

Purchaseorder PDM2012 with embedded XML

M.Balster 16-11-2022

PDM2012 offers by app PDFP309XML.exe (v.15-11-2022 XML Datapoint) purchase orders in PDF with embedded XML. These PDF documents have been marked with **“PDF+XMLdata”** at the corner right below. An easy to understand XML file containing the lines of the purchaseorder with explaining XML tag in English.

The **Excel/VBA application** (download from www.PDM2012.nl) and a download from Xpdfreader PDFtotext64.exe as shown at page 3 enables extracting the XML from the PDF. Next step is to prepare the XML for order entry to your own administration or ERP system. We recommend the use of the embedded data and appreciate your response with same identifiers to enable goods received, materials charge and certificate administration and invoice handling at individual part handling. Besides, the XML is embedded as plain text between identifiers (read the VBA module1). Some easy effort at your ICT Dept. can do the same... or better! Use the form at www.pdm2012.nl for questions and more information.

Example Excel/VBA application: (in Dutch)

	K + verzendlocatie	XML verzenden van de orderentry	XML verzenden van de klantorder	
1				<p>meldingen</p> <p>PDF'en verzenden en verspreiden in kolom 5.</p> <p>verzenden XML's naar orderentry inclusief orderen en klanten, ook in kolom 6 een klant en orderen en klanten beide zijn ingesold.</p>
2	verzoeken	9003	333	
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				

Sheet 'instellingen' shows the settings for reading and writing at your location.:

	A	B
1	geldigheid locatie wordt niet getest	altijd \ op einde plaatsen
2	location is NOT validated; check your access rights	terminate path with \
3	locatie .pdf inkoopopdrachten PDM2012	C:\Demo\pdm\xntvb6\PDFInkoop\
4	verzendlocatie xml uit deze .pdf'en (orderentry)	d:\snijafdeling\xmlin\
5	verzendlocatie xml backup	d:\snijafdeling\xmlreservekopie\
6		

Line 3: locatie .pdf inkoopopdrachten PDM2012 -> source pdf to read from

Line 4: verzendlocatie xml uit deze .pdf'en(orderentry) -> destination of XML for postprocessing or orderentry

Line 5: verzendlocatie xml backup -> archive of processed XML files

Content of the XML basic layout:

All lines have been tagged in XML. Even the text lines. PDM2012 distinguishes three type of lines, to be recognized by the 'recordtype':

- RS1 articles purchased directly to order to supply JIT parts delivery management
- RS2 articles purchased with article code to supply to stock and MRP
- RS3 text lines

Your ordernumber and clientcode will be merged into the XML by the Excel/VBA. View the VBAcode to discover that also a working plan can be merged.

Below an example of the embedded XML basic layout⁽¹⁾:

Article purchased directly to order. 'UnitPrice' 14,40 is fantasy.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="true"?>
- <PPSImport xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Parts>
    - <Part Line="080530001002">
      <CustomerOrder>K09300 </CustomerOrder>
      <Recordtype>RS1</Recordtype>
      <Linenumber> 001 002</Linenumber>
      <Articlecode>* </Articlecode>
      <Description>LASERGESN.PLAAT ø140X2 + GAT 26 </Description>
      <Material>1.4307 </Material>
      <Quantity> 1 </Quantity>
      <Unit>stuks</Unit>
      <UnitPrice> 14,40</UnitPrice>
      <Deliveryweek>2022464</Deliveryweek>
      <Deliverydate>17-11-2022</Deliverydate>
      <CreatedBy>PDM2012v20221110basis</CreatedBy>
      <SupOrderNo>b101 </SupOrderNo>
      <SupCustomerNo>333</SupCustomerNo>
      <SupWorkingPlan>MyWorkPlan</SupWorkingPlan>
    </Part>
  - <Part Line="080530001003">
```

Textline:

```
- <Part Line="080530 000">
  <CustomerOrder>K09300 </CustomerOrder>
  <Recordtype>RS3</Recordtype>
  <Linenumber> </Linenumber>
  <Articlecode>textline </Articlecode>
  <Description>- REGEL 1 T M 12 VLGS TEKENING K21307-205 </Description>
  <Material> </Material>
  <Quantity> </Quantity>
  <Unit> </Unit>
  <UnitPrice> </UnitPrice>
  <Deliveryweek> </Deliveryweek>
  <Deliverydate> </Deliverydate>
  <CreatedBy>PDM2012v20221110basis</CreatedBy>
  <SupOrderNo>b101 </SupOrderNo>
  <SupCustomerNo>333</SupCustomerNo>
  <SupWorkingPlan>MyWorkPlan</SupWorkingPlan>
</Part>
```

1) per supplier a XML layout can be composed. We focus here to the Basic layout.

For developing and testing we used below PDFtoText.exe and renamed this app to pdftotext64.exe

Naam	Type	Gecomprimeerde gr...	Met wa
pdfdetach.exe	Toepassing	451 kB	Nee
pdffonts.exe	Toepassing	460 kB	Nee
pdfimages.exe	Toepassing	488 kB	Nee
pdfinfo.exe	Toepassing	464 kB	Nee
pdftohtml.exe	Toepassing	1.000 kB	Nee
pdftopng.exe	Toepassing	961 kB	Nee
pdftoppm.exe	Toepassing	917 kB	Nee
pdftops.exe	Toepassing	995 kB	Nee
<u>pdftotext.exe</u>	Toepassing	516 kB	Nee

Source:

<https://www.xpdfreader.com> > pdfto... · Vertaal deze pagina ⋮

pdftotext(1) - XpdfReader

Pdftotext converts Portable Document Format (PDF) files to plain text. **Pdftotext** reads the PDF file, PDF-file, and writes a text file, text-file.

<https://www.xpdfreader.com> · Vertaal deze pagina ⋮

XpdfReader

Xpdf is a free PDF viewer and toolkit, including a text extractor, image converter, HTML converter, and more. Most of the **tools** are available as open source ...

Programmers only: Content of Excel/VBA module 1 at 16-11-2022.

This code refers to two subs for the two buttons.

'M.Balster 2022

Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal lpszOp As String, ByVal lpszFile As String, ByVal lpszParams As String, ByVal lpszDir As String, ByVal FsShowCmd As Long) As Long

Sub BLAD2_OVERZICHT_PDF_BESTANDEN()

' selectie vermelden op blad 2

'* cellen selectief leegmaken

Blad2.Range("A2:E999").ClearContents

'* reset aantal regels op blad2

eind_blad2 = 1

'* string om FAT regel in te zetten (FAT=File Allocation Table)

Dim StrFile_str As String

'* string om bestandenpad met de Pdf bestanden te bewaren

Dim bestandenpad_str As String

'* ophalen bestandenpad uit blad3, instellingen regel 3 kolom 2

bestandenpad_str = Blad3.Cells(3, 2) + ".pdf"

'* eerste FAT regel ophalen op locatie bestandenpad met filter op .pdf

StrFile_str = Dir(bestandenpad_str)

'/1/ zolang er iets in de stringvariabele StrFile_str staat

Do While Len(StrFile_str) > 0

'* extra regel op blad2 toevoegen

eind_blad2 = eind_blad2 + 1

'* gevonden bestandsnaam toevoegen op regel eind_blad2 kolom 1

Blad2.Cells(eind_blad2, 1) = StrFile_str

'* en een volgende FAT regel laden in StrFile_str

StrFile_str = Dir

Loop 'verder naar /1/

'* aantal gevonden bestanden bewaren

Blad3.Cells(3, 3) = eind_blad2

End Sub 'lezen_xml

Sub read_xml_from_pdf_basis()

Dim bestandenpad_str As String

Dim bestandenpad_RESERVEKOPIE_str As String

Dim bestandenpad_XMLORDERINVOER_str As String

Dim This_file_str As String

Dim sfilein As String

Dim vArguments As Variant

Dim vDirectory As Variant

Dim vOperation As Variant

Dim vShow As Variant

Dim deelnummer_int%

bestandenpad_str = Blad3.Cells(3, 2)

bestandenpad_XMLORDERINVOER_str = Blad3.Cells(4, 2)

bestandenpad_RESERVEKOPIE_str = Blad3.Cells(5, 2)

deelnummer_int% = 0 ' =0 geeft elke regel van de XML geen eigen deelordernummer, =1 wel

```

'* lezen aantal regels met pdf files
eind_blad2& = Blad3.Cells(3, 3)

'* regelnummerteller per order
regelnummer_lng& = 0

'* bekijk alle regels met een pdf
For T& = 2 To eind_blad2&

    '* is deze regel met een X geselecteerd?
    If UCase(Trim(Blad2.Cells(T&, 2))) = "X" Then

        This_file_str = Blad2.Cells(T&, 1)
        This_MyOrder_replacement_str$ = Trim(Blad2.Cells(T&, 3))
        This_MyCustomer_replacement_str$ = Trim(Blad2.Cells(T&, 4))
        Blad2.Cells(T&, 5) = "" 'errorlog

        'vfilein = "pdftotext32.exe" 'voor XP
        vfilein = "pdftotext64.exe"
        '* hDot% -> waar staat de punt voor de file extensie?
        hDot% = InStr(This_file_str, ".")

        '* reset RawFilename_str$ naar een ?
        RawFilename_str$ = "?"

        '* staat er een file met een . voor de file extensie ? zo ja, dan HDot% is groter gelijk 1
        If hDot% > 1 Then
            '* bestandsnaam teruglezen vanuit Excel naar stringvariabele RawFilename_str$
            RawFilename_str$ = Mid$(This_file_str, 1, hDot% - 1)
        Else
            '* als dit niet lukt dit op het scherm melden met een MessageBox
            ' en precies met 'END' beëindigen
            MsgBox "cannot proces filename = " + This_file_str
            End
        End If ' hDot% > 1

        '* output argumenten definiëren voor PDFtoTEXT
        vArguments = "-raw " + This_file_str + " " + RawFilename_str$ + ".rawtext"
        '* path zetten
        vDirectory = bestandenpad_str
        'vOperation= nvt
        'vShow= nvt
        '* uitvoeren tekst afsplitsen uit PDF
        Dim objShell
        Set objShell = CreateObject("Shell.Application")
        Call objShell.ShellExecute(vfilein, vArguments, vDirectory, "", 1)
        Set objShell = Nothing
        Close #99
        '* raw text file als binary openen
        Open bestandenpad_str + RawFilename_str$ + ".rawtext" For Binary Shared As #99
        '* aantal chars bepalen in dit file
        hFilesize_lng& = LOF(99)
        '* stringbuffer bepalen ter grootte van file
        hFileStrinbuffer_str$ = String$(hFilesize_lng&, Chr$(32))
        '* file inhoud naar stringbuffer lezen
        Get #99, 1, hFileStrinbuffer_str$
        '* startpositie ingesloten XML bepalen - let op offset +27 en -1 (alias 28)
        hStartByte_in_Buffer_lng& = InStr(hFileStrinbuffer_str$, "PDM2012_XML_DATAPOINT_START") 'START MARKER
        '* eindpositie ingesloten XML bepalen
        hEndByte_in_Buffer_lng& = InStr(hFileStrinbuffer_str$, "PDM2012_XML_DATAPOINT_END") 'START MARKER

        '* mb13042022 1 is ook ok If hStartByte_in_Buffer_lng& <= 1 Or hEndByte_in_Buffer_lng& <= 1 Then
        If hStartByte_in_Buffer_lng& < 1 Or hEndByte_in_Buffer_lng& < 1 Then 'mb13042022
            Blad2.Cells(T&, 5) = Blad2.Cells(T&, 5) + "FATAL ERROR, no XMLdata found, NO OrderEntry file processed"
            GoTo error_no_XMLdata
        End If 'hStartByte_in_Buffer_lng& <= 1 Or hEndByte_in_Buffer_lng& <= 1

        '* bepaal buffer met XMLdata uit inkoopopdrachts
        hExtract_startbyte_lng& = hStartByte_in_Buffer_lng& + 27 '29
        hExtract_stringlength_lng& = hEndByte_in_Buffer_lng& - hExtract_startbyte_lng& - 29 - hStartByte_in_Buffer_lng& - 2

```

```
XMLbuffer_str$ = Mid$(hFileStrinbuffer_str$, hStartByte_in_Buffer_lng& + 29, hEndByte_in_Buffer_lng& - 29 - hStartByte_in_Buffer_lng& - 2) 'mb15112022 test -2
```

```
'>> programmers only *****
'* even onder water kijken hoe die buffer er uit ziet
' bij voorkeur met Notepad++ bekijken
' en zelf c:\test even aanmaken
'Close #501
'Open "c:\test\XMLbuffer.txt" For Output As #501
'Print #501, XMLbuffer_str$; "|buffereinde|"
'Close #501
'<< programmers only *****

'* eenmalige aktie om 1 diameterteken van chr$(221) te vervangen door chr$(248)
'/1/ 221
hTestChar_str$ = "Y"
While InStr(XMLbuffer_str$, hTestChar_str$) >= 1
  hTest_int% = InStr(XMLbuffer_str$, hTestChar_str$)
  If hTest_int% >= 1 Then '2nd check
    Mid$(XMLbuffer_str$, hTest_int%, 1) = Chr$(248)
    nop = 1
  End If 'hTest_int% >= 1
Wend 'InStr(XMLbuffer_str$, hTestChar_str$) >= 1

'IN DEZE BUFFER ORDERNR EN KLANTNR PLAATSEN
'This_MyOrder_replacement_str$ = Blad2.Cells(T&, 3)
'This_MyCustomer_replacement_str$ = Blad2.Cells(T&, 4)

'* mb0402022 This_MyOrder_replacement_str$
If This_MyOrder_replacement_str$ > "" Then
  '* lengte van in te voegen ordernummer
  hLen_lng& = Len(This_MyOrder_replacement_str$)

  '* lengte van string "MYORDER" die vervangen gaat wordt door echt ordernummer van de leverancier
  hLen_MYORDER_lng& = 7

  '* voor elke "MYORDER" die in deze buffer voorkomt (kunnen er dus tientallen zijn..)
  While InStr(XMLbuffer_str$, "MYORDER") >= 1

    '* regelteller verhogen. Nummer wordt alleen ingevoegd als variabele deelnummer_int%=1
    regelnummer_lng& = regelnummer_lng& + 1
    '* waar begint MYORDER?
    hCharpos_lng& = InStr(XMLbuffer_str$, "MYORDER")
    '* linkerdeel bepalen voor MYORDER
    hLeftBuffer_str$ = Mid$(XMLbuffer_str$, 1, hCharpos_lng& - 1) '-1
    '* rechterdeel bepalen na MYORDER
    hrightBuffer_str$ = Mid$(XMLbuffer_str$, hCharpos_lng& + hLen_MYORDER_lng&, Len(XMLbuffer_str$) - hCharpos_lng& - hLen_MYORDER_lng& + 1)
    '* opgevenen ordernummer invoegen en buffer weer in elkaar plakken
    regelnummer_str$ = Format$(regelnummer_lng&, "000")
    If deelnummer_int% <> 1 Then regelnummer_str$ = "" 'en weer leegmaken

    XMLbuffer_str$ = hLeftBuffer_str$ + This_MyOrder_replacement_str$ + " " + regelnummer_str$ + hrightBuffer_str$
  'mb31052022 format via regelnr_str$
  Wend 'instr(XMLbuffer_str$, "MYORDER") >=1
End If 'This_MyOrder_replacement_str$>"" then

'* mb0402022 MyCustomer_replacement_str$
If This_MyCustomer_replacement_str$ > "" Then
  hLen_lng& = Len(This_MyCustomer_replacement_str$)
  hLen_MYCUSTOMER_lng& = 10
  While InStr(XMLbuffer_str$, "MYCUSTOMER") >= 1
    'waar begint MYCUSTOMER?
    hCharpos_lng& = InStr(XMLbuffer_str$, "MYCUSTOMER")
    'linkerdeel bepalen voor MYCUSTOMER
    hLeftBuffer_str$ = Mid$(XMLbuffer_str$, 1, hCharpos_lng& - 1)
    'rechterdeel bepalen na MYCUSTOMER
    hrightBuffer_str$ = Mid$(XMLbuffer_str$, hCharpos_lng& + hLen_MYCUSTOMER_lng&, Len(XMLbuffer_str$) - hCharpos_lng& - hLen_MYCUSTOMER_lng& + 1)
    'opgegeven KLANtnummer invoegen en buffer weer in elkaar plakken
```

```

XMLbuffer_str$ = hLeftBuffer_str$ + This_MyCustomer_replacement_str$ + hrightBuffer_str$
Wend 'instr(XMLbuffer_str$, "MYCUSTOMER") >=1
End If 'This_MyCustomer_replacement_str$>"" then

```

test:

```

'Debug.Print XMLbuffer_str$
'* gewijzigde stringbuffer wegzetten als xml in reservekopie locatie
' mb31052022 RW RawFilename_str$ gewijzigd naar "MasterDataImp" + This_MyOrder_replacement_str$
Close #99
Open bestandenpad_RESERVEKOPIE_str + "MasterDataImp" + This_MyOrder_replacement_str$ + ".xml" For Binary Shared As #99
Put #99, 1, XMLbuffer_str$
Close #99

'#99 wordt geopend voor het regel voor regel lezen. de regel staat in record99_str$
Open bestandenpad_RESERVEKOPIE_str + "MasterDataImp" + This_MyOrder_replacement_str$ + ".xml" For Input As #99
'#100 is werkkopie waarin worksteps worden weggeschreven
Open bestandenpad_RESERVEKOPIE_str + "_new_MasterDataImp" + This_MyOrder_replacement_str$ + ".xml" For Output As #100
While Not EOF(99) 'zoland einde van #99 niet bereikt is steeds een regel lezen
  Line Input #99, record99_str$ 'regel lezen
  workstep_str$ = "werkvoorbereiding " 'default workstep
  hcharpos1_int% = InStr(record99_str$, "<Note>") 'kijk waar deze tag en eindtag zich bevinden
  hcharpos2_int% = InStr(record99_str$, "</Note>")
  note_str$ = ""
  '* note in variabele zetten die nu bekende kan worden
  If hcharpos1_int% > 1 And hcharpos2_int% > hcharpos1_int% Then
    note_str$ = UCase(Mid$(record99_str$, hcharpos1_int% + 6, hcharpos2_int% - hcharpos1_int% - 6 - 1))
    If InStr(note_str$, "WATER") Then workstep_str$ = workstep_str$ + "waterstraal snijden "
    If InStr(note_str$, "ZETTEN") Then workstep_str$ = workstep_str$ + "zetbank "
  End If 'hcharpos1_int% > 1 And hcharpos2_int% > hcharpos1_int% Then
  '* kijken waar de te vervangen mywordorder staat om op die plaats de workstep(s) te plaatsen
  hcharpos3_int% = InStr(record99_str$, "MyWorkOrder")
  If hcharpos3_int% >= 1 Then
    hlen_int% = Len("MyWorkOrder")
    hleft_record99$ = Mid$(record99_str$, 1, hcharpos3_int% - 1)
    hlenrecord_int% = Len(record99_str$)
    hright_record99$ = Mid$(record99_str$, hcharpos3_int% + hlen_int%, hlenrecord_int% - hcharpos3_int% - hlen_int% + 1)
    'Debug.Print record99_str$
    'Debug.Print hleft_record99$
    'Debug.Print hright_record99$
    record99_str$ = hleft_record99$ + workstep_str$ + hright_record99$
    nop = 1
  End If 'hcharpos3_int% >= 1 Then
  '* aangepaste record99_str$ naar werkbestand #100 overzetten
  Print #100, record99_str$
Wend '#99
Close #99
Close #100

'#100 terugzetten naar #99
XMLbuffer_str$ = ""
Close #100
'kanaal 100 openen for binary
Open bestandenpad_RESERVEKOPIE_str + "_new_MasterDataImp" + This_MyOrder_replacement_str$ + ".xml" For Binary Shared As
#100
'buffer gelijk aan filelengte in bytes
new_XMLbuffer_str$ = String$(LOF(100), Chr$(32))
'buffer vullen met gehele filecontent
Get #100, 1, new_XMLbuffer_str$
Close #100

'* replica eveneens in reservekopie wegzetten. Is een keuze!
Close #99
Open bestandenpad_RESERVEKOPIE_str + "MasterDataImp" + This_MyOrder_replacement_str$ + ".xml" For Binary Shared As #99
Put #99, 1, new_XMLbuffer_str$
Close #99

'* gewijzigde stringbuffer wegzetten als xml in orderinvoer locatie of err=81
If This_MyOrder_replacement_str$ > "" Then

```

```

        Close #99
        Open bestandenpad_XMLORDERINVOER_str + "CustomerOrderImp" + This_MyOrder_replacement_str$ + ".xml" For Binary
Shared As #99
        Put #99, 1, new_XMLbuffer_str$
        Close #99
        Blad2.Cells(T&, 5) = Blad2.Cells(T&, 5) + "file " + This_MyOrder_replacement_str$ + ".xml to OrderEntry"
    Else
        Blad2.Cells(T&, 5) = Blad2.Cells(T&, 5) + "err=81/no This_MyOrder_replacement_str$, no export to OrderEntry " 'errorlog"
    End If 'This_MyOrder_replacement_str$ > "" Then

        Blad2.Cells(T&, 2) = "verzonden"
error_no_XMLdata:
        End If 'UCase(Trim(Blad2.Cells(T&, 2))) = "X"
    Next T&

End Sub 'read_xml_from_pdf

```

```

'spare
Sub ShellAndWait(pathFile$)
    With CreateObject("WScript.Shell")
        .Run pathFile, 1, True
    End With
End Sub

```