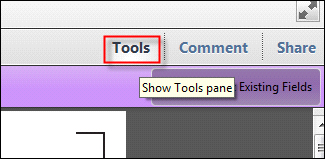


To populate PDF form from Excel file we need to create the below activities:

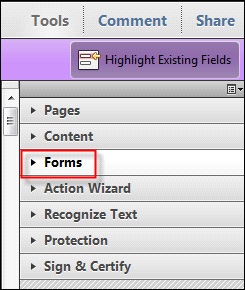
* Export the XML file from PDF Form:

Fill the fields in PDF form first for which you need to populate the data from excel file.

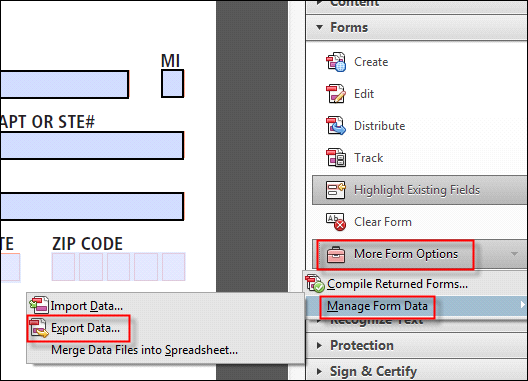
Open this PDF file in “Adobe Acrobat Professional Version”. Click on the Tool located at the right corner of Adobe Acrobat.



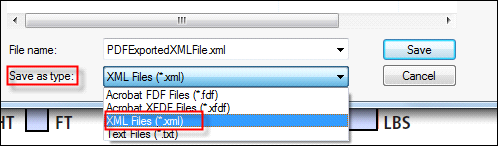
Expand the Forms.



Go to the “Export Data…” using More Form Option > Manage Form Data Export Data

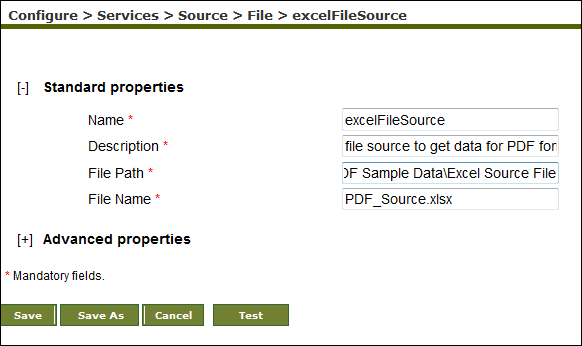


Now select “Save as type” as XML files and save exported xml file.



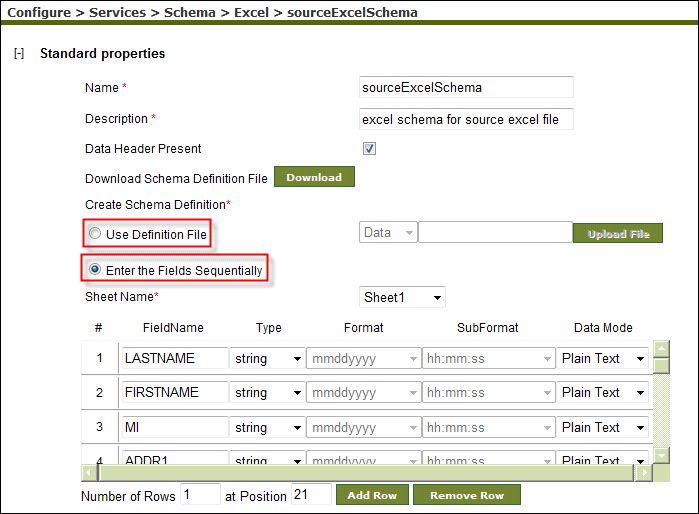
* Create a File Source activity:

Go to the link **Configure > Services > Source > File.** Create on the New link to create the new file source activity having the location for the Excel file location.



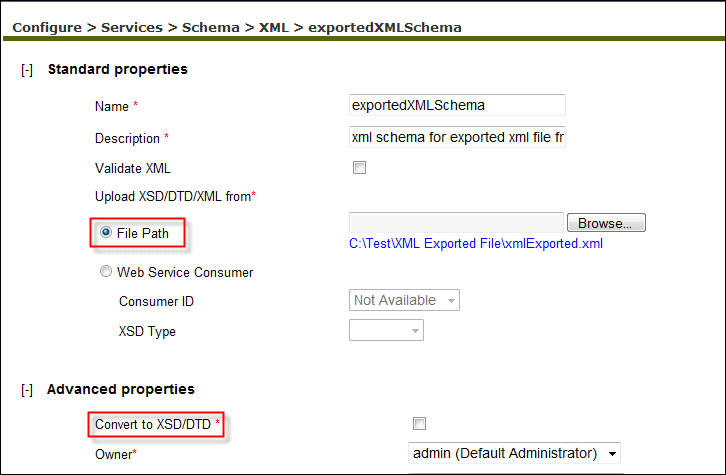
* Create Excel Schema for the source:

Go to the link **Configure > Services > Schema > Excel**. Click the New link to create new excel schema. You can create the schema manually or by “Use Definition File”.



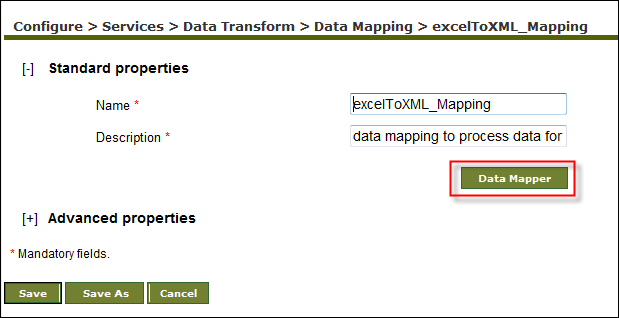
* Create XML Schema for the target:

Go to the link **Configure > Services > Schema > XML.** Click on the New link to create new XML schema. Give the File Path for the PDF exported file in the Step 1. Also unchecked the property “Convert to XSD/DTD” under Advance properties.

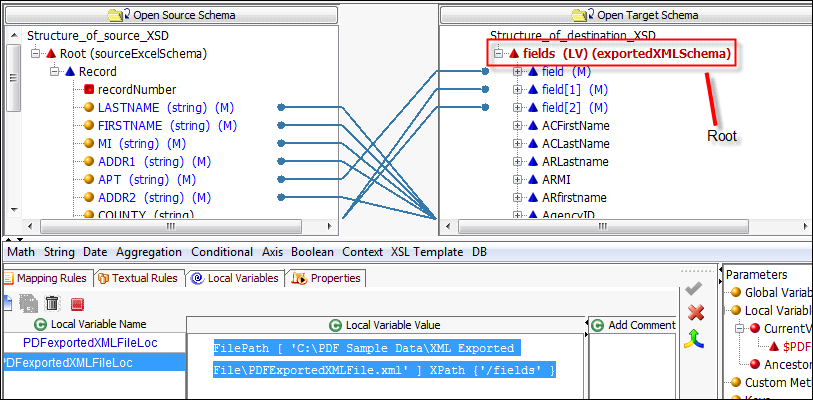


* Create Data Mapping:

Go to the link **Configure > Services > Data Transform > Data Mapping**. Click on the New link to create new data mapping. Click on the Data Mapper to open the applet. Load source and target schema.



Create a local variable at Root level named “PDFexportedXMLFileLoc” having value “FilePath [ '**C:\PDF Sample Data\XML Exported File\PDFExportedXMLFile.xml'** ] XPath {'/fields' }”.



“C:\PDF Sample Data\XML Exported File\PDFExportedXMLFile.xml” is the location for the PDF form exported XML file and “/fields” is the root element for the target XML schema.

To apply the Mapping:

Select the element. For example if we select LastName. Map this element to the LASTNAME of the Source schema as given in the mapping rule. Now map the attribute “xfdf:original” to “$PDFexportedXMLFileLoc/LastName[1]/@xfdf:original”. Here you can see “$PDFexportedXMLFileLoc” is the variable which is already created and “LastName[1]/@xfdf:original” is the XPath for the attribute for the element “LastName”. If any element doesn’t have the attribute then we need to do only mapping for the element. Please note while doing mapping please use “IIF” condition to filter out the element if the source element doesn’t have any value.

* Create Custom Plugin:

Go to the link **Configure > Services > Extensions > Custom Plugin**. Click on the New link to create new Custom Plugin.

**Note: please place the attached “ConvertXMLtoPDF.class” file to the location**

**<Adeptia Installed Directory>\ServerKernel\customClasses and restart the kernels.**

Place the below code in the custom plugin (this plugin is already built in the EnvironmentPromotion zip file that contains this example flow):

import java.io.InputStream;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;

import org.xml.sax.InputSource;

Document doc = null;

ArrayList arrayList = null;

DocumentBuilderFactory factory = null;

DocumentBuilder builder = null;

Document ret = null;

try {

java.util.Date date = new java.util.Date();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd-hhmmss");

String outputDateFormat = sdf.format(date);

InputStream inputStream = service.getSourceStream("excelToXML\_Mapping");

factory = DocumentBuilderFactory.newInstance();

builder = factory.newDocumentBuilder();

doc = builder.parse(new InputSource(inputStream));

arrayList = new ArrayList();

ConvertXMLtoPDF obj = new ConvertXMLtoPDF();

obj.getNodeDetials(doc.getDocumentElement(), arrayList);

// setFormDataIntoPDF(String pdfSourceFile, String pdfTargetFile, ArrayList arrayList)

String pdfSourceFile = context.get("PDFFormLocation");

String pdfTargetFile = context.get("generatedPDFFormLocation");

if (pdfTargetFile.contains(".")) {

pdfTargetFile = pdfTargetFile.substring(0,pdfTargetFile.indexOf(".")) + outputDateFormat + pdfTargetFile.substring(pdfTargetFile.indexOf("."),pdfTargetFile.length());

}

obj.setFormDataIntoPDF(pdfSourceFile, pdfTargetFile , arrayList );

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

if (arrayList != null) {

arrayList.clear();

}

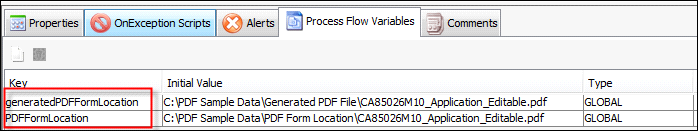
}

Here in the custom plugin we have appended the time stamp with the generated PDF file to generate unique file for each record.

* Create the Process Flow:

For the Flow design please go through attached PDF file “PopulatePDFform\_ProcessFlow.pdf”.

In the process Flow You can configure the location for the actual PDF from and the for generated PDF form.



Note: For the reference please find the attached zip file.