

ZUGFeRD Composer Documentation

What does the Composer do?

The Composer enables the creation of valid **ZUGFeRD** (including embedding the data in a PDF file) and **XRechnung** files from a **text file** using a **configuration file**. The composer uses the extensive **ZUGFerd-csharp library** from Stephan Stapel

(source code at <https://github.com/stephanstapel/ZUGFeRD-csharp>) If you manage to successfully generate digital invoices using the composer, it would be appropriate to support the library author:

<https://github.com/sponsors/stephanstapel>

Starting with version 1.0.0.4, it is possible to automatically check the created XRechnung file using the official tool from the Coordination Office for IT Standards (KoSIT). The new options /v and /vH have been added for this purpose.

[Link to command line options and configuration settings](#)

Examples included

Preparation

Unzip the archive into the directory '**c:\ZUGFeRDComposer**' so that the directories

- C:\ZUGFeRDComposer\bin
- C:\ZUGFeRDComposer\Examples
- ...

are created. Only in this way do the references to the macros in the Word sample files and the PrintMulti configuration file. If you copy the files to a different directory, these have to be adjusted.

First attempt with hidden texts

There are several Word files with macros in the Examples\Word directory.

Incidentally, the example for the first attempt with hidden text is now also available as a LibreOffice Writer macro in the corresponding directory. To make it easy to call up, you should also link the macros to interface elements here.

If you would like to try out the composer, then

- Open the Word file with the macros **invoice_two_vat.docm** (or **invoice_two_vat.odt**) and click on the symbol  or on the symbol with the two rings. You may need to create links to the macros once.
- Alternatively, create a link from the desktop to the file 'RunFromClipboard.cmd' in the Examples subdirectory, open one of the Word files without macros or the LibreOffice file and copy the complete contents of the invoice to the clipboard. Now execute the link to the RunFromClipboard.cmd and a digital invoice should be created in the 'XRechnung' subdirectory.

Second attempt with mail merge functionality

There is another way to create digital invoices in Word, in this case completely without hidden texts.

- Open the file `MailMergeNoPattern.docm` and link the document to the Excel spreadsheet of the same name. Click on the only macro with the ☹️ symbol. The currently displayed invoice is saved as a ZUGFeRD invoice and the target directory is opened.

Translated with DeepL.com (free version)

How it works

The method with hidden texts

The trick with Word and other word processing programmes is to hide the text passages with the patterns so that they are not visible when printing. This is very easy with style sheets.

The Word macro file contains two style sheets and the following four macros:

- One for hiding the text with the style sheets 'Hidden' (mandatory fields) and 'Hidden optional'.
- To display them.
- Creation of a ZUGFeRD file (PDF) and an XRechnung (XML) with IFilter technology.
- Creation of an XRechnung with clipboard transfer.

The LibreOffice Writer macro file also contains two style sheets and four macros. With LibreOffice, it is possible to export the content as a UTF-8 text document and use it. If you try this in Word (regardless of the version), the name of the current document will always be changed. This method is therefore not suitable for Word.

The following macros are available in the LibreOffice examples:

- One for hiding the text with the style sheets 'Hidden' (mandatory fields) and 'Hidden optional'.
- To display them.
- Creation of a ZUGFeRD file (PDF) and an XRechnung file with a saved UTF-8 text file.
- Creation of both files with additional validation of the results, including an HTML report.

Visible

[162]Verk.·Postfach·(Adr3)[]

[40]DE[38][Verk.·PLZ]:[37][Verk.·Ort]:·

+

RECHNUNG·[01]2025-0002[03]INVOICE

Käuferreferenz: → [10]N/A·(Leitweg-ID)

Projektreferenz: → [11]P1234/[11a]Ref.·Name

Vertragsnummer: → [12]V1234

[05]EUR[NF]DE-de

RECHNUNGSADRESSE

[44][Käuf.·Firmenname]

DATUM·[02]01.03.2024

Zeitraum:·[73]2025-07-01-[74]10/31/2025

Lieferdatum: → [72]2025-03-05

Bestellnummer: → [13]B1234

Auftragsnummer: → [14]A1234

[58]kaeuf@mail.de

Hidden:

RECHNUNG· 2025-0002	DA
Käuferreferenz: → N/A (Leitweg-ID)	Zeit
Projektreferenz: → P1234/ <u>Ref.</u> ·Name	Lief
Vertragsnummer: → V1234	Bes
	Auf

To avoid destroying the layout, you would have to use tabs for formatting or use a fixed-pitch font for the hidden text, which must be of equal length.

The method with fields with defined names

In LibreOffice it is possible to create digital invoices. This is based on fields and data in a database (I assume).

[Here you find a video describing it](#)

The LibreOffice idea of using fields can also be applied to Word.

The example `MailMergeNoPattern.docm` shows how it works. You can configure fixed document fields (DOCPROPERTY) under the menu item `File/Information` and then reference them in the document.

This applies to all fixed components of the invoice, such as the seller's data or the bank details. Only the fields beginning with an underscore "_" are used for the digital invoice. Only precisely defined texts can be used. These are listed in the file 'DefaultExtractConfiguration.txt'.

The dynamic fields are integrated using the mail merge function (MERGEFIELD). You can still use hidden texts for unusual fields if you do not want them to appear on the invoice. However, it is easy to define other names for constants using an alias table.

For example, for a normal invoice, the value 'invoice' (BT-3) must be transferred. However, you do not necessarily want to see this on German invoices.

You can solve this problem by making entries in the alias table 'AliasTable.txt'. Always write the key in lower case! The replacements apply to all fields except text fields such as names, streets, etc.

```
# BT-3
rechnung:invoice
facture:invoice
```

The following image shows the mixture of fields

Empfänger

```
{ MERGEFIELD _buyer_name }
MERGEFORMAT }
{ MERGEFIELD _buyer_contact_name }
_seller_contact_name \* MERGEFORMAT }
{ MERGEFIELD _buyer_adr1 }
MERGEFORMAT }
{ MERGEFIELD _buyer_country }-{ MERGEFIELD _buyer_postcode } { MERGEFIELD _buyer_city
}
\* MERGEFORMAT }-{ DOCPROPERTY _seller_postcode \* MERGEFORMAT } {
DOCPROPERTY _seller_city \* MERGEFORMAT }
```

```
E-Mail: { MERGEFIELD _buyer_email }
_seller_email \* MERGEFORMAT }
USt-IdNr: { MERGEFIELD _buyer_vat }
_seller_phone \* MERGEFORMAT }
Leitweg-ID: { MERGEFIELD _invoice_buyer_reference }
\* MERGEFORMAT }
```

Absender

```
{ DOCPROPERTY _seller_name \*
{ DOCPROPERTY
{ DOCPROPERTY _seller_adr1 \*
{ DOCPROPERTY _seller_country
DOCPROPERTY _seller_postcode \* MERGEFORMAT } {
DOCPROPERTY _seller_vat }
```

```
E-Mail: { DOCPROPERTY
Tel.: { DOCPROPERTY
USt-IdNr. { DOCPROPERTY _seller_vat }
```

What about other file formats?

- **Libre Office**

Two examples are provided with macros that use hidden text. One document has a simple structure, while the other uses the MailMerge functionality and the supplied 'MailMerge.ods' file as its data source. You can also use the clipboard functionality or the technology via PrintMulti.

- **Excel**

Excel is of course very well suited for the automatic calculation of totals, taxes, etc., but it has a disadvantage here...

For an automatic calculation to work, the hidden texts must not be in the cells with numbers for calculation. So you would have to create a large number of columns for the hidden texts and then hide them using format templates.

However, a lot of hard work is involved, because in the general area you have to combine a lot of cells. This is certainly possible, but there is another option.

You can just define the invoice items in Excel and then embed the sheet in Word.

However, there is the disadvantage that neither the clipboard solution nor the IFilter technology work on the Word document.

Take a look at the example [invoice_excel.docx](#). The Excel files to be embedded all start with `embed_*.xlsx`.

A change in how the text file is saved supports PrintMulti from version 2.0.0.8.

There is another example for Excel without Word, in which the patterns are defined in comments.

However, the invoice creation then works exclusively via an Excel macro. Further information on how this works can be found in the example itself ([invoice_using_comments.xlsm](#)).

- **Html**

Invoices in this format are also widespread. There you can simply use **css** to switch between visible and invisible. Text extraction with IFilter and the clipboard seems to work. However, success with PrintMulti depends on the application doing the printing. Chrome-based browsers don't seem to work. Firefox does.

There is an incomplete example of embedding and using CSS in the examples directory.

How does the composer work?

The configuration file contains keys and patterns for searching the text file from which the data is extracted. The configuration file is structured line by line and divided into columns by separators. Comments start with `//` or `#`. Blank lines are ignored.

Here is an example:

```
// BT-1 invoice_nr:0:c:[01]:: Entries are normally separated by ';'. With sep=: the colon is used (as in Excel).
```

The structure is

```
[fixed key][sep][optional][sep][operator][sep][arg1][sep][arg2]
```

The possible keys are defined in the programme. The supplied configuration file contains all supported keys. Whether or not an argument is optional is ultimately determined by the official ZUGFeRD/XRechnung rules. If 'O' is specified here in the configuration file, the composer checks for the pattern and generates an error if necessary. Possible operators are

- **'c'**: Stands for Character. The pattern ends with the occurrence of one of the specified characters in **arg2**. Here we have selected a character from the Unicode character set **'∴'**, which normally does not occur in the text. You can, of course, choose a different character by changing the configuration file and the source files.
- **'='**: A constant in **arg1** (**arg2** is not used). It can also contain escape sequences such as **\r\n** if a line break is required.
- **'#'**: Probably not needed. In this case, **arg2** contains an exact string for the end of the pattern. **arg1** is the beginning of the pattern. In the example, the text '[01]' is searched for in the file, including the brackets. Most patterns (with the exception of invoice items) are constructed in such a way that the BT number is in brackets. The text file generated from the Word example contains, for example:
∴RECHNUNG [01]2025-0001∴[03]INVOICE The text **'2025-0001'** is set here as the invoice number, for example. Spaces at the beginning and end are removed. By using constants, parts such as the information about the supplier can be predefined and do not always have to be included in the text file.

Where does the text file come from?

The text file is the be-all and end-all. It must contain the patterns and texts in the correct order for the evaluation. There are several possibilities:

1. You can create the file yourself with a program and use the composer to avoid having to integrate the library yourself.
2. You can use [PrintMulti](#). More about this in the next chapter or in the help files in the PrintMulti subdirectory.

3. You pass a document with the option `'/IF'`. The composer then applies the `IFilter` associated with the document to generate the text file. This technique is used by Windows for full-text search. Unfortunately, this does not always work as desired. It is a requirement that the text is output in the correct order. This is not important for the full-text search.
4. You create a text file from the document and transfer it using the `'/IT'` option. You can specify the encoding of the file using the `'/CE'` option. The default encoding is UTF8. The LibreOffice macros use this option. Unfortunately, Word causes problems here. It is best if the text file begins with a `BOM` code. This clearly defines the encoding. Otherwise, the default is UTF-8 format. For unusual formats, there is the `/CE` option to specify the text format. Possible formats are displayed with `/?E`. PrintMulti generates UTF-16 and IFilter UTF-8 BOM files.

PrintMulti and ZUGFeRD

With `PrintMulti` it is possible to output a print job with many options on several printers at the same time. It is often used to simultaneously generate a PDF with the output in addition to printing. Since it is also possible to execute scripts, PrintMulti can take over both necessary steps

1. Create a PDF from the print job.
2. In many cases, PrintMulti can extract the text into a file that can then be used with the composer for automatic digital invoicing. This can be done without IFilter technology.

You can, for example, create a virtual printer 'ZUGFeRD Printer' and then automatically generate compliant digital invoices directly from your invoicing software by printing to this printer, generate compliant digital invoices.

Of course, this only works if your invoicing program allows you to insert and hide the necessary patterns.

PrintMulti is free of charge and without restrictions on client operating systems A sample configuration file is included in the archive in the **PrintMulti** folder.

ZUGFeRDComposer

Description of the command line parameters for ZUGFeRDComposer

If you call up the tool with `/?`, you will get help on the parameters. Here is a table with all options. The most important are shown in **bold**.

Path arguments can also include environment variables, e.g. `%temp%`.

Option	Description
<code>/?</code>	Help output
<code>/?V</code>	Output of allowed pattern and codes from the ZUGFeRD_csharp library for different input types
<code>/?E</code>	Output of supported encodings for special text files
Input	
<code>/IT</code>	<i>Text file to be used</i>

Option	Description
/IF	Source file for IFilter call
/iC	The clipboard content is used as text
/iE	No text file is used, e.g. when all data is passed as constants
/IP	PDF file for embedding the digital invoice
Output	
/OX	XML file to store the digital invoice
/OP	PDF output file with embedded invoice
Messages	
/mO	Show a message when the invoice has been successfully generated
/mE	Show a message when an error occurs
File/Directory	
/fO	Open the output path in Explorer when successful
/fL	Open the log4net file when an error occurs
Validation	
/v	Check the XRechnung document after creation
/vH	Create an HTML report file with the verification results
Logging	
/OF	IFilter text output
/OC	File with the content of the clipboard at /iC
/OJ	File that contains a line with various information for each call
/OE	File that contains a list of all recognised keys and values.
Miscellaneous	
	Changing standard settings
/ZV	ZUGFeRD version (see ZUGFerd-csharp)
/ZP	ZUGFeRD profile (see ZUGFerd-csharp)
/ZF	ZUGFeRD format (see ZUGFerd-csharp) Default: CII (alternative: UBL)
/CF	Name of a configuration file if the default file is not to be used

If the digital invoice is saved successfully, '0' is returned. If an error occurs, a different value is returned.

Configuration settings

Some settings can be configured in the `ZUGFeRDComposer.exe.config` file.

In addition to the default values for creating digital invoices, there are also some settings for validation:

Setting	Description
Log4NetFile	Log file from Log4Net
ZUGFeRDVersion*	ZUGFeRD version, if not overwritten via command line
ZUGFeRDProfile*	ZUGFeRD profile, if not overwritten via command line
ZUGFeRDFormat*	ZUGFeRD format, CII or UBL
SumCheckTolerance	Tolerance for rounding errors in the total sum
ValidatorResultsPath	Output path for the validation results. The default is the path of the generated XRechnung xml file
ValidatorJarMask	Filter on the jar file for calling the validator
ValidatorPath	Search path for the validator. If empty, all parent directories are searched for a directory named "Validator"
ValidatorScenarioFile	Name of the required "Scenario" file if you want to use a different one than the default
ValidatorExtOptions	Additional options for calling the validator. Call " <i>java -jar validator-1.6.0-standalone.jar</i> " for possible options.
JavaHome	Base path of the Java installation. You should use a version ≥ 21 . The default here is "%JAVA_HOME%".
JavaOptions	Depending on the Java and validation library used, you may need additional options.

* The values come from the [ZUGFerd-csharp](#) library. Use the **/?H** option to display the possible values.

Use of macros

Macros can be used in the names of **output** files. In this case, macros are fixed abbreviations that are replaced by other texts. They are enclosed in curly brackets, e.g. '{invoice_nr}' is replaced by the invoice number. Characters that are not allowed in a path are replaced by an underscore. In addition to the keys for the invoice texts, the following macros are available:

Macro	Description
{I}	name of the input file from /IF or /IT without file extension
{R}	result OK or ERROR
{Date}	yyyyMMdd
{Time}	HHmmss
{d}	day two digits

Macro	Description
{M}	month two digits
{y}	year four digits
{H}	Two-digit hour
{m}	Two-digit minute
{s}	Two-digit second

Examples of command lines can be found in the file 'Examples\Word\RunExamples.cmd' and 'Examples\RunFromClipboard.cmd'. Composer uses log4net technology. By default, a log file is created in C:\ProgramData\ZUGFeRDComposer\ZUGFeRDComposer.log with the level INFO. Log4net is controlled (file sizes, level, etc.) via the configuration file log4net.config in the program directory. The name of the log file is specified in the ZUGFeRDComposer.exe.config file.

What else is important

Digital invoices are international and require certain formats for dates and numbers.

Handling date fields

The composer automatically recognises dates in the format templates

- **dd.mm.yyyy** (e.g. 31.01.2025)
- **mm/dd/yyyy** (e.g. 01/31/2025)
- **yyyy-mm-dd** (e.g. 2025-01-31) In the sample Word calculation, all formats are mixed wildly.

Handling numbers

The rules apply to all values that expect floating point numbers. There are two possibilities

1. You specify the format using a (hidden) internal field 'number_format':
By default, it is defined as follows: number_format:1:c:[NF]:. In the example image with the visible patterns, you can see the field [NF]DE-de:.
2. Automatic detection based on the net amount of the first invoice item. Two decimal places are expected and either a comma or a period is used as a decimal point, or the other value are used as thousands separators. In principle, the extracted amounts are scanned from the end until a number is recognised. Everything up to that point is ignored. These are usually currency symbols.

Calculating the totals

The composer independently calculates all the totals (net, gross, tax) from all the invoice items and the tax

ZWISCHENSUMME	▣	[109]1.102,00€	▣
MWST-7%	▣	3,92€	▣
MWST-19%	▣	198,74€	▣
GESAMTBETRAG	▣	[112]1.304,66€	▣
BEREITS-BEGLICHEN	▣	[113]300,00€	▣
FÄLLIGER-BETRAG	▣	[115]1004,66€	▣

rates.

If you pass the optional amounts, the composer checks whether they match the calculated values and returns an error if they do not. The tolerance can be configured in the : `SumCheckTolerance`. See [chapter: 'Configuration Settings'](#)

Troubleshooting

It is unlikely that everything will work at the first attempt, and even if an invoice can be generated, it does not mean that it conforms to all the rules.

Log files from the composer

1. the option `/OE` creates a file that contains all patterns with values. The beginning looks like this for the example file

The grand total was not entered in the form for the last value and the composer calculated and added it

```
invoice_nr:(0)='2025-0001'
invoice_date:(0)='01.03.2024'
invoice_type:(0)='INVOICE'
invoice_currency:(0)='EUR'
invoice_payment_due_date:(0)='10.03.2024'
invoice_payment_description:(0)='Please settle the amount
by'
invoice_buyer_reference:(0)='12345678-12345-83'
invoice_project_reference:(0)='P1234'
invoice_project_reference_name:(0)='Ref Name'
...
item_id:(1)='1'
item_id:(2)='2'
item_name:(1)='Inflatable boat'
item_name:(2)='Water'
...
total_paid_amount:(0)='300,00€'
total_to_pay:(0)='(added)1004,66'
```

2. When using `IFilter` with `/IF`, a log file can be generated with the `/OF` option. This contains all the recognised texts in direct succession (unlike when calling `ifiltercmd`, which lists the texts one below the other).

Invoice [01]2025-0001;[03]INVOICE;Date [02]01.03.2024;Buyer reference:-[1 0]N/A;
(route ID)Project reference:-[11]P1234;/[11a]Ref. Name;Contract number:-[12]...

3. then there is the job log file, which contains a line for each call and possibly also error messages. The file currently has a fixed format.

```
Date;Result;Message;Invoice Nr; Invoice Date;Buyer name;Invoice sum; Invoice to pay
2025-01-16_19-11-56;OK;Successful;2025:-0002;01.03.2024;[Buyer company name];
€1,304.66;
2025-01-16_19-20-24;ERROR;Creator faulty: invoice_nr: BT-1 is required;;01.03.2024;
[Purch. Company name];€1,304.66;
```

4. The log4net file contains a lot of debug output

```
2025-01-16 19:11:57.307 [1] INFO ZUGFeRDComposer.Program - XML Data created.
Length: 9872 with options: Version:Version23, Profile:XRechnung, Format:CII
2025-01-16 19:11:57.307 [1] INFO ZUGFeRDComposer.Program - Write xml data to
file:C:\ZUGFeRDComposer\2025_-0002\invoice_two_vat_20250116_191156.xml
2025-01-16 19:11:57.449 [1] INFO ZUGFeRDComposer.Program - Xml data has been
successfully embedded in the PDF
file:C:\ZUGFeRDComposer\2025_-0002\invoice_two_vat_20250116_191156.pdf
2025-01-16 19:11:57.449 [1] INFO ZUGFeRDComposer.Program - 2025-01-16_19-1 1-
56;OK;Successful;2025:-0002;01.03.2024;[Buyer. Company Name];€1,304.66;
2025-01-16 19:20:24.181 [1] INFO ZUGFeRDComposer.Program - Start Application with
parameters: 'C:\ZUGFeRDComposer\bin\ZUGFeRDComposer.exe' /IF
'C:\ZUGFeRDComposer\Examples\invoice_two_vat.docm' /IP '
C:\ZUGFeRDComposer\Temp\invoice_two_vat20250116_192023.pdf' /OX
"C:\ZUGFeRDComposer{invoice_nr}\invoice_two_vat_{Date}{Time}.xml" /OP
'C:\ZUGFeRDComposer{invoice_nr}\invoice_two_vat{ Date}{Time}.pdf' /OF
"C:\ZUGFeRDComposer\Temp\invoice_two_vat{Date}{Time}ifilter.txt" /OJ
"C:\ZUGFeRDComposer\Log\Jobs.csv" /OE
"C:\ZUGFeRDComposer\Temp\invoice_two_vat{Date}{Time}_keys.txt"
2025-01-16 19:20:24.702 [1] ERROR ZUGFeRDComposer.Program - 2025-01-16_19-2 0-
24;ERROR;Creator faulty: invoice_nr: BT-1 is required;;01.03.2024;[Buyer. Company
name];€1,304.66;
```

Links

The following are suitable for checking and displaying the finished digital invoices

- [Portinvoice](#)
Very accurate inspection.
- [PDF24](#)
This can be used to create invoices with a GUI, import invoices and also save them as PDF (invoice view) or XML. The toolbox is free and really very helpful.
- [OpenXRechnungToolbox](#)
(Must be installed - can be called up for verification via command line.)

- In the c't in issue 26/2024 there is a detailed article and a [link list](#) descriptions of the BT fields
- <https://www.e-rechnung-bund.de/wp-content/uploads/2023/03/Uebersichtslisten-Eingabefelder-ZRE.pdf>
- <https://easyfirma.net/e-rechnung/zugferd/bt-felder>