

The zugferd package*

Creating electronic and hybrid invoices using L^AT_EX

Marei Peischl
marei@peitex.de

October 23, 2024

Abstract

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

Sponsors & Supporters

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

This project was financially supported by:

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

*This document corresponds to zugferd v0.9, dated 2024-10-23.

Contents

1	Quick start	3
1.1	Disclaimer concerning the zugferd-invoice Package	3
2	Package Options	3
3	User Commands	4
4	Commands for template authors	4
4.1	Interfaces to write the XML contents	5
4.2	Commands to temporary disable/re-enable the XML writing interfaces	7
4.3	Escaping macros inside XML data	7
5	Adding data to the XML	7
5.1	General Invoicing Data	8
5.1.1	Invoice number/document ID	8
5.1.2	Currency	8
5.1.3	Dates	8
5.1.4	Payment terms	8
5.1.5	Notes: Adding additional information	9
5.2	Trade parties	9
5.2.1	Buyer Reference	11
5.2.2	Payment Means	11
5.3	Variables which may be changed per invoice item	12
5.3.1	Units	12
5.3.2	Tax category and rate	12
6	Implementation	13
6.1	Preparation to write the .xml file	14
6.2	Number rounding	21
6.3	XML indentation	22
6.4	Invoice Items	33
6.4.1	SpecifiedLineTradeDelivery	34
	Change History	40
	Index	41

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 [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 3.0.1 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`) (default: `xrechnung`)

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

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

`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 L^AT_EX releases than this package requires. There will be a warning, but the visible part should be okay.

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

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

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

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

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

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

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

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

3 User Commands

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

`\SetZUGFeRDData*{key value list}`

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

`\InsertZUGFeRDData`

`\InsertZugferdData`

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

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

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

As special modes the command currently supports the following:

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

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

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

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

4 Commands for template authors

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

This environment opens the XML file using `\startWritingZUGFeRDxml` and afterwards writes the XML header including the File and Scheme information, the `ExchangedDocumentContext` and information of the `ExchangedDocument`. Notes will

also be written within this step. Afterwards the environment should include all the mechanisms to write the invoice positions as well as summation.

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

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

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

4.1 Interfaces to write the XML contents

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

```
\begin{ZUGFeRD}
  \zugferd_write_Item:nnnnnn {1} {} {Plushie~\TeX\ lion} {31.98} {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} {31.98} {2}
  → {63.78}
  % Code using the data for visual representation
\group_end:
```

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

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

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

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

The example above could then be replaced by

```
\zugferd_write_Item:nnnnnnn {tax/rate=19,tax/category=S} {1} {Plushie-01}
→ {Plushie \TeX\ lion} {31.98} {2} {63.78}
% visual representation now may not refer to the data
```

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

The starting includes the so called “ApplicableHeaderTradeAgreement” which contains the address data of both trade parties, see [subsection 5.2](#) And this will also print the “SpecifiedTradeSettlementPaymentMeans”, see [subsubsection 5.2.2](#).

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

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

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

`\zugferd_write_Summation:nnnnnnnn`

The total values are all collected with a single macro.

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

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

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

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

`\zugferd_enable_XML_interfaces:`
`\zugferd_disable_XML_interfaces:`

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

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

4.3 Escaping macros inside XML data

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

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

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

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. The corresponding issue can be found at <https://github.com/TeXhackse/LaTeX-ZUGFeRD/issues/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= (commercial-invoice)` (default: `commercial-invoice`)

Select the document type. The only supported value currently is `commercial-invoice`. This will select the corresponding type code, which is 380.

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: "20241023" (October 23, 2024).

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

5.1.4 Payment terms

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

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

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

5.1.5 Notes: Adding additional information

```
subject= (komavar/⟨Tokenlist⟩)                ⟨initially unset⟩
fromaddress= (komavar/⟨Tokenlist⟩)           ⟨initially unset⟩
add-note= ⟨Tokenlist⟩                         ⟨initially unset⟩
```

The ZUGFeRD example files[10] use all visible data to add them to the XML as a note. `subject` and `fromaddress` are used to support this. The data should not be too relevant but zugferd wants 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/name = {peiTeX (Marei Peischl)},
  seller/email = {invoicing@peitex.de},
  seller/vatid = {DE123456789},
  seller/contact= {Marei\\+4900000000\\marei@peitex.de},
  seller/address = {Address Line 1\\Address Line 2},
  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>/vaid= <VAT ID>                                   <initially unset>
    <party>/address= <address>                               <initially unset>

```

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

Alternatively it's also possible to use `<party>/lineone` and `<party/linetwo` separately. This may be helpful if you use a custom input format. In any way you should ensure that all macros used within the data either are expandable or disabled using `\zugferd_disable_macros:`.

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

5.2.1 Buyer Reference

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

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

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

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

5.2.2 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 a 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. The macros are called `\zugferd@paymentMeans@<language name>`. 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 [8].

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 [8].

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,0`. This can either be added using a text (`exemption-reason`) or a predefined code (`exemption-reason-code`). The codes are listed at [9].

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 0.00 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_conformance_level_str 5 \bool_new:N \g__zugferd_writePaymentMeans_bool
\g__zugferd_conformance_level_str 6 \str_new:N \g__zugferd_conformance_level_str

(End of definition for \l__zugferd_tmp_tl and others.)

format
xrechnung      7 \char_set_catcode_other:N \#%
write-xml      8 \keys_define:nn {zugferd} {
zugferd      9 xrechnung .code:n = {
xml-file     10 \bool_gset_true:N \g__zugferd_writeTradeContact_bool
auto-exemption 11 \bool_gset_true:N \g__zugferd_writePaymentMeans_bool
              12 \str_gset:Nn \g__zugferd_conformance_level_str {XRECHNUNG}
              13 },
              14 format .choice:,
              15 format / xrechnung3.0 .code:n = {

```

```

16 \str_gset:Nx \g__zugferd_format_str {
17   urn:cen.eu:en16931:2017#compliant#urn:xeinkauf.de:kosit:xrechnung_3.0
18 }
19 \str_gset:Nx \g__zugferd_businessProcessId_str {
20   urn:fdc:peppol.eu:2017:poacc:billing:01:1.0
21 }
22 \keys_set:nn {zugferd}{xrechnung}
23 },
24 format / xrechnung2.3 .code:n = {
25   \str_gset:Nx \g__zugferd_format_str {
26     urn:cen.eu:en16931:2017#compliant#urn:xoev-de:kosit:standard:xrechnung_2.3
27   }
28   \keys_set:nn {zugferd}{xrechnung}
29 },
30 format / basic .code:n = {
31   \str_gset:Nx \g__zugferd_format_str {
32     urn:cen.eu:en16931:2017#compliant#urn:factur-x.eu:ip0:basic
33   }
34   \bool_gset_false:N \g__zugferd_writeTradeContact_bool
35   \bool_gset_false:N \g__zugferd_writePaymentMeans_bool
36   \str_gset:Nn \g__zugferd_conformance_level_str {BASIC}
37 },
38 format / xrechnung .meta:n = { format = xrechnung3.0 },
39 format .initial:n = xrechnung,
40 format .usage:n = load,
41 write-xml .bool_gset:N = \g__zugferd_write_xml_bool,
42 write-xml .initial:n = true,
43 write-xml .usage:n = load,
44 zugferd .bool_gset:N = \g__zugferd_active_bool,
45 zugferd .initial:n = true,
46 zugferd .default:n = true,
47 zugferd .usage:n = load,
48 ZUGFerD .meta:n = {zugferd = #1},
49 xml-file .tl_gset:N = \g__zugferd_xml_file_tl,
50 xml-file .initial:n = \jobname _zugferd.xml,
51 xml-file .usage:n = load,
52 auto-exemption .bool_gset:N = \g__zugferd_auto_exemption_bool,
53 auto-exemption .initial:n = true,
54 auto-exemption .default:n = true,
55 auto-exemption .usage:n =load,
56 }
57 \char_set_catcode_parameter:N \#%
58
59 \ProcessKeyOptions[zugferd]

```

6.1 Preparation to write the .xml file

```
\__zugferd_xml_writer_iow
```

```
60 \iow_new:N \__zugferd_xml_writer_iow
```

(End of definition for __zugferd_xml_writer_iow.)

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

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

```

61 \bool_if:NT \g__zugferd_active_bool {
62 \cs_if_exist:NF \pdfmeta_xmp_xmlns_new:nn {
63 \msg_new:nnnn {zugferd} {PDFmanagement-not-active} {
64 The~\LaTeX~PDF~management~is~not~active.\\
65 Activate~it~using~\string\DocumentMetadata.
66 } {
67 See~ZUGFeRD~or~PDFmanagement~documentation~for~more~information.
68 }
69 \msg_error:nn{zugferd} {PDFmanagement-not-active}
70 }
71 %% based on experiments for l3pdfmeta by Ulrike Fischer
72 \pdfmeta_xmp_xmlns_new:nn {fx}{
73 urn:factur-x:pdfa:CrossIndustryDocument:invoice:1p0\c_hash_str
74 }
75
76 \pdfmeta_xmp_schema_new:nnn
77 {Factur-X-PDFA-Extension-Schema}
78 {fx}
79 {urn:factur-x:pdfa:CrossIndustryDocument:invoice:1p0\c_hash_str}
80
81 \pdfmeta_xmp_property_new:nnnnn
82 {fx}
83 {DocumentFileName}
84 {Text}
85 {external}
86 {name-of-the-embedded-XML-invoice-file}
87
88 \pdfmeta_xmp_property_new:nnnnn
89 {fx}
90 {DocumentType}
91 {Text}
92 {external}
93 {INVOICE}
94
95 \pdfmeta_xmp_property_new:nnnnn
96 {fx}
97 {Version}
98 {Text}
99 {external}
100 {The-actual-version-of-the-factur-x-schema}
101
102 \pdfmeta_xmp_property_new:nnnnn
103 {fx}
104 {ConformanceLevel}
105 {Text}
106 {external}
107 {The-conformance-level-of-the-factur-x-data}
108
109 \exp_args:Ne \pdfmeta_xmp_add:n {
110 % fix INVOICE
111 <fx:DocumentType>INVOICE</fx:DocumentType>\iow_newline:
112 % fix factur-x.xml
113 <fx:DocumentFileName>factur-x.xml</fx:DocumentFileName>\iow_newline:

```

```

114 % fix schema version
115 <fx:Version>1.0</fx:Version>\iow_newline:
116 % zulässige Werte MINIMUM, BASIC WL, BASIC, EN 16931, EXTENDED, XRECHNUNG
117 <fx:ConformanceLevel>\g_zugferd_conformance_level_str</fx:ConformanceLevel>%
118 \iow_newline:
119 %
120 }
121 }

```

\SetZUGFeRData

```

122 \NewDocumentCommand{\SetZUGFeRData}{sm}{
123 \IfBooleanTF{#1}
124   {\keys_set:ne}
125   {\keys_set:nn}
126   {zugferd} {#2}
127 }
128 \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.

```

129 \NewDocumentCommand{\InsertZUGFeRDData}{om}{
130   \str_case:nnF {#1} {
131     {AddressData} {
132       \clist_map_inline:nn {name, lineone, linetwo} {
133         \prop_if_in:cnT {g__zugferd_#2_AddressData_prop } {##1} {
134           \prop_item:cn {g__zugferd_#2_AddressData_prop } {##1}\
135         }
136       }
137       \prop_item:cn {g__zugferd_#2_AddressData_prop } {postcode}
138       \space
139       \prop_item:cn {g__zugferd_#2_AddressData_prop } {city}
140     }
141     {set-today} {
142       \__zugferd_set_today:v {g__zugferd_#2_t1}
143     }
144   } {

```

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

```

145   \str_set:Ne \l_tmpa_str {#2}
146   \str_replace_all:Nnn \l_tmpa_str {-} {_}
147   \str_replace_all:Nnn \l_tmpa_str {/} {_}
148   \bool_set_true:N \g_tmpa_bool
149   \clist_map_inline:nn {t1, str} {
150     \clist_map_inline:nn {g, l} {
151       \use:c {##1_if_exist:cT} {####1__zugferd_ \l_tmpa_str _##1}
152       {
153         \use:c {####1__zugferd_ \l_tmpa_str _##1}
154         \bool_gset_false:N \g_tmpa_bool
155       }
156       \bool_if:NF \g_tmpa_bool {\clist_map_break:}
157     }
158     \bool_if:NF \g_tmpa_bool {\clist_map_break:}
159   }
160 }
161 }
162 \providecommand{\InsertZugferdData}{\InsertZUGFeRDData}
163 \providecommand{\insertZugferdData}{\InsertZUGFeRDData}

```

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

```

164 \cs_new:Nn \\_zugferd_set_today:n {
165   \\_zugferd_set_today_aux:w #1 \q_stop
166 }
167 \cs_generate_variant:Nn \\_zugferd_set_today:n {v}
168 \cs_new:Npn \\_zugferd_set_today_aux:w #1 #2 #3 #4 #5 #6 #7 #8 \q_stop{
169   \int_set:Nn \year {#1#2#3#4}
170   \int_set:Nn \month {#5#6}
171   \int_set:Nn \day {#7#8}
172 }

173 \NewDocumentEnvironment{ZUGFeRD}{o}{
174   \IfNoValueF{#1}{
175     \SetZUGFeRDData{#1}
176   }
177   \zugferd_enable_XML_interfaces:
178   \startWritingZUGFeRDxml
179   \zugferd_write_Header:
180   \ignorespaces
181 }{
182   \zugferd_enable_XML_interfaces:
183   \zugferd_write_Footer:
184   \stopWritingZUGFeRDxml
185   \zugferd_disable_XML_interfaces:
186 }

187 \newcommand*{\startWritingZUGFeRDxml}{
188   \begingroup
189   \bool_if:NTF \g__zugferd_write_xml_bool {
190     \char_set_active_eq:nN {13} \\_zugferd_xml_newline_indent:
191     \iow_open:Nn \\_zugferd_xml_writer_iow {\g__zugferd_xml_file_tl}
192   }{
193     \msg_info:nn {zugferd} {no-xml-write}
194   }
195 }
196 \msg_new:nnn {zugferd} {no-xml-write} {
197   The-option-write-xml=false-was-set.\\
198   Writing-of~XML~file~is~deactivated.
199 }

```

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

```

200 %% The metadata elements are taken by Ulrike Fischer's faktur-x experiments
201 %% https://github.com/latex3/pdfresources/tree/main/experiments/faktur-x-bills
202 \newcommand*{\stopWritingZUGFeRDxml}{%
203   \bool_if:NT \g__zugferd_write_xml_bool
204   {\iow_close:N \\_zugferd_xml_writer_iow}%
205 }
206 \bool_if:NT \g__zugferd_active_bool {
207   \group_begin:
208   \pdfdict_put:nnn {l_pdffile/Filespec} {AFRelationship}{/Alternative}
209   %or /Source in some cases
210   \pdfdict_put:nnn {l_pdffile/Filespec} {Desc}{(Faktur-X/ZUGFeRD-Rechnung)}
211   \pdffile_embed_file:nnn {\g__zugferd_xml_file_tl}{faktor-x.xml}
212   {zugferd/rechnung}

```

```
213 \group_end:
214 \pdfmanagement_add:nx
215   {Catalog/Names}
216   {EmbeddedFiles}
217   {\pdf_object_ref:n{zugferd/rechnung}}
218   % steht in der docu ist aber pdf 2.0 ....
219 \pdfmanagement_add:nx{Catalog}{AF}{\pdf_object_ref:n{zugferd/rechnung}}
220 }
221 }
```

Provide public interfaces and the ZUGFeRD environment.

\zugferd_enable_XML_interfaces:
\zugferd_disable_XML_interfaces:

```
222 \cs_new:Nn \zugferd_enable_XML_interfaces: {
223 \bool_if:NT \g__zugferd_write_xml_bool {
224 \cs_gset:Nn \zugferd_write_Header: {
225 \__zugferd_insert_Header:
226 \__zugferd_insert_FrontMatter:
227 }
228 \cs_gset:Nn \zugferd_write_Footer: {
229 \__zugferd_insert_Footer:
230 }
231 \cs_gset_eq:NN \zugferd_write_Item:nnnnnn
232 \__zugferd_insert_TradeLineItem:nnnnnn
233 \cs_gset:Nn \zugferd_startInvoiceSums: {
234 \__zugferd_ApplicableHeaderTradeAgreement:
235 \__zugferd_ApplicableHeaderTradeSettlement_start:
236 \__zugferd_SpecifiedTradeSettlementPaymentMeans:
237 }
238 \cs_gset:Nn \zugferd_stopInvoiceSums: {
239 \__zugferd_ApplicableHeaderTradeSettlement_stop:
240 }
241 \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \__zugferd_ApplicableTradeTax:nnnn
242 \cs_gset:Nn \zugferd_write_Summation:nnnnnnnn {
243 \__zugferd_SpecifiedTradePaymentTerms:
244 \__zugferd_SpecifiedTradeSettlementHeaderMonetarySummation:nnnnnnnn
245 {##1} {##2} {##3} {##4} {##5} {##6} {##7} {##8}
246 }
247 }
248 }
249 \cs_new:Nn \zugferd_disable_XML_interfaces: {
250 \cs_gset_eq:NN \zugferd_write_Header: \prg_do_nothing:
251 \cs_gset_eq:NN \zugferd_write_Footer: \prg_do_nothing:
252 \cs_gset_eq:NN \zugferd_write_Item:nnnnnn \use_none:nnnnnn
253 \cs_gset_eq:NN \zugferd_startInvoiceSums: \prg_do_nothing:
254 \cs_gset_eq:NN \zugferd_stopInvoiceSums: \prg_do_nothing:
255 \cs_gset_eq:NN \zugferd_write_TaxEntry:nnnn \use_none:nnnn
256 \cs_gset_eq:NN \zugferd_write_Summation:nnnnnnnn \use_none:nnnnnnnn
257 }
258 \bool_if:NTF \g__zugferd_write_xml_bool {
259 \zugferd_enable_XML_interfaces:
260 }{
261 \zugferd_disable_XML_interfaces:
262 }
```

__zugferd_write_Item:nnnnnnn

```
263 \cs_new:Nn \zugferd_write_Item:nnnnnnn {
264 \group_begin:
265 \keys_set:nn {zugferd/item} {#1}
266 \zugferd_write_Item:nnnnnn {#2} {#3} {#4} {#5} {#6} {#7}
267 \group_end:
268 }
269 \cs_generate_variant:Nn \zugferd_write_Item:nnnnnnn {ennnnnn}
```

```

\__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`.

```

270 \bool_if:NTF \g__zugferd_write_xml_bool {
271 \cs_new:Nn \__zugferd_write_xml:n {
272 \iow_now:Ne \__zugferd_xml_writer_iow {\__zugferd_xml_auto_indent: #1}
273 }
274 \cs_new_eq:NN \__zugferd_define_xml_writer:Nn \cs_new:Nn
275 \cs_new_eq:NN \__zugferd_define_xml_content:Nn \cs_new:Nn
276 } {
277 \cs_set_eq:NN \__zugferd_write_xml:n \use_none:n
278 \cs_set:Nn \__zugferd_define_xml_writer:Nn {\cs_new:Nn #1 {}}
279 \cs_set:Nn \__zugferd_define_xml_content:Nn {\cs_new:Nn #1 {}}
280 }
281
282 \cs_generate_variant:Nn \__zugferd_write_xml:n {e}

```

6.2 Number rounding

As `siunitx` is implementing this, we use it instead of building our own mechanism.

```

283 \RequirePackage{siunitx}

```

```

\__zugferd_number_format:nNn
\__zugferd_number_format:nNe

```

```

284 \cs_new:Nn \__zugferd_number_format:nNn {
285 \sisetup{
286 parse-numbers=true,
287 round-mode=places,
288 round-precision=#1,
289 round-pad = false,
290 group-digits=false,
291 minimum-decimal-digits=#1,
292 output-decimal-marker=.
293 }
294 \siunitx_number_format:nN {#3} #2
295 }
296 \cs_generate_variant:Nn \__zugferd_number_format:nNn {nNe}

```

```

\__zugferd_write_rounded:nNnn
\__zugferd_write_rounded:nnn

```

```

297 \cs_new:Nn \__zugferd_write_rounded:nNnn {
298 \__zugferd_number_format:nNe {#1} \l__zugferd_tmp_tl {#4}
299 \__zugferd_write_xml:e {<ram:#2#3>\l__zugferd_tmp_tl</ram:#2>}
300 }
301 \cs_new:Nn \__zugferd_write_rounded:nnn {
302 \__zugferd_write_rounded:nNnn {#1} {#2} {} {#3}
303 }

```

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
304 \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:
```

```
305 \cs_new:Nn \__zugferd_indent: {
306   \space\space
307 }
308 \cs_new:Nn \__zugferd_indent:n {
309   \prg_replicate:nn {#1} {\__zugferd_indent:}
310 }
311 \cs_new:Nn \__zugferd_xml_auto_indent: {
312   \__zugferd_indent:n {\g__zugferd_indent_int}
313 }
```

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

```
314 \cs_new:Nn \__zugferd_xml_newline_indent: {
315   \iow_newline: \__zugferd_xml_auto_indent:
316 }
```

Setting up the catcodes to simplify the XML indentation.

```
\zugferd_disable_macros:
```

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

```
317 \cs_new:Nn \zugferd_disable_macros: {
318   \let\{\textbraceleft
319   \let\}\textbraceright
320   \let\\\textbackslash
321   \let\#\textnumbersign
322   \let\$\textdollar
```

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

```
\g__zugferd_notes_seq
  \g__zugferd_id_tl
  \g__zugferd_date_tl
  \g__zugferd_subject_tl
\g__zugferd_fromaddress_tl
  \g__zugferd_DocumentTypeCode_tl
323 \seq_new:N \g__zugferd_notes_seq
324 \tl_new:N \g__zugferd_id_tl
325 \tl_new:N \g__zugferd_date_tl
326 \tl_new:N \g__zugferd_delivery_date_tl
327 \tl_new:N \g__zugferd_subject_tl
328 \tl_new:N \g__zugferd_fromaddress_tl
```

```

329 \tl_new:N \g__zugferd_DocumentTypeCode_tl
330 \tl_new:N \l__zugferd_currency_tl

(End of definition for \g__zugferd_notes_seq and others.)

331 \keys_define:nn { zugferd } {
332 %TODO define others

id
document-type 333 document-type .choice:,
334 document-type / commercial-invoice .code:n =
335 \tl_gset:Nn \g__zugferd_DocumentTypeCode_tl {380},
336 document-type .initial:n = commercial-invoice,

337 id .choice:,
338 id / komavar .code:n = \tl_gset:Nf \g__zugferd_id_tl {\scr@invoice@var},
339 id / unknown .code:n = \tl_gset:Nn \g__zugferd_id_tl {#1},

340 currency .choices:nn = {EUR, USD, CHF} {
341 \tl_set_eq:NN \l__zugferd_currency_tl \l_keys_choice_tl
342 },
343 currency / € .meta:n = {currency = EUR},
344 currency / unknown .code:n = {
345 \exp_args:Nnnx \keys_if_choice_exist:nnnTF {zugferd} {unit} {
346 \str_uppercase:f {#1}
347 }
348 {\keys_set:nx { zugferd } {unit= {\str_uppercase:f {#1}}}}
349 {
350 \msg_warning:nnn { zugferd } {unknown-value} {currency} {#1}
351 \tl_set:Ne \l__zugferd_currency_tl {\str_uppercase:f {#1}}
352 }
353 },
354 currency .initial:n = EUR,

date
delivery-date 355 date .choice:,
356 date / auto .code:n = \tl_gset:Ne \g__zugferd_date_tl {
357 \the\year
358 \int_compare:nNnT {\month} < {10} {0} \the\month
359 \int_compare:nNnT {\day} < {10} {0}\the\day
360 },
361 date / unknown .code:n = \tl_gset:Nn \g__zugferd_date_tl {#1},
362 date .initial:n = auto,
363 delivery-date .choice:,
364 delivery-date / auto .code:n = \tl_gset:Ne \g__zugferd_delivery_date_tl {
365 \the\year
366 \int_compare:nNnT {\month} < {10} {0} \the\month
367 \int_compare:nNnT {\day} < {10} {0}\the\day
368 },
369 delivery-date / unknown .code:n = {
370 \tl_gset:Nn \g__zugferd_delivery_date_tl {#1}
371 },
372 delivery-date .initial:n = auto,
373 due-date .tl_gset:N = \g__zugferd_due_date_tl,
374 due-date .initial:n =,

```

payment-terms

```
375 payment-terms .str_gset:N = \g__zugferd_payment_terms_str,  
376 payment-terms .initial:n =,
```

subject
fromaddress
add-note

```
377 subject .choice:,  
378 subject / komavar .code:n = {  
379   \tl_gset:Nf \g__zugferd_subject_tl {\scr@subject@var}  
380 },  
381 subject / unknown .code:n = \tl_gset:Nn \g__zugferd_subject_tl {#1},  
382 fromaddress .choice:,  
383 fromaddress / komavar .code:n = \tl_gset:Nf \g__zugferd_fromaddress_tl  
384   {\scr@fromaddress@var},  
385 fromaddress / unknown .code:n = \tl_gset:Nn \g__zugferd_fromaddress_tl {#1},  
386 add-note .code:n = \seq_gput_right:Nn \g__zugferd_notes_seq {#1},  
387 }  
388 \msg_new:nnnn {zugferd} {unknown-value} {  
389   You-selected-a-#1-which-was-not-predefined.\  
390   I-will-directly-use-your-selection-'#1=#2'.  
391   Please-ensure-the-selection-is-valid!  
392 } {  
393   For-more-information-see-the-zugferd-documentation.  
394 }
```

Macro to write notes

```
395 \cs_new:Nn \__zugferd_note_if_not_empty:N {%  
396   \tl_if_empty:NF #1 {%  
397     \__zugferd_write_note:n {#1}%  
398   }%  
399 }%
```

ApplicableHeaderTradeAgreement

Contains information on seller and buyer trade party:

- BuyerReference
- SellerTradeParty
- BuyerTradeParty

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

```
\g__zugferd_buyer_AddressData_prop  
\g__zugferd_shipto_AddressData_prop  
400 \clist_map_inline:nn {seller,buyer,shipto} {  
401   \prop_new:c {g__zugferd_#1_AddressData_prop}  
402   \keys_define:nn {zugferd / #1} {  
403     address .code:n = {  
404       \seq_set_split:Nnn \l_tmpa_seq {\} {##1}  
405       %TODO error if more than 2 lines or only 1  
406       \keys_set:nx {zugferd / #1} {  
407         lineone=\seq_item:Nn \l_tmpa_seq {1},  
408         linetwo=\seq_item:Nn \l_tmpa_seq {2}  
409       }  
410     },  
411     contact .code:n = {  
412       \seq_set_split:Nnn \l_tmpa_seq {\} {##1}  
413       \int_compare:nNnTF {\seq_count:N \l_tmpa_seq} > {3} {
```



```

414     \keys_set:ne {zugferd/#1} {
415         contact-name = \seq_item:Nn \l_tmpa_seq {1},
416         contact-department = \seq_item:Nn \l_tmpa_seq {2},
417         contact-phone = \seq_item:Nn \l_tmpa_seq {3},
418         contact-email= \seq_item:Nn \l_tmpa_seq {4}
419     }
420 }{
421     \keys_set:ne {zugferd/#1} {
422         contact-name = \seq_item:Nn \l_tmpa_seq {1},
423         contact-phone = \seq_item:Nn \l_tmpa_seq {2},
424         contact-email= \seq_item:Nn \l_tmpa_seq {3}
425     }
426 }
427 },
428 unknown .code:n = \tl_if_blank:nF {##1} {
429     \prop_gput:cVe {g__zugferd_#1_AddressData_prop}
430     \l_keys_key_tl {\tl_trim_spaces:n {##1}}
431 }
432 }
433 }

```

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

```

\__zugferd_PostalTradeAddress:N
\__zugferd_DefinedTradeContact:N

```

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

```

434 \cs_new:Nn \__zugferd_PostalTradeAddress_short:N {
435     \exp_args:Ne \tl_if_blank:nF {\prop_item:Nn #1 {name}}
436     {<ram:Name>\prop_item:Nn #1 {name}</ram:Name>\iow_newline:\__zugferd_xml_auto_indent:}
437     \__zugferd_PostalTradeAddress:eeeeee
438     {\prop_item:Nn #1 {postcode}}
439     {\prop_item:Nn #1 {lineone}}
440     {\prop_item:Nn #1 {linetwo}}
441     {\prop_item:Nn #1 {city}}
442     {\prop_item:Nn #1 {country}}
443 }
444 \cs_new:Nn \__zugferd_PostalTradeAddress:N {%
445     \__zugferd_PostalTradeAddress:eeeeeee
446     {\prop_item:Nn #1 {postcode}}
447     {\prop_item:Nn #1 {lineone}}
448     {\prop_item:Nn #1 {linetwo}}
449     {\prop_item:Nn #1 {city}}
450     {\prop_item:Nn #1 {country}}
451     {\prop_item:Nn #1 {email}}
452     {\prop_item:Nn #1 {vatid}}
453 }%
454 \cs_new:Nn \__zugferd_DefinedTradeContact:N {%
455     \__zugferd_DefinedTradeContact:eeee
456     {\prop_item:Nn #1 {contact-name}}
457     {\prop_item:Nn #1 {contact-department}}
458     {\prop_item:Nn #1 {contact-phone}}
459     {\prop_item:Nn #1 {contact-email}}
460 }%

```

```

461 \cctab_begin:N \g__zugferd_xml_cctab%
462 \__zugferd_define_xml_content:Nn \__zugferd_PostalTradeAddress:nnnnn {%
463 <ram:PostalTradeAddress>
464 \__zugferd_indent: <ram:PostcodeCode>#1</ram:PostcodeCode>
465 \tl_if_blank:nF {#2} {%
466 \__zugferd_indent: <ram:LineOne>#2</ram:LineOne>
467 }%
468 \tl_if_blank:nF {#3} {%
469 \__zugferd_indent: <ram:LineTwo>#3</ram:LineTwo>
470 }%
471 \__zugferd_indent: <ram:CityName>#4</ram:CityName>
472 \__zugferd_indent: <ram:CountryID>#5</ram:CountryID>
473 </ram:PostalTradeAddress>%
474 }
475 \__zugferd_define_xml_content:Nn \__zugferd_PostalTradeAddress:nnnnnnn {%
476 \__zugferd_PostalTradeAddress:nnnnn {#1} {#2} {#3} {#4} {#5}
477 \tl_if_empty:nF {#6} {%
478 <ram:URIUniversalCommunication>
479 \__zugferd_indent: <ram:URIID~schemeID="EM">#6</ram:URIID>
480 </ram:URIUniversalCommunication>%
481 }%
482 % TODO add support local tax id: schemaID="FC"
483 \tl_if_empty:nF {#7} {
484 <ram:SpecifiedTaxRegistration>
485 \__zugferd_indent: <ram:ID~schemeID="VA">#7</ram:ID>
486 </ram:SpecifiedTaxRegistration>%
487 }%
488 }%
489 %
490 % Contact data phone/email to a specific contact person
491 \__zugferd_define_xml_content:Nn \__zugferd_DefinedTradeContact:nnnn {%
492 % Do not print if name is empty
493 \tl_if_blank:nT {#1} {\use_none:nnn} %
494 \bool_if:NT \g__zugferd_writeTradeContact_bool {%
495 <ram:DefinedTradeContact>
496 \__zugferd_indent: <ram:PersonName>#1</ram:PersonName>
497 \tl_if_blank:nF {#2} {%
498 \__zugferd_indent: <ram:DepartmentName>#2</ram:DepartmentName>
499 }%
500 \tl_if_blank:nF {#3} {%
501 \__zugferd_indent: <ram:TelephoneUniversalCommunication>
502 \__zugferd_indent:n {2} <ram:CompleteNumber>#3</ram:CompleteNumber>
503 \__zugferd_indent: </ram:TelephoneUniversalCommunication>
504 }%
505 \tl_if_blank:nF {#4} {%
506 \__zugferd_indent: <ram:EmailURIUniversalCommunication>
507 \__zugferd_indent:n {2} <ram:URIID>#4</ram:URIID>
508 \__zugferd_indent: </ram:EmailURIUniversalCommunication>
509 }%
510 </ram:DefinedTradeContact>%
511 }%
512 }%
513 %
514 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeAgreement: {%

```

```

515 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeAgreement>}%
516 \int_gincr:N \g__zugferd_indent_int%
517 \__zugferd_write_xml:e {%
518 <ram:BuyerReference>\g__zugferd_buyer_reference_t1</ram:BuyerReference>
519 <ram:SellerTradeParty>%
520 }%
521 \int_gincr:N \g__zugferd_indent_int%
522 \__zugferd_write_xml:e {%
523 <ram:Name>\prop_item:Nn \g__zugferd_seller_AddressData_prop {name}</ram:Name>
524 \__zugferd_DefinedTradeContact:N \g__zugferd_seller_AddressData_prop%
525 \__zugferd_PostalTradeAddress:N \g__zugferd_seller_AddressData_prop%
526 }%
527 \int_gdecr:N \g__zugferd_indent_int%
528 \__zugferd_write_xml:e {%
529 </ram:SellerTradeParty>
530 <ram:BuyerTradeParty>%
531 }%
532 \int_gincr:N \g__zugferd_indent_int%
533 \__zugferd_write_xml:e {%
534 <ram:Name>\prop_item:Nn \g__zugferd_buyer_AddressData_prop {name}</ram:Name>
535 \__zugferd_DefinedTradeContact:N \g__zugferd_buyer_AddressData_prop%
536 \__zugferd_PostalTradeAddress:N \g__zugferd_buyer_AddressData_prop%
537 }%
538 \int_gdecr:N \g__zugferd_indent_int%
539 \__zugferd_write_xml:n {</ram:BuyerTradeParty>}%
540 \int_gdecr:N \g__zugferd_indent_int%
541 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeAgreement>}%
542 \__zugferd_ApplicableHeaderTradeDelivery:V \g__zugferd_delivery_date_t1%
543 }%
544 %
545 %
546 % delivery date
547 \__zugferd_define_xml_content:Nn \__zugferd_ApplicableHeaderTradeDelivery:n {%
548 \bool_lazy_and:nnF {\prop_if_empty_p:N \g__zugferd_shipto_AddressData_prop}%
549 {\t1_if_blank_p:n {#1}}}%
550 {%
551 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeDelivery>}%
552 \prop_if_empty:NF \g__zugferd_shipto_AddressData_prop {%
553 \int_gincr:N \g__zugferd_indent_int%
554 \__zugferd_write_xml:n {<ram:ShipToTradeParty>}%
555 \int_gincr:N \g__zugferd_indent_int%
556 \__zugferd_write_xml:e {%
557 \__zugferd_PostalTradeAddress_short:N \g__zugferd_shipto_AddressData_prop%
558 }%
559 \int_gdecr:N \g__zugferd_indent_int%
560 \__zugferd_write_xml:n {</ram:ShipToTradeParty>}%
561 \int_gdecr:N \g__zugferd_indent_int%
562 }%
563 \t1_if_blank:nF {#1} {%
564 \__zugferd_write_xml:n {%
565 \__zugferd_indent: <ram:ActualDeliverySupplyChainEvent>
566 \__zugferd_indent:n {2} <ram:OccurrenceDateTime>
567 \__zugferd_indent:n {3}<udt:DateTimeString~format="102">#1</udt:DateTimeString>
568 \__zugferd_indent:n {2} </ram:OccurrenceDateTime>

```

```

569     \__zugferd_indent: </ram:ActualDeliverySupplyChainEvent>%
570     }%
571     }%
572     \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeDelivery>}%
573     }%
574     }%
575     %
576     \cctab_end:
577     \cs_generate_variant:Nn \__zugferd_DefinedTradeContact:nmmm {eeee}
578     \cs_generate_variant:Nn \__zugferd_PostalTradeAddress:nmmmm {eeeee}%
579     \cs_generate_variant:Nn \__zugferd_PostalTradeAddress:nmmmmmm {eeeeeee}
580     \cs_generate_variant:Nn \__zugferd_ApplicableHeaderTradeDelivery:n {V}

```

buyer/reference

```

581     \tl_new:N \g__zugferd_buyer_reference_tl
582     \keys_define:nn {zugferd/buyer} {
583       reference .choice:,
584       reference / komavar .code:n = {
585         \tl_gset:Nf \g__zugferd_buyer_reference_tl {\scr@yourref@var}
586       },
587       reference / unknown .code:n = {
588         \tl_gset:Nn \g__zugferd_buyer_reference_tl {#1}
589       }
590     }

```

payment-means

```

591     \ExplSyntaxOff
592     \providecommand*{\zugferd@paymentMeans@german}{
593       1 = Keine Zahlungsart definiert,
594       10 = Bargeld,
595       30 = Überweisung,
596       42 = Zahlung an Bankkonto,
597       48 = Kartenzahlung,
598       49 = Lastschriftverfahren,
599       57 = Dauerauftrag,
600       58 = Zahlung per SEPA Überweisung.,
601       59 = SEPA Lastschrift,
602       97 = Ausgleich zwischen Partnern
603     }
604     \def\zugferd@paymentMeans@ngerman{\zugferd@paymentMeans@german}
605     \providecommand*{\zugferd@paymentMeans@english}{
606       1 = Instrument not defined,
607       10 = In cash,
608       30 = Credit Transfer,
609       31 = Debit Transfer,
610       42 = Payment to bank account,
611       48 = Bank card,
612       49 = Direct Debit,
613       57 = Standing agreement,
614       58 = SEPA credit transfer,
615       59 = SEPA direct debit,
616       97 = Clearing between partners
617     }
618     \ExplSyntaxOn

```

```

619 \tl_new:N \g__zugferd_payment_means_tl
620 \keys_define:nn {zugferd} {
621   payment-means / type .choices:nn = {1,10,30,42,48,49,57,58,59,97} {
622     \tl_gset_eq:NN \g__zugferd_payment_means_tl \l_keys_choice_tl
623   },
624   payment-means / unknown .code:n = {
625     \msg_warning:nnn { zugferd } {unknown-value} {payment-means} {#1}
626     \tl_gset:Nn \g__zugferd_payment_means_tl {\int_eval:n {#1}}
627   }
628 }
629
630 \clist_map_inline:nn {iban,account-holder,bic} {
631   \keys_define:nn {zugferd/payment-means} {
632     #1 .tl_gset:c = {g__zugferd_payment_#1_tl}
633   }
634 }
635
636 \prop_new:c {g__zugferd_payment-means_names_prop}
637 \newcommand*{\setupZUGFeRDStrings}[2]{
638   \prop_gset_from_keyval:cn {g__zugferd_#1_names_prop} {
639     #2
640   }
641 }
642
643 \hook_gput_code:nnn {begindocument/end}{zugferd/payment-means}{
644   \prop_if_empty:cT {g__zugferd_payment-means_names_prop} {
645     \exp_args:Nne \setupZUGFeRDStrings{payment-means}{
646       \use:c {zugferd@paymentMeans@
647         \cs_if_exist:cTF {zugferd@paymentMeans@\languagename} {\languagename} {english}
648     }
649   }
650 }
651 }

```

unit

```

652 \keys_define:nn { zugferd } {
653   unit .choices:nn = {HUR,DAY,C62,H87} {
654     \tl_set_eq:NN \l__zugferd_unit_code_tl \l_keys_choice_tl
655   },
656   unit / hour .meta:n = {unit=HUR},
657   unit / day .meta:n = {unit=DAY},
658   unit / one .meta:n = {unit= C62},
659   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

```

660   unit / unknown .code:n = {
661     \tl_set:Nx \l_tmpa_tl {\str_uppercase:f {#1}}
662     \exp_args:NnnV \keys_if_choice_exist:nnnTF {zugferd} {unit}
663     \l_tmpa_tl
664     {
665       \keys_set:ne { zugferd } {unit= {\l_tmpa_tl}}
666     } {
667       \msg_warning:nnn { zugferd } {unknown-unit} {#1}

```

```

668     \tl_set_eq:NN \l__zugferd_unit_code_tl \l_tmpa_tl
669   }
670 },
671 unit .usage:n = general,
672 }
673
674 \msg_new:nnnn {zugferd} {unknown-unit} {
675 You-selected-a-unit-which-was-not-predefined.\}
676 I~will-directly-use-your-selection~'unit=#1'~as~Common~Code.
677 Please-ensure-the-selection-is-valid!
678 } {
679 For-more-information-see-the-zugferd-documentation\}
680 and-the~/UN/CEFACT-Common-Code-list.
681 }

```

category

```

exemption-reason
rate
682 \msg_new:nnnn {zugferd} {unknown-tax-category} {
683 You-selected-an-unknown-tax-category.\}
684 I~will-directly-use-your-selection~'category=#1'~as~Code.\}
685 Please-ensure-the-selection-is-valid!
686 } {
687 For-more-information-see-the-zugferd-documentation\}
688 and-the~corresponding~code~list.
689 }
690 \msg_new:nnnn {zugferd} {no-auto-exemption} {
691 You-selected-tax/category=#1-together-with-the-auto-exemption-option.\}
692 I-don't-have-any-pre-configured-exemption-setting-for-category=#1.\}
693 Please-make-sure-you-add-a-valid-setting-yourself.
694 } {
695 For-more-information-see-the-zugferd-documentation.
696 }
697 \keys_define:nn { zugferd / tax } {
698 category .choices:nn = {S,Z,E,AE,K,G,O,L,M} {
699 \tl_set_eq:NN \l__zugferd_tax_category_code_tl \l_keys_choice_tl
700 \bool_if:NT \g__zugferd_auto_exemption_bool {
701 \keys_if_choice_exist:nnTF {zugferd/tax} {exemption-reason-auto} {#1} {
702 \keys_set:nn {zugferd/tax} {exemption-reason-auto=#1}
703 } {
704 \msg_warning:nnn { zugferd } {no-auto-exemption} {#1}
705 }
706 }
707 },

```

```

708 exemption-reason .tl_set:N = \l__zugferd_tax_exemption_reason_tl,
709 exemption-reason .initial:V = \c_empty_tl,
710 exemption-reason .usage:n = general,
711 exemption-reason-code .tl_set:N = \l__zugferd_tax_exemption_code_tl,
712 exemption-reason-code .initial:V = \c_empty_tl,
713 exemption-reason .usage:n = general,

```

```

714 exemption-reason-auto .choice:,
715 exemption-reason-auto / S .code:n = {
716 \keys_set:nn {zugferd/tax} {exemption-reason=,exemption-reason-code=}
717 },
718 exemption-reason-auto / K .code:n = {

```

```

719 \keys_set:nn {zugferd/tax} {
720   exemption-reason= Intra-Community-Supply,
721   exemption-reason-code={vatex-eu-ic}
722 }
723 },
724 exemption-reason-auto / AE .code:n = {
725   \keys_set:nn {zugferd/tax}{
726     exemption-reason=Reverse-Charge,
727     exemption-reason-code={vatex-eu-ae}
728   }
729 },
730 exemption-reason-auto / G .code:n = {
731   \keys_set:nn {zugferd/tax}{
732     exemption-reason=Export-outside-the-EU,
733     exemption-reason-code={vatex-eu-g}
734   }
735 },
736 exemption-reason-auto / O .code:n = {
737   \keys_set:nn {zugferd/tax}{
738     exemption-reason=No-subject-to-VAT,
739     exemption-reason-code={vatex-eu-o}
740   }
741 },
742 standard .meta:n = {category=S},
743 zero .meta:n = {category=Z},
744 exempt .meta:n = {category=E},
745 reverse-charge .meta:n = {category=AE},
746 intra-community .meta:n = {category=K},
747 EEA .meta:n = {category=K},
748 export .meta:n = {category=G},
749 canary-islands .meta:n = {category=L},
750 ceuta .meta:n = {category=M},
751 melilla .meta:n = {category=M},
752 category / unknown .code:n = {
753   \exp_args:Nnnx \keys_if_choice_exist:nnnTF {zugferd} {category}
754   {\str_uppercase:f {#1}}
755   {
756     \keys_set:nx { zugferd } {category= {\str_uppercase:f {#1}}}
757   } {
758     \msg_warning:nnn { zugferd } {unknown-tax-category} {#1}
759   }
760 },
761 category .initial:n = S,
762 category .usage:n = general,
763 rate .fp_set:N = \l__zugferd_tax_rate_fp,
764 rate .initial:n = 19,
765 rate .usage:n = general
766 }%
767 \keys_define:nn {zugferd/item} {
768   tax .choice:,
769   tax / unknown .code:n = \keys_set:ne {zugferd/tax} { \l_keys_key_str = \exp_not:V \l_keys_v
770   unknown .code:n = \keys_set:ne {zugferd} { \l_keys_key_str = \exp_not:V \l_keys_value_tl}
771 }

```

start-date

end-date

```
772 \keys_define:nn { zugferd / item } {
773   start-date .tl_gset:N = \l__zugferd_start_date_tl,
774   start-date .initial:n =,
775   end-date .tl_gset:N = \l__zugferd_end_date_tl,
776   end-date .initial:n =,
777 }%

778 \__zugferd_define_xml_writer:Nn \__zugferd_write_inline:nn {
779   \tl_if_blank:nF {#2} {
780     <#1>#2</#1>
781   }
782 }
783 \cs_generate_variant:Nn \__zugferd_write_inline:nn {ne}
784 \__zugferd_define_xml_writer:Nn \__zugferd_write_inline_i:nn {
785   \__zugferd_indent: \__zugferd_write_inline:nn {#1} {#2}
786 }
787 \cctab_begin:N \g__zugferd_xml_cctab%
788 %
789 \__zugferd_define_xml_writer:Nn \__zugferd_write_note:n {%
790   \begingroup%
791     \let\\iow_newline:%
792     \__zugferd_write_xml:e {%
793       <ram:IncludedNote>
794         \__zugferd_indent: <ram:Content>
795         #1
796         \__zugferd_indent: </ram:Content>
797       </ram:IncludedNote>%
798     }%
799   \endgroup%
800 }%
801 %
802 %
803 \begingroup%
804 \char_set_catcode_other:N \#%
805 \char_set_catcode:nn {32}{10}%
806 \__zugferd_define_xml_writer:Nn \__zugferd_insert_Header: {%
807   \__zugferd_write_xml:e {%
808     <?xml version='1.0' encoding='UTF-8' ?>
809     <rsm:CrossIndustryInvoice %
810       xmlns:rsm="urn:un:unece:uncefact:data:standard:CrossIndustryInvoice:100" %
811       xmlns:qdt="urn:un:unece:uncefact:data:standard:QualifiedDataType:100" %
812       xmlns:ram="urn:un:unece:uncefact:data:standard:ReusableAggregateBusinessInformationEntity:
813       xmlns:xs="http://www.w3.org/2001/XMLSchema" %
814       xmlns:udt="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:100">%
815   }%
816   \int_gincr:N \g__zugferd_indent_int%
817   \__zugferd_write_xml:n {<rsm:ExchangedDocumentContext>}%
818   \int_gincr:N \g__zugferd_indent_int%
819   \__zugferd_write_xml:e {%
820     \str_if_empty:NF \g__zugferd_businessProcessId_str {%
821       <ram:BusinessProcessSpecifiedDocumentContextParameter>
822       \__zugferd_indent: <ram:ID>\g__zugferd_businessProcessId_str</ram:ID>
823       </ram:BusinessProcessSpecifiedDocumentContextParameter>
```



```

824 }%
825 <ram:GuidelineSpecifiedDocumentContextParameter>
826 \__zugferd_indent: <ram:ID>\g__zugferd_format_str</ram:ID>
827 </ram:GuidelineSpecifiedDocumentContextParameter>%
828 }%
829 \int_gdecr:N \g__zugferd_indent_int%
830 \__zugferd_write_xml:n {</rsm:ExchangedDocumentContext>}%
831 }%
832 \endgroup%
833 \__zugferd_define_xml_writer:Nn \__zugferd_insert_FrontMatter: {%
834 \__zugferd_write_xml:n {<rsm:ExchangedDocument>}%
835 \int_gincr:N \g__zugferd_indent_int%
836 \__zugferd_write_xml:e {%
837 <ram:ID>\g__zugferd_id_t1</ram:ID>
838 <ram:TypeCode>\g__zugferd_DocumentTypeCode_t1</ram:TypeCode>
839 <ram:IssueDateTime>
840 % space required!
841 \__zugferd_indent:<udt:DateTimeString~format="102">\g__zugferd_date_t1</udt:DateTimeString>
842 </ram:IssueDateTime>%
843 }%
844 \__zugferd_note_if_not_empty:N \g__zugferd_subject_t1%
845 \__zugferd_note_if_not_empty:N \g__zugferd_fromaddress_t1%
846 \seq_map_inline:Nn \g__zugferd_notes_seq {%
847 \__zugferd_write_note:n {##1}%
848 }%
849 \int_gdecr:N \g__zugferd_indent_int%
850 \__zugferd_write_xml:e {%
851 </rsm:ExchangedDocument>
852 <rsm:SupplyChainTradeTransaction>%
853 }%
854 \int_gincr:N \g__zugferd_indent_int%
855 }%
856 %
857 % footer
858 \__zugferd_define_xml_writer:Nn \__zugferd_insert_Footer: {%
859 \int_gdecr:N \g__zugferd_indent_int%
860 \__zugferd_write_xml:n {</rsm:SupplyChainTradeTransaction>}%
861 \int_gdecr:N \g__zugferd_indent_int%
862 \__zugferd_write_xml:n {</rsm:CrossIndustryInvoice>}%
863 }%
864 \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-

```
865 \cctab_begin:N \g__zugferd_xml_cctab%
```

`__zugferd_AssociatedDocumentLineDocument:n`

```
866 \__zugferd_define_xml_writer:Nn \__zugferd_AssociatedDocumentLineDocument:n {%
867 <ram:AssociatedDocumentLineDocument>
868 \__zugferd_indent: <ram:LineID>#1</ram:LineID>
869 </ram:AssociatedDocumentLineDocument>%
870 }%
```

`__zugferd_SpecifiedTradeProduct:nn`

```
871 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeProduct:nn {%
872 <ram:SpecifiedTradeProduct>
873 \tl_if_empty:nF {#1} {%
874 \__zugferd_indent: <ram:SellerAssignedID>#1</ram:SellerAssignedID>
875 }%
876 \__zugferd_indent: <ram:Name>#2</ram:Name>
877 </ram:SpecifiedTradeProduct>%
878 }%
```

`__zugferd_ProductTradePrice:nn`

```
879 \__zugferd_define_xml_writer:Nn \__zugferd_ProductTradePrice:nn {%
880 <ram:\str_uppercase:n #1PriceProductTradePrice>
881 \__zugferd_indent: <ram:ChargeAmount>#2</ram:ChargeAmount>
882 </ram:\str_uppercase:n #1PriceProductTradePrice>%
883 }%
```

`__zugferd_SpecifiedLineTradeAgreement:nn`

```
884 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedLineTradeAgreement:nn {%
885 <ram:SpecifiedLineTradeAgreement>
886 \__zugferd_indent:<ram:GrossPriceProductTradePrice>
887 \__zugferd_indent:n {2} <ram:ChargeAmount>#1</ram:ChargeAmount>
888 \__zugferd_indent:</ram:GrossPriceProductTradePrice>
889 \__zugferd_indent:<ram:NetPriceProductTradePrice>
890 \__zugferd_indent:n {2} <ram:ChargeAmount>#2</ram:ChargeAmount>
891 \__zugferd_indent:</ram:NetPriceProductTradePrice>
892 </ram:SpecifiedLineTradeAgreement>
893 }%
```

6.4.1 SpecifiedLineTradeDelivery

```
:nn <unit code > <number>
```

_zugferd_SpecifiedLineTradeDelivery:nn

```
894 \_zugferd_define_xml_content:Nn \_zugferd_SpecifiedLineTradeDelivery:nn {%  
895 <ram:SpecifiedLineTradeDelivery>  
896 % SPACE!  
897 \_zugferd_indent: <ram:BilledQuantity~unitCode="#1">#2</ram:BilledQuantity>  
898 </ram:SpecifiedLineTradeDelivery>  
899 }%
```

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

```
900 \__zugferd_define_xml_writer:Nn \__zugferd_write_SpecifiedLineTradeSettlement:nnn {%
901 \__zugferd_write_xml:n {<ram:SpecifiedLineTradeSettlement>}%
902 \int_gincr:N \g__zugferd_indent_int%
903 \__zugferd_write_xml:e {\__zugferd_Line_ApplicableTradeTax:nn {#1} {#2}}%
904 \__zugferd_write_xml:e {\__zugferd_BillingSpecifiedPeriod:VV \l__zugferd_start_date_tl \l
905 \__zugferd_write_xml:e {\__zugferd_SpecifiedTradeSettlementLineMonetarySummation:n {#3}}%
906 \int_gdecr:N \g__zugferd_indent_int%
907 \__zugferd_write_xml:n {</ram:SpecifiedLineTradeSettlement>}%
908 }%
909 \cs_generate_variant:Nn \__zugferd_write_SpecifiedLineTradeSettlement:nnn {Vnn}%
910 \__zugferd_define_xml_content:Nn \__zugferd_Line_ApplicableTradeTax:nn {%
911 <ram:ApplicableTradeTax>
912 %BT-151
913 \__zugferd_indent: <ram:TypeCode>VAT</ram:TypeCode>
914 %BT-151
915 \__zugferd_indent: <ram:CategoryCode>#1</ram:CategoryCode>
916 %BT-152
917 \__zugferd_indent: <ram:RateApplicablePercent>#2</ram:RateApplicablePercent>
918 </ram:ApplicableTradeTax>
919 }%
920 \__zugferd_define_xml_content:Nn \__zugferd_BillingSpecifiedPeriod:nn {%
921 \bool_lazy_or:nnF {\tl_if_blank_p:n {#1}} {\tl_if_blank_p:n {#2}} {%
922 <ram:BillingSpecifiedPeriod>
923 %<!-- BT-134 -->
924 \__zugferd_indent:<ram:StartDateTime>
925 \__zugferd_indent:n {2} <udt:DateTimeString~format="102">#1</udt:DateTimeString>
926 \__zugferd_indent:</ram:StartDateTime>
927 %<!-- BT-135 -->
928 \__zugferd_indent:<ram:EndDateTime>
929 \__zugferd_indent:n {2} <udt:DateTimeString~format="102">#2</udt:DateTimeString>
930 \__zugferd_indent:</ram:EndDateTime>
931 </ram:BillingSpecifiedPeriod>
932 }%
933 }%
934 \__zugferd_define_xml_content:Nn \__zugferd_SpecifiedTradeSettlementLineMonetarySummation:n
935 <ram:SpecifiedTradeSettlementLineMonetarySummation>
936 % BT-131
937 \__zugferd_indent: <ram:LineTotalAmount>#1</ram:LineTotalAmount>
938 </ram:SpecifiedTradeSettlementLineMonetarySummation>
939 }
940
941 \cctab_end:
942 \cs_generate_variant:Nn \__zugferd_SpecifiedLineTradeSettlement:nnn {Vnn}
943 \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.

```
944 \cs_new:Nn \__zugferd_SpecifiedLineTradeSettlement:nnnn {
945 \msg_warning:nnnn {zugferd} {macro-deprecated}
```

```

946 { \__zugferd_SpecifiedLineTradeSettlement:nnnn }
947 { \__zugferd_SpecifiedLineTradeSettlement:nnn }
948 \__zugferd_SpecifiedLineTradeSettlement:nnn {#2} {#3} {#4}
949 }
950 \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.

```

951 \__zugferd_define_xml_writer:Nn \__zugferd_insert_TradeLineItem:nnnnnn {
952 \__zugferd_write_xml:n {<ram:IncludedSupplyChainTradeLineItem>}
953 \int_gincr:N \g__zugferd_indent_int
954 \__zugferd_write_xml:e {\__zugferd_AssociatedDocumentLineDocument:n {#1}}%
955
956 \begingroup
957 \zugferd_disable_macros:
958 \__zugferd_write_xml:e {\__zugferd_SpecifiedTradeProduct:nn {#2} {#3}}
959 \endgroup
960
961 \__zugferd_write_xml:n {<ram:SpecifiedLineTradeAgreement>}
962 \int_gincr:N \g__zugferd_indent_int%
963 \__zugferd_write_xml:e {\__zugferd_ProductTradePrice:nn {net} {#4}}
964 \int_gdecr:N \g__zugferd_indent_int%
965 \__zugferd_write_xml:n {</ram:SpecifiedLineTradeAgreement>}
966 \__zugferd_write_xml:e {%
967 \__zugferd_SpecifiedLineTradeDelivery:nn { \l__zugferd_unit_code_tl } {#5}
968 }%
969 \__zugferd_write_SpecifiedLineTradeSettlement:Vnn \l__zugferd_tax_category_code_tl %
970 { \fp_use:N \l__zugferd_tax_rate_fp } {#6}%
971 \int_gdecr:N \g__zugferd_indent_int%
972 \__zugferd_write_xml:n {</ram:IncludedSupplyChainTradeLineItem>}
973 }

```

ApplicableHeaderTradeSettlement

```

974 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeSettlementHeaderMonetarySummation:
975 \__zugferd_write_xml:n {<ram:SpecifiedTradeSettlementHeaderMonetarySummation>}%
976 \int_gincr:N \g__zugferd_indent_int%
977 \__zugferd_write_rounded:nnn {2} {LineTotalAmount} {#1}
978 \__zugferd_write_rounded:nnn {2} {ChargeTotalAmount} {#2}
979 \__zugferd_write_rounded:nnn {2} {AllowanceTotalAmount} {#3}
980 \__zugferd_write_rounded:nnn {2} {TaxBasisTotalAmount} {#4}
981 \__zugferd_write_rounded:nnnn {2} {TaxTotalAmount} {~currencyID="\l__zugferd_currency_tl "}
982 \__zugferd_write_rounded:nnn {2} {GrandTotalAmount} {#6}
983 \__zugferd_write_rounded:nnn {2} {TotalPrepaidAmount} {#7}
984 \__zugferd_write_rounded:nnn {2} {DuePayableAmount} {#8}
985 \int_gdecr:N \g__zugferd_indent_int%
986 \__zugferd_write_xml:n {</ram:SpecifiedTradeSettlementHeaderMonetarySummation>}%
987 }%
988 % ApplicableTradeTax CategoryCode Rate BaseAmount Result
989 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableTradeTax:nnnn {%
990 \__zugferd_write_xml:n {<ram:ApplicableTradeTax>}
991 \int_gincr:N \g__zugferd_indent_int%
992 \__zugferd_write_rounded:nnn {2} {CalculatedAmount} {#4}%BT117

```

```

993 \__zugferd_write_xml:n {<ram:TypeCode>VAT</ram:TypeCode>}%BT118
994 \tl_if_blank:VF \l__zugferd_tax_exemption_reason_tl {%BT-120
995 \__zugferd_write_xml:e {<ram:ExemptionReason>
996 \l__zugferd_tax_exemption_reason_tl
997 </ram:ExemptionReason>}
998 }
999 {{\__zugferd_write_rounded:nnn {2} {BasisAmount} {#3}}}%BT-116
1000 \__zugferd_write_xml:e {<ram:CategoryCode>#1</ram:CategoryCode>}%BT-118
1001 \tl_if_blank:VF \l__zugferd_tax_exemption_code_tl {%BT121
1002 \__zugferd_write_xml:e {<ram:ExemptionReasonCode>
1003 \l__zugferd_tax_exemption_code_tl
1004 </ram:ExemptionReasonCode>}
1005 }
1006 \__zugferd_write_xml:n {<ram:RateApplicablePercent>#2</ram:RateApplicablePercent>}%BT-
119
1007 \int_gdecr:N \g__zugferd_indent_int%
1008 \__zugferd_write_xml:n {</ram:ApplicableTradeTax>}
1009 }%
1010 %
1011 \cctab_begin:N \g__zugferd_xml_cctab%
1012 % ApplicableHeaderTradeSettlement
1013 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeSettlement:nnnnnnnn {%
1014
1015 }%

```

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

```

1016 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeSettlement_start: {%
1017 \__zugferd_write_xml:n {<ram:ApplicableHeaderTradeSettlement>}%
1018 \int_gincr:N \g__zugferd_indent_int%
1019 \__zugferd_write_xml:e {<ram:InvoiceCurrencyCode>\l__zugferd_currency_tl</ram:InvoiceCurren
1020 }%
1021 \__zugferd_define_xml_writer:Nn \__zugferd_ApplicableHeaderTradeSettlement_stop: {%
1022 \int_gdecr:N \g__zugferd_indent_int%
1023 \__zugferd_write_xml:n {</ram:ApplicableHeaderTradeSettlement>}%
1024 }%

```

SpecifiedTradePaymentTerms

```

1025 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradePaymentTerms:nn {%
1026 \__zugferd_write_xml:n {<ram:SpecifiedTradePaymentTerms>}%
1027 \int_gincr:N \g__zugferd_indent_int%
1028 \__zugferd_write_xml:e {%
1029 \tl_if_blank:nF {#1} {%
1030 <ram:Description>#1</ram:Description>%
1031 }%
1032 \tl_if_blank:nF {#2} {%
1033 <ram:DueDateDateTime>
1034 \__zugferd_indent: <udt:DateTimeString~format="102">#2</udt:DateTimeString>
1035 </ram:DueDateDateTime>%
1036 }%
1037 }%
1038 \int_gdecr:N \g__zugferd_indent_int%
1039 \__zugferd_write_xml:n {</ram:SpecifiedTradePaymentTerms>}%
1040 }%
1041 %

```

```

1042 %
1043 % sums
1044 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeSettlementPaymentMeans:nnn {%
1045 \bool_if:NT \g__zugferd_writePaymentMeans_bool {%
1046 \__zugferd_write_xml:n {<ram:SpecifiedTradeSettlementPaymentMeans>}%
1047 \int_gincr:N \g__zugferd_indent_int%
1048 \__zugferd_write_xml:e {%
1049 <ram:TypeCode>\g__zugferd_payment_means_tl</ram:TypeCode>
1050 \__zugferd_write_inline:ne {ram:Information} {\prop_item:cV {g__zugferd_payment-
means_names_prop} \g__zugferd_payment_means_tl}%
1051 \tl_if_blank:nF {#1#2} {%
1052 <ram:PayeePartyCreditorFinancialAccount>
1053 \__zugferd_write_inline_i:nn {ram:IBANID} {#2}
1054 \__zugferd_write_inline_i:nn {ram:AccountName} {#1}
1055 </ram:PayeePartyCreditorFinancialAccount>
1056 }%
1057 \tl_if_blank:nF {#3} {%
1058 <ram:PayeeSpecifiedCreditorFinancialInstitution>
1059 \__zugferd_write_inline_i:nn {ram:BICID} {#3}
1060 </ram:PayeeSpecifiedCreditorFinancialInstitution>%
1061 }%
1062 }%
1063 \int_gdecr:N \g__zugferd_indent_int%
1064 \__zugferd_write_xml:n {</ram:SpecifiedTradeSettlementPaymentMeans>}%
1065 }%
1066 }%
1067 \cctab_end:
1068 \cs_generate_variant:Nn \__zugferd_SpecifiedTradePaymentTerms:nn {VV}
1069 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradePaymentTerms: {
1070 \__zugferd_SpecifiedTradePaymentTerms:VV \g__zugferd_payment_terms_str \g__zugferd_due_date
1071 }

```

__zugferd_SpecifiedTradeSettlementPaymentMeans:

```

1072 \cs_generate_variant:Nn \__zugferd_SpecifiedTradeSettlementPaymentMeans:nnn {vvv}
1073 \__zugferd_define_xml_writer:Nn \__zugferd_SpecifiedTradeSettlementPaymentMeans: {
1074 \tl_if_blank:VF \g__zugferd_payment_means_tl {
1075 \__zugferd_SpecifiedTradeSettlementPaymentMeans:vvv
1076 {g__zugferd_payment_account-holder_tl} {g__zugferd_payment_iban_tl} {g__zugferd_payment_b
1077 }
1078 }

1079 \msg_new:nnn {zugferd} {macro-deprecated} {
1080 The-function-#1-is-deprecated.\
1081 It-was-replaced-by-#2.\
1082 Please-adjust-your-mechanism-to-use-the-new-version.
1083 }

```

Change History

v0.4	end-date: Deprecate old syntax and add public interfaces.	36	v0.8	General: First CTAN version	1, 13
v0.6	General: Provide public interfaces and first version of the documentation.	5	v0.8a	General: Use the new public interface for l3pdfmeta.	15
v0.7	General: Added exemption-reason-auto key for pre-configured exemption-reasons.	30	v0.9	General: Add interface using an argument for the keyval options locally per item	20
	auto-exemption: Added auto-exemption option	13	end-date: Split SpecifiedLineTradeSettlement to be more flexible and add support for BillingSpecifiedPeriod	36	

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] *Mustangproject*. URL: <https://github.com/ZUGFeRD/mustangproject> (visited on 08/20/2024).
- [8] *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).
- [9] *VAT exemption reason code list*. URL: https://www.xrepository.de/details/urn:xoev-de:kosit:codeliste:vatex_1 (visited on 08/20/2024).
- [10] *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	
<code>\#</code>	7, 57, 321, 804
<code>\\$</code>	322
<code>\%</code>	64, 134, 197, 320, 389, 404, 412, 675, 679, 683, 684, 687, 691, 692, 791, 1080, 1081
<code>\{</code>	318
<code>\}</code>	319
<code><party>/address</code> (option)	10
<code><party>/city</code> (option)	10
<code><party>/contact</code> (option)	10
<code><party>/country</code> (option)	10
<code><party>/email</code> (option)	10
<code><party>/name</code> (option)	10
<code><party>/postcode</code> (option)	10
<code><party>/vavid</code> (option)	10
<code><party>/address</code> (option)	<u>400</u>
A	
<code>add-note</code> (option)	9, <u>377</u>
<code>AddressData</code> (option)	4
<code>auto-exemption</code> (option)	4, <u>7</u>
B	
<code>\beginngroup</code>	188, 790, 803, 956
bool commands:	
<code>\bool_gset_false:N</code>	34, 35, 154
<code>\bool_gset_true:N</code>	10, 11
<code>\bool_if:NTF</code>	61, 156, 158, 189, 203, 206, 223, 258, 270, 494, 700, 1045
<code>\bool_lazy_and:nnTF</code>	548
<code>\bool_lazy_or:nnTF</code>	921
<code>\bool_new:N</code>	4, 5
<code>\bool_set_true:N</code>	148
<code>\g_tmpa_bool</code>	148, 154, 156, 158
<code>buyer/reference</code> (option)	11, <u>581</u>
C	
<code>category</code> (option)	<u>682</u>
cctab commands:	
<code>\cctab_begin:N</code> ...	461, 787, 865, 1011
<code>\cctab_end:</code>	576, 864, 941, 1067
char commands:	
<code>\char_set_active_eq:nN</code>	190
<code>\char_set_catcode:nn</code>	805
<code>\char_set_catcode_other:N</code> ...	7, 804
<code>\char_set_catcode_parameter:N</code> ...	57
clist commands:	
<code>\clist_map_break:</code>	156, 158
<code>\clist_map_inline:nn</code>	132, 149, 150, 400, 630
cs commands:	
<code>\cs_generate_variant:Nn</code>	167, 269, 282, 296, 577, 578, 579, 580, 783, 909, 942, 943, 950, 1068, 1072
<code>\cs_gset:Nn</code>	224, 228, 233, 238, 242
<code>\cs_gset_eq:NN</code>	231, 241, 250, 251, 252, 253, 254, 255, 256
<code>\cs_if_exist:NTF</code>	62, 647
<code>\cs_new:Nn</code> 164, 222, 249, 263, 271, 274, 275, 278, 279, 284, 297, 301, 305, 308, 311, 314, 317, 395, 434, 444, 454, 944
<code>\cs_new:Npn</code>	168
<code>\cs_new_eq:NN</code>	274, 275
<code>\cs_set:Nn</code>	278, 279
<code>\cs_set_eq:NN</code>	277
<code>currency</code> (option)	8
D	
<code>date</code> (option)	8, <u>355</u>
<code>\day</code>	171, 359, 367
<code>\def</code>	604
<code>delivery-date</code> (option)	8, <u>355</u>
<code>document-type</code> (option)	8, <u>333</u>
<code>\DocumentMetadata</code>	65
<code>due-date</code> (option)	8
E	
<code>end-date</code> (option)	<u>772</u>
<code>\endgroup</code>	205, 799, 832, 959
environments:	
<code>ZUGFeRD</code>	4
<code>exemption-reason</code> (option)	<u>682</u>
exp commands:	
<code>\exp_args:Ne</code>	109, 435
<code>\exp_args:Nne</code>	645
<code>\exp_args:NnnV</code>	662
<code>\exp_args:Nnnx</code>	345, 753
<code>\exp_not:n</code>	769, 770
<code>\ExplSyntaxOff</code>	591
<code>\ExplSyntaxOn</code>	618
F	
<code>format</code> (option)	3, <u>7</u>
fp commands:	
<code>\fp_use:N</code>	970
<code>fromaddress</code> (option)	9, <u>377</u>

	G		<code>\l_keys_value_tl</code> 769, 770
group commands:			L
<code>\group_begin:</code> 207, 264			<code>\languagename</code> 647
<code>\group_end:</code> 213, 267			<code>\LaTeX</code> 64
	H		<code>\let</code> 128, 318, 319, 320, 321, 322, 791
hook commands:			M
<code>\hook_gput_code:nmn</code> 643			<code>\month</code> 170, 358, 366
	I	msg commands:	
<code>id</code> (option) 8, <u>333</u>		<code>\msg_error:nn</code> 69	
<code>\IfBooleanTF</code> 123		<code>\msg_info:nn</code> 193	
<code>\IfNoValueF</code> 174		<code>\msg_new:nnn</code> 196, 1079	
<code>\ignorespaces</code> 180		<code>\msg_new:nmnn</code> . . . 63, 388, 674, 682, 690	
<code>\InsertZUGFeRDData</code> 4		<code>\msg_warning:nnn</code> 350, 625, 667, 704, 758	
<code>\InsertZUGFeRDData</code> 17, 129, 162, 163		<code>\msg_warning:nmnn</code> 945	
<code>\InsertZugferdData</code> 17, 162			N
<code>\insertZugferdData</code> 163			<code>\NewDocumentCommand</code> 122, 129
int commands:			<code>\NewDocumentEnvironment</code> 173
<code>\int_compare:nNnTF</code>			O
. 358, 359, 366, 367, 413		options:	
<code>\int_eval:n</code> 626		<code><party>/address</code> 10	
<code>\int_gdecr:N</code> 527, 538, 540,		<code><party>/city</code> 10	
559, 561, 829, 849, 859, 861, 906,		<code><party>/contact</code> 10	
964, 971, 985, 1007, 1022, 1038, 1063		<code><party>/country</code> 10	
<code>\int_gincr:N</code> 516, 521, 532,		<code><party>/email</code> 10	
553, 555, 816, 818, 835, 854, 902,		<code><party>/name</code> 10	
953, 962, 976, 991, 1018, 1027, 1047		<code><party>/postcode</code> 10	
<code>\int_new:N</code> 304		<code><party>/vaid</code> 10	
<code>\int_set:Nn</code> 169, 170, 171		<code><party>/address</code> <u>400</u>	
iow commands:		<code>add-note</code> 9, <u>377</u>	
<code>\iow_close:N</code> 204		<code>AddressData</code> 4	
<code>\iow_new:N</code> 60		<code>auto-exemption</code> 4, <u>7</u>	
<code>\iow_newline:</code>		<code>buyer/reference</code> 11, <u>581</u>	
. . . . 111, 113, 115, 118, 315, 436, 791		<code>category</code> <u>682</u>	
<code>\iow_now:Nn</code> 272		<code>currency</code> 8	
<code>\iow_open:Nn</code> 191		<code>date</code> 8, <u>355</u>	
item/end-date (option) 13		<code>delivery-date</code> 8, <u>355</u>	
item/start-date (option) 13		<code>document-type</code> 8, <u>333</u>	
	J	<code>due-date</code> 8	
<code>\jobname</code> 50		<code>end-date</code> <u>772</u>	
	K	<code>exemption-reason</code> <u>682</u>	
keys commands:		<code>format</code> 3, <u>7</u>	
<code>\l_keys_choice_tl</code> . . 341, 622, 654, 699		<code>fromaddress</code> 9, <u>377</u>	
<code>\keys_define:nn</code> 8, 331,		<code>id</code> 8, <u>333</u>	
402, 582, 620, 631, 652, 697, 767, 772		<code>item/end-date</code> 13	
<code>\keys_if_choice_exist:nmnTF</code>		<code>item/start-date</code> 13	
. 345, 662, 701, 753		<code>payment-means</code> <u>591</u>	
<code>\l_keys_key_str</code> 769, 770		<code>payment-terms</code> 8, <u>375</u>	
<code>\l_keys_key_tl</code> 430		<code>rate</code> <u>682</u>	
<code>\keys_set:nn</code> 22, 28, 124, 125,		<code>set-today</code> 4	
265, 348, 406, 414, 421, 665, 702,		<code>start-date</code> <u>772</u>	
716, 719, 725, 731, 737, 756, 769, 770		<code>subject</code> 9, <u>377</u>	

tax/category	12	\seq_gput_right:Nn	386
tax/exemption-reason	12	\seq_item:Nn	407, 408, 415, 416, 417, 418, 422, 423, 424
tax/exemption-reason-code	12	\seq_map_inline:Nn	846
tax/rate	13	\seq_new:N	323
unit	12, 652	\seq_set_split:Nnn	404, 412
write-xml	3, 7	\l_tmpa_seq	404, 407, 408, 412, 413, 415, 416, 417, 418, 422, 423, 424
xml-file	3, 7	set-today (option)	4
xrechnung	7	\setupZUGFeRDStrings	637, 645
zugferd	3, 7	\SetZUGFeRDData	4
		\SetZUGFeRDData	7, 16, 122, 128, 175
		\SetZugferdData	128
		\sisetup	285
		siunitx commands:	
		\siunitx_number_format:nN	294
		\space	138, 306
		start-date (option)	772
		\startWritingZUGFeRDxml	5
		\startWritingZUGFeRDxml	4, 178, 187
		\stopWritingZUGFeRDxml	5
		\stopWritingZUGFeRDxml	5, 184, 202
		str commands:	
		\c_hash_str	73, 79
		\str_case:nNTF	130
		\str_gset:Nn	12, 16, 19, 25, 31, 36
		\str_if_empty:NTF	820
		\str_new:N	2, 3, 6
		\str_replace_all:Nnn	146, 147
		\str_set:Nn	145
		\str_uppercase:n	346, 348, 351, 661, 754, 756, 880, 882
		\l_tmpa_str	145, 146, 147, 151, 153
		\string	65
		subject (option)	9, 377
		T	
		tax/category (option)	12
		tax/exemption-reason (option)	12
		tax/exemption-reason-code (option)	12
		tax/rate (option)	13
		TeX and L ^A T _E X 2 _ε commands:	
		\scr@fromaddress@var	384
		\scr@invoice@var	338
		\scr@subject@var	379
		\scr@yourref@var	585
		\zugferd@paymentMeans@english	605
		\zugferd@paymentMeans@german	592, 604
		\zugferd@paymentMeans@ngerman	604
		\textbackslash	320
		\textbraceleft	318
		\textbraceright	319
		\textdollar	322
		\textnumbersign	321
		P	
payment-means (option)	591		
payment-terms (option)	8, 375		
pdf commands:			
\pdf_object_ref:n	217, 219		
pdfdict commands:			
\pdfdict_put:nnn	208, 210		
pdffile commands:			
\pdffile_embed_file:nnn	211		
pdfmanagement commands:			
\pdfmanagement_add:nnn	214, 219		
pdfmeta commands:			
\pdfmeta_xmp_add:n	109		
\pdfmeta_xmp_property_new:nnnnn	81, 88, 95, 102		
\pdfmeta_xmp_schema_new:nnn	76		
\pdfmeta_xmp_xmlns_new:nn	62, 72		
prg commands:			
\prg_do_nothing:	250, 251, 253, 254		
\prg_replicate:nn	309		
\ProcessKeyOptions	59		
prop commands:			
\prop_gput:Nnn	429		
\prop_gset_from_keyval:Nn	638		
\prop_if_empty:NTF	552, 644		
\prop_if_empty_p:N	548		
\prop_if_in:NnTF	133		
\prop_item:Nn	134, 137, 139, 435, 436, 438, 439, 440, 441, 442, 446, 447, 448, 449, 450, 451, 452, 456, 457, 458, 459, 523, 534, 1050		
\prop_new:N	401, 636		
\providecommand	162, 163, 592, 605		
		Q	
quark commands:			
\q_stop	165, 168		
		R	
rate (option)	682		
\RequirePackage	283		
		S	
seq commands:			
\seq_count:N	413		

\l__zugferd_end_date_tl ... 775, 904
\g__zugferd_format_str
..... 1, 16, 25, 31, 826
\g__zugferd_fromaddress_tl
..... 323, 383, 385, 845
\g__zugferd_id_tl . 323, 338, 339, 837
__zugferd_indent:
..... 22, 305, 309, 464, 466, 469,
471, 472, 479, 485, 496, 498, 501,
503, 506, 508, 565, 569, 785, 794,
796, 822, 826, 841, 868, 874, 876,
881, 886, 888, 889, 891, 897, 913,
915, 917, 924, 926, 928, 930, 937, 1034
__zugferd_indent:n . 308, 312, 502,
507, 566, 567, 568, 887, 890, 925, 929
\g__zugferd_indent_int . 304, 312,
516, 521, 527, 532, 538, 540, 553,
555, 559, 561, 816, 818, 829, 835,
849, 854, 859, 861, 902, 906, 953,
962, 964, 971, 976, 985, 991, 1007,
1018, 1022, 1027, 1038, 1047, 1063
__zugferd_insert_Footer: . 229, 858
__zugferd_insert_FrontMatter: .
..... 226, 833
__zugferd_insert_Header: . 225, 806
__zugferd_insert_TradeLineItem:nmnnnn
..... 5, 37, 232, 951
__zugferd_Line_ApplicableTradeTax:nn
..... 903, 910
__zugferd_note_if_not_empty:N .
..... 395, 844, 845
\g__zugferd_notes_seq .. 323, 386, 846
__zugferd_number_format:nNn ...
..... 21, 284, 296, 298
\g__zugferd_payment_means_tl ...
..... 619, 622, 626, 1049, 1050, 1074
\g__zugferd_payment_terms_str ..
..... 375, 1070
__zugferd_PostalTradeAddress:N
..... 25, 444, 525, 536
__zugferd_PostalTradeAddress:nmnnn
..... 437, 462, 476, 578
__zugferd_PostalTradeAddress:nmnnmnn
..... 445, 475, 579
__zugferd_PostalTradeAddress_
short:N 434, 557
__zugferd_ProductTradePrice:nn
..... 34, 879, 963
\g__zugferd_seller_AddressData_
prop 400, 523, 524, 525
__zugferd_set_today:N 18
__zugferd_set_today:n . 142, 164, 167
__zugferd_set_today_aux:w 165, 168
\g__zugferd_shipto_AddressData_
prop 400, 548, 552, 557
__zugferd_SpecifiedLineTradeAgreement:nn
..... 34, 884
__zugferd_SpecifiedLineTradeDelivery:nn
..... 35, 894, 967
__zugferd_SpecifiedLineTradeSettlement:nnn
..... 36, 942, 947, 948
__zugferd_SpecifiedLineTradeSettlement:nmnn
..... 944, 946, 950
__zugferd_SpecifiedTradePaymentTerms:
..... 243, 1069
__zugferd_SpecifiedTradePaymentTerms:nn
..... 1025, 1068, 1070
__zugferd_SpecifiedTradeProduct:nn
..... 34, 871, 958
__zugferd_SpecifiedTradeSettlementHeaderMonetarySumm
..... 244, 974
__zugferd_SpecifiedTradeSettlementLineMonetarySumm
..... 905, 934
__zugferd_SpecifiedTradeSettlementPaymentMeans:
..... 39, 236, 1073
__zugferd_SpecifiedTradeSettlementPaymentMeans:nmnn
..... 1044, 1072, 1075
\l__zugferd_start_date_tl . 773, 904
\g__zugferd_subject_tl
..... 323, 379, 381, 844
\l__zugferd_tax_category_code_tl
..... 699, 969
\l__zugferd_tax_exemption_code_
tl 711, 1001, 1003
\l__zugferd_tax_exemption_reason_
tl 708, 994, 996
\l__zugferd_tax_rate_fp ... 763, 970
\l__zugferd_tmp_tl 1, 298, 299
\l__zugferd_unit_code_tl 654, 668, 967
__zugferd_write_inline:nn
..... 778, 783, 785, 1050
__zugferd_write_inline_i:nn ...
..... 784, 1053, 1054, 1059
__zugferd_write_Item:nmnnmnnn ... 20
__zugferd_write_note:n 397, 789, 847
__zugferd_write_rounded:nmnn ...
..... 21, 301, 977,
978, 979, 980, 982, 983, 984, 992, 999
__zugferd_write_rounded:nmnnn ...
..... 21, 297, 302, 981
__zugferd_write_SpecifiedLineTradeSettlement:nmn
..... 900, 909, 969
__zugferd_write_TradeLineItem:nmnnmnn
..... 37
__zugferd_write_xml:n
..... 21, 271, 277, 282, 299,
515, 517, 522, 528, 533, 539, 541,

551, 554, 556, 560, 564, 572, 792, 461, 787, 865, 1011
807, 817, 819, 830, 834, 836, 850,	\g_zugferd_xml_file_tl . 49, 191, 211
860, 862, 901, 903, 904, 905, 907,	__zugferd_xml_newline_indent: .
952, 954, 958, 961, 963, 965, 966, 22, 190, 314
972, 975, 986, 990, 993, 995, 1000,	_zugferd_xml_writer_iow
1002, 1006, 1008, 1017, 1019, 1023, 60, 191, 204, 272
1026, 1028, 1039, 1046, 1048, 1064	\zugferd_disable_macros:
\g_zugferd_write_xml_bool 7
..... 41, 189, 203, 223, 258, 270	\zugferd_disable_XML_interfaces:
\g_zugferd_writePaymentMeans_ 7
bool	\zugferd_enable_XML_interfaces:
..... 1, 11, 35, 1045 6
\g_zugferd_writeTradeContact_	\zugferd_startInvoiceSums:
bool 6
..... 1, 10, 34, 494	\zugferd_stopInvoiceSums:
_zugferd_xml_auto_indent: 6
..... 22, 272, 311, 315, 436	\zugferd_write_Item:ennnnnn
\g_zugferd_xml_cctab 5
	\zugferd_write_Item:nnnnnn
 6
	\zugferd_write_Summation:nnnnnnnn ...
 6
	\zugferd_write_TaxEntry:nnnn
 6