

Validating schema for electronic invoices with ZUGFeRD

When receiving invoices in electronic formats, you have several tasks to implement:

1. Receiving invoice via email.

You may use our [CURL](#) functions to regularly download emails from your IMAP postbox and use our [EmailParser](#) functions to get the attachments. That may include the ZUGFeRD PDF invoice file. Or a pure XML file with invoice data.

2. Extract XML for invoice from PDF

Use [DynaPDF](#) functions (Lite license or higher) to extract XML from a PDF file using functions for embedded files. We have sample projects for this included with the plugin download.

3. Validate the XML to match the schema using the ZUGFeRD XSD files

Make sure the XML is syntactically okay, doesn't produce a parse error or a schema error.

Below we show how to do this with [MBS FileMaker Plugin](#) in FileMaker.

4. Validate whether the content of the invoice is validate for the business rules

There is a file included with ZUGFeRD download with the rules (.sch file). For example this specifies that the VAT ID starts with the country code, which is not part of the schema in step 3.

Implementing this in FileMaker can be a lot of work, but at least you can use our [XML.Query](#) function to help.

5. Read values from XML.

Several people coded the import and the script depends on your field schema in FileMaker. Check our example file and maybe use it as a start. Our [XML](#) functions can help and you may read a lot of files simply with a call to [XML.GetPathValue](#).

Validate against XML schema

Since the extended format includes the basic one, we can validate it in FileMaker directly using [MBS FileMaker Plugin](#) and the XSD files for the extended format. For that we use the [XML.Validate](#) function where we can pass the XML and the schema as text. Since the schema contains four files, we put them all in a folder and reference the folder as current directory, so the schema validator finds them. Technically you could also merge them all into one schema file.

Here is the sample script, which reads the sample invoice and the schema from the folder:

```
# you got the XML from the invoice
Set Variable [ $xml ; Value: MBS("Text.ReadTextFile"; "/Users/cs/ZUGFeRD/factor-x.xml"; "UTF-8") ]
#
# you have schema files in a folder
Set Variable [ $path ; Value: "/Users/cs/ZUGFeRD" ]
Set Variable [ $r ; Value: MBS("Process.SetCurrentDirectory"; $path) ]
#
# validate
Set Variable [ $Schema ; Value: MBS("Text.ReadTextFile"; "/Users/cs/ZUGFeRD/Factor-X_1.0.07_EXTENDED.xsd"; "UTF-8") ]
Set Variable [ $result ; Value: MBS( "XML.Validate"; $XML; $schema ) ]
Show Custom Dialog [ "Result" ; $result ]
```

Once run, the example should be "OK" if everything is fine. If you get an error, the format is like this:

```
[MBS] Validate failed with error 1871.
```

```
Error 1871 in line 95: Element
```

```
'{urn:un:unece:uncefact:data:standard:ReusableAggregateBusinessInformationEntity:100}IDd': This element is not expected. Expected is ( {urn:un:unece:uncefact:data:standard:ReusableAggregateBusinessInformationEntity:100}ID ).
```

The result should provide the error code, number and description lines. If multiple things are reported by the validator, you may get multiple lines of error messages there.

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