ZUGFeRD and PDF/A-3

Introduction

ZUGFeRD is the German acronym for Zentraler User Guide des Forums elektronische Rechnungen Deutschland which means the Central User Guide of the Forum for Electronic Invoicing in Germany.

ZUGFeRD utilizes structured XML to represents electronic invoice data. In addition, it chooses PDF format as a carrier to embed the XML invoice for exchanging and visualization. PDF can be recognized as a visualization layer and container of the electronic invoice. Combining the advantages of structured XML and PDF representation, the electronic invoice can be viewed as a traditional paper invoice, exchanged electronically, or further processed automatically in billing workflow depending on the requirements from the sender or receiver side.

In simple words, ZUGFeRD requires the PDF file compliant with PDF/A-3 standard. Each ZUGFeRD invoice must be a PDF/A-3 compliant file containing (embedding) one structured XML file representing the electronic invoice data. The main feature PDF/A-3 compliance is the capability to allow embedding extra files (e.g. XML data file) in the PDF.

eDocPrinter supports creating PDF files compliant with PDF/A-3 and various ZUGFeRD modes. ZUGFeRD mode of eDocPrinter **requires the feature and license of Ent. pack installed**. It also requires a ZUGFeRD XML assigned from the SaveAs dialog or by embedded commands to embed into the destination PDF.

Since Ver 9, eDocPrinter supports [ZUGFeRD 2.0], [ZUGFeRD 2.1], and [Factur-X] modes for PDF compliance with the latest ZUFFeRD e-invoice standard. It requires an ZUGFeRD XML assigned from the SaveAs dialog or by embedded commands. By default, in ZUGFeRD and Factur-X modes, it will use PDF/A-3b conformance. It will use PDF/A-3a when [Tagged PDF] is turned ON.

It also extends the embedded command eczf with more optional arguments for embedding XML invoice data in ZUGFeRD and Factur-X modes. The command syntax of eczf is [%%eczf: ZUGFeRD or Factur-X xml file path to be embedded, (optional) Filename if not using the default, (optional) Description, (optional) Data relationship, (optional) XML invoice data profile%%]. It adds a new parameter to support custom data relationship in the embedded commands eczf. The default value of the data relationship is Alternative for ZUGFeRD and Factur-x. Possible values include Data, Source, Alternative, Supplement, and Unspecified. It also adds new parameter to support specifying the profile name of the conformance level profile in the embedded command. Its default value is BASIC for ZUGFeRD and Factur-X. Possible values are MINIMUM, BASIC WL, BASIC, EN 16931, and EXTENDED.

In addition, an embedded command ezcx is added to support embedding extra supplement files in the ZUGFeRD or Factur-X PDF invoices. This syntax of this eczx embedded commands syntax is [%%eczx: the extra file path to be embedded, (optional) Filename if not using the default, (optional) Description, (optional) Data relationship]. The default value of the filename and description is the filename extracted from the path specified. The default value of the data relationship is Supplement if not specified. This

command is provided to help to embed extra files like delivery report, tracking information, product spec, or others in the PDF invoice as a file attachment.

For compliance, embedded commands eczf and eczx will automatically add extra file information (ModDate and Size) in the PDF EF's Params dictionary since the xml invoice data and extra supplement files are all embedded in the PDF as file attachments.

A new registry setting [DocInfoAutoVersion] with its default value True is added. It will automatically set the proper pdf version header from the PDF compliance standard selected. (PDF/A, ZUGFeRD, Factur-X) When it is True, PDF 1.7 is automatically set for the and Factur-X mode and ZUGFeRD 2.x modes.

Since Ver 10, eDocPrinter adds the support of xml template instantiation (expansion) for resolving variables and data items defined in the xml file from the eczf, eczf1, or eczfd commands (or from AnnZUGFeRD registry settings) when there are data items set by the special command %%zx. Basically, the xml file specified by the ZUGFeRD commands or from the SaveAs dialog will be treated as an invoice template. eDocPrinter uses the data from zx commands to expand the template to a compliant xml invoice.

The special embedded command zx. Is used to assign data items or variables referenced in the template xml file for generating the proper factur-x.xml dynamically by expanding (instantiating) the template xml supplied when printing jobs. For example, %%zx.doc.id: 280081%% can used to specify the id for in the invoice xml. %%zx.line.total: 99.90%% can be used as the amount of the line item of the invoice. %%zx.ahts.due: 230.48%% represents the amount receivable to be set in the xml. When instantiating, these parameters will be expanded in the template xml to replace the matched items with the command values assigned.

In addition, a new embedded command eczf1 to support "EN 16931" profile directly without specifying the profile parameter by the syntax [%%eczf1: ZUGFeRD or Factur-X xml file path to be embedded, (optional) Filename if not using the default, (optional) Description, (optional) Data relationship]. In contrast, the default profile is BASIC for ZUGFeRD and Factur-X of the original command eczf.

It also adds a new embedded command eczfd to support "EXTENDED" profile directly without specifying the profile parameter by the syntax [%%eczfd: ZUGFeRD or Factur-X xml file path to be embedded, (optional) Filename if not using the default, (optional) Description, (optional) Data relationship].

For flexibility, a registry setting and command AnnZUGXVar with default value True is added to allow users to use the environment variables and adv command variables in the %%zx. commands for instantiating the xml file with variables defined. For example, %%zx.doc.date: %#DATEY#%% will be dynamically expanded into the current date in YYYYMMDD when printing the job. Another registry setting and command AnnZUGXEsc with default value True is added to allow new lines in escaped backslash-n '\n' in the data items assigned by the %%zx. will be expanded into the new line in the factur-x.xml file.

Furthermore, [ZUGFeRD 2.2] and [ZUGFeRD 2.3] modes are added in the [Compliant with] options in the [Doc Info] property page for PDF compliance with the latest ZUFFeRD e-invoice standard. It requires an ZUGFeRD XML assigned from the SaveAs dialog or by embedded commands eczf, eczf1, or eczfd.

In Ver 10.04, the ZUGFeRD Word Addin is added to support handy ZUGFeRD PDF conversion using Word Mail Merge (Merge Fields) feature to load sales(order) data source specified to a Word (invoice) template to create a ZUGFeRD PDF containing the dynamically instantiated factur-x.xml automatically.

- [ZUGFeRD/Factur-X Addin- Word]: (32bit/64bit Addin) Add the ribbon button [Create ZUGFeRD PDF] to automate the workflow from starting Word mailing merge mode, merge the invoice fields in the template Word document, using these field values to instantiate data items or variables referenced in the template xml file for generating the factur-x.xml, and create ZUGFeRD PDF files by iterating all records from the data source containing the invoice details.

- [ZUGFeRD/Factur-X Addin- Word]: (32bit/64bit Addin) Add the ribbon button [Create/Email ZUGFeRD PDF] to automate the workflow from starting Word mailing merge mode, merge the invoice fields in the template Word document, using these field values to instantiate data items or variables referenced in the template xml file for generating the factur-x.xml, and create/email ZUGFeRD PDF files by iterating all records from the data source containing the invoice details.

- [ZUGFeRD/Factur-X Addin- Word]: (32bit/64bit Addin) Add the ZUGFeRD property page of the Word-Addin to configure related settings. It adds an option to select the factur-x.xml template file path. (AddinZUGXMLPath) It also adds an option to select the ZUGFeRD compliance profile. (AddinZUGProfile, default value is "EN 16931")

- [ZUGFeRD/Factur-X- commands]: Improve the embedded command zx. to support assigning data items or variables in a flatten way (without nested structure) in the template xml file for generating the proper factur-x.xml dynamically by expanding (instantiating) the template xml supplied when printing jobs. For example, %%zx.docid: 280081%% can specify the id for the invoice. %%zx.lntotal: 99.90%% can be used the amount of the line item of the invoice. %%zx.totaldue: 230.48%% will represent the amount receivable to be set in the xml.

- [ZUGFeRD/Factur-X Addin- Word]: (32bit/64bit Addin) Add a registry setting AddinSkipNormalDoc (with default value False) for ZUGFeRD and MailMerge Addin to automatically start MailMerge mode for normal document having no merge fields inserted. Users need to select a data source to proceed the converting process.

- [ZUGFeRD/Factur-X Addin- Word]: (32bit/64bit Addin) It will automatically turn ON detecting embedded commands and selecting the PDF/A-3 modes for users to convert to PDF using embedded commands in Word documents.

In Ver 10.08, the ZUGFeRD Excel Addin is added to support handy ZUGFeRD PDF conversion from an Excel worksheet containing formulas loading data from the table specified to create the invoice containing the dynamically populated factur-x.xml automatically.

- [ZUGFeRD/Factur-X- Excel Addin]: (32bit/64bit Addin) Add the ribbon button [Create ZUGFeRD PDF] to automate the PDF creation workflow of merging (filling) the invoice fields in the invoice Excel Worksheet, using these field values to instantiate data items or variables referenced in the template xml file for generating the factur-x.xml, and create ZUGFeRD PDF files of the active worksheet by iterating all records from the data source containing the invoice fields values. The

data source is an Excel table with default name "orders" imported from a Excel worksheet or others supported by Excel.

- [ZUGFeRD/Factur-X- Excel Addin]: (32bit/64bit Addin) Add the ribbon button [Create ZUGFeRD/Email PDF] to automate the PDF creation/email workflow of merging (filling) the invoice fields in the invoice Excel Worksheet, using these field values to instantiate data items or variables referenced in the template xml file for generating the factur-x.xml, and create and email ZUGFeRD PDF files of the active worksheet by iterating all records from the data source containing the invoice fields values. The data source is a Excel table with default name "orders" imported from a Excel worksheet or others supported by Excel.

- [ZUGFeRD/Factur-X- Excel Addin]: (32bit/64bit Addin) Add the ZUGFeRD property page of the Excel-Addin to configure related settings. It adds an option to select the factur-x.xml template file path. (AddinZUGXMLPath) It also adds an option to select the ZUGFeRD compliance profile. (AddinZUGProfile, default value is "EN 16931")

- [ZUGFeRD/Factur-X- Excel Addin]: (32bit/64bit Addin) Add the option [Data Table] for specifying the table name to load the invoice data (field values) for instantiating the template factur-x.xml. Its registry settting is AddinZUGTable in the corresponding registries of the addin settings with the default value "orders".

- [Digital Signature/PDFSealer- PDF/A Validation]: Support PDF/A validation compatibility by 3rd party validators (like veraPDF). Some 3rd party validators use fixed scanning of ByteRange key entry directly supposing that ByteRange must be in front of Contents in the digital signature dictionary. PDF spec does not assume key entries ordering. It requires only that ByteRange includes the whole PDF data excluding the hex signature itself.

- [ZUGFeRD/Factur-X- commands]: Support resolving the xml file path using the destination PDF directory in embedded commands eczf, eczf1, and eczfd when there is no directory specified in the xml filename in these commands. [%%eczf1: ZUGFeRD or Factur-X xml filename or full path to be embedded, ...]

- [ZUGFeRD/Factur-X- Office Addins]: (32bit/64bit Addin) Support resolving the xml file path using the destination PDF directory when there is no directory specified in the xml filename in the addin ZUGFeRD setting. (AddinZUGXMLPath can be the template xml filename or the template xml full path to be intantiated for embedding.)

For more on zx commands, please refer the links to zx command sample documents and its corresponding xml templates.

Features related to ZUGFeRD modes

1. The installer supports options in UI for installing ZUGFeRD features and adding printer instances enabling embedded commands detection and xml invoice data insertion compliant with ZUGFeRD directly.

2. A utility command line [addprn az3z "printer profile name"] is available for adding edocprinter instances with turning on [ZUGFeRD] mode.

3. [ZUGFeRD] modes are available to select in Doc Info property page. For creating ZUGFeRD invoices, please select PDF compliant with the preferred [ZUGFeRD] mode in Doc Info property page. It also requires a ZUGFeRD XML assigned from the SaveAs dialog or by embedded commands.

4. An embedded commands eczf is added for embedding the ZUGFeRD invoice XML file by commands without using the Saveas dialog for workflow automation. It requires the Ent. pack license installed. The command and its parameter list are like

   %%eczf: ZURFeRD xml file path to be embedded, filename if not using default, descriptions%%.

   For example,

   %%eczf: c:\invoices\ZURFeRD1.xml%%.
   It will embed ZURFeRD1.xml file as the invoice XML data and use the default standard ZUGFeRD-invoice.xml as the embedded filename for compliance. It requires the printer profile set in compliance mode ZUGFeRD Basic. By contrast, Adv. commands acef and acex are used for embedding other documents or files into the destination PDF with normal attributes.The command acef is used to embed a file with a visible icon in the PDF. Instead, the command acex is used to embed a file invisibly in the PDF. For example,

   %%acef: file path, icon type, [R,G,B], [flag, w, h, ox, oy], filename, description%%
   %%acef: c:\images\logo.jpg, Paperclip, 0,255,0%%

   %%acex: file path, filename , description%%
   %%acex: c:\data\report.xls%%

5. (since Ver. 9) [ZUGFeRD 2.0/2.1] modes are available to select in Doc Info property page for creating ZUGFeRD invoices complaint with ZUGFeRD 2.0/2.1.

6. (since Ver. 9) [Factur-X] modes are available to select in Doc Info property page for creating ZUGFeRD invoices complaint with Factur-X 1.0.

7. (since Ver. 10) [ZUGFeRD 2.2/2.3] modes are available to select in Doc Info property page for creating ZUGFeRD invoices complaint with ZUGFeRD 2.2/2.3.

Features related to PDF/A-3 mode

Listed below are new features related to PDF/A modes.

1. The installer supports options in UI for installing PDF/A-3 features and adding printer instances compliant with PDF/A-3 Basic directly.

2. New modes [az3a] and [az3b] of addprn command line utility are added to support adding printer instances compliant with PDF/A-3. For example, administrators may run [addprn az3a "printer profile name"] or [addprn az3b "printer profile name"].

3. PDF/A-2a, PDF/A-2b, PDF/A-2u, PDF/A-3a, PDF/A-3b, and PDF/A-3u modes are added in the [Compliant with] option in the Doc Info property page. Users can select to create PDF files compliant with the specific standard.

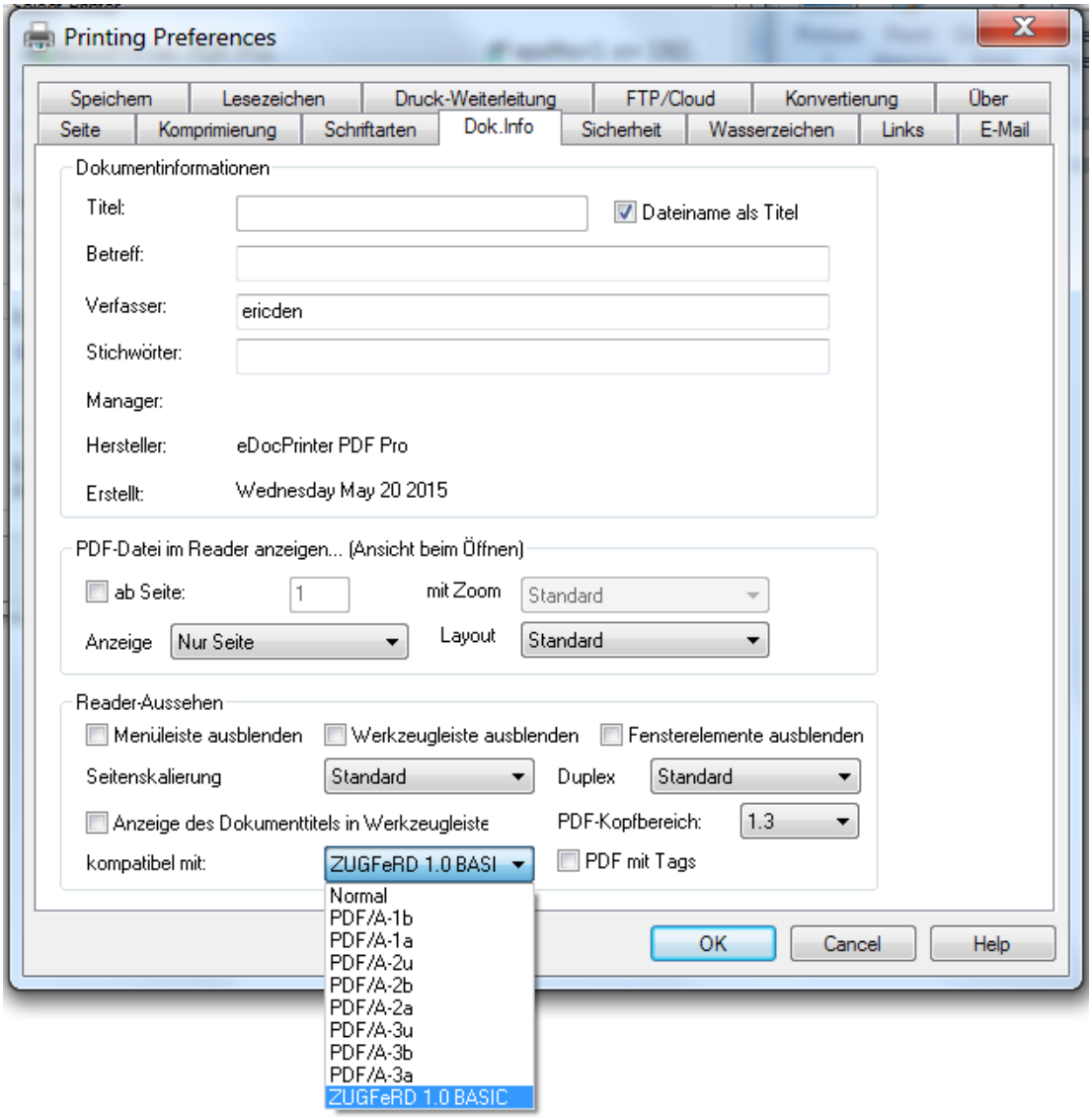
4. [addprn.exe]: Add command line arguments for adding printer instances compliant with ZUGFeRD 1.0 with/without enabling embedding commands. The syntax is [addprn az3z "ZUGFeRD 1.0 printer"] and [addprn az3u "ZUGFeRD 1.0 printer"] correspondingly.

5. [addprn.exe- since Ver 9.x]: Add command line arguments for adding printer instances compliant with ZUGFeRD 2.1 with/without enabling embedding commands. The syntax is [addprn az3z2 "ZUGFeRD 2.1 printer"] and [addprn az3u2 "ZUGFeRD 2.1 printer"] correspondingly.

6. [addprn.exe- since Ver 9.x]: Add command line arguments for adding printer instances compliant with Factur-X with/without enabling embedding commands. The syntax is [addprn az3zx "Factur-X 1.0 printer"] and [addprn az3ux "Factur-X 1.0 printer"] correspondingly.

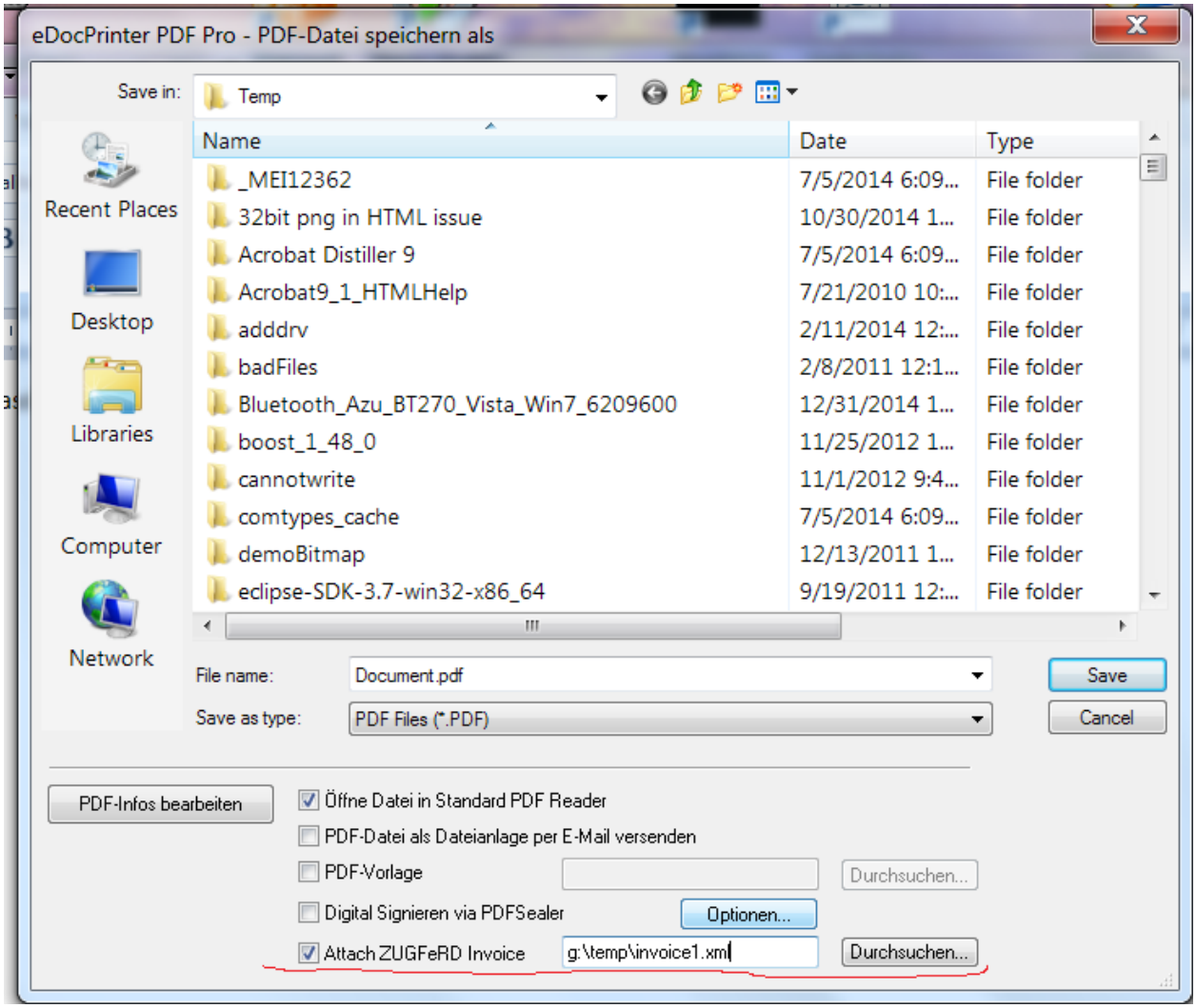
Illustrations for using ZUGFeRD mode

For creating PDF files compliant with PDF/A-3 or ZUGFeRD, please select the compliant with field properly in the Doc Info property page as illustrated below. For convenience, administrators can create extra edocprinter profiles with enabling PDF/A-3 or PDF ZUGFeRD by default from the Installer feature selection or by addprn command line utility. ZUGFeRD mode is an Ent. pack feature, which requires the Ent. pack license installed.



For ZUGFeRD invoices, users need to embed the invoice xml file. It will show the optional ZUGFeRD invoice field in Saveas dialog asd illustrated when in PDF/A-3 or ZUGFeRD compatible mode. Users can browse to add a valid structured XML file of the electronic invoice to embed into the PDF.

In addition, a new Ent. pack command %%eczf: path to invoice.xml%% is added to support directly assigning the path of ZUGFeRD invoice xml when doing silently printing without showing SaveAs dialog. For example, %%eczf: c:\invoices\ZURFeRD1.xml%%.



Step by step guides for creating a PDF/A file with embedding factur-x.xml

eDocPrinter Ent. Pack supports generating PDF/A-3 complaint PDF files. For creating PDF files compliant with PDF/A-3 or ZUGFeRD, users need to select the compliant mode PDF/A-3 or ZUGFeRD in the Doc Info property page. For convenience, administrators can create optional edocprinter profiles with enabling PDF/A-3 or ZUGFeRD by default from the Installer feature selection or by addprn command line utility. It requires the invoice xml is existing to embed as the factur-x.xml in the PDF.

Below are video links for illustrating adding extra PDF ZUGFeRD profiles and embedding the invoice xml manually or by the embedded command.

- eDocPrinter Installer - Add extra eDocPrinter instances(profiles), e.g., "PDF ZUGFeRD" profile which enables the ZUGFeRD compliant mode and detecting embedded commands.
  https://youtu.be/Mfx_wPPpeQI

- eDocPrinter ZUGFeRD - Print to create PDF files following ZUGFeRD 2.1 compliance with embedding a factur-x.xml invoice file in SaveAs dialog.
  https://youtu.be/dhlufoGgDbE

- eDocPrinter- Print to create PDF files using the command eczf1 to embed the invoice file factur-x.xml without prompting for the Save As dialog. For example:
  %%eczf1: c:\invoices\facturx-customer1%%
  https://youtu.be/LrqjDPbV3TQ

Using zx. commands to populate fields in the template invoice xml

Since Ver 10, eDocPrinter adds the support of xml template instantiation (expansion) for resolving variables and fields defined in the xml file. Basically, the xml file provided will be treated as an invoice template. During conversion, the data from zx commands will be used to populate (expand) the template to a compliant xml invoice.

The xml template is using the Mustache template system, which is a popular xml and html template format used globally. The field (variable) is defined by wrapping its name with curly braces, e.g., {{name}}. The embedded command, e.g., %%zx.name: My Product%%, provides the data to expand all root level {{name}} fields into the value My Product.

For more on Mustache template system, please refer https://mustache.github.io/

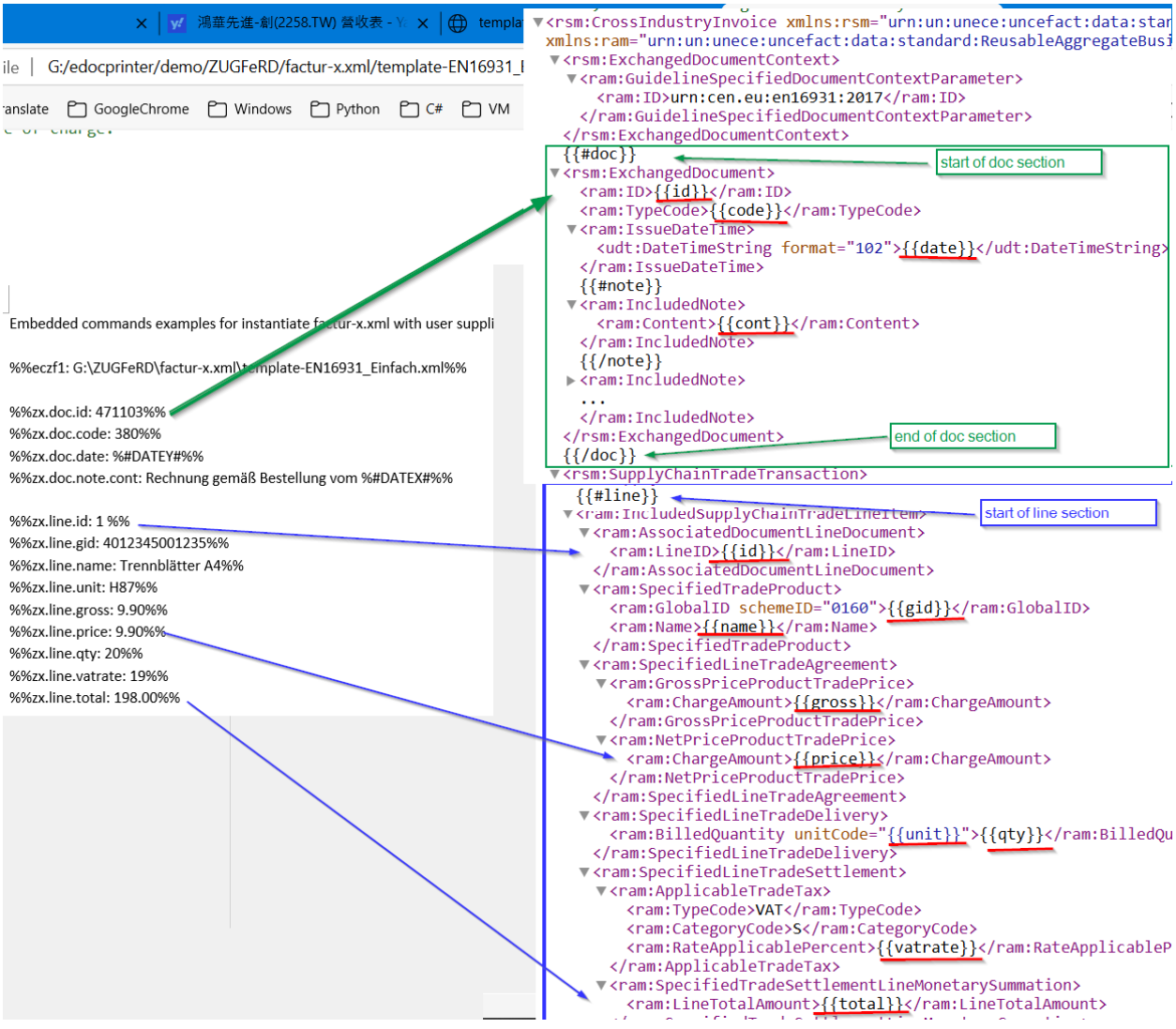
Using sections (levels) is to organize data structures and avoid field name conflicts. For example, the {{name}} in the line item section is different from the {{name}} field in the buyer section. The start and end of the section will be marked by the special character # and /. For example {{#line}} represents the start of the line section in the template. And {{/line}} section marks the end of the section. Accordingly, {{#buyer}} {{/buyer}} will wrap the whole buyer information section in the xml.

The field names are user defined in the template xml. As a result, the trailer of the command zx behind the dot will be composed of the section name and field name defined in the template xml. Hence zx commands may be template dependent. Users may also define their own field names for convenience. For example, for simple invoice containing one item from specific seller, users may design a template xml with prefixed seller information and others. Users may also use a flattened fields without levels in their own xml invoice template.

The dot in the key name of zx command is used for nested levels separation.

For example,
%%zx.doc.id: 280081%% is used to specify the id value inside the doc section in the template xml.
%%zx.line.total: 99.90%% represents the amount of the total field of the line item of the invoice.
%%zx.ahts.due: 230.48%% assigns the amount receivable field {{due}} of the section ahts in the xml.

When instantiating, these values will be populated into the template xml to replace the matched fields as illustrated below.



Using sections (levels) is to organize data structures and avoid field name conflicts. For example, the {{name}} in the line item section is different from the {{name}} field in the buyer section.

For simplifying commands, the template xml may be designed with a prefilled seller info. If necessary, users may use eczf to choose the proper xml for different seller. Of course, users may also define fields for the seller section and use zx commands to populate these fields. For example, user may define the SellerTradeParty properties as follows:

```
      <ram:SellerTradeParty>
        {{#seller}}
        <ram:ID>{{id}}</ram:ID>
        <ram:GlobalID schemeID="0088">{{gid}}</ram:GlobalID>
        <ram:Name>{{name}}</ram:Name>
        <ram:PostalTradeAddress>
          <ram:PostcodeCode>{{post}}</ram:PostcodeCode>
          <ram:LineOne>{{line1}}</ram:LineOne>
          <ram:LineTwo>{{line2}}</ram:LineTwo>
          <ram:CityName>{{city}}</ram:CityName>
          <ram:CountryID>{{country}}</ram:CountryID>
        </ram:PostalTradeAddress>
        <ram:SpecifiedTaxRegistration>
          <ram:ID schemeID="FC">{{regnum}}</ram:ID>
        </ram:SpecifiedTaxRegistration>
        <ram:SpecifiedTaxRegistration>
          <ram:ID schemeID="VA">{{vatnum}}</ram:ID>
        </ram:SpecifiedTaxRegistration>
        {{/seller}}
      </ram:SellerTradeParty>
```

Examples for using zx. commands to instantiate the template invoice xml

Please find examples below for using new commands for generating factur-x.xml from a template (The rtf document contains a command to use the template xml specified. Users may need to modify the path when testing) (The rtf document contains the commands for filling the invoice data into the template xml dynamically when printing)

These invoice xml template files are modified from the invoice data of examples in the ZUGFeRD spec.

Example 1:
EN 16931 compliance from EN16931_Einfach

The sample document using embedded commands zx:
https://edocprinter.info/download/zugferd/demo-zugcmd-EN16931_Einfach.rtf

The xml invoice template:
https://edocprinter.info/download/zugferd/template-EN16931_Einfach.xml

Example 2:
EN 16931 compliance from EN16931_Reisekostenabrechnung

The sample document using embedded commands zx:
https://edocprinter.info/download/zugferd/demo-zugcmd-EN16931_Reisekostenabrechnung.rtf

The xml invoice template:
https://edocprinter.info/download/zugferd/template-EN16931_Reisekostenabrechnung.xml

Example 3:
BASIC compliance from BASIC_Einfach

The sample document using embedded commands zx:
https://edocprinter.info/download/zugferd/demo-zugcmd-BASIC_Einfach.rtf

The xml invoice template:
https://edocprinter.info/download/zugferd/template-BASIC_Einfach.xml