

ZUGFeRD Composer Dokumentation

Was leistet der Composer?

Der Composer ermöglicht die Erstellung von gültigen **ZUGFeRD**- (inkl. Einbettung der Daten in eine PDF-Datei) und **XRechnung**-Dateien aus einer

- **Textdatei** mit Hilfe einer
- **Konfigurationsdatei**

Der Composer verwendet die umfangreiche Bibliothek **ZUGFerd-csharp** von Stephan Stapel (Quellcode unter <https://github.com/stephanstapel/ZUGFeRD-csharp>)

Wenn es Ihnen gelingt, mit dem Composer erfolgreich digitale Rechnungen zu erzeugen, wäre es angebracht den Autor der Bibliothek zu unterstützen: <https://github.com/sponsors/stephanstapel>

Enthaltene Beispiele

Vorbereitung

Entpacken Sie das Archiv in das Verzeichnis '**c:\ZUGFeRDComposer**', so dass die Verzeichnisse


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

erzeugt werden. Nur so passen die Verweise auf die Makros in den Word-Beispieldateien und der PrintMulti-Konfigurationsdatei. Beim Kopieren in ein anderes Verzeichnis müssen diese angepasst werden.

Erster Versuch mit versteckten Texten

Im Verzeichnis Examples\Word befinden sich mehrere Word-Dateien mit Makros.

Wenn Sie den Composer ausprobieren möchten, dann

- Öffnen Sie die Word Datei mit den Makros **invoice_two_vat.docm** und klicken oben Links auf das Symbol  oder auf das Zeichen mit den beiden Ringen.
- Alternativ erzeugen Sie einen Link vom Desktop auf die Datei **RunFromClipboard.cmd** im Examples Unterverzeichnis, öffnen eine der Worddateien ohne Makros oder die LibreOffice Datei und kopieren den kompletten Inhalt der Rechnung in die Zwischenablage. Jetzt noch den Link auf die RunFromClipboard.cmd ausführen und es sollte eine digitale Rechnung im Unterverzeichnis **XRechnung** erzeugt werden.

Zweiter Versuch mit Serienbriefunktionalität

Es gibt noch eine weitere Möglichkeit in Word digitale Rechnungen zu erstellen, in diesem Fall komplett ohne versteckte Texte.

- Öffnen Sie die Datei `MailMergeNoPattern.docm` und verknüpfen Sie das Dokument mit der gleichnamigen Excel-Tabelle. Klicken Sie auf das einzige Makro mit dem Symbol ☹️. Die aktuell angezeigte Rechnung wird als ZUGFeRD-Rechnung gespeichert und das Zielverzeichnis geöffnet.

Das steckt dahinter

Die Methode mit versteckten Texten

Der Trick mit Word oder anderen Textverarbeitungsprogrammen ist, die Textpassagen mit den Mustern so zu verstecken, dass sie beim Drucken nicht sichtbar sind. Mit Formatvorlagen ist das ganz einfach. Die Makro-Datei enthält zwei Formatvorlagen und die folgenden 4 Makros.

- Eines um Ausblenden der Texte mit den Formatvorlagen **Hidden** (Pflichtfelder) und **Hidden optional**.
- Um sie anzuzeigen
- Erzeugung einer ZUGFeRD-Datei (PDF) mit eingebetteter Rechnung mit IFilter-Technologie
- Erzeugung einer XRechnung Datei (XML) mit Clipboard Übergabe

Nachfolgend zwei Bilder mit verstecktem und sichtbarem Muster.

Sichtbar

[162]Verk.·Postfach-(Adr3);¶	
[40]DE;[38]Verk.·PLZ;[37]Verk.·Ort;·¶	
+	
RECHNUNG·[01]2025-0002;[03]INVOICE;¶	DATUM·[02]01.03.2024;¶
Käuferreferenz: → [10]N/A;·(Leitweg-ID);	Zeitraum:·[73]2025-07-01;-[74]10/31/2025;¶
Projektreferenz: → [11]P1234;/[11a]Ref.·Name;¶	Lieferdatum:· → [72]2025-03-05;¶
Vertragsnummer:→[12]V1234;¶	Bestellnummer: → [13]B1234;¶
[05]EUR;[NF]DE-de;¶	Auftragsnummer:· → [14]A1234;¶
RECHNUNGSADRESSE¶	¶
[44][Käuf.·Firmenname];¶	[58]kaeuf@mail.de;¶

Versteckt:

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

Um das Layout nicht zu zerstören, müssten Sie Tabulatoren für die Formatierung verwenden oder eine Schriftart mit festen Abständen für die versteckten Texte verwenden, die dann die gleiche Länge

haben müssen.

Die Methode mit Feldern mit definierten Namen

In LibreOffice ist es möglich, digitale Rechnungen zu erstellen. Dies basiert auf Feldern und Daten in einer Datenbank (vermute ich mal).

[Hier gibt es ein Video dazu](#)

Die LibreOffice Idee mit Feldern kann man auch auf Word übertragen.

Das Beispiel `MailMergeNoPattern.docm` zeigt wie es funktioniert. Man kann fest definierte Dokumentenfelder (DOCPROPERTY) konfigurieren beim Menüpunkt `Datei/Informationen` und diese dann im Dokument referenzieren.

Dies gilt für alle festen Bestandteile der Rechnung, wie z.B. die Daten des Verkäufers oder die Bankverbindung. Es werden nur die Felder für die digitale Rechnung verwendet, die mit einem Unterstrich "_" beginnen. Es können nur genau definierte Texte verwendet werden. Diese sind in der Datei `DefaultExtractConfiguration.txt` aufgelistet.

Die dynamischen Felder werden über die Seriendruckfunktion (MERGEFIELD) eingebunden. Man kann für ungewöhnliche Felder immer noch versteckte Texte verwenden, wenn diese nicht auf der Rechnung vorkommen sollen. Durch eine `Aliastabelle` ist es aber einfach möglich, andere Namen für Konstanten zu definieren.

Für eine normale Rechnung z.B.: muss der Wert `invoice` (BT-3) übergeben werden. Den will man auf deutschen Rechnungen aber nicht unbedingt sehen.

Durch Einträge in der Aliastabelle `AliasTable.txt` kann man das Problem lösen. Den Key immer in Kleinbuchstaben schreiben! Die Ersetzungen gelten für alle Felder bis auf Textfelder wie Namen, Straßen, ...

```
# BT-3
```

```
rechnung:invoice
```

```
facture:invoice
```

Im folgenden Bild sieht man die Mischung der Felder

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
}
{ DOCPROPERTY _seller_vat
}
```

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

Was ist mit anderen Dateiformaten?

- **Libre-Office**

Funktioniert mit Zwischenablage. Libre-Office erlaubt die Erstellung von Makros. Damit sollte die gleiche Funktionalität wie in Word möglich sein. Wie oben schon geschrieben, unterstützt LibreOffice auch direkt eine Erstellung von digitalen Rechnungen.

Es gibt auch eine Libre-Office Beispieldatei im Examples Verzeichnis - ohne Makros.

- **Excel**

Excel eignet sich natürlich sehr gut für eine automatische Berechnung der Summen, Steuern, ... , hat aber hier einen Nachteil...

Damit eine automatische Berechnung funktioniert, dürfen die versteckten Texte nicht in den Zellen mit Zahlen zur Berechnung stehen. Man müßte also eine Vielzahl an Spalten für die versteckten Texte erzeugen und diese dann über Formatvorlagen verstecken.

Es steckt aber sehr viel Fleißarbeit dahinter, denn im allgemeinen Bereich muss man sehr viele Zellen verbinden. Das ist sicherlich möglich, aber es gibt noch eine weitere Möglichkeit.

Man kann nur die Rechnungspositionen in Excel definiert und das Sheet dann in Word einbetten. Allerdings gibt es den Nachteil, dass weder die Zwischenablagelösung noch die IFilter-Technik auf dem Word-Dokument funktionieren.

Schauen Sie sich die das Beispiel [invoice_excel.docx](#) an. Die einzubettenden Excel Dateien fangen alle mit `embed_*.xlsx` an.

Durch eine Änderung beim Speichern der Textdatei unterstützt PrintMulti ab Version 2.0.0.8 diese Möglichkeit.

Es gibt ein weiteres Beispiel für Excel ohne Word, bei der die Pattern in Kommentaren definiert sind. Die Rechnungserstellung funktioniert dann aber ausschließlich über ein Excel-Makro. Weitere Informationen wie das funktioniert, finden Sie im Beispiel selber ([invoice_using_comments.xlsm](#))

- **Html**

Rechnungen in diesem Format sind ebenfalls weit verbreitet. Dort kann man mit **css** einfach zwischen sichtbar und unsichtbar umgeschaltet werden. Die Textextraktion mit IFilter und Clipboard scheint zu funktionieren. Der Erfolg mit PrintMulti hängt jedoch von der druckenden Anwendung ab. Chrome-basierte Browser scheinen nicht zu funktionieren. Firefox hingegen schon.

Es gibt ein unvollständiges Beispiel für die Einbettung und CSS-Verwendung im Examples-Verzeichnis.

Wie funktioniert der Composer?

Die Konfigurationsdatei enthält Schlüssel und Muster für die Suche in der Textdatei, aus der die Daten extrahiert werden.

Die Konfigurationsdatei ist zeilenweise aufgebaut und durch Trennzeichen in Spalten unterteilt. Kommentare beginnen mit // bzw. #. Leerzeilen werden ignoriert.

Hier ein Beispiel:

```
// BT-1
invoice_nr:0:c:[01]:;
```

Einträge werden normalerweise durch ';' getrennt. Mit **sep=:** wird der Doppelpunkt verwendet (wie in Excel).

Der Aufbau ist

```
[fester Key][sep][optional][sep][operator][sep][arg1][sep][arg2]
```

Die möglichen Schlüssel sind im Programm fest definiert. Die mitgelieferte Konfigurationsdatei enthält alle unterstützten Schlüssel. Ob ein Argument optional ist oder nicht, wird letztlich durch die offiziellen ZUGFeRD/XRechnung-Regeln bestimmt. Steht hier in der Konfigurationsdatei "0", so prüft der Composer das Vorhandensein des Patterns und generiert ggf. einen Fehler.

Mögliche Operatoren sind

- **'c'**: Steht für Character. Das Muster endet mit dem Auftreten eines der angegebenen Zeichen in **arg2**. Hier haben wir ein Zeichen aus dem Unicode-Zeichensatz **'⋮'** ausgewählt, das normalerweise nicht im Text vorkommt. Sie können selbstverständlich auch ein anderes Zeichen wählen, indem Sie die Konfigurationsdatei und die Quelldateien ändern.
- **'='**: Eine Konstante in **arg1** (**arg2** wird nicht verwendet). Sie kann auch Escapesequenzen wie **\r\n** enthalten, wenn ein Zeilenumbruch erforderlich ist.
- **'#'**: Wird wahrscheinlich nicht benötigt. In diesem Fall enthält **arg2** einen exakten String für das Pattern-Ende.

arg1 ist der Anfang des Patterns. Im Beispiel wird also nach dem Text **"[01]"** in der Datei inklusive der Klammern gesucht. Die meisten Muster (mit Ausnahme der Rechnungspositionen) sind so aufgebaut, dass die BT-Nummer in Klammern steht.

Die aus dem Word-Beispiel generierte Textdatei enthält z.B:

```
⋮RECHNUNG [01]2025-0001⋮[03]INVOICE
```

Der Text "2025-0001" wird hier z.B. als Rechnungsnummer gesetzt. Leerzeichen am Anfang und Ende werden entfernt.

Durch die Verwendung von Konstanten können Teile, wie z.B. die Angaben zum Lieferanten, fest vorgegeben werden und müssen nicht immer in der Textdatei enthalten sein.

Woher kommt die Textdatei?

Die Textdatei ist das A und O. Sie muss die Muster und Texte in der richtigen Reihenfolge für die Auswertung enthalten.

Es gibt mehrere Möglichkeiten:

1. Sie können die Datei selbst mit einem Programm erzeugen und den Composer verwenden, um die Bibliothek nicht selbst einbinden zu müssen.
2. Sie benutzen [PrintMulti](#). Dazu mehr im nächsten Kapitel oder in den Hilfedateien im PrintMulti Unterverzeichnis
3. Sie übergeben ein Dokument mit der Option '/IF'. Der Composer wendet dann den zum Dokument gehörenden [IFilter](#) an, um die Textdatei zu erzeugen. Diese Technik wird von Windows für die Volltextsuche verwendet.
Leider funktioniert dies nicht immer wie gewünscht. Voraussetzung ist, dass der Text in der richtigen Reihenfolge ausgegeben wird. Für die Volltextsuche spielt das keine Rolle.

Am besten ist, wenn die Textdatei mit einem [BOM](#)-Code beginnt. Dadurch wird die Kodierung eindeutig festgelegt. Standard ist ansonsten das UTF-8-Format.
Für ungewöhnliche Formate gibt es die Option [/CE](#), um das Format der Textformat vorzugeben. Mögliche Formate werden mit [/?E](#) ausgegeben. PrintMulti erzeugt UTF-16 und IFilter UTF-8 BOM-Dateien.

PrintMulti und ZUGFeRD

Mit [PrintMulti](#) ist es möglich einen Druckjob mit vielen Möglichkeiten auf mehreren Druckern gleichzeitig auszugeben.

Häufig wird es verwendet, um neben dem Druck gleichzeitig ein PDF mit der Ausgabe zu erzeugen. Da es auch möglich ist Skripte auszuführen, kann PrintMulti beide notwendigen Schritte übernehmen.

1. Erzeugen einer PDF aus dem Druckjob.
2. PrintMulti kann in vielen Fällen den Text in eine Datei extrahieren, die dann mit dem Composer zur automatischen digitalen Rechnungserstellung verwendet werden kann. Das geht also in dem Fall ohne IFilter Technik.

Sie können also z.B. einen virtuellen Drucker "ZUGFeRD Printer" erzeugen und dann automatisch direkt aus Ihrem Rechnungsprogramm durch Drucken auf diesen Drucker, konforme digitale Rechnungen erzeugen.

Das geht selbstverständlich nur, falls Ihr Rechnungsprogramm es ermöglicht die notwendigen Muster einzufügen und zu verstecken.

PrintMulti ist auf Clientbetriebssystemen kostenfrei ohne Einschränkungen

Im Order **PrintMulti** ist im Archiv eine Konfigurationsbeispieldatei enthalten.

ZUGFeRDComposer

Beschreibung der Kommandozeilenparameter von ZUGFeRDComposer

Wenn Sie das Tool mit `/?` aufrufen, erhalten Sie eine Hilfe zu den Parametern.

Hier ist eine Tabelle mit allen Optionen. Die wichtigsten sind **fett** dargestellt.

Pfade können auch Umgebungsvariablen z.B. `%temp%` enthalten.

Option	Beschreibung
<code>/?</code>	Ausgabe der Hilfe
<code>/?V</code>	Ausgabe der erlaubten Pattern Types und der Codes von der ZUGFeRD_csharp lib
<code>/?E</code>	Ausgabe der unterstützten Kodierungen für spezielle Textdateien
Input	
<code>/IT</code>	<i>Zu verwendenden Textdatei</i>
<code>/IF</code>	<i>Quelldatei für IFilter-Aufruf</i>
<code>/iC</code>	<i>Der Inhalt der Zwischenablage wird als Text verwendet</i>
<code>/iE</code>	<i>Es wird keine Textdatei verwendet z.B.: bei Übergabe aller Daten als Konstanten</i>
<code>/IP</code>	<i>PDF-Datei zur Einbettung der digitalen Rechnung</i>
Output	
<code>/OX</code>	<i>XML-Datei zur Speicherung der digitalen Rechnung</i>
<code>/OP</code>	<i>PDF-Ausgabedatei mit eingebetteter Rechnung</i>
Meldungen	
<code>/mO</code>	Zeige eine Meldung bei erfolgreicher Rechnungserzeugung
<code>/mE</code>	Zeige eine Meldung bei einem Fehler
Datei/Verzeichnis	
<code>/fO</code>	Den Ausgabepfad im Explorer bei Erfolg öffnen
<code>/fL</code>	Die log4net Datei bei einem Fehler öffnen
Logging	
<code>/OF</code>	IFilter Textausgaben
<code>/OC</code>	Datei mit dem Inhalt des Clipboards bei <code>/iC</code>

Option	Beschreibung
/OJ	Datei, die für jeden Aufruf eine Zeile mit diversen Informationen enthält
/OE	Datei, die eine Liste aller erkannten Schlüssel und Werte enthält.
Sonstiges	Ändern von Standardeinstellungen
/ZV	ZUGFeRD Version (siehe ZUGFerd-csharp)
/ZP	ZUGFeRD Profile (siehe ZUGFerd-csharp)
/ZF	ZUGFeRD Format (siehe ZUGFerd-csharp) Standard: CII (Alternative: UBL)
/CF	Name einer Konfigurationsdatei, falls die Standarddatei nicht verwendet werden soll

Bei erfolgreicher Speicherung der digitalen Rechnung wird \emptyset zurückgegeben. Im Fehlerfall ein anderer Wert.

Verwendung von Makros

In den Namen der **Ausgabedateien** können Makros verwendet werden. Makros sind in diesem Fall festgelegte Abkürzungen, die durch andere Texte ersetzt werden. Sie werden in geschweifte Klammern eingeschlossen, z.B. `{invoice_nr}` wird durch die Rechnungsnummer ersetzt.

Zeichen, die in einem Pfad nicht erlaubt sind, werden durch einen Unterstrich ersetzt.

Neben den Schlüsseln für die Rechnungstexte gibt es die folgenden Makros:

Makro	Beschreibung
{I}	Name der Eingabedatei aus /IF oder /IT ohne Dateiendung
{R}	Ergebnis OK oder ERROR
{Date}	yyyyMMdd
{Time}	HHmmss
{d}	Tag zweistellig
{M}	Monat zweistellig
{y}	Jahr vierstellig
{H}	Stunde zweistellig
{m}	Minute zweistellig
{s}	Sekunde zweistellig

Beispiele für die Kommandozeilen sind in der Datei `Examples\Word\RunExamples.cmd` und `Examples\RunFromClipboard.cmd` zu finden.

Der Composer verwendet die log4net-Technologie. Standardmäßig wird eine Protokolldatei in `C:\ProgramData\ZUGFeRDComposer\ZUGFeRDComposer.log` mit dem Level **INFO** erzeugt.

Die Steuerung von log4net (Dateigrößen, Level, ...) geschieht über die Konfigurationsdatei: `log4net.config` im Programmverzeichnis.

Der Name der Logdatei wird in der Datei `ZUGFeRDComposer.exe.config` festgelegt.

Was ist noch wichtig

Die digitalen Rechnungen sind international und verlangen bestimmte Formate von Datum und Zahlen.

Behandlung von Datumsfelder

Der Composer erkennt automatisch Datumsangaben in den Formatvorlagen

- **tt.mm.yyyy** (z.B. 31.01.2025)
- **mm/dd/yyyy** (z.B. 01/31/2025)
- **yyyy-mm-dd** (z.B. 2025-01-31)

In der Wordbeispielrechnung werden alle Formate wild gemischt.

Behandlung von Zahlen

Die Regeln gelten für alle Werte, die Fließkommazahlen erwarten.

Es gibt zwei Möglichkeiten

1. Sie geben durch ein (verstecktes) internes Feld `number_format` das Format vor:

Standardmäßig ist es folgendermaßen definiert:

```
number_format:1:c:[NF]:;
```

Im Beispielbild mit den sichtbaren Pattern, können Sie das Feld `[NF]DE-de;` erkennen.

2. Automatische Erkennung anhand des Nettobetrags der ersten Rechnungsposition.

Es werden zwei Dezimalstellen erwartet und Komma oder Punkt als Dezimalpunkt bzw. der jeweils andere Werte als tausender Trennzeichen verwendet.

Prinzipiell gilt, dass die extrahierten Beträge von hinten gescannt werden bis eine Zahl erkannt wird. Alles bis dahin wird ignoriert. Das sind normalerweise Währungssymbole.

Berechnung der Gesamtsummen

Der Composer berechnet aus allen Rechnungspositionen und den Steuersätzen selbstständig alle Summen (Netto, Brutto, Steuer).

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€;▣

Wenn Sie die optionalen Beträge übergeben, dann überprüft der Composer, ob diese mit den berechneten Werten übereinstimmen und liefert falls nicht einen Fehler zurück.

Fehlersuche

Es ist unwahrscheinlich, dass alles auf den ersten Anlauf funktioniert und selbst wenn eine Rechnung erzeugt werden kann, heisst das ja nicht, dass diese konform mit allen Regeln ist.

Logdateien vom Composer

1. mit der Option `/OE` wird eine Datei erzeugt, die alle Muster mit Werten enthält. Der Anfang sieht für die Beispieldatei so aus
Beim letzten Wert wurde die Gesamtsumme nicht im Formular hinterlegt und der Composer hat diese berechnet und hinzugefügt

```
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)='Bitte begleichen Sie den Betrag
bis zum'
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)='Schlauchboot'
item_name:(2)='Wasser'
...
total_paid_amount:(0)='300,00€'
total_to_pay:(0)='(added)1004,66'
```

2. bei der Verwendung von IFilter mit `/IF` kann mit der Option `/OF` eine Logdatei erzeugt werden. Diese enthält alle erkannten Texte direkt hintereinander (im Gegensatz zum Aufruf von `ifiltercmd`, welche die Texte untereinander auflistet)

```
Rechnung [01]2025-0001;[03]INVOICE;Datum [02]01.03.2024;Käuferreferenz:-[10]N/A;
(Leitweg-ID)Projektreferenz:-[11]P1234;/[11a]Ref. Name;Vertragsnummer:-[12]...
```

3. dann gibt es noch die Job-Logdatei, die für jeden Aufruf eine Zeile enthält und u.U. auch Fehlermeldungen. Die Datei hat aktuell ein festes 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;[Kauf.
Firmenname];1.304,66€;
```

```
2025-01-16_19-20-24;ERROR;Creator faulty: invoice_nr: BT-1 is required;;01.03.2024;
[Käuf. Firmenname];1.304,66€;
```

4. In der log4net Datei sind viele Debugausgaben enthalten

```
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_19-11-
56;OK;Successful;2025:-0002;01.03.2024;[Käuf. Firmenname];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_19-20-
24;ERROR;Creator faulty: invoice_nr: BT-1 is required;;01.03.2024;[Käuf.
Firmenname];1.304,66€;
```

Links

Zur Überprüfung und Anzeige der fertigen digitalen Rechnungen eignen sich

- [Portinvoice](#)
Sehr genaue Überprüfung.
- [PDF24](#)
Damit kann man Rechnungen mit einer Oberfläche erstellen, Rechnungen importieren und diese auch speichern als PDF (Rechnungsansicht) oder XML. Die Toolbox ist frei und wirklich sehr hilfreich.
- [OpenXRechnungToolbox](#)
(Muss installiert werden - kann per Kommandozeile zur Überprüfung aufgerufen werden.)
- In der c't im Heft 26/2024 ist ein ausführlicher Artikel und eine [Linkliste](#)

Beschreibungen der BT-Felder

- <https://www.e-rechnung-bund.de/wp-content/uploads/2023/03/Uebersichtslisten-Eingabefelder-ZRE.pdf>
- <https://easyfirma.net/e-rechnung/zugferd/bt-felder>