

Extract invoice for ZUGFeRD and Facture-X

Let's say you received an electronic invoice in Germany or France as a PDF document with embedded XML. You like to use DynaPDF to extract the XML and we have a sample file for you, which we like to explain here. So let's take a look on the main script to extract the attachment with the XML.

First we initialize DynaPDF and have the script load the dynapdf library. Then we open the invoice PDF with [DynaPDF.OpenPDFFromContainer](#) function and then use [DynaPDF.ImportPDFFile](#) function to import the content of the PDF into memory into the working PDF.

```
# Make new PDF environments
```

```
Set Variable [ $pdf ; Value: MBS("DynaPDF.New") ]
```

```
# Load PDF from container
```

```
Set Variable [ $r ; Value: MBS("DynaPDF.OpenPDFFromContainer"; $pdf;  
ZUGFeRD Extract XML::Input PDF) ]
```

```
Set Variable [ $r ; Value: MBS("DynaPDF.ImportPDFFile"; $pdf) ]
```

Next step is to check how many attachments are in the PDF document. For a ZUGFeRD invoice, this should be one as there is usually only a XML file attached. But there could technically be more attachments like an additional PDF.

```
Set Variable [ $FileCount ; Value: MBS("DynaPDF.GetEmbeddedFileCount";  
$pdf) ]
```

We then loop over the entries counting \$index from 0 to \$FileCount-1 and query metadata for each file on the way. The file name is usually something like "zugferd-invoice.xml" or "factur-x.xml". The mime type is usually "text/xml" for the XML documents. Description text may be something like "Factur-X/ZUGFeRD-Rechnung" to inform someone looking on the attachment what it may be.

For ZUGFeRD and Factor-X, we have these file names:

- ZUGFeRD 1.x: "ZUGFeRD-invoice.xml"
- ZUGFeRD 2.x: "zugferd-invoice.xml"
- Factur-X / ZUGFeRD 2.1: "factur-x.xml"
- XRechnung: "xrechnung.xml"

```
Set Variable [ $Name ; Value: MBS( "DynaPDF.GetEmbeddedFile"; $pdf; $index;  
"Name") ]
```

```
Set Variable [ $MimeType ; Value: MBS( "DynaPDF.GetEmbeddedFile"; $pdf;  
$index; "MimeType") ]
```

```
Set Variable [ $Description ; Value: MBS( "DynaPDF.GetEmbeddedFile"; $pdf;  
$index; "Description") ]
```

Next we ask for the content of the embedded file as text with UTF-8 encoding. This gives us the XML directly, which we then can use for the XML functions. As an example we use XML.Query to just check the IncludedNote nodes and extract them as text. This gives us some textual description of the invoice for the records. Also we use DynaPDF to query the version information for the PDF Version, so we see whether it is PDF/A 3b and Facture-X:

```
Set Variable [ $Content ; Value: MBS( "DynaPDF.GetEmbeddedFile"; $pdf; $index; "Content"; "UTF8" ) ]
Set Variable [ $Content ; Value: MBS( "Text.ReplaceNewline"; $Content; 1 ) // CR for line ending ]
Set Field [ ZUGFeRD Extract XML::Output XML ; $Content ]
Set Field [ ZUGFeRD Extract XML::FileName ; $Name ]
Set Field [ ZUGFeRD Extract XML::MimeType ; $MimeType ]
Set Field [ ZUGFeRD Extract XML::Description ; $Description ]
Set Field [ ZUGFeRD Extract XML::Included Notes ; MBS("XML.Query"; $Content; "//IncludedNote"; "" ; 8+2) ]
Set Field [ ZUGFeRD Extract XML::PDF Version ; MBS( "DynaPDF.GetPDFVersionEx"; $PDF ) ]
```

Once you are done with the PDF document, please don't forget to release the DynaPDF environment. Alternatively you can use [DynaPDF.Clear](#) function to start over and then import the next PDF document and reuse the environment.

cleanup memory

```
Set Variable [ $r ; Value: MBS("DynaPDF.Release"; $pdf) ]
```

That's it. Once you have the XML, you can parse it with our XML functions and fill it in fields and records.

The updated examples will be included in next [MBS FileMaker Plugin](#) pre-release. Let us know if you like a copy in advance. Please try them and add the reading of electronic invoices for your solution.

See also [DynaPDF Licenses and ZUGFeRD invoices](#), [The new ZUGFeRD example](#) and [FileMaker with ZUGFeRD 2.0 and Factur-X](#).