

The zugferd package*

Creating electronic and hybrid invoices using \LaTeX

Marei Peischl
marei@peitex.de

January 2, 2025

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 \LaTeX 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 \TeX Live setup.
- Frederik Schwan ([Schwan.IT](https://www.schwan.it))
- byte physics e.K., <https://www.byte-physics.de>

*This document corresponds to zugferd v0.9d, dated 2025-01-02.

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	5
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.

`unknown-value-warning=` (*boolean*) (default: `true`)

There are predefined values for units, tax categories and payment-means. Since there are a lot of options `zugferd` does not define all of them. In case you want to use another one as the predefined options you there will a warning to make you aware of the fact, that the package does not know if that's a valid selection. This is to help you avoiding typos. In case you are sure about your selection and don't want to define an own value for this you can also disable the warning by setting `unknown-value-warning` to `false`.

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 data is processed. Depending on the source of the data this might be necessary. In that case activate the expansion by using the star argument. 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
\zugferd_write_Item:ennnnnn options.

```

```

\zugferd_write_Item:nnnnnnn
  {<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
\zugferd_stopInvoiceSums: been placed. Typically, in LATEX 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 command 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\newline{\iow_newline:}}
```

`\zugferd_tl_set_escaped_xml:` As described before and also in #9 there are characters which are special in XML but
`\zugferd_tl_gset_escaped_xml:` 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: “20250102” (January 2, 2025).

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^[12] use all visible data to add them to the XML as a note. `subject` and `fromaddress` are used to support this. Please be aware that `subject-code` is not the same as `subject` even if some viewers might display it like that, [#35](#). 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 `add-note` corresponds to the mechanism provided by `scrletter`. It accesses the variable `add-note` and 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:`.

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

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

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

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

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

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 [10].

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 [10].

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 [11].

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
              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,

```

auto-exemption

```

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,

```

unknown-value-warning

```

70 unknown-value-warning .bool_gset:N = \g__zugferd_unknown_value_warning_bool,

```

```

71 unknown-value-warning .initial:n = true,
72 }
73 \char_set_catcode_parameter:N \#%
74
75 \ProcessKeyOptions[zugferd]

```

6.1 Preparation to write the .xml file

`_zugferd_xml_writer_iow`

```

76 \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 Ulrike Fischer from the L^AT_EX Project Team. She had prepared some experiment files for the PDF attachment in the experiments of the repository. We use these to embed the XML file #7.

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

```

77 \bool_if:NT \g_zugferd_active_bool {
78   \cs_if_exist:NF \pdfmeta_xmp_xmlns_new:nn {
79     \msg_new:nnnn {zugferd} {PDFmanagement-not-active} {
80       The~\LaTeX~PDF~management~is~not~active.\\
81       Activate~it~using~\string\DocumentMetadata.
82     } {
83       See~ZUGFeRD~or~PDFmanagement~documentation~for~more~information.
84     }
85     \msg_error:nn {zugferd} {PDFmanagement-not-active}
86   }

87   \IfPackageAtLeastTF{pdfmanagement-testphase}{2024/09/13}{}{
88     \msg_new:nnn {zugferd} {PDFmanagement-too-old} {
89       Your~version~of~\LaTeX's~PDF~management~is~too~old.
90       You~need~to~update~your~LaTeX~distribution~to~be~able~to~use~the~zugferd~package~correctly.
91     }
92     \msg_error:nn{zugferd} {PDFmanagement-too-old}
93   }
94   %% based on experiments for l3pdfmeta by Ulrike Fischer
95   \pdfmeta_xmp_xmlns_new:nn {fx}{
96     urn:factur-x:pdfa:CrossIndustryDocument:invoice:1p0\c_hash_str
97   }

98
99   \pdfmeta_xmp_schema_new:nnn
100     {Factur-X-PDFA-Extension-Schema}
101     {fx}
102     {urn:factur-x:pdfa:CrossIndustryDocument:invoice:1p0\c_hash_str}
103
104   \pdfmeta_xmp_property_new:nnnnn
105     {fx}
106     {DocumentFileName}
107     {Text}
108     {external}
109     {name-of-the-embedded-XML-invoice-file}
110
111   \pdfmeta_xmp_property_new:nnnnn

```



```

112     {fx}
113     {DocumentType}
114     {Text}
115     {external}
116     {INVOICE}
117
118     \pdfmeta_xmp_property_new:nnnnn
119     {fx}
120     {Version}
121     {Text}
122     {external}
123     {The~actual~version~of~the~factur-x~schema}
124
125     \pdfmeta_xmp_property_new:nnnnn
126     {fx}
127     {ConformanceLevel}
128     {Text}
129     {external}
130     {The~conformance~level~of~the~factur-x~data}
131
132     \exp_args:Ne \pdfmeta_xmp_add:n {
133     % fix INVOICE
134     <fx:DocumentType>INVOICE</fx:DocumentType>\iow_newline:
135     % fix factur-x.xml
136     <fx:DocumentFileName>\g__zugferd_xml_embedded_file_tl</fx:DocumentFileName>\iow_newline:
137     % fix schema version
138     <fx:Version>1.0</fx:Version>\iow_newline:
139     % zulässige Werte MINIMUM, BASIC WL, BASIC, EN 16931, EXTENDED, XRECHNUNG
140     <fx:ConformanceLevel>\g__zugferd_conformance_level_str</fx:ConformanceLevel>%
141     \iow_newline:
142     %
143     }
144     }

```

\SetZUGFeRData

```

145     \NewDocumentCommand{\SetZUGFeRData}{sm}{
146     \IfBooleanTF{#1}
147     {\keys_set:ne}
148     {\keys_set:nn}
149     {zugferd} {#2}
150     }
151     \let\SetZugferdData\SetZUGFeRData

```

`\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.

```

152 \NewExpandableDocumentCommand{\UseZUGFeRDElement}{m}{
153   \tl_use:c {l__zugferd_#1_tl}
154 }
155 \NewDocumentCommand{\InsertZUGFeRDData}{om}{
156   \str_case:nnF {#1} {
157     {AddressData} {
158       \clist_map_inline:nn {name, lineone, linetwo, linethree} {
159         \prop_if_in:cnT {g__zugferd_#2_AddressData_prop } {##1} {
160           \prop_item:cn {g__zugferd_#2_AddressData_prop } {##1}\
161         }
162       }
163       \prop_item:cn {g__zugferd_#2_AddressData_prop } {postcode}
164       \space
165       \prop_item:cn {g__zugferd_#2_AddressData_prop } {city}
166     }
167     {set-today} {
168       \__zugferd_set_today:v {g__zugferd_#2_tl}
169     }
170   } {

```

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

```

171   \str_set:Ne \l_tmpa_str {#2}
172   \str_replace_all:Nnn \l_tmpa_str {-} {_}
173   \str_replace_all:Nnn \l_tmpa_str {/} {_}
174   \bool_set_true:N \g_tmpa_bool
175   \clist_map_inline:nn {tl, str} {
176     \clist_map_inline:nn {g, l} {
177       \use:c {##1 _if_exist:cT} {###1__zugferd_ \l_tmpa_str _##1}
178       {
179         \use:c {###1__zugferd_ \l_tmpa_str _##1}
180         \bool_gset_false:N \g_tmpa_bool
181       }
182       \bool_if:NF \g_tmpa_bool {\clist_map_break:}
183     }
184     \bool_if:NF \g_tmpa_bool {\clist_map_break:}
185   }
186 }
187 }
188 \providecommand{\InsertZugferdData}{\InsertZUGFeRDData}
189 \providecommand{\insertZugferdData}{\InsertZUGFeRDData}

```

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

```
190 \cs_new:Nn \_zugferd_set_today:n {
191   \_zugferd_set_today_aux:w #1 \q_stop
192 }
193 \cs_generate_variant:Nn \_zugferd_set_today:n {v}
194 \cs_new:Npn \_zugferd_set_today_aux:w #1 #2 #3 #4 #5 #6 #7 #8 \q_stop{
195   \int_set:Nn \year {#1#2#3#4}
196   \int_set:Nn \month {#5#6}
197   \int_set:Nn \day {#7#8}
198 }

199 \NewDocumentEnvironment{ZUGFeRD}{o}{
200   \IfNoValueF{#1}{
201     \SetZUGFeRDData{#1}
202   }
203   \zugferd_enable_XML_interfaces:
204   \startWritingZUGFeRDxml
205   \zugferd_write_Header:
206   \ignorespaces
207 }{
208   \zugferd_enable_XML_interfaces:
209   \zugferd_write_Footer:
210   \stopWritingZUGFeRDxml
211   \zugferd_disable_XML_interfaces:
212 }

213 \newcommand*{\startWritingZUGFeRDxml}{
214   \begingroup
215   \bool_if:NTF \g__zugferd_write_xml_bool {
216     \char_set_active_eq:nN {13} \_zugferd_xml_newline_indent:
217     \iow_open:Nn \_zugferd_xml_writer_iow {\g__zugferd_xml_file_tl}
218   }{
219     \msg_info:nn {zugferd} {no-xml-write}
220   }
221 }
222 \msg_new:nnn {zugferd} {no-xml-write} {
223   The~option~write~xml=false~was~set.\\
224   Writing~of~XML~file~is~deactivated.
225 }
```

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

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

```
239 \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).

```
240 \pdfmanagement_add:nee
241   {Catalog/Names/EmbeddedFiles}
242   {\g__zugferd_xml_embedded_file_t1}
243   {\pdf_object_ref:n{zugferd/rechnung}}
244   % steht in der doku ist aber pdf 2.0 ....
245   \pdfmanagement_add:nx{Catalog}{AF}{\pdf_object_ref:n{zugferd/rechnung}}
246 }
247 }
```

Provide public interfaces and the ZUGFeRD environment.

\zugferd_enable_XML_interfaces:
\zugferd_disable_XML_interfaces:

```
248 \cs_new:Nn \zugferd_enable_XML_interfaces: {
249   \bool_if:NT \g__zugferd_write_xml_bool {
250     \cs_gset:Nn \zugferd_write_Header: {
251       \__zugferd_insert_Header:
252       \__zugferd_insert_FrontMatter:
253     }
254     \cs_gset:Nn \zugferd_write_Footer: {
255       \__zugferd_insert_Footer:
256     }
257     \bool_if:NTF \g__zugferd_minimum_bool {
258       \cs_gset_eq:NN \zugferd_write_Item:nnnnnn \use_none:nnnnnn
259     }{
260       \cs_gset_eq:NN \zugferd_write_Item:nnnnnn
261       \__zugferd_insert_TradeLineItem:nnnnnn
262     }
263     \cs_gset:Nn \zugferd_startInvoiceSums: {
264       \__zugferd_ApplicableHeaderTradeAgreement:
265       \__zugferd_ApplicableHeaderTradeSettlement_start:
266       \__zugferd_SpecifiedTradeSettlementPaymentMeans:
267     }
268     \cs_gset:Nn \zugferd_stopInvoiceSums: {
269       \__zugferd_ApplicableHeaderTradeSettlement_stop:
270     }
271     \bool_if:NTF \g__zugferd_minimum_bool {
272       \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \use_none:nnnn
273     } {
274       \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \__zugferd_ApplicableTradeTax:nnnn
275     }
276     \cs_gset:Nn \zugferd_write_Summation:nnnnnnnn {
277       \__zugferd_SpecifiedTradePaymentTerms:
278       \__zugferd_SpecifiedTradeSettlementHeaderMonetarySummation:nnnnnnnn
279       {##1} {##2} {##3} {##4} {##5} {##6} {##7} {##8}
280     }
281   }
282 }
283 \cs_new:Nn \zugferd_disable_XML_interfaces: {
284   \cs_gset_eq:NN \zugferd_write_Header: \prg_do_nothing:
285   \cs_gset_eq:NN \zugferd_write_Footer: \prg_do_nothing:
286   \cs_gset_eq:NN \zugferd_write_Item:nnnnnn \use_none:nnnnnn
287   \cs_gset_eq:NN \zugferd_startInvoiceSums: \prg_do_nothing:
288   \cs_gset_eq:NN \zugferd_stopInvoiceSums: \prg_do_nothing:
289   \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \use_none:nnnn
290   \cs_gset_eq:NN \zugferd_write_Summation:nnnnnnnn \use_none:nnnnnnnn
291 }
292 \bool_if:NTF \g__zugferd_write_xml_bool {
293   \zugferd_enable_XML_interfaces:
294 }{
295   \zugferd_disable_XML_interfaces:
296 }
```

`\zugferd_write_Item:nnnnnnn`

```
297 \cs_new:Nn \zugferd_write_Item:nnnnnnn {
298   \group_begin:
299   \keys_set:nn {zugferd/item} {#1}
300   \zugferd_write_Item:nnnnnnn {#2} {#3} {#4} {#5} {#6} {#7}
301   \group_end:
302 }
303 \cs_generate_variant:Nn \zugferd_write_Item:nnnnnnn {ennnnnn}
```

`\zugferd_write_TaxEntry:ennn`

```
304 \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`.

```
305 \bool_if:NTF \g__zugferd_write_xml_bool {
306   \cs_new:Nn \__zugferd_write_xml:n {
307     \iow_now:Ne \__zugferd_xml_writer_iow {\__zugferd_xml_auto_indent: #1}
308   }
309   \cs_new_eq:NN \__zugferd_define_xml_writer:Nn \cs_new:Nn
310   \cs_new_eq:NN \__zugferd_define_xml_content:Nn \cs_new:Nn
311 } {
312   \cs_set_eq:NN \__zugferd_write_xml:n \use_none:n
313   \cs_set:Nn \__zugferd_define_xml_writer:Nn {\cs_new:Nn #1 {}}
314   \cs_set:Nn \__zugferd_define_xml_content:Nn {\cs_new:Nn #1 {}}
315 }
316
317 \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.

```
318 \RequirePackage{siunitx}
```

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

```
319 \cs_new:Nn \__zugferd_number_format:nNn {
320   \sisetup{
321     parse-numbers=true,
322     round-mode=places,
323     round-precision=#1,
324     round-pad = false,
325     group-digits=none,
326     minimum-decimal-digits=#1,
327     output-decimal-marker=.
328   }
329   \siunitx_number_format:nN {#3} #2
330 }
331 \cs_generate_variant:Nn \__zugferd_number_format:nNn {nNe}
```

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

```
332 \cs_new:Nn \__zugferd_write_rounded:nnnn {
333   \__zugferd_number_format:nNe {#1} \l__zugferd_tmp_tl {#4}
334   \__zugferd_write_xml:e {<ram:#2#3>\l__zugferd_tmp_tl</ram:#2>}
335 }
336 \cs_new:Nn \__zugferd_write_rounded:nnn {
337   \__zugferd_write_rounded:nnnn {#1} {#2} {} {#3}
338 }
```

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
```

```
339 \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:
```

```
340 \cs_new:Nn \__zugferd_indent: {
341   \space\space
342 }
343 \cs_new:Nn \__zugferd_indent:n {
344   \prg_replicate:nn {#1} {\__zugferd_indent:}
345 }
346 \cs_new:Nn \__zugferd_xml_auto_indent: {
347   \__zugferd_indent:n {\g__zugferd_indent_int}
348 }
```

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

```

349 \cs_new:Nn \__zugferd_xml_newline_indent: {
350 \iow_newline: \__zugferd_xml_auto_indent:
351 }

```

Setting up the catcodes to simplify the XML indentation.

`\zugferd_disable_macros:` The definition was mostly taken from `hyperref` [9]. 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.

```

352 \cs_new:Nn \zugferd_disable_macros: {
353 \let\{\c_left_brace_str
354 \let\textbraceleft\c_left_brace_str
355 \let\}\c_right_brace_str
356 \let\textbraceright\c_right_brace_str
357 \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.

```

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

```



```

391 \char_set_catcode_active:N &
392 }

```

```

\zugferd_fp_set_rounded:Nn
\zugferd_fp_gset_rounded:Nn

```

```

393 % \begin{function}{\zugferd_fp_set_rounded:Nn, \zugferd_fp_gset_rounded:Nn}
394 \cs_set:Nn \zugferd_fp_set_rounded:Nn {
395   \__zugferd_number_format:nNn {2} \l__zugferd_tmp_tl {\fp_eval:n {#2}}
396   \fp_set:Nn #1 {\l__zugferd_tmp_tl }
397 }
398 \cs_set:Nn \zugferd_fp_gset_rounded:Nn {
399   \__zugferd_number_format:nNn {2} \l__zugferd_tmp_tl {\fp_eval:n {#2}}
400   \fp_gset:Nn #1 {\l__zugferd_tmp_tl }
401 }

```

```

\g__zugferd_notes_seq
  \g__zugferd_id_tl
  \g__zugferd_date_tl
\g__zugferd_subject_tl
\g__zugferd_fromaddress_tl
  \g__zugferd_DocumentTypeCode_tl
402 \seq_new:N \g__zugferd_notes_seq
403 \tl_new:N \g__zugferd_id_tl
404 \tl_new:N \g__zugferd_date_tl
405 \tl_new:N \g__zugferd_delivery_date_tl
406 \tl_new:N \g__zugferd_subject_tl
407 \tl_new:N \g__zugferd_fromaddress_tl
408 \tl_new:N \g__zugferd_DocumentTypeCode_tl
409 \tl_new:N \l__zugferd_currency_tl

```

(End of definition for \g__zugferd_notes_seq and others.)

```

410 \keys_define:nn { zugferd } {
411   %TODO define others

```

```

id
document-type

```

```

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

```

```

433 currency .choices:nn = {EUR, USD, CHF} {
434   \tl_set_eq:NN \l__zugferd_currency_tl \l_keys_choice_tl
435 },
436 currency / € .meta:n = {currency = EUR},
437 currency / unknown .code:n = {
438   \exp_args:Nnnx \keys_if_choice_exist:nnnTF {zugferd} {unit} {
439     \str_uppercase:f {#1}
440   }
441   {\keys_set:nx { zugferd } {unit= {\str_uppercase:f {#1}}}}
442   {
443     \bool_if:NT \g__zugferd_unknown_value_warning_bool {
444       \msg_warning:nnn { zugferd } {unknown-value} {currency} {#1}
445     }
446     \tl_set:Ne \l__zugferd_currency_tl {\str_uppercase:f {#1}}
447   }
448 },
449 currency .initial:n = EUR,

date
delivery-date 450 date .choice:,
451 date / auto .code:n = \tl_gset:Ne \g__zugferd_date_tl {
452   \the\year
453   \int_compare:nNnT {\month} < {10} {0} \the\month
454   \int_compare:nNnT {\day} < {10} {0}\the\day
455 },
456 date / unknown .code:n = \tl_gset:Nn \g__zugferd_date_tl {#1},
457 date .initial:n = auto,
458 delivery-date .choice:,
459 delivery-date / auto .code:n = \tl_gset:Ne \g__zugferd_delivery_date_tl {
460   \the\year
461   \int_compare:nNnT {\month} < {10} {0} \the\month
462   \int_compare:nNnT {\day} < {10} {0}\the\day
463 },
464 delivery-date / unknown .code:n = {
465   \tl_gset:Nn \g__zugferd_delivery_date_tl {#1}
466 },
467 delivery-date .initial:n = auto,
468 due-date .tl_gset:N = \g__zugferd_due_date_tl,
469 due-date .initial:n =,

payment-terms

subject
fromaddress
add-note 472 subject .choice:,
473 subject / komavar .code:n = {
474   \tl_gset:Nf \g__zugferd_subject_tl {\scr@subject@var}
475 },
476 subject / unknown .code:n = \tl_gset:Nn \g__zugferd_subject_tl {#1},
477 fromaddress .choice:,
478 fromaddress / komavar .code:n = \tl_gset:Nf \g__zugferd_fromaddress_tl
479   {\scr@fromaddress@var},
480 fromaddress / unknown .code:n = \tl_gset:Nn \g__zugferd_fromaddress_tl {#1},

```

```

481 add-note .code:n = \seq_gput_right:Nn \g__zugferd_notes_seq {#1},
482 }
483 \msg_new:nnnn {zugferd} {unknown-value} {
484 You-selected-a-~#1-which-was-not-predefined.\\
485 I-will-directly-use-your-selection-~'~#1=#2' .
486 Please-ensure-the-selection-is-valid!
487 } {
488 For-more-information-see-the-zugferd-documentation.
489 }

```

Macro to write notes

```

490 \cs_new:Nn \__zugferd_note_if_not_empty:N {%
491 \tl_if_empty:NF #1 {%
492 \__zugferd_write_note:n {#1}%
493 }%
494 }%

```

ApplicableHeaderTradeAgreement

Contains information on seller and buyer trade party:

- BuyerReference
- SellerTradeParty
- BuyerTradeParty

~~\g__zugferd_trade_party_address~~ Seller and Buyer are specified the same way.

```

\g__zugferd_buyer_AddressData_prop
\g__zugferd_shipto_AddressData_prop
495 \clist_map_inline:nn {seller,buyer,shipto} {
496 \prop_new:c {g__zugferd_#1_AddressData_prop}
497 \keys_define:nn {zugferd / #1} {
498 address .code:n = {
499 \seq_set_split:Nnn \l_tmpa_seq {\} {##1}
500 %TODO error if more than 2 lines or only 1
501 \keys_set:nx {zugferd / #1} {
502 lineone={\seq_item:Nn \l_tmpa_seq {1}},
503 linetwo={\seq_item:Nn \l_tmpa_seq {2}},
504 linethree={\seq_item:Nn \l_tmpa_seq {3}}
505 }
506 },
507 contact .code:n = {
508 \seq_set_split:Nnn \l_tmpa_seq {\} {##1}
509 \int_compare:nNnTF {\seq_count:N \l_tmpa_seq} > {3} {
510 \keys_set:ne {zugferd/#1} {
511 contact-name = {\seq_item:Nn \l_tmpa_seq {1}},
512 contact-department = {\seq_item:Nn \l_tmpa_seq {2}},
513 contact-phone = {\seq_item:Nn \l_tmpa_seq {3}},
514 contact-email= {\seq_item:Nn \l_tmpa_seq {4}},
515 }
516 }{
517 \keys_set:ne {zugferd/#1} {
518 contact-name = {\seq_item:Nn \l_tmpa_seq {1}},
519 contact-phone = {\seq_item:Nn \l_tmpa_seq {2}},
520 contact-email= {\seq_item:Nn \l_tmpa_seq {3}},
521 }
522 }
523 },

```

```

524   unknown .code:n = \tl_if_blank:nF {##1} {
525     \prop_gput:cVe {g__zugferd_#1_AddressData_prop}
526     \l_keys_key_tl {\tl_trim_spaces:n {##1}}
527   }
528 }
529 }

```

(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.

```

530 \cs_new:Nn \__zugferd_TradePartyIdentity:N {
531   \__zugferd_TradePartyIdentity:eeee
532   {\prop_item:Nn #1 {id}}
533   {\prop_item:Nn #1 {name}}
534   {\prop_item:Nn #1 {legal-description}}
535   {\prop_item:Nn #1 {legal-id}}
536   {\prop_item:Nn #1 {trading-name}}
537 }
538 \cs_new:Nn \__zugferd_PostalTradeAddress_short:N {
539   \exp_args:Ne \tl_if_blank:nF {\prop_item:Nn #1 {name}}
540   {<ram:Name>\prop_item:Nn #1 {name}</ram:Name>\iow_newline:\__zugferd_xml_auto_indent:}
541   \__zugferd_PostalTradeAddress:eeeeeee
542   {\prop_item:Nn #1 {postcode}}
543   {\prop_item:Nn #1 {lineone}}
544   {\prop_item:Nn #1 {linetwo}}
545   {\prop_item:Nn #1 {linethree}}
546   {\prop_item:Nn #1 {city}}
547   {\prop_item:Nn #1 {country}}
548   {\prop_item:Nn #1 {subdivision}}
549 }
550 \cs_new:Nn \__zugferd_PostalTradeAddress:N {%
551   \__zugferd_PostalTradeAddress:eeeeeee
552   {\prop_item:Nn #1 {postcode}}
553   {\prop_item:Nn #1 {lineone}}
554   {\prop_item:Nn #1 {linetwo}}
555   {\prop_item:Nn #1 {linethree}}
556   {\prop_item:Nn #1 {city}}
557   {\prop_item:Nn #1 {country}}
558   {\prop_item:Nn #1 {subdivision}}
559   \__zugferd_UniversalCommunication:n {\prop_item:Nn #1 {email}}
560   \__zugferd_TaxRegistration:nn {\prop_item:Nn #1 {vatid}} {\prop_item:Nn #1 {taxid}}
561 }%
562 \cs_new:Nn \__zugferd_DefinedTradeContact:N {%
563   \__zugferd_DefinedTradeContact:eeee
564   {\prop_item:Nn #1 {contact-name}}
565   {\prop_item:Nn #1 {contact-department}}
566   {\prop_item:Nn #1 {contact-phone}}
567   {\prop_item:Nn #1 {contact-email}}
568 }%
569 \cctab_begin:N \g__zugferd_xml_cctab%

```

```

570 \__zugferd_define_xml_content:Nn \__zugferd_TradePartyIdentity:nnnnn {%
571 %%<!-- BT-29 -->
572 \__zugferd_write_inline:nn {ram:ID} {#1}
573 %%<!-- BT-27 -->
574 \__zugferd_write_inline:nn {ram:Name} {#2}
575 %%<!-- BT-33 -->
576 \__zugferd_write_inline:nn {ram:Description} {#3}
577 \tl_if_blank:eF {#4#5} {%
578 <ram:SpecifiedLegalOrganization>
579 %%<!-- BT-30 -->
580 \__zugferd_write_inline_i:nnn {ram:ID} {~schemeID="0088"} {#4}
581 %%<!-- BT-28 -->
582 \__zugferd_write_inline_i:nn {ram:TradingBusinessName} {#5}
583 </ram:SpecifiedLegalOrganization>
584 }%
585 }%

586 \__zugferd_define_xml_content:Nn \__zugferd_PostalTradeAddress:nnnnnnn {%
587 <ram:PostalTradeAddress>
588 \bool_if:NF \g__zugferd_minimum_bool {%
589 \__zugferd_write_inline_i:nn {ram:PostcodeCode} {#1}
590 \__zugferd_write_inline_i:nn {ram:LineOne} {#2}
591 \__zugferd_write_inline_i:nn {ram:LineTwo} {#3}
592 \__zugferd_write_inline_i:nn {ram:LineThree} {#4}
593 \__zugferd_write_inline_i:nn {ram:CityName} {#5}
594 }%
595 \__zugferd_indent: <ram:CountryID>#6</ram:CountryID>
596 \bool_if:NF \g__zugferd_minimum_bool {%
597 \__zugferd_write_inline_i:nn {ram:CountrySubDivisionName} {#7}
598 }%
599 </ram:PostalTradeAddress>%
600 }%
601 \bool_if:NTF \g__zugferd_minimum_bool {%
602 \cs_set_eq:NN \__zugferd_UniversalCommunication:n \use_none:n%
603 } {%
604 \__zugferd_define_xml_content:Nn \__zugferd_UniversalCommunication:n {%
605 \tl_if_blank:eF {#1} {
606 <ram:URIUniversalCommunication>
607 \__zugferd_indent: <ram:URIID~schemeID="EM">#1</ram:URIID>
608 </ram:URIUniversalCommunication>%
609 }%
610 }%
611 }%
612 \__zugferd_define_xml_content:Nn \__zugferd_TaxRegistration:nn {%
613 %%<!-- BT-34 -->
614 \tl_if_blank:eF {#1} {
615 <ram:SpecifiedTaxRegistration>
616 \__zugferd_indent: <ram:ID~schemeID="VA">#1</ram:ID>
617 </ram:SpecifiedTaxRegistration>%
618 }%
619 %%<!-- BT-32 -->
620 \tl_if_blank:eF {#2} {
621 <ram:SpecifiedTaxRegistration>
622 \__zugferd_indent: <ram:ID~schemeID="FC">#2</ram:ID>
623 </ram:SpecifiedTaxRegistration>%

```

```

624 }%
625 }%
626 %
627 % Contact data phone/email to a specific contact person
628 \__zugferd_define_xml_content:Nn \__zugferd_DefinedTradeContact:nnnn {%
629 % Do not print if name is empty
630 \tl_if_blank:nT {#1} {\use_none:nnn} %
631 \bool_if:NT \g__zugferd_writeTradeContact_bool {%
632 <ram:DefinedTradeContact>
633 \__zugferd_indent: <ram:PersonName>#1</ram:PersonName>
634 \__zugferd_write_inline_i:nn {ram:DepartmentName} {#2}
635 \__zugferd_write_inline_wrapped_i:nnn {ram:TelephoneUniversalCommunication} {ram:CompleteN
636 \__zugferd_write_inline_wrapped_i:nnn {ram:EmailURIUniversalCommunication} {ram:URIID} {#4
637 </ram:DefinedTradeContact>%
638 }%
639 }%
640 %
641 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeAgreement: {%
642 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeAgreement>}%
643 \int_gincr:N \g__zugferd_indent_int%
644 \bool_lazy_and:nnF%
645 {\tl_if_blank_p:V \g__zugferd_buyer_reference_tl}%
646 {\g__zugferd_minimum_bool}%
647 {%
648 \__zugferd_write_xml:e {%
649 <ram:BuyerReference>\g__zugferd_buyer_reference_tl</ram:BuyerReference>
650 }%
651 }%
652 \__zugferd_write_xml:n {<ram:SellerTradeParty>}%
653 \int_gincr:N \g__zugferd_indent_int%
654 \__zugferd_write_xml:e {%
655 \__zugferd_TradePartyIdentity:N \g__zugferd_seller_AddressData_prop
656 \__zugferd_DefinedTradeContact:N \g__zugferd_seller_AddressData_prop%
657 \__zugferd_PostalTradeAddress:N \g__zugferd_seller_AddressData_prop%
658 }%
659 \int_gdecr:N \g__zugferd_indent_int%
660 \__zugferd_write_xml:e {%
661 </ram:SellerTradeParty>
662 <ram:BuyerTradeParty>%
663 }%
664 \int_gincr:N \g__zugferd_indent_int%
665 \__zugferd_write_xml:e {%
666 \__zugferd_TradePartyIdentity:N \g__zugferd_buyer_AddressData_prop
667 \bool_if:NF \g__zugferd_minimum_bool {
668 \__zugferd_DefinedTradeContact:N \g__zugferd_buyer_AddressData_prop%
669 \__zugferd_PostalTradeAddress:N \g__zugferd_buyer_AddressData_prop%
670 }%
671 }%
672 \int_gdecr:N \g__zugferd_indent_int%
673 \__zugferd_write_xml:n {</ram:BuyerTradeParty>}%
674 \clist_map_inline:nn {SellerOrder,BuyerOrder,Contract} {%
675 \__zugferd_write_ReferencedDocument:n {##1}%
676 }%
677 \int_gdecr:N \g__zugferd_indent_int%

```

```

678 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeAgreement>}%
679 \bool_if:NTF \g__zugferd_minimum_bool%
680 {\__zugferd_write_xml:n {<ram:ApplicableHeaderTradeDelivery/>}}%
681 {\__zugferd_ApplicableHeaderTradeDelivery:V \g__zugferd_delivery_date_tl}%
682 }%
683 %
684 %
685 % delivery date
686 \__zugferd_define_xml_content:Nn \__zugferd_ApplicableHeaderTradeDelivery:n {%
687 \bool_lazy_and:nnF {\prop_if_empty_p:N \g__zugferd_shipto_AddressData_prop}%
688 {\tl_if_blank_p:n {#1}}}%
689 {%
690 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeDelivery>}%
691 \prop_if_empty:Nf \g__zugferd_shipto_AddressData_prop {%
692 \int_gincr:N \g__zugferd_indent_int%
693 \__zugferd_write_xml:n {<ram:ShipToTradeParty>}%
694 \int_gincr:N \g__zugferd_indent_int%
695 \__zugferd_write_xml:e {%
696 \__zugferd_PostalTradeAddress_short:N \g__zugferd_shipto_AddressData_prop%
697 }%
698 \int_gdecr:N \g__zugferd_indent_int%
699 \__zugferd_write_xml:n {</ram:ShipToTradeParty>}%
700 \int_gdecr:N \g__zugferd_indent_int%
701 }%
702 \tl_if_blank:nF {#1} {%
703 \__zugferd_write_xml:n {%
704 \__zugferd_indent: <ram:ActualDeliverySupplyChainEvent>
705 \__zugferd_indent:n {2} <ram:OccurrenceDateTime>
706 \__zugferd_indent:n {3}<udt:DateTimeString-format="102">#1</udt:DateTimeString>
707 \__zugferd_indent:n {2} </ram:OccurrenceDateTime>
708 \__zugferd_indent: </ram:ActualDeliverySupplyChainEvent>%
709 }%
710 }%
711 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeDelivery>}%
712 }%
713 }%
714 %
715 \cctab_end:
716 \cs_generate_variant:Nn \__zugferd_TradePartyIdentity:nnnnn {eeee}
717 \cs_generate_variant:Nn \__zugferd_DefinedTradeContact:nnnn {eeee}
718 \cs_generate_variant:Nn \__zugferd_PostalTradeAddress:nnnnnn {eeeeee}
719 \cs_generate_variant:Nn \__zugferd_ApplicableHeaderTradeDelivery:n {V}

```

buyer/reference

```

720 \tl_new:N \g__zugferd_buyer_reference_tl
721 \keys_define:nn {zugferd/buyer} {
722 reference .choice:,
723 reference / komavar .code:n = {
724 \tl_gset:Nf \g__zugferd_buyer_reference_tl {\scr@yourref@var}
725 },
726 reference / unknown .code:n = {
727 \tl_gset:Nn \g__zugferd_buyer_reference_tl {#1}
728 }
729 }

```

```

730 \prop_new:N \g__zugferd_document_references_prop
731 \clist_new:N \g__zugferd_buyer_document_references_clist
732 \keys_define:nn {zugferd} {
733   contract-reference .prop_gput:N = \g__zugferd_document_references_prop,
734   buyerorder-reference .prop_gput:N = \g__zugferd_document_references_prop,
735   purchase-order-reference .meta:n = {buyerorder-reference=#1},
736   sellerorder-reference .prop_gput:N = \g__zugferd_document_references_prop,
737   sales-order-reference .meta:n = {sellerorder-reference=#1},
738 }
739 \cctab_begin:N \g__zugferd_xml_cctab%
740 \__zugferd_define_xml_writer:Nn \__zugferd_write_ReferencedDocument:n {%
741   \prop_if_in:NeT \g__zugferd_document_references_prop {\str_lowercase:n {#1}-reference} {%
742     \__zugferd_write_xml:e {%
743       <ram:#1ReferencedDocument>
744       \__zugferd_indent:<ram:IssuerAssignedID>\prop_item:Ne \g__zugferd_document_references_prop
745       reference}</ram:IssuerAssignedID>
746     }%
747   }%
748 }%
749 \cctab_end:

```

payment-means

```

750 \ExplSyntaxOff
751 \providecommand*{\zugferd@paymentMeans@german}{
752   1 = Keine Zahlungsart definiert,
753   10 = Bargeld,
754   30 = Überweisung,
755   42 = Zahlung an Bankkonto,
756   48 = Kartenzahlung,
757   49 = Lastschriftverfahren,
758   57 = Dauerauftrag,
759   58 = Zahlung per SEPA Überweisung.,
760   59 = SEPA Lastschrift,
761   97 = Ausgleich zwischen Partnern
762 }
763 \def\zugferd@paymentMeans@ngerman{\zugferd@paymentMeans@german}
764 \providecommand*{\zugferd@paymentMeans@english}{
765   1 = Instrument not defined,
766   10 = In cash,
767   30 = Credit Transfer,
768   31 = Debit Transfer,
769   42 = Payment to bank account,
770   48 = Bank card,
771   49 = Direct Debit,
772   57 = Standing agreement,
773   58 = SEPA credit transfer,
774   59 = SEPA direct debit,
775   97 = Clearing between partners
776 }
777 \ExplSyntaxOn
778 \tl_new:N \g__zugferd_payment_means_tl
779 \keys_define:nn {zugferd} {
780   payment-means / type .choices:nn = {1,10,30,42,48,49,57,58,59,97} {

```



```

781 \tl_gset_eq:NN \g__zugferd_payment_means_tl \l_keys_choice_tl
782 },
783 payment-means / unknown .code:n = {
784 \bool_if:NT \g__zugferd_unknown_value_warning_bool {
785 \msg_warning:nnn { zugferd } {unknown-value} {payment-means} {#1}
786 }
787 \tl_gset:Nn \g__zugferd_payment_means_tl {\int_eval:n {#1}}
788 }
789 }
790
791 \clist_map_inline:nn {iban,account-holder,bic} {
792 \keys_define:nn {zugferd/payment-means} {
793 #1 .tl_gset:c = {g__zugferd_payment_#1_tl}
794 }
795 }
796
797 \prop_new:c {g__zugferd_payment-means_names_prop}
798 \newcommand*{\setupZUGFeRDStrings}[2]{
799 \prop_gset_from_keyval:cn {g__zugferd_#1_names_prop} {
800 #2
801 }
802 }
803
804 \hook_gput_code:nnn {begindocument/end}{zugferd/payment-means}{
805 \prop_if_empty:cT {g__zugferd_payment-means_names_prop} {
806 \exp_args:Nne \setupZUGFeRDStrings{payment-means}{
807 \use:c {zugferd@paymentMeans@
808 \cs_if_exist:cTF {zugferd@paymentMeans@\languagename} {\languagename} {english}
809 }
810 }
811 }
812 }

```

unit

```

813 \keys_define:nn { zugferd } {
814 unit .choices:nn = {HUR,DAY,C62,H87} {
815 \tl_set_eq:NN \l__zugferd_unit_code_tl \l_keys_choice_tl
816 },
817 unit / hour .meta:n = {unit=HUR},
818 unit / day .meta:n = {unit=DAY},
819 unit / one .meta:n = {unit=C62},
820 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

```

821 unit / unknown .code:n = {
822 \tl_set:Nx \l_tmpa_tl {\str_uppercase:f {#1}}
823 \exp_args:NnnV \keys_if_choice_exist:nnnTF {zugferd} {unit}
824 \l_tmpa_tl
825 {
826 \keys_set:ne { zugferd } {unit= {\l_tmpa_tl}}
827 } {
828 \bool_if:NT \g__zugferd_unknown_value_warning_bool {
829 \msg_warning:nnn { zugferd } {unknown-unit} {#1}

```

```

830     }
831     \tl_set_eq:NN \l__zugferd_unit_code_tl \l_tmpa_tl
832   }
833 },
834 unit .usage:n = general,
835 }
836
837 \msg_new:nnnn {zugferd} {unknown-unit} {
838 You~selected~a~unit~which~was~not~predefined.\\
839 I~~will~directly~use~your~selection~'unit=#1'~as~Common~Code.
840 Please~ensure~the~selection~is~valid!
841 } {
842 For~more~information~see~the~zugferd~documentation\\
843 and~the~/UN/CEFACT~Common~Code~list.
844 }

```

category

```

exemption-reason
rate
845 \msg_new:nnnn {zugferd} {unknown-tax-category} {
846 You~selected~an~unknown~tax~category.\\
847 I~~will~directly~use~your~selection~'category=#1'~as~Code.\\
848 Please~ensure~the~selection~is~valid!
849 } {
850 For~more~information~see~the~zugferd~documentation\\
851 and~the~corresponding~code~list.
852 }
853 \msg_new:nnnn {zugferd} {no-auto-exemption} {
854 You~selected~tax/category=#1~together~with~the~auto~exemption~option.\\
855 I~don't~have~any~pre~configured~exemption~setting~for~category=#1.\\
856 Please~make~sure~you~add~a~valid~setting~yourself.
857 } {
858 For~more~information~see~the~zugferd~documentation.
859 }
860 \keys_define:nn { zugferd / tax } {
861 category .choices:nn = {S,Z,E,AE,K,G,O,L,M} {
862 \tl_set_eq:NN \l__zugferd_tax_category_code_tl \l_keys_choice_tl
863 \bool_if:NT \g__zugferd_auto_exemption_bool {
864 \keys_if_choice_exist:nnTF {zugferd/tax} {exemption-reason-auto} {#1} {
865 \keys_set:nn {zugferd/tax} {exemption-reason-auto=#1}
866 } {
867 \msg_warning:nnn { zugferd } {no-auto-exemption} {#1}
868 }
869 }
870 },

```

```

871 exemption-reason .tl_set:N = \l__zugferd_tax_exemption_reason_tl,
872 exemption-reason .initial:V = \c_empty_tl,
873 exemption-reason .usage:n = general,
874 exemption-reason-code .tl_set:N = \l__zugferd_tax_exemption_code_tl,
875 exemption-reason-code .initial:V = \c_empty_tl,
876 exemption-reason .usage:n = general,

```

```

877 exemption-reason-auto .choice:,
878 exemption-reason-auto / S .code:n = {
879 \keys_set:nn {zugferd/tax} {exemption-reason=,exemption-reason-code=}
880 },

```

```

881 exemption-reason-auto / K .code:n = {
882   \keys_set:nn {zugferd/tax} {
883     exemption-reason= Intra-Community-Supply,
884     exemption-reason-code={vatex-eu-ic}
885   }
886 },
887 exemption-reason-auto / AE .code:n = {
888   \keys_set:nn {zugferd/tax}{
889     exemption-reason=Reverse-Charge,
890     exemption-reason-code={vatex-eu-ae}
891   }
892 },
893 exemption-reason-auto / G .code:n = {
894   \keys_set:nn {zugferd/tax}{
895     exemption-reason=Export-outside-the-EU,
896     exemption-reason-code={vatex-eu-g}
897   }
898 },
899 exemption-reason-auto / O .code:n = {
900   \keys_set:nn {zugferd/tax}{
901     exemption-reason=No-subject-to-VAT,
902     exemption-reason-code={vatex-eu-o}
903   }
904 },
905 standard .meta:n = {category=S},
906 zero .meta:n = {category=Z},
907 exempt .meta:n = {category=E},
908 reverse-charge .meta:n = {category=AE},
909 intra-community .meta:n = {category=K},
910 EEA .meta:n = {category=K},
911 export .meta:n = {category=G},
912 canary-islands .meta:n = {category=L},
913 ceuta .meta:n = {category=M},
914 melilla .meta:n = {category=M},
915 category / unknown .code:n = {
916   \exp_args:Nnnx \keys_if_choice_exist:nnnTF {zugferd} {category}
917   {\str_uppercase:f {#1}}
918   {
919     \keys_set:nx { zugferd } {category= {\str_uppercase:f {#1}}}
920   } {
921     \tl_set:Ne \l__zugferd_tax_category_code_tl {\str_uppercase:f {#1}}
922     \bool_if:NT \g__zugferd_unknown_value_warning_bool {
923       \msg_warning:nnn { zugferd } {unknown-tax-category} {#1}
924     }
925   }
926 },
927 category .initial:n = S,
928 category .usage:n = general,
929 rate .fp_set:N = \l__zugferd_tax_rate_fp,
930 rate .initial:n = 19,
931 rate .usage:n = general
932 }%
933 \keys_define:nn {zugferd/item} {
934   tax .choice:,

```

```

935 tax / unknown .code:n = \keys_set:ne {zugferd/tax} { \l_keys_key_str = \exp_not:V \l_keys_v
936 unknown .code:n = \keys_set:ne {zugferd} { \l_keys_key_str = \exp_not:V \l_keys_value_tl}
937 }

```

start-date

end-date

```

938 \keys_define:nn { zugferd / item } {
939 start-date .tl_gset:N = \l__zugferd_start_date_tl,
940 start-date .initial:n =,
941 end-date .tl_gset:N = \l__zugferd_end_date_tl,
942 end-date .initial:n =,
943 }%
944 \__zugferd_define_xml_content:Nn \__zugferd_write_inline:nn {
945 \__zugferd_write_inline:nnn {#1} {} {#2}
946 }
947 \__zugferd_define_xml_content:Nn \__zugferd_write_inline:nnn {
948 \tl_if_blank:nF {#3} {
949 <#1#2>#3</#1>
950 }
951 }
952 \__zugferd_define_xml_content:Nn \__zugferd_write_inline_wrapped:nnnn {
953 \__zugferd_indent:n {#1} <#2>\__zugferd_xml_newline_indent:
954 \__zugferd_indent:n {#1+1} <#3>#4</#3>\__zugferd_xml_newline_indent:
955 \__zugferd_indent:n {#1} </#2>
956 }
957 \__zugferd_define_xml_content:Nn \__zugferd_write_inline_wrapped_i:nnn {
958 \__zugferd_write_inline_wrapped:nnnn {1} {#1} {#2} {#3}
959 }
960
961 \cs_generate_variant:Nn \__zugferd_write_inline:nn {ne}
962 \__zugferd_define_xml_content:Nn \__zugferd_write_inline_i:nnn {
963 \__zugferd_indent: \__zugferd_write_inline:nnn {#1} {#2} {#3}
964 }
965 \__zugferd_define_xml_content:Nn \__zugferd_write_inline_i:nn {
966 \__zugferd_write_inline_i:nnn {#1} {} {#2}
967 }
968 \cctab_begin:N \g__zugferd_xml_cctab%
969 %
970 \bool_if:NTF \g__zugferd_minimum_bool {%
971 \cs_set_eq:NN \__zugferd_write_note:n \use_none:n%
972 } {%
973 \__zugferd_define_xml_writer:Nn \__zugferd_write_note:n {%
974 \begingroup%
975 \let\\\iow_newline:%
976 \__zugferd_write_xml:e {%
977 <ram:IncludedNote>
978 \__zugferd_indent: <ram:Content>
979 #1
980 \__zugferd_indent: </ram:Content>
981 </ram:IncludedNote>%
982 }%
983 \endgroup%
984 }%
985 }%
986 %

```

```

987 \begingroup%
988 \char_set_catcode_other:N \#%
989 \char_set_catcode:nn {32}{10}%
990 \__zugferd_define_xml_writer:Nn \__zugferd_insert_Header: {%
991 \__zugferd_write_xml:e {%
992 <?xml version='1.0' encoding='UTF-8' ?>
993 <rsm:CrossIndustryInvoice %
994 xmlns:rsm="urn:un:unece:unefact:data:standard:CrossIndustryInvoice:100" %
995 xmlns:qdt="urn:un:unece:unefact:data:standard:QualifiedDataType:100" %
996 xmlns:ram="urn:un:unece:unefact:data:standard:ReusableAggregateBusinessInformationEntity:100" %
997 xmlns:xs="http://www.w3.org/2001/XMLSchema" %
998 xmlns:udt="urn:un:unece:unefact:data:standard:UnqualifiedDataType:100">%
999 }%
1000 \int_gincr:N \g__zugferd_indent_int%
1001 \__zugferd_write_xml:n {<rsm:ExchangedDocumentContext>}%
1002 \int_gincr:N \g__zugferd_indent_int%
1003 \__zugferd_write_xml:e {%
1004 \str_if_empty:NF \g__zugferd_businessProcessId_str {%
1005 <ram:BusinessProcessSpecifiedDocumentContextParameter>
1006 \__zugferd_indent: <ram:ID>\g__zugferd_businessProcessId_str</ram:ID>
1007 </ram:BusinessProcessSpecifiedDocumentContextParameter>
1008 }%
1009 <ram:GuidelineSpecifiedDocumentContextParameter>
1010 \__zugferd_indent: <ram:ID>\g__zugferd_format_str</ram:ID>
1011 </ram:GuidelineSpecifiedDocumentContextParameter>%
1012 }%
1013 \int_gdecr:N \g__zugferd_indent_int%
1014 \__zugferd_write_xml:n {</rsm:ExchangedDocumentContext>}%
1015 }%
1016 \endgroup%
1017 \__zugferd_define_xml_writer:Nn \__zugferd_insert_FrontMatter: {%
1018 \__zugferd_write_xml:n {<rsm:ExchangedDocument>}%
1019 \int_gincr:N \g__zugferd_indent_int%
1020 \__zugferd_write_xml:e {%
1021 <ram:ID>\g__zugferd_id_tl</ram:ID>
1022 <ram:TypeCode>\g__zugferd_DocumentTypeCode_tl</ram:TypeCode>
1023 <ram:IssueDateTime>
1024 % space required!
1025 \__zugferd_indent:<udt:DateTimeString~format="102">\g__zugferd_date_tl</udt:DateTimeString>
1026 </ram:IssueDateTime>%
1027 }%
1028 \__zugferd_note_if_not_empty:N \g__zugferd_subject_tl%
1029 \__zugferd_note_if_not_empty:N \g__zugferd_fromaddress_tl%
1030 \seq_map_inline:Nn \g__zugferd_notes_seq {%
1031 \__zugferd_write_note:n {##1}%
1032 }%
1033 \int_gdecr:N \g__zugferd_indent_int%
1034 \__zugferd_write_xml:e {%
1035 </rsm:ExchangedDocument>
1036 <rsm:SupplyChainTradeTransaction>%
1037 }%
1038 \int_gincr:N \g__zugferd_indent_int%
1039 }%
1040 %

```

```

1041 % footer
1042 \__zugferd_define_xml_writer:Nn \__zugferd_insert_Footer: {%
1043 \int_gdecr:N \g__zugferd_indent_int%
1044 \__zugferd_write_xml:n {</rsm:SupplyChainTradeTransaction>}%
1045 \int_gdecr:N \g__zugferd_indent_int%
1046 \__zugferd_write_xml:n {</rsm:CrossIndustryInvoice>}%
1047 }%
1048 \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-

```

1049 \cctab_begin:N \g__zugferd_xml_cctab%

```

```

\underline{\__zugferd_AssociatedDocumentLineDocument:n}

```

```

1050 \__zugferd_define_xml_writer:Nn \__zugferd_AssociatedDocumentLineDocument:n {%
1051 <ram:AssociatedDocumentLineDocument>
1052 \__zugferd_indent: <ram:LineID>#1</ram:LineID>
1053 </ram:AssociatedDocumentLineDocument>%
1054 }%

```

```

\underline{\__zugferd_SpecifiedTradeProduct:nn}

```

```

1055 \__zugferd_define_xml_content:Nn \__zugferd_SpecifiedTradeProduct:nn {%
1056 <ram:SpecifiedTradeProduct>
1057 \bool_lazy_or:nnF%
1058 {\tl_if_blank_p:n {#1}}%
1059 {\str_if_eq_p:Vn \g__zugferd_conformance_level_str {BASIC}} {%
1060 \__zugferd_indent: <ram:SellerAssignedID>#1</ram:SellerAssignedID>
1061 }%
1062 \__zugferd_indent: <ram:Name>#2</ram:Name>
1063 </ram:SpecifiedTradeProduct>%
1064 }%

```

_zugferd_ProductTradePrice:nn

```
1065 \\_zugferd_define_xml_writer:Nn \\_zugferd_ProductTradePrice:nn {%
1066 <ram:\str_uppercase:n #1PriceProductTradePrice>
1067 \\_zugferd_indent: <ram:ChargeAmount>#2</ram:ChargeAmount>
1068 </ram:\str_uppercase:n #1PriceProductTradePrice>%
1069 }%
```

_zugferd_SpecifiedLineTradeAgreement:nn

```
1070 \\_zugferd_define_xml_writer:Nn \\_zugferd_SpecifiedLineTradeAgreement:nn {%
1071 <ram:SpecifiedLineTradeAgreement>
1072 \\_zugferd_indent:<ram:GrossPriceProductTradePrice>
1073 \\_zugferd_indent:n {2} <ram:ChargeAmount>#1</ram:ChargeAmount>
1074 \\_zugferd_indent:</ram:GrossPriceProductTradePrice>
1075 \\_zugferd_indent:<ram:NetPriceProductTradePrice>
1076 \\_zugferd_indent:n {2} <ram:ChargeAmount>#2</ram:ChargeAmount>
1077 \\_zugferd_indent:</ram:NetPriceProductTradePrice>
1078 </ram:SpecifiedLineTradeAgreement>
1079 }%
```

6.4.1 SpecifiedLineTradeDelivery

:nn <unit code > <number>

_zugferd_SpecifiedLineTradeDelivery:nn

```
1080 \\_zugferd_define_xml_content:Nn \\_zugferd_SpecifiedLineTradeDelivery:nn {%
1081 <ram:SpecifiedLineTradeDelivery>
1082 % SPACE!
1083 \\_zugferd_indent: <ram:BilledQuantity~unitCode="#1">#2</ram:BilledQuantity>
1084 </ram:SpecifiedLineTradeDelivery>
1085 }%
```

```
\__zugferd_SpecifiedLineTradeSettlement:nnn
\__zugferd_SpecifiedLineTradeSettlement:Vnn
```

```
1086 \__zugferd_define_xml_writer:Nn \__zugferd_write_SpecifiedLineTradeSettlement:nnn {%
1087 \__zugferd_write_xml:n {<ram:SpecifiedLineTradeSettlement>}%
1088 \int_gincr:N \g__zugferd_indent_int%
1089 \__zugferd_write_xml:e {\__zugferd_Line_ApplicableTradeTax:nn {#1} {#2}}%
1090 \__zugferd_write_xml:e {\__zugferd_BillingSpecifiedPeriod:VV \l__zugferd_start_date_tl \l
1091 \__zugferd_write_xml:e {\__zugferd_SpecifiedTradeSettlementLineMonetarySummation:n {#3}}%
1092 \int_gdecr:N \g__zugferd_indent_int%
1093 \__zugferd_write_xml:n {</ram:SpecifiedLineTradeSettlement>}%
1094 }%
1095 \cs_generate_variant:Nn \__zugferd_write_SpecifiedLineTradeSettlement:nnn {Vnn}%
1096 \__zugferd_define_xml_content:Nn \__zugferd_Line_ApplicableTradeTax:nn {%
1097 <ram:ApplicableTradeTax>
1098 %BT-151
1099 \__zugferd_indent: <ram:TypeCode>VAT</ram:TypeCode>
1100 %BT-151
1101 \__zugferd_indent: <ram:CategoryCode>#1</ram:CategoryCode>
1102 %BT-152
1103 \__zugferd_indent: <ram:RateApplicablePercent>#2</ram:RateApplicablePercent>
1104 </ram:ApplicableTradeTax>
1105 }%
1106 \__zugferd_define_xml_content:Nn \__zugferd_BillingSpecifiedPeriod:nn {%
1107 \bool_lazy_or:nnF {\tl_if_blank_p:n {#1}} {\tl_if_blank_p:n {#2}} {%
1108 <ram:BillingSpecifiedPeriod>
1109 %<!-- BT-134 -->
1110 \__zugferd_indent:<ram:StartDateTime>
1111 \__zugferd_indent:n {2} <udt:DateTimeString~format="102">#1</udt:DateTimeString>
1112 \__zugferd_indent:</ram:StartDateTime>
1113 %<!-- BT-135 -->
1114 \__zugferd_indent:<ram:EndDateTime>
1115 \__zugferd_indent:n {2} <udt:DateTimeString~format="102">#2</udt:DateTimeString>
1116 \__zugferd_indent:</ram:EndDateTime>
1117 </ram:BillingSpecifiedPeriod>%
1118 }%
1119 }%
1120 \__zugferd_define_xml_content:Nn \__zugferd_SpecifiedTradeSettlementLineMonetarySummation:n
1121 <ram:SpecifiedTradeSettlementLineMonetarySummation>
1122 % BT-131
1123 \__zugferd_indent: <ram:LineTotalAmount>#1</ram:LineTotalAmount>
1124 </ram:SpecifiedTradeSettlementLineMonetarySummation>%
1125 }
1126
1127 \cctab_end:
1128 \cs_generate_variant:Nn \__zugferd_SpecifiedLineTradeSettlement:nnn {Vnn}
1129 \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.

```
1130 \cs_new:Nn \__zugferd_SpecifiedLineTradeSettlement:nnnn {
1131 \msg_warning:nnnn {zugferd} {macro-deprecated}
```



```

1132 { \__zugferd_SpecifiedLineTradeSettlement:nnnn }
1133 { \__zugferd_SpecifiedLineTradeSettlement:nnn }
1134 \__zugferd_SpecifiedLineTradeSettlement:nnn {#2} {#3} {#4}
1135 }
1136 \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.

```

1137 \__zugferd_define_xml_writer:Nn \__zugferd_insert_TradeLineItem:nnnnnn {
1138 \__zugferd_write_xml:n {<ram:IncludedSupplyChainTradeLineItem>}
1139 \int_gincr:N \g__zugferd_indent_int
1140 \__zugferd_write_xml:e {\__zugferd_AssociatedDocumentLineDocument:n {#1}}%
1141
1142 \begingroup
1143 \zugferd_disable_macros:
1144 \__zugferd_write_xml:e {\__zugferd_SpecifiedTradeProduct:nn {#2} {#3}}
1145 \endgroup
1146
1147 \__zugferd_write_xml:n {<ram:SpecifiedLineTradeAgreement>}
1148 \int_gincr:N \g__zugferd_indent_int%
1149 \__zugferd_write_xml:e {\__zugferd_ProductTradePrice:nn {net} {#4}}
1150 \int_gdecr:N \g__zugferd_indent_int%
1151 \__zugferd_write_xml:n {</ram:SpecifiedLineTradeAgreement>}
1152 \__zugferd_write_xml:e {%
1153 \__zugferd_SpecifiedLineTradeDelivery:nn { \l__zugferd_unit_code_tl } {#5}
1154 }%
1155 \__zugferd_write_SpecifiedLineTradeSettlement:Vnn \l__zugferd_tax_category_code_tl %
1156 { \fp_use:N \l__zugferd_tax_rate_fp } {#6}%
1157 \int_gdecr:N \g__zugferd_indent_int%
1158 \__zugferd_write_xml:n {</ram:IncludedSupplyChainTradeLineItem>}
1159 }

```

ApplicableHeaderTradeSettlement

```

1160 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeSettlementHeaderMonetarySummation:nnnnnn
1161 \__zugferd_write_xml:n {<ram:SpecifiedTradeSettlementHeaderMonetarySummation>}%
1162 \int_gincr:N \g__zugferd_indent_int%
1163 \bool_if:NF \g__zugferd_minimum_bool {
1164 \__zugferd_write_rounded:nnn {2} {LineTotalAmount} {#1}
1165 \__zugferd_write_rounded:nnn {2} {ChargeTotalAmount} {#2}
1166 \__zugferd_write_rounded:nnn {2} {AllowanceTotalAmount} {#3}
1167 }
1168 \__zugferd_write_rounded:nnn {2} {TaxBasisTotalAmount} {#4}
1169 \__zugferd_write_rounded:nnnn {2} {TaxTotalAmount} {~currencyID="\l__zugferd_currency_tl "}
1170 \__zugferd_write_rounded:nnn {2} {GrandTotalAmount} {#6}
1171 \bool_if:NF \g__zugferd_minimum_bool {
1172 \__zugferd_write_rounded:nnn {2} {TotalPrepaidAmount} {#7}
1173 }
1174 \__zugferd_write_rounded:nnn {2} {DuePayableAmount} {#8}
1175 \int_gdecr:N \g__zugferd_indent_int%
1176 \__zugferd_write_xml:n {</ram:SpecifiedTradeSettlementHeaderMonetarySummation>}%
1177 }%
1178 % ApplicableTradeTax CategoryCode Rate BaseAmount Result

```

```

1179 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableTradeTax:nnnn {%
1180 \__zugferd_write_xml:n {<ram:ApplicableTradeTax>}
1181 \int_gincr:N \g__zugferd_indent_int%
1182 \__zugferd_write_rounded:nnn {2} {CalculatedAmount} {#4}%BT117
1183 \__zugferd_write_xml:n {<ram:TypeCode>VAT</ram:TypeCode>}%BT118

1184 %
1185 \begingroup
1186 \bool_if:NT \g__zugferd_auto_exemption_bool {%
1187 \keys_set:nn { zugferd / tax } {exemption-reason-auto=#1}
1188 }%
1189 \tl_if_blank:VF \l__zugferd_tax_exemption_reason_tl {%BT-120
1190 \__zugferd_write_xml:e {<ram:ExemptionReason>
1191 \l__zugferd_tax_exemption_reason_tl
1192 </ram:ExemptionReason>}
1193 }
1194 {{\__zugferd_write_rounded:nnn {2} {BasisAmount} {#3}}}%BT-116
1195 \__zugferd_write_xml:e {<ram:CategoryCode>#1</ram:CategoryCode>}%BT-118

1196 \bool_lazy_or:nnF% BT121
1197 {\tl_if_blank_p:V \l__zugferd_tax_exemption_code_tl}%
1198 {\str_if_eq_p:Vn \g__zugferd_conformance_level_str {BASIC}}%
1199 {%
1200 \__zugferd_write_xml:e {%
1201 <ram:ExemptionReasonCode>\l__zugferd_tax_exemption_code_tl</ram:ExemptionReasonCode>%
1202 }%
1203 }%
1204 \endgroup
1205 \__zugferd_write_xml:n {<ram:RateApplicablePercent>#2</ram:RateApplicablePercent>}%BT-
119
1206 \int_gdecr:N \g__zugferd_indent_int%
1207 \__zugferd_write_xml:n {</ram:ApplicableTradeTax>}
1208 }%
1209 %
1210 \cctab_begin:N \g__zugferd_xml_cctab%
1211 % ApplicableHeaderTradeSettlement

```

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

```

1212 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeSettlement_start: {%
1213 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeSettlement>}%
1214 \int_gincr:N \g__zugferd_indent_int%
1215 \__zugferd_write_xml:e {<ram:InvoiceCurrencyCode>\l__zugferd_currency_tl</ram:InvoiceCurren
1216 }%
1217 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeSettlement_stop: {%
1218 \int_gdecr:N \g__zugferd_indent_int%
1219 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeSettlement>}%
1220 }%

```

SpecifiedTradePaymentTerms

```

1221 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradePaymentTerms:nn {%
1222 \__zugferd_write_xml:n {<ram:SpecifiedTradePaymentTerms>}%
1223 \int_gincr:N \g__zugferd_indent_int%
1224 \__zugferd_write_xml:e {%
1225 \tl_if_blank:nF {#1} {%

```

```

1226     <ram:Description>#1</ram:Description>%
1227   }%
1228   \tl_if_blank:nF {#2} {%
1229     <ram:DueDateDateTime>
1230     \__zugferd_indent: <udt:DateTimeString~format="102">#2</udt:DateTimeString>
1231     </ram:DueDateDateTime>%
1232   }%
1233 }%
1234 \int_gdecr:N \g__zugferd_indent_int%
1235 \__zugferd_write_xml:n {</ram:SpecifiedTradePaymentTerms>}%
1236 }%
1237 %
1238 %
1239 % sums
1240 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeSettlementPaymentMeans:nnn {%
1241 \bool_if:NT \g__zugferd_writePaymentMeans_bool {%
1242   \__zugferd_write_xml:n {<ram:SpecifiedTradeSettlementPaymentMeans>}%
1243   \int_gincr:N \g__zugferd_indent_int%
1244   \__zugferd_write_xml:e {%
1245     <ram:TypeCode>\g__zugferd_payment_means_tl</ram:TypeCode>
1246     \__zugferd_write_inline:ne {ram:Information} {\prop_item:cV {g__zugferd_payment-
means_names_prop} \g__zugferd_payment_means_tl}%
1247     \tl_if_blank:nF {#1#2} {%
1248       <ram:PayeePartyCreditorFinancialAccount>
1249       \__zugferd_write_inline_i:nn {ram:IBANID} {#2}
1250       \__zugferd_write_inline_i:nn {ram:AccountName} {#1}
1251       </ram:PayeePartyCreditorFinancialAccount>
1252     }%
1253     \tl_if_blank:nF {#3} {%
1254       <ram:PayeeSpecifiedCreditorFinancialInstitution>
1255       \__zugferd_write_inline_i:nn {ram:BICID} {#3}
1256       </ram:PayeeSpecifiedCreditorFinancialInstitution>%
1257     }%
1258   }%
1259   \int_gdecr:N \g__zugferd_indent_int%
1260   \__zugferd_write_xml:n {</ram:SpecifiedTradeSettlementPaymentMeans>}%
1261 }%
1262 }%
1263 \cctab_end:
1264 \cs_generate_variant:Nn \__zugferd_SpecifiedTradePaymentTerms:nn {VV}
1265 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradePaymentTerms: {
1266 \bool_if:NF \g__zugferd_minimum_bool {
1267   \__zugferd_SpecifiedTradePaymentTerms:VV \g__zugferd_payment_terms_str \g__zugferd_due_dat
1268 }
1269 }

```

_zugferd_SpecifiedTradeSettlementPaymentMeans:

```

1270 \cs_generate_variant:Nn \_zugferd_SpecifiedTradeSettlementPaymentMeans:nnn {vvv}
1271 \_zugferd_define_xml_writer:Nn \_zugferd_SpecifiedTradeSettlementPaymentMeans: {
1272 \tl_if_blank:VF \g_zugferd_payment_means_tl {
1273 \_zugferd_SpecifiedTradeSettlementPaymentMeans:vvv
1274 {g_zugferd_payment_account-holder_tl} {g_zugferd_payment_iban_tl} {g_zugferd_payment_b
1275 }
1276 }

1277 \msg_new:nnn {zugferd} {macro-deprecated} {
1278 The-function-#1-is-deprecated.\\
1279 It-was-replaced-by-#2.\\
1280 Please-adjust-your-mechanism-to-use-the-new-version.
1281 }

```

Change History

v0.4	file for the zugferd profile.	20
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 auto-exemption option Added exemption-reason-auto key for pre-configured exemption-reasons.	15 34
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
xml-file: minimum		14
v0.9b	General: Fix file name of embedded	
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
v0.9d	General: Add note on subject not being the same as subject-code #35.	9
Added unknown-value-warning option		15

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] Sebastian Rahtz, Heiko Oberdiek, and The L^AT_EX Project. *CTAN: Package hyperref*. Nov. 5, 2024. URL: <https://ctan.org/pkg/hyperref> (visited on 01/02/2025).
- [10] *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).
- [11] *VAT exemption reason code list*. URL: https://www.xrepository.de/details/urn:xoev-de:kosit:codeliste:vatex_1 (visited on 08/20/2024).
- [12] *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	
\#	9, 73, 357, 988
\&	375, 384
\\	80, 160, 223, 484, 499, 508, 838, 842, 846, 847, 850, 854, 855, 975, 1278, 1279
\{	353
\}	355
<i><party>/address</i> (option)	10
<i><party>/city</i> (option)	11
<i><party>/contact</i> (option)	11
<i><party>/country</i> (option)	11
<i><party>/email</i> (option)	10
<i><party>/name</i> (option)	10
<i><party>/postcode</i> (option)	11
<i><party>/subdivision</i> (option)	11
<i><party>/taxid</i> (option)	10
<i><party>/vatid</i> (option)	10
<i><party>/address</i> (option)	<u>495</u>

A	
add-note (option)	9, 472
AddressData (option)	4
auto-exemption (option)	4, 66
B	
\begin	393
\begingroup	214, 974, 987, 1142, 1185
bool commands:	
\bool_gset_false:N	37, 38, 47, 48, 180
\bool_gset_true:N	12, 13, 46
\bool_if:NTF	
.	77, 182, 184, 215, 229, 232, 249,
.	257, 271, 292, 305, 443, 588, 596,
.	601, 631, 667, 679, 784, 828, 863,
.	922, 970, 1163, 1171, 1186, 1241, 1266
\bool_lazy_and:nnTF	644, 687
\bool_lazy_or:nnTF	1057, 1107, 1196
\bool_new:N	4, 5, 6
\bool_set_true:N	174
\g_tmpa_bool	174, 180, 182, 184
buyer/reference (option)	11, 720
C	
category (option)	845
cctab commands:	
\cctab_begin:N	569, 739, 968, 1049, 1210
\cctab_end:	715, 749, 1048, 1127, 1263
\cctab_select:N	364
char commands:	
\char_set_active_eq:NN	
.	380, 381, 382, 383, 384
\char_set_active_eq:nN	216
\char_set_catcode:nn	989
\char_set_catcode_active:N	
.	387, 388, 389, 390, 391
\char_set_catcode_other:N	9, 988
\char_set_catcode_parameter:N	73
clist commands:	
\clist_map_break:	182, 184
\clist_map_inline:nn	
.	158, 175, 176, 495, 674, 791
\clist_new:N	731
cs commands:	
\cs_generate_variant:Nn	193, 303,
.	304, 317, 331, 716, 717, 718, 719,
.	961, 1095, 1128, 1129, 1136, 1264, 1270
\cs_gset:Nn	250, 254, 263, 268, 276
\cs_gset_eq:NN	258, 260, 272,
.	274, 284, 285, 286, 287, 288, 289, 290
\cs_if_exist:NTF	78, 808
\cs_new:Nn	190, 248, 283, 297, 306,
.	309, 310, 313, 314, 319, 332, 336,
.	340, 343, 346, 349, 352, 359, 370,
.	374, 386, 490, 530, 538, 550, 562, 1130
\cs_new:Npn	194
\cs_new_eq:NN	309, 310
\cs_set:Nn	313, 314, 394, 398
\cs_set_eq:NN	312, 602, 971
currency (option)	9
D	
date (option)	9, 450
\day	197, 454, 462
\def	763
delivery-date (option)	9, 450
document-type (option)	8, 412
\DocumentMetadata	81
due-date (option)	9
E	
\edef	375, 376, 377, 378, 379
end-date (option)	938
\endgroup	231, 983, 1016, 1145, 1204
environments:	
ZUGFeRD	5
exemption-reason (option)	845
exp commands:	
\exp_args:Ne	132, 539
\exp_args:Nne	806
\exp_args:NnnV	823
\exp_args:Nnnx	438, 916
\exp_args:NnV	237
\exp_not:n	935, 936
\ExplSyntaxOff	750
\ExplSyntaxOn	777
F	
format (option)	3, 9
fp commands:	
\fp_eval:n	395, 399
\fp_gset:Nn	400
\fp_set:Nn	396
\fp_use:N	1156
fromaddress (option)	9, 472
G	
group commands:	
\group_begin:	233, 298, 360
\group_end:	239, 301, 368
H	
hook commands:	
\hook_gput_code:mnn	804
I	
id (option)	8, 412
\IfBooleanTF	146

\IfNoValueF	200	\msg_info:nn	219
\IfPackageAtLeastTF	87	\msg_new:nnn	88, 222, 1277
\ignorespaces	206	\msg_new:nnnn	79, 483, 837, 845, 853
\InsertZUGFeRData	4	\msg_warning:nnn	444, 785, 829, 867, 923
\InsertZUGFeRData	18, 155, 188, 189	\msg_warning:nnnn	1131
\InsertZugferdData	18, 188		
\insertZugferdData	189		
int commands:		N	
\int_compare:nNnTF	453, 454, 461, 462, 509	\NewDocumentCommand	145, 155
\int_eval:n	787	\NewDocumentEnvironment	199
\int_gdecr:N	659, 672, 677, 698, 700, 1013, 1033, 1043, 1045, 1092, 1150, 1157, 1175, 1206, 1218, 1234, 1259	\NewExpandableDocumentCommand	152
\int_gincr:N	643, 653, 664, 692, 694, 1000, 1002, 1019, 1038, 1088, 1139, 1148, 1162, 1181, 1214, 1223, 1243		
\int_new:N	339	O	
\int_set:Nn	195, 196, 197	options:	
iow commands:		<party>/address	10
\iow_close:N	230	<party>/city	11
\iow_new:N	76	<party>/contact	11
\iow_newline:	134, 136, 138, 141, 350, 540, 975	<party>/country	11
\iow_now:Nn	307	<party>/email	10
\iow_open:Nn	217	<party>/name	10
item/end-date (option)	14	<party>/postcode	11
item/start-date (option)	14	<party>/subdivision	11
		<party>/taxid	10
		<party>/vatid	10
		<party>/address	495
		add-note	9, 472
		AddressData	4
		auto-exemption	4, 66
		buyer/reference	11, 720
		category	845
		currency	9
		date	9, 450
		delivery-date	9, 450
		document-type	8, 412
		due-date	9
		end-date	938
		exemption-reason	845
		format	3, 9
		fromaddress	9, 472
		id	8, 412
		item/end-date	14
		item/start-date	14
		payment-means	750
		payment-terms	9, 470
		rate	845
		set-today	4
		start-date	938
		subject	9, 472
		tax/category	13
		tax/exemption-reason	13
		tax/exemption-reason-code	13
		tax/rate	14
		unit	13, 813
		unknown-value-warning	4, 70
		write-xml	3, 9
J			
\jobname	64		
K			
keys commands:			
\l_keys_choice_tl	434, 781, 815, 862		
\keys_define:nn	10, 410, 497, 721, 732, 779, 792, 813, 860, 933, 938		
\keys_if_choice_exist:nnnTF	438, 823, 864, 916		
\l_keys_key_str	935, 936		
\l_keys_key_tl	526		
\keys_set:nn	25, 31, 147, 148, 299, 441, 501, 510, 517, 826, 865, 879, 882, 888, 894, 900, 919, 935, 936, 1187		
\l_keys_value_tl	935, 936		
L			
\languagename	808		
\LaTeX	80, 89		
\let	151, 353, 354, 355, 356, 357, 975		
M			
\month	196, 453, 461		
msg commands:			
\msg_error:nn	85, 92		

xml-file	3, 9	<code>\seq_set_split:Nnn</code>	499, 508
xrechnung	9	<code>\l_tmpa_seq</code>	499, 502, 503, 504, 508, 509, 511, 512, 513, 514, 518, 519, 520
zugferd	3, 9	<code>set-today (option)</code>	4
P			
payment-means (option)	750	<code>\setupZUGFeRDStrings</code>	798, 806
payment-terms (option)	9, 470	<code>\SetZUGFeRDData</code>	4
pdf commands:		<code>\SetZUGFeRDData</code>	8, 17, 145, 151, 201
<code>\pdf_object_ref:n</code>	243, 245	<code>\SetZugferdData</code>	151
pdfdict commands:		<code>\sisetup</code>	320
<code>\pdfdict_put:nnn</code>	234, 236	siunitx commands:	
pdfmfile commands:		<code>\siunitx_number_format:nN</code>	329
<code>\pdfmfile_embed_file:nnn</code>	237	<code>\space</code>	164, 341
pdfmanagement commands:		<code>start-date (option)</code>	938
<code>\pdfmanagement_add:nnn</code>	240, 245	<code>\startWritingZUGFeRDxml</code>	5
pdfmeta commands:		<code>\startWritingZUGFeRDxml</code>	5, 204, 213
<code>\pdfmeta_xmp_add:n</code>	132	<code>\stopWritingZUGFeRDxml</code>	5
<code>\pdfmeta_xmp_property_new:nnnn</code>	104, 111, 118, 125	<code>\stopWritingZUGFeRDxml</code>	5, 210, 228
<code>\pdfmeta_xmp_schema_new:nnn</code>	99	str commands:	
<code>\pdfmeta_xmp_xmlns_new:nn</code>	78, 95	<code>\c_hash_str</code>	96, 102, 357
prg commands:		<code>\c_left_brace_str</code>	353, 354
<code>\prg_do_nothing:</code>	284, 285, 287, 288	<code>\c_right_brace_str</code>	355, 356
<code>\prg_replicate:nn</code>	344	<code>\str_case:nnTF</code>	156
<code>\ProcessKeyOptions</code>	75	<code>\c_str_cctab</code>	364
prop commands:		<code>\str_gset:Nn</code>	14, 19, 22, 28, 34, 39, 43, 49
<code>\prop_gput:Nnn</code>	525	<code>\str_if_empty:NTF</code>	1004
<code>\prop_gset_from_keyval:Nn</code>	799	<code>\str_if_eq_p:nn</code>	1059, 1198
<code>\prop_if_empty:NTF</code>	691, 805	<code>\str_lowercase:n</code>	741, 744
<code>\prop_if_empty_p:N</code>	687	<code>\str_new:N</code>	2, 3, 7
<code>\prop_if_in:NnTF</code>	159, 741	<code>\str_replace_all:Nnn</code>	172, 173
<code>\prop_item:Nn</code>	160, 163, 165, 532, 533, 534, 535, 536, 539, 540, 542, 543, 544, 545, 546, 547, 548, 552, 553, 554, 555, 556, 557, 558, 559, 560, 564, 565, 566, 567, 744, 1246	<code>\str_set:Nn</code>	171
<code>\prop_new:N</code>	496, 730, 797	<code>\str_uppercase:n</code>	439, 441, 446, 822, 917, 919, 921, 1066, 1068
<code>\providecommand</code>	188, 189, 751, 764	<code>\l_tmpa_str</code>	171, 172, 173, 177, 179
Q			
quark commands:		<code>\string</code>	81
<code>\q_stop</code>	191, 194	<code>subject (option)</code>	9, 472
R			
rate (option)	845	T	
<code>\RequirePackage</code>	318	<code>tax/category (option)</code>	13
S			
seq commands:		<code>tax/exemption-reason (option)</code>	13
<code>\seq_count:N</code>	509	<code>tax/exemption-reason-code (option)</code>	13
<code>\seq_gput_right:Nn</code>	481	<code>tax/rate (option)</code>	14
<code>\seq_item:Nn</code>	502, 503, 504, 511, 512, 513, 514, 518, 519, 520	\TeX and $\LaTeX 2_{\epsilon}$ commands:	
<code>\seq_map_inline:Nn</code>	1030	<code>\scr@fromaddress@var</code>	479
<code>\seq_new:N</code>	402	<code>\scr@invoice@var</code>	431
		<code>\scr@subject@var</code>	474
		<code>\scr@yourref@var</code>	724
		<code>\zugferd@paymentMeans@english</code>	764
		<code>\zugferd@paymentMeans@german</code>	751, 763
		<code>\zugferd@paymentMeans@ngerman</code>	763
		<code>\textbraceleft</code>	354
		<code>\textbraceright</code>	356
		<code>\textgreater</code>	376, 380
		<code>\textless</code>	377, 381
		<code>\textquotedbl</code>	378, 382

`\textquotesingle` 379, 383
`\the` 452, 453, 454, 460, 461, 462
tl commands:
`\c_empty_tl` 872, 875
`\tl_gset:Nn` 15, 40, 50,
367, 414, 416, 418, 420, 422, 424,
426, 428, 431, 432, 451, 456, 459,
465, 474, 476, 478, 480, 724, 727, 787
`\tl_gset_eq:NN` 781
`\tl_gset_rescan:Nnn` 363
`\tl_if_blank:nTF` 524,
539, 577, 605, 614, 620, 630, 702,
948, 1189, 1225, 1228, 1247, 1253, 1272
`\tl_if_blank_p:n`
..... 645, 688, 1058, 1107, 1197
`\tl_if_empty:NTF` 491
`\tl_new:N` 1, 8, 358, 403,
404, 405, 406, 407, 408, 409, 720, 778
`\tl_set:Nn` 446, 822, 921
`\tl_set_eq:NN` .. 372, 434, 815, 831, 862
`\tl_to_str:n` ... 375, 376, 377, 378, 379
`\tl_trim_spaces:n` 526
`\tl_use:N` 153
`\l_tmpa_tl` 822, 824, 826, 831
`\today` 4

U

unit (option) 13, 813
unknown-value-warning (option) 4, 70
use commands:
`\use:N` 177, 179, 807
`\use_none:n` 312, 602, 971
`\use_none:nnn` 630
`\use_none:nnnn` 272, 289
`\use_none:nnnnn` 258, 286
`\use_none:nnnnnnn` 290
`\UseZUGFeRDElement` 152

W

write-xml (option) 3, 9

X

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

Y

`\year` 195, 452, 460

Z

ZUGFeRD (env.) 5
zugferd (option) 3, 9
zugferd commands:
`\zugferd_disable_macros:`
..... 24, 352, 361, 1143
`\zugferd_disable_XML_interfaces:`
..... 21, 211, 283, 295
`\zugferd_enable_XML_interfaces:`
..... 21, 203, 208, 248, 293
`\zugferd_fp_gset_rounded:Nn`
..... 25, 393, 398
`\zugferd_fp_set_rounded:Nn`
..... 25, 393, 394
`\zugferd_startInvoiceSums:` 263, 287
`\zugferd_stopInvoiceSums:` . 268, 288
`\zugferd_tl_gset_escaped_xml:Nn`
..... 359, 371
`\zugferd_tl_set_escaped_xml:Nn` . 370
`\zugferd_write_Footer:` . 209, 254, 285
`\zugferd_write_Header:` . 205, 250, 284
`\zugferd_write_Item:nnnnn`
..... 258, 260, 286, 300
`\zugferd_write_Item:nnnnnnn`
..... 22, 297, 303
`\zugferd_write_Summation:nnnnnnn`
..... 276, 290
`\zugferd_write_TaxEntry:nnnn` ...
..... 22, 44, 272, 274, 289, 304
zugferd internal commands:
`\g__zugferd_active_bool` .. 58, 77, 232
`__zugferd_ApplicableHeaderTradeAgreement:`
..... 264, 641
`__zugferd_ApplicableHeaderTradeDelivery:n`
..... 681, 686, 719
`__zugferd_ApplicableHeaderTradeSettlement_`
`start:` 265, 1212
`__zugferd_ApplicableHeaderTradeSettlement_`
`stop:` 269, 1217
`__zugferd_ApplicableTradeTax:nnnn`
..... 274, 1179
`__zugferd_AssociatedDocumentLineDocument:n`
..... 38, 1050, 1140
`\g__zugferd_auto_exemption_bool`
..... 66, 863, 1186
`__zugferd_BillingSpecifiedPeriod:nn`
..... 1090, 1106, 1129
`\g__zugferd_businessProcessId_`
`str` 1, 22, 1004, 1006
`\g__zugferd_buyer_AddressData_`
`prop` 495, 666, 668, 669
`\g__zugferd_buyer_document_`
`references_clist` 731
`\g__zugferd_buyer_reference_tl` .
..... 645, 649, 720, 724, 727
`\g__zugferd_conformance_level_`
`str` .. 1, 14, 39, 49, 140, 1059, 1198
`\l__zugferd_currency_tl`
..... 409, 434, 446, 1169, 1215
`\g__zugferd_date_tl` 402, 451, 456, 1025

__zugferd_define_xml_content:Nn	\g__zugferd_minimum_bool
. 22, 310, 314, 570, 586, 604, 1, 46, 257, 271, 588, 596, 601,
612, 628, 686, 944, 947, 952, 957,	646, 667, 679, 970, 1163, 1171, 1266
962, 965, 1055, 1080, 1096, 1106, 1120	__zugferd_note_if_not_empty:N .
__zugferd_define_xml_writer:Nn 490, 1028, 1029
. 22, 309, 313, 641,	\g__zugferd_notes_seq . 402, 481, 1030
740, 973, 990, 1017, 1042, 1050,	__zugferd_number_format:nNn . . .
1065, 1070, 1086, 1137, 1160, 1179, 23, 319, 331, 333, 395, 399
1212, 1217, 1221, 1240, 1265, 1271	\g__zugferd_payment_means_tl
__zugferd_DefinedTradeContact:N 778, 781, 787, 1245, 1246, 1272
. 28, 562, 656, 668	\g__zugferd_payment_terms_str
__zugferd_DefinedTradeContact:nmnn 470, 1267
. 563, 628, 717	__zugferd_PostalTradeAddress:N
\g__zugferd_delivery_date_tl 550, 657, 669
. 405, 459, 465, 681	__zugferd_PostalTradeAddress:nmnnmnn
\g__zugferd_document_references_ 541, 551, 586, 718
prop . . 730, 733, 734, 736, 741, 744	__zugferd_PostalTradeAddress_
\g__zugferd_DocumentTypeCode_tl	short:N 538, 696
. 402, 414,	__zugferd_ProductTradePrice:nm
416, 418, 420, 422, 424, 426, 428, 1022 39, 1065, 1149
\g__zugferd_due_date_tl . . . 468, 1267	\g__zugferd_seller_AddressData_
\l__zugferd_end_date_tl . . . 941, 1090	prop 495, 655, 656, 657
__zugferd_escape_xml_activate:	__zugferd_set_today:N 19
. 365, 386	__zugferd_set_today:n . 168, 190, 193
__zugferd_escape_xml_aux: 362, 374	__zugferd_set_today_aux:w 191, 194
\g__zugferd_format_str	\g__zugferd_shipto_AddressData_
. 1, 19, 28, 34, 43, 1010	prop 495, 687, 691, 696
\g__zugferd_fromaddress_tl	__zugferd_SpecifiedLineTradeAgreement:nm
. 402, 478, 480, 1029 39, 1070
\g__zugferd_id_tl 402, 431, 432, 1021	__zugferd_SpecifiedLineTradeDelivery:nm
__zugferd_indent: 23, 340, 344, 595, 39, 1080, 1153
607, 616, 622, 633, 704, 708, 744,	__zugferd_SpecifiedLineTradeSettlement:nmnn
963, 978, 980, 1006, 1010, 1025, 40, 1128, 1133, 1134
1052, 1060, 1062, 1067, 1072, 1074,	__zugferd_SpecifiedLineTradeSettlement:nmnnn
1075, 1077, 1083, 1099, 1101, 1103, 1130, 1132, 1136
1110, 1112, 1114, 1116, 1123, 1230	__zugferd_SpecifiedTradePaymentTerms:
__zugferd_indent:n 277, 1265
. 343, 347, 705, 706, 707,	__zugferd_SpecifiedTradePaymentTerms:nm
953, 954, 955, 1073, 1076, 1111, 1115 1221, 1264, 1267
\g__zugferd_indent_int	__zugferd_SpecifiedTradeProduct:nm
. 339, 347, 643, 653, 659, 38, 1055, 1144
664, 672, 677, 692, 694, 698, 700,	__zugferd_SpecifiedTradeSettlementHeaderMonetarySumm
1000, 1002, 1013, 1019, 1033, 1038, 278, 1160
1043, 1045, 1088, 1092, 1139, 1148,	__zugferd_SpecifiedTradeSettlementLineMonetarySumm
1150, 1157, 1162, 1175, 1181, 1206, 1091, 1120
1214, 1218, 1223, 1234, 1243, 1259	__zugferd_SpecifiedTradeSettlementPaymentMeans:
__zugferd_insert_Footer: . 255, 1042 44, 266, 1271
__zugferd_insert_FrontMatter:	__zugferd_SpecifiedTradeSettlementPaymentMeans:nmnn
. 252, 1017 1240, 1270, 1273
__zugferd_insert_Header: . 251, 990	\l__zugferd_start_date_tl . 939, 1090
__zugferd_insert_TradeLineItem:nmnnmnn	\g__zugferd_subject_tl
. 5, 41, 261, 1137 402, 474, 476, 1028
__zugferd_Line_ApplicableTradeTax:nm	\l__zugferd_tax_category_code_tl
. 1089, 1096 862, 921, 1155

\l__zugferd_tax_exemption_code_	__zugferd_write_TradeLineItem:nnnnnn
tl 874, 1197, 1201 41
\l__zugferd_tax_exemption_reason_	__zugferd_write_xml:n
tl 871, 1189, 1191 22, 306, 312, 317, 334, 642,
\l__zugferd_tax_rate_fp ... 929, 1156	648, 652, 654, 660, 665, 673, 678,
__zugferd_TaxRegistration:mn ..	680, 690, 693, 695, 699, 703, 711,
..... 560, 612	742, 976, 991, 1001, 1003, 1014,
\g__zugferd_tmp_tl 358, 371, 372	1018, 1020, 1034, 1044, 1046, 1087,
\l__zugferd_tmp_tl	1089, 1090, 1091, 1093, 1138, 1140,
..... 1, 333, 334, 395, 396, 399, 400	1144, 1147, 1149, 1151, 1152, 1158,
__zugferd_TradePartyIdentity:N	1161, 1176, 1180, 1183, 1190, 1195,
..... 530, 655, 666	1200, 1205, 1207, 1213, 1215, 1219,
__zugferd_TradePartyIdentity:N\	1222, 1224, 1235, 1242, 1244, 1260
_zugferd_PostalTradeAddress:N 28	\g__zugferd_write_xml_bool
__zugferd_TradePartyIdentity:nnnn 55, 215, 229, 249, 292, 305
..... 531, 570, 716	\g__zugferd_writePaymentMeans_
\l__zugferd_unit_code_tl	bool 1, 13, 38, 48, 1241
..... 815, 831, 1153	\g__zugferd_writeTradeContact_
__zugferd_UniversalCommunication:n	bool 1, 12, 37, 47, 631
..... 559, 602, 604	__zugferd_xml_auto_indent:
\g__zugferd_unknown_value_ 23, 307, 346, 350, 540
warning_bool 70, 443, 784, 828, 922	\g__zugferd_xml_cctab
__zugferd_write_inline:nn 569, 739, 968, 1049, 1210
..... 572, 574, 576, 944, 961, 1246	\g__zugferd_xml_embedded_file_tl
__zugferd_write_inline:nnn 1, 15, 40, 50, 136, 237, 242
..... 945, 947, 963	\g__zugferd_xml_file_tl . 63, 217, 237
__zugferd_write_inline_i:nn ...	__zugferd_xml_newline_indent: .
..... 582, 589, 590, 591, 592, 23, 216, 349, 953, 954
593, 597, 634, 965, 1249, 1250, 1255	__zugferd_xml_writer_iow
__zugferd_write_inline_i:nnn 76, 217, 230, 307
..... 580, 962, 966	\zugferd_disable_macros:
__zugferd_write_inline_wrapped:nnnn 7
..... 952, 958	\zugferd_disable_XML_interfaces: 7
__zugferd_write_inline_wrapped_	\zugferd_enable_XML_interfaces: 7
i:nnn 635, 636, 957	\zugferd_fp_gset_rounded:Nn 7
__zugferd_write_note:n	\zugferd_fp_set_rounded:Nn 7
..... 492, 971, 973, 1031	\zugferd_startInvoiceSums:
__zugferd_write_ReferencedDocument:n 6
..... 675, 740	\zugferd_stopInvoiceSums:
__zugferd_write_rounded:nnn 6
..... 23, 336, 1164, 1165, 1166,	\zugferd_tl_gset_escaped_xml:Nn 7
1168, 1170, 1172, 1174, 1182, 1194	\zugferd_tl_set_escaped_xml:Nn 7
__zugferd_write_rounded:nnnn ..	\zugferd_write_Item:ennnnnn 6
..... 23, 332, 337, 1169	\zugferd_write_Item:nnnnnn 5
__zugferd_write_SpecifiedLineTradeSett	\zugferd_write_Item:nnnnnnn 6
lements 1086, 1095, 1155	\zugferd_write_Summation:nnnnnnnn ... 6
	\zugferd_write_TaxEntry:ennn 6
	\zugferd_write_TaxEntry:nnnn 6