write_xml:n {<ram:SpecifiedTradeSettlementHeaderMonetarySummation>}%
1157 \int_gincr:N \g__zugferd_indent_int%
1158 \bool_if:NF \g__zugferd_minimum_bool {
1159 \__zugferd_write_rounded:nnn {2} {LineTotalAmount} {#1}
1160 \__zugferd_write_rounded:nnn {2} {ChargeTotalAmount} {#2}
1161 \__zugferd_write_rounded:nnn {2} {AllowanceTotalAmount} {#3}
1162 }
1163 \__zugferd_write_rounded:nnn {2} {TaxBasisTotalAmount} {#4}
1164 \__zugferd_write_rounded:nnnn {2} {TaxTotalAmount} {~currencyID="\l__zugferd_currency_tl "}
1165 \__zugferd_write_rounded:nnn {2} {GrandTotalAmount} {#6}
1166 \bool_if:NF \g__zugferd_minimum_bool {
1167 \__zugferd_write_rounded:nnn {2} {TotalPrepaidAmount} {#7}
1168 }
1169 \__zugferd_write_rounded:nnn {2} {DuePayableAmount} {#8}
1170 \int_gdecr:N \g__zugferd_indent_int%
1171 \__zugferd_write_xml:n {</ram:SpecifiedTradeSettlementHeaderMonetarySummation>}%
1172 }%
1173 % ApplicableTradeTax CategoryCode Rate BaseAmount Result

```

```

1174 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableTradeTax:nmmn {%
1175 \__zugferd_write_xml:n {<ram:ApplicableTradeTax>}
1176 \int_gincr:N \g__zugferd_indent_int%
1177 \__zugferd_write_rounded:nnn {2} {CalculatedAmount} {#4}%BT117
1178 \__zugferd_write_xml:n {<ram:TypeCode>VAT</ram:TypeCode>}%BT118

1179 %
1180 \begingroup
1181 \bool_if:NT \g__zugferd_auto_exemption_bool {%
1182 \keys_set:nn { zugferd / tax } {exemption-reason-auto=#1}
1183 }%
1184 \tl_if_blank:VF \l__zugferd_tax_exemption_reason_tl {%BT-120
1185 \__zugferd_write_xml:e {<ram:ExemptionReason>
1186 \l__zugferd_tax_exemption_reason_tl
1187 </ram:ExemptionReason>}
1188 }
1189 {{\__zugferd_write_rounded:nnn {2} {BasisAmount} {#3}}}%BT-116
1190 \__zugferd_write_xml:e {<ram:CategoryCode>#1</ram:CategoryCode>}%BT-118

1191 \bool_lazy_or:nnF% BT121
1192 {\tl_if_blank_p:V \l__zugferd_tax_exemption_code_tl}%
1193 {\str_if_eq_p:Vn \g__zugferd_conformance_level_str {BASIC}}%
1194 {%
1195 \__zugferd_write_xml:e {%
1196 <ram:ExemptionReasonCode>\l__zugferd_tax_exemption_code_tl</ram:ExemptionReasonCode>%
1197 }%
1198 }%
1199 \endgroup
1200 \__zugferd_write_xml:n {<ram:RateApplicablePercent>#2</ram:RateApplicablePercent>}%BT-
1201 119
1202 \int_gdecr:N \g__zugferd_indent_int%
1203 \__zugferd_write_xml:n {</ram:ApplicableTradeTax>}
1204 }%
1205 %
1206 \cctab_begin:N \g__zugferd_xml_cctab%
1207 % ApplicableHeaderTradeSettlement

```

ApplicableHeaderTradeSettlement needs to be splitted to be used in separate parts of the tabular.

```

1207 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeSettlement_start: {%
1208 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeSettlement>}%
1209 \int_gincr:N \g__zugferd_indent_int%
1210 \__zugferd_write_xml:e {<ram:InvoiceCurrencyCode>\l__zugferd_currency_tl</ram:InvoiceCurren
1211 }%
1212 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeSettlement_stop: {%
1213 \int_gdecr:N \g__zugferd_indent_int%
1214 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeSettlement>}%
1215 }%

```

SpecifiedTradePaymentTerms

```

1216 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradePaymentTerms:nn {%
1217 \__zugferd_write_xml:n {<ram:SpecifiedTradePaymentTerms>}%
1218 \int_gincr:N \g__zugferd_indent_int%
1219 \__zugferd_write_xml:e {%
1220 \tl_if_blank:nF {#1} {%

```

```

1221     <ram:Description>#1</ram:Description>%
1222   }%
1223   \tl_if_blank:nF {#2} {%
1224     <ram:DueDateDateTime>
1225     \__zugferd_indent: <udt:DateTimeString~format="102">#2</udt:DateTimeString>
1226     </ram:DueDateDateTime>%
1227   }%
1228 }%
1229 \int_gdecr:N \g__zugferd_indent_int%
1230 \__zugferd_write_xml:n {</ram:SpecifiedTradePaymentTerms>}%
1231 }%
1232 %
1233 %
1234 % sums
1235 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeSettlementPaymentMeans:nnn {%
1236 \bool_if:NT \g__zugferd_writePaymentMeans_bool {%
1237   \__zugferd_write_xml:n {<ram:SpecifiedTradeSettlementPaymentMeans>}%
1238   \int_gincr:N \g__zugferd_indent_int%
1239   \__zugferd_write_xml:e {%
1240     <ram:TypeCode>\g__zugferd_payment_means_tl</ram:TypeCode>
1241     \__zugferd_write_inline:ne {ram:Information} {\prop_item:cV {g__zugferd_payment-
means_names_prop} \g__zugferd_payment_means_tl}%
1242     \tl_if_blank:nF {#1#2} {%
1243       <ram:PayeePartyCreditorFinancialAccount>
1244       \__zugferd_write_inline_i:nn {ram:IBANID} {#2}
1245       \__zugferd_write_inline_i:nn {ram:AccountName} {#1}
1246       </ram:PayeePartyCreditorFinancialAccount>
1247     }%
1248     \tl_if_blank:nF {#3} {%
1249       <ram:PayeeSpecifiedCreditorFinancialInstitution>
1250       \__zugferd_write_inline_i:nn {ram:BICID} {#3}
1251       </ram:PayeeSpecifiedCreditorFinancialInstitution>%
1252     }%
1253   }%
1254   \int_gdecr:N \g__zugferd_indent_int%
1255   \__zugferd_write_xml:n {</ram:SpecifiedTradeSettlementPaymentMeans>}%
1256 }%
1257 }%
1258 \cctab_end:
1259 \cs_generate_variant:Nn \__zugferd_SpecifiedTradePaymentTerms:nn {VV}
1260 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradePaymentTerms: {
1261 \bool_if:NF \g__zugferd_minimum_bool {
1262   \__zugferd_SpecifiedTradePaymentTerms:VV \g__zugferd_payment_terms_str \g__zugferd_due_dat
1263 }
1264 }

```

_zugferd_SpecifiedTradeSettlementPaymentMeans:

```
1265 \cs_generate_variant:Nn \_zugferd_SpecifiedTradeSettlementPaymentMeans:nnn {vfv}
1266 \_zugferd_define_xml_writer:Nn \_zugferd_SpecifiedTradeSettlementPaymentMeans: {
1267 \tl_if_blank:VF \g_zugferd_payment_means_tl {
1268 \_zugferd_SpecifiedTradeSettlementPaymentMeans:vfv
1269 {g_zugferd_payment_account-holder_tl} {g_zugferd_payment_iban_tl} {g_zugferd_payment_b
1270 }
1271 }

1272 \msg_new:nnn {zugferd} {macro-deprecated} {
1273 The-function-#1-is-deprecated.\\
1274 It-was-replaced-by-#2.\\
1275 Please-adjust-your-mechanism-to-use-the-new-version.
1276 }
```

Change History

v0.4		auto-exemption: minimum	14
	end-date: Deprecate old syntax and add public interfaces.		40
v0.6	General: Provide public interfaces and first version of the documentation.		5
v0.7	General: Added exemption-reason-auto key for pre-configured exemption-reasons.		35
	auto-exemption: Added auto-exemption option		14
v0.8	General: First CTAN version		1, 14
v0.8a	General: Use the new public interface for l3pdfmeta.		16
v0.9	General: Add interface using an argument for the keyval options locally per item		21
	end-date: Split SpecifiedLineTradeSettlement to be more flexible and add support for BillingSpecifiedPeriod		40
v0.9a	General: Add error message if pdfmanagement is outdated.		16
	Add public interface for the rounding mechanism.		7
		auto-exemption: minimum	14
v0.9b	General: Fix file name of embedded file for the zugferd profile.		20
v0.9c	General: Add support for extended Addresses including additional legal data.		10
	Add support for other invoice types.		8
	Add support for subdivision.		11
	Add support for taxid.		10
	Add support for third address line.		10
	Add TradePartyName to support additional legal information.		28
	Added \zugferd_write_TaxEntry:enmn variant		6
	Added mechanism to escape XML special characters.		7
	Call exemption-reason-auto also locally inside ApplicableTradeTaxEntry		42
	Compatibility to show the error message in older formats versions.		16
	BASIC #28		42
	Remove second catalog entry linking to xml #32		20
	end-date: basic, #28		38

References

- [1] URL: <https://xeinkauf.de/dokumente/> (visited on 08/20/2024).
- [2] *ECE/TRADE/201. ISO COUNTRY CODE for Representation of Names of Countries*. URL: <https://unece.org/trade/documents/iso-country-code-representation-names-countries> (visited on 08/20/2024).
- [3] United Nations Economic Commission for Europe (UNECE). *Code List Recommendations*. URL: <https://unece.org/trade/uncefact/cl-recommendations> (visited on 08/20/2024).
- [4] Markus Kohm. *The scrletter package. Letter extension to KOMA-Script classes*. Version 3.41. July 7, 2023. URL: <https://komascript.de/> (visited on 09/03/2024).
- [5] Koordinierungsstelle für IT-Standards. *Spezifikation Standard XRechnung. CIUS und Extension*. June 20, 2024. URL: <https://xeinkauf.de/app/uploads/2024/07/302-XRechnung-2024-06-20.pdf> (visited on 08/20/2024).
- [6] Koordinierungsstelle für IT Standards (KoSIT). *Xeinkauf FAQ. Leitweg ID*. URL: <https://xeinkauf.de/faq/xrechnung#leitweg-id> (visited on 09/04/2024).
- [7] *Mustang Project*. URL: <https://github.com/ZUGFeRD/mustangproject> (visited on 08/20/2024).
- [8] Marei Peischl. *ZUGFeRD - Create ZUGFeRD and other kinds of E-invoices using L^AT_EX*. *GitHub Repository*. URL: <https://github.com/TeXhackse/LaTeX-ZUGFeRD>.
- [9] *Rec 20 – Codes for Units of Measure Used in International Trade*. Link directly to xlsx. [see 3, for all revisions]. URL: https://unece.org/sites/default/files/2023-10/rec20_Rev17e-2021.xlsx (visited on 08/20/2024).
- [10] *VAT exemption reason code list*. URL: https://www.xrepository.de/details/urn:xoev-de:kosit:codelist:vatex_1 (visited on 08/20/2024).
- [11] *ZUGFeRD 2.2. Download page*. URL: <https://ferd-net.de/standards/zugferd-2.2/zugferd-2.2.html> (visited on 09/02/2024).

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	
$\backslash\#$	9, 71, 352, 983
$\backslash&$	370, 379
$\backslash\backslash$	78, 155, 218, 477, 492, 501, 845, 849, 853, 854, 857, 861, 862, 970, 1273, 1274
$\backslash\{$	348
$\backslash\}$	350
$\langle party \rangle / \text{address}$ (option)	10
$\langle party \rangle / \text{city}$ (option)	11
$\langle party \rangle / \text{contact}$ (option)	11
$\langle party \rangle / \text{country}$ (option)	11
$\langle party \rangle / \text{email}$ (option)	10
$\langle party \rangle / \text{name}$ (option)	10
$\langle party \rangle / \text{postcode}$ (option)	11
$\langle party \rangle / \text{subdivision}$ (option)	11
$\langle party \rangle / \text{taxid}$ (option)	10
$\langle party \rangle / \text{vatid}$ (option)	10
$\langle party \rangle / \text{address}$ (option)	<u>488</u>
A	
add-note (option)	9, <u>465</u>
AddressData (option)	4
auto-exemption (option)	4, <u>9</u>

B		D	
<code>\begin</code>	388	<code>date</code> (option)	9, 443
<code>\begingroup</code>	209, 969, 982, 1137, 1180	<code>\day</code>	192, 447, 455
bool commands:		<code>\def</code>	774
<code>\bool_gset_false:N</code>	37, 38, 47, 48, 175	<code>delivery-date</code> (option)	9, 443
<code>\bool_gset_true:N</code>	12, 13, 46	<code>document-type</code> (option)	8, 407
<code>\bool_if:NTF</code>	75, 177, 179, 210, 224, 227, 244, 252, 266, 287, 300, 581, 595, 602, 632, 678, 690, 870, 965, 1158, 1166, 1181, 1236, 1261	<code>\DocumentMetadata</code>	79
<code>\bool_lazy_and:nnTF</code>	655, 698	<code>due-date</code> (option)	9
<code>\bool_lazy_or:nnTF</code>	1052, 1102, 1191	E	
<code>\bool_new:N</code>	4, 5, 6	<code>\edef</code>	370, 371, 372, 373, 374
<code>\bool_set_true:N</code>	169	<code>end-date</code> (option)	942
<code>\g_tmpa_bool</code>	169, 175, 177, 179	<code>\endgroup</code>	226, 978, 1011, 1140, 1199
buyer/reference (option)	11, 731	environments:	
C		<code>ZUGFeRD</code>	4
<code>category</code> (option)	852	<code>exemption-reason</code> (option)	852
cctab commands:		exp commands:	
<code>\cctab_begin:N</code> 562, 750, 963, 1044, 1205		<code>\exp_args:Ne</code>	130, 532
<code>\cctab_end:</code> . 726, 760, 1043, 1122, 1258		<code>\exp_args:Nne</code>	815
<code>\cctab_select:N</code>	359	<code>\exp_args:NnnV</code>	832
char commands:		<code>\exp_args:Nnnx</code>	433, 923
<code>\char_set_active_eq:NN</code>	375, 376, 377, 378, 379	<code>\exp_args:NnV</code>	232
<code>\char_set_active_eq:nN</code>	211	<code>\exp_not:n</code>	939, 940
<code>\char_set_catcode:nn</code>	984	<code>\ExplSyntaxOff</code>	761
<code>\char_set_catcode_active:N</code>	382, 383, 384, 385, 386	<code>\ExplSyntaxOn</code>	788
<code>\char_set_catcode_other:N</code>	9, 983	F	
<code>\char_set_catcode_parameter:N</code>	71	<code>format</code> (option)	3, 9
clist commands:		fp commands:	
<code>\clist_map_break:</code>	177, 179	<code>\fp_eval:n</code>	390, 394
<code>\clist_map_inline:nn</code>	153, 170, 171, 488, 685, 800	<code>\fp_gset:Nn</code>	395
<code>\clist_new:N</code>	742	<code>\fp_set:Nn</code>	391
cs commands:		<code>\fp_use:N</code>	1151
<code>\cs_generate_variant:Nn</code> 188, 298, 299, 312, 326, 727, 728, 729, 730, 956, 1090, 1123, 1124, 1131, 1259, 1265		<code>fromaddress</code> (option)	9, 465
<code>\cs_gset:Nn</code>	245, 249, 258, 263, 271	G	
<code>\cs_gset_eq:NN</code>	253, 255, 267, 269, 279, 280, 281, 282, 283, 284, 285	group commands:	
<code>\cs_if_exist:NTF</code>	76, 817	<code>\group_begin:</code>	228, 293, 355
<code>\cs_new:Nn</code> 185, 243, 278, 292, 301, 304, 305, 308, 309, 314, 327, 331, 335, 338, 341, 344, 347, 354, 365, 369, 381, 483, 523, 531, 543, 555, 1125		<code>\group_end:</code>	234, 296, 363
<code>\cs_new:Npn</code>	189	H	
<code>\cs_new_eq:NN</code>	304, 305	hook commands:	
<code>\cs_set:Nn</code>	308, 309, 389, 393	<code>\hook_gput_code:nnn</code>	813
<code>\cs_set_eq:NN</code>	307, 603, 966	I	
currency (option)	9	<code>id</code> (option)	8, 407
		<code>\IfBooleanTF</code>	144
		<code>\IfNoValueF</code>	195
		<code>\IfPackageAtLeastTF</code>	85
		<code>\ignorespaces</code>	201
		<code>\InsertZUGFeRDData</code>	4
		<code>\InsertZUGFeRDData</code>	18, 150, 183, 184
		<code>\InsertZugferdData</code>	18, 183
		<code>\insertZugferdData</code>	184

int commands:	
\int_compare:nNnTF	446, 447, 454, 455, 502
\int_eval:n	796
\int_gdecr:N 670, 683, 688, 709, 711, 1008, 1028, 1038, 1040, 1087, 1145, 1152, 1170, 1201, 1213, 1229, 1254	
\int_gincr:N 654, 664, 675, 703, 705, 995, 997, 1014, 1033, 1083, 1134, 1143, 1157, 1176, 1209, 1218, 1238	
\int_new:N	334
\int_set:Nn	190, 191, 192
iow commands:	
\iow_close:N	225
\iow_new:N	74
\iow_newline:	132, 134, 136, 139, 345, 533, 970
\iow_now:Nn	302
\iow_open:Nn	212
item/end-date (option)	14
item/start-date (option)	14
J	
\jobname	64
K	
keys commands:	
\l_keys_choice_tl	429, 792, 824, 869
\keys_define:nn	10, 405, 490, 732, 743, 790, 801, 822, 867, 937, 942
\keys_if_choice_exist:nnnTF	433, 832, 871, 923
\l_keys_key_str	939, 940
\l_keys_key_tl	519
\keys_set:nn	25, 31, 145, 146, 294, 436, 494, 503, 510, 835, 872, 886, 889, 895, 901, 907, 926, 939, 940, 1182
\l_keys_value_tl	939, 940
L	
\languagename	817
\LaTeX	78, 87
\let	149, 348, 349, 350, 351, 352, 970
M	
\month	191, 446, 454
msg commands:	
\msg_error:nn	83, 90
\msg_info:nn	214
\msg_new:nnn	86, 217, 1272
\msg_new:nnnn	77, 476, 844, 852, 860
\msg_warning:nnn 438, 795, 837, 874, 928	
\msg_warning:nnnn	1126
N	
\NewDocumentCommand	143, 150
\NewDocumentEnvironment	194
O	
options:	
<party>/address	10
<party>/city	11
<party>/contact	11
<party>/country	11
<party>/email	10
<party>/name	10
<party>/postcode	11
<party>/subdivision	11
<party>/taxid	10
<party>/vaid	10
<party>/address	488
add-note	9, 465
AddressData	4
auto-exemption	4, 9
buyer/reference	11, 731
category	852
currency	9
date	9, 443
delivery-date	9, 443
document-type	8, 407
due-date	9
end-date	942
exemption-reason	852
format	3, 9
fromaddress	9, 465
id	8, 407
item/end-date	14
item/start-date	14
payment-means	761
payment-terms	9, 463
rate	852
set-today	4
start-date	942
subject	9, 465
tax/category	13
tax/exemption-reason	13
tax/exemption-reason-code	13
tax/rate	14
unit	13, 822
write-xml	3, 9
xml-file	3, 9
xrechnung	9
zugferd	3, 9
P	
payment-means (option)	761
payment-terms (option)	9, 463

421, 423, 426, 427, 444, 449, 452,
 458, 467, 469, 471, 473, 735, 738, 796
 \tl_gset_eq:NN 792
 \tl_gset_rescan:Nnn 358
 \tl_if_blank:nTF 517,
 532, 570, 583, 586, 589, 596, 606,
 615, 621, 631, 635, 638, 643, 713,
 952, 1184, 1220, 1223, 1242, 1248, 1267
 \tl_if_blank_p:n
 656, 699, 1053, 1102, 1192
 \tl_if_empty:NTF 484
 \tl_new:N 1, 8, 353, 398,
 399, 400, 401, 402, 403, 404, 731, 789
 \tl_set:Nn 439, 831
 \tl_set_eq:NN .. 367, 429, 824, 838, 869
 \tl_to_str:n ... 370, 371, 372, 373, 374
 \tl_trim_spaces:n 519
 \l_tmpa_tl 831, 833, 835, 838
 \today 4

U

unit (option) 13, 822
 use commands:
 \use:N 172, 174, 816
 \use_none:n 307, 603, 966
 \use_none:nnn 631
 \use_none:nnnn 267, 284
 \use_none:nnnnn 253, 281
 \use_none:nnnnnnn 285

W

write-xml (option) 3, 9

X

xml-file (option) 3, 9
 xrechnung (option) 9

Y

\year 190, 445, 453

Z

ZUGFeRD (env.) 4
 zugferd (option) 3, 9
 zugferd commands:
 \zugferd_disable_macros:
 24, 347, 356, 1138
 \zugferd_disable_XML_interfaces:
 21, 206, 278, 290
 \zugferd_enable_XML_interfaces:
 21, 198, 203, 243, 288
 \zugferd_fp_gset_rounded:Nn
 25, 388, 393
 \zugferd_fp_set_rounded:Nn
 25, 388, 389
 \zugferd_startInvoiceSums: 258, 282

\zugferd_stopInvoiceSums: . 263, 283
 \zugferd_tl_gset_escaped_xml:Nn
 354, 366
 \zugferd_tl_set_escaped_xml:Nn . 365
 \zugferd_write_Footer: . 204, 249, 280
 \zugferd_write_Header: . 200, 245, 279
 \zugferd_write_Item:nnnnnn
 253, 255, 281, 295
 \zugferd_write_Item:nnnnnnn
 22, 292, 298
 \zugferd_write_Summation:nnnnnnnn
 271, 285
 \zugferd_write_TaxEntry:nnnn ...
 22, 44, 267, 269, 284, 299

zugferd internal commands:

\g_zugferd_active_bool .. 58, 75, 227
 __zugferd_ApplicableHeaderTradeAgreement:
 259, 652
 __zugferd_ApplicableHeaderTradeDelivery:n
 692, 697, 730
 __zugferd_ApplicableHeaderTradeSettlement_
 start: 260, 1207
 __zugferd_ApplicableHeaderTradeSettlement_
 stop: 264, 1212
 __zugferd_ApplicableTradeTax:nnnn
 269, 1174
 __zugferd_AssociatedDocumentLineDocument:n
 38, 1045, 1135
 \g_zugferd_auto_exemption_bool
 66, 870, 1181
 __zugferd_BillingSpecifiedPeriod:nn
 1085, 1101, 1124
 \g_zugferd_businessProcessId_
 str 1, 22, 999, 1001
 \g_zugferd_buyer_AddressData_
 prop 488, 677, 679, 680
 \g_zugferd_buyer_document_
 references_clist 742
 \g_zugferd_buyer_reference_tl .
 656, 660, 731, 735, 738
 \g_zugferd_conformance_level_
 str .. 1, 14, 39, 49, 138, 1054, 1193
 \l_zugferd_currency_tl
 404, 429, 439, 1164, 1210
 \g_zugferd_date_tl 397, 444, 449, 1020
 __zugferd_define_xml_content:Nn
 22, 305, 309, 563,
 579, 605, 613, 629, 697, 948, 951,
 957, 960, 1050, 1075, 1091, 1101, 1115
 __zugferd_define_xml_writer:Nn
 22, 304, 308, 652,
 751, 968, 985, 1012, 1037, 1045,
 1060, 1065, 1081, 1132, 1155, 1174,
 1207, 1212, 1216, 1235, 1260, 1266

__zugferd_DefinedTradeContact:N
 28, 555, 667, 679
 __zugferd_DefinedTradeContact:nnnn
 556, 629, 728
 \g__zugferd_delivery_date_t1 ...
 400, 452, 458, 692
 \g__zugferd_document_references_
 prop .. 741, 744, 745, 747, 752, 755
 \g__zugferd_DocumentTypeCode_t1
 397, 409,
 411, 413, 415, 417, 419, 421, 423, 1017
 \g__zugferd_due_date_t1 ... 461, 1262
 \l__zugferd_end_date_t1 ... 945, 1085
 __zugferd_escape_xml_activate:
 360, 381
 __zugferd_escape_xml_aux: 357, 369
 \g__zugferd_format_str
 1, 19, 28, 34, 43, 1005
 \g__zugferd_fromaddress_t1
 397, 471, 473, 1024
 \g__zugferd_id_t1 397, 426, 427, 1016
 __zugferd_indent: 23,
 335, 339, 582, 584, 587, 590, 592,
 594, 597, 608, 617, 623, 634, 636,
 639, 641, 644, 646, 715, 719, 755,
 958, 973, 975, 1001, 1005, 1020,
 1047, 1055, 1057, 1062, 1067, 1069,
 1070, 1072, 1078, 1094, 1096, 1098,
 1105, 1107, 1109, 1111, 1118, 1225
 __zugferd_indent:n
 338, 342, 640, 645,
 716, 717, 718, 1068, 1071, 1106, 1110
 \g__zugferd_indent_int
 334, 342, 654, 664, 670,
 675, 683, 688, 703, 705, 709, 711,
 995, 997, 1008, 1014, 1028, 1033,
 1038, 1040, 1083, 1087, 1134, 1143,
 1145, 1152, 1157, 1170, 1176, 1201,
 1209, 1213, 1218, 1229, 1238, 1254
 __zugferd_insert_Footer: . 250, 1037
 __zugferd_insert_FrontMatter: .
 247, 1012
 __zugferd_insert_Header: . 246, 985
 __zugferd_insert_TradeLineItem:nnnnn
 5, 41, 256, 1132
 __zugferd_Line_ApplicableTradeTax:nn
 1084, 1091
 \g__zugferd_minimum_bool
 ... 1, 46, 252, 266, 581, 595, 602,
 657, 678, 690, 965, 1158, 1166, 1261
 __zugferd_note_if_not_empty:N .
 483, 1023, 1024
 \g__zugferd_notes_seq . 397, 474, 1025
 __zugferd_number_format:nNn ...
 23, 314, 326, 328, 390, 394
 \g__zugferd_payment_means_t1 ...
 789, 792, 796, 1240, 1241, 1267
 \g__zugferd_payment_terms_str ..
 463, 1262
 __zugferd_PostalTradeAddress:N
 543, 668, 680
 __zugferd_PostalTradeAddress:nnnnnnn
 534, 544, 579, 729
 __zugferd_PostalTradeAddress_
 short:N 531, 707
 __zugferd_ProductTradePrice:nn
 39, 1060, 1144
 \g__zugferd_seller_AddressData_
 prop 488, 666, 667, 668
 __zugferd_set_today:N 19
 __zugferd_set_today:n . 163, 185, 188
 __zugferd_set_today_aux:w 186, 189
 \g__zugferd_shipto_AddressData_
 prop 488, 698, 702, 707
 __zugferd_SpecifiedLineTradeAgreement:nn
 39, 1065
 __zugferd_SpecifiedLineTradeDelivery:nn
 39, 1075, 1148
 __zugferd_SpecifiedLineTradeSettlement:nnn
 40, 1123, 1128, 1129
 __zugferd_SpecifiedLineTradeSettlement:nnnn
 1125, 1127, 1131
 __zugferd_SpecifiedTradePaymentTerms:
 272, 1260
 __zugferd_SpecifiedTradePaymentTerms:nn
 1216, 1259, 1262
 __zugferd_SpecifiedTradeProduct:nn
 38, 1050, 1139
 __zugferd_SpecifiedTradeSettlementHeaderMonetarySu
 273, 1155
 __zugferd_SpecifiedTradeSettlementLineMonetarySumm
 1086, 1115
 __zugferd_SpecifiedTradeSettlementPaymentMeans:
 44, 261, 1266
 __zugferd_SpecifiedTradeSettlementPaymentMeans:nnn
 1235, 1265, 1268
 \l__zugferd_start_date_t1 . 943, 1085
 \g__zugferd_subject_t1
 397, 467, 469, 1023
 \l__zugferd_tax_category_code_t1
 869, 1150
 \l__zugferd_tax_exemption_code_
 t1 881, 1192, 1196
 \l__zugferd_tax_exemption_reason_
 t1 878, 1184, 1186
 \l__zugferd_tax_rate_fp ... 933, 1151

__zugferd_TaxRegistration:nn ..	722, 753, 971, 986, 996, 998, 1009,
.....	1013, 1015, 1029, 1039, 1041, 1082,
\g__zugferd_tmp_tl	1084, 1085, 1086, 1088, 1133, 1135,
\l__zugferd_tmp_tl	1139, 1142, 1144, 1146, 1147, 1153,
.....	1156, 1171, 1175, 1178, 1185, 1190,
__zugferd_TradePartyIdentity:N	1195, 1200, 1202, 1208, 1210, 1214,
.....	1217, 1219, 1230, 1237, 1239, 1255
__zugferd_TradePartyIdentity:N\ __zugferd_PostalTradeAddress:N	28
__zugferd_TradePartyIdentity:nnnnn	524, 563, 727
\l__zugferd_unit_code_tl	824, 838, 1148
__zugferd_UniversalCommunication:n	552, 603, 605
__zugferd_write_inline:nn	565, 567, 569, 948, 956, 1241
__zugferd_write_inline:nnn	949, 951, 958
__zugferd_write_inline_i:nn	575, 960, 1244, 1245, 1250
__zugferd_write_inline_i:nnn ..	573, 957, 961
__zugferd_write_note:n	485, 966, 968, 1026
__zugferd_write_ReferencedDocument:n	686, 751
__zugferd_write_rounded:nnn ...	23, 331, 1159, 1160, 1161,
.....	1163, 1165, 1167, 1169, 1177, 1189
__zugferd_write_rounded:nnnn ..	23, 327, 332, 1164
__zugferd_write_SpecifiedLineTradeSettlement:nnn	1081, 1090, 1150
__zugferd_write_TradeLineItem:nnnnnn	41
__zugferd_write_xml:n	22, 301, 307, 312, 329,
.....	653, 659, 663, 665, 671, 676, 684,
.....	689, 691, 701, 704, 706, 710, 714,
\g__zugferd_write_xml_bool	55, 210, 224, 244, 287, 300
\g__zugferd_writePaymentMeans_ bool	1, 13, 38, 48, 1236
\g__zugferd_writeTradeContact_ bool	1, 12, 37, 47, 632
__zugferd_xml_auto_indent:	23, 302, 341, 345, 533
\g__zugferd_xml_cctab	562, 750, 963, 1044, 1205
\g__zugferd_xml_embedded_file_tl	1, 15, 40, 50, 134, 232, 237
\g__zugferd_xml_file_tl .	63, 212, 232
__zugferd_xml_newline_indent: .	23, 211, 344
__zugferd_xml_writer_iow	74, 212, 225, 302
\zugferd_disable_macros:	7
\zugferd_disable_XML_interfaces:	7
\zugferd_enable_XML_interfaces:	7
\zugferd_fp_gset_rounded:Nn	7
\zugferd_fp_set_rounded:Nn	7
\zugferd_startInvoiceSums:	6
\zugferd_stopInvoiceSums:	6
\zugferd_tl_gset_escaped_xml:Nn	7
\zugferd_tl_set_escaped_xml:Nn	7
\zugferd_write_Item:ennnnnn	6
\zugferd_write_Item:nnnnnnn	5
\zugferd_write_Item:nnnnnnn	6
\zugferd_write_Summation:nnnnnnnn ..	6
\zugferd_write_TaxEntry:ennn	6
\zugferd_write_TaxEntry:nnnn	6