

How to convert XFA form to XML in PDF in VB.NET using ByteScout PDF Extractor SDK

How to convert XFA form to XML in PDF in VB.NET

With this source code sample you may quickly learn how to convert XFA form to XML in PDF in VB.NET. What is ByteScout PDF Extractor SDK? It is the SDK that helps developers to extract data from unstructured documents, pdf, images, scanned and electronic forms. Includes AI functions like automatic table detection, automatic table extraction and restructuring, text recognition and text restoration from pdf and scanned documents. Includes PDF to CSV, PDF to XML, PDF to JSON, PDF to searchable PDF functions as well as methods for low level data extraction. It can help you to convert XFA form to XML in PDF in your VB.NET application.

This rich sample source code in VB.NET for ByteScout PDF Extractor SDK includes the number of functions and options you should do calling the API to convert XFA form to XML in PDF. Follow the instructions from the scratch to work and copy the VB.NET code. Implementing VB.NET application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

Free trial version of ByteScout PDF Extractor SDK is available for download from our website. Get it to try other source code samples for VB.NET.

VB.NET - Program.vb

```
Imports Bytescout.PDFExtractor

Class Program

    Friend Shared Sub Main(args As String())

        ' Create Bytescout.PDFExtractor.XFAFormExtractor instance
        Dim extractor As New XFAFormExtractor()
        extractor.RegistrationName = "demo"
        extractor.RegistrationKey = "demo"

        ' Load PDF document
        ' (!) We do not provide the sample document, please load your own.
        extractor.LoadDocumentFromFile("samplexfa.pdf")

        ' Enumerate XFA form content part types
        For Each contentType As XFAFormContentType In
[Enum].GetValues(GetType(XFAFormContentType))

            'Get count of content parts of specified type
            Dim partCount As Integer = extractor.GetCount(contentType)
```

```
        ' Save parts as XML files
        For i As Integer = 0 To partCount - 1
            Dim fileName As String = contentType.ToString() + i.ToString() +
".xml"
            extractor.SaveToFile(contentType, i, fileName)
            Console.WriteLine("Saved form part " + fileName)
        Next
    Next

    ' Cleanup
    extractor.Dispose()

    Console.WriteLine()
    Console.WriteLine("Press any key to continue...")
    Console.ReadLine()

End Sub

End Class
```

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout PDF Extractor SDK](#)

[Explore documentation](#)

[Visit www.ByteScout.com](#)

or

[Get Your Free API Key for www.PDF.co Web API](#)