How to convert XLS to XML and XML to XLS in VB.NET with ByteScout Spreadsheet SDK

The tutorial below will demonstrate how to convert XLS to XML and XML to XLS in VB.NET

These sample source codes on this page below are demonstrating how to convert XLS to XML and XML to XLS in VB.NET. What is ByteScout Spreadsheet SDK? It is the SDK component for writing, reading, modifying and calculating Excel and CSV spreadsheets. Can calculate and reculculate formulas with Excel installed. You may import or export data to and from CSV, XML, JSON. Supports export to databases, arrays, streams. It can help you to convert XLS to XML and XML to XLS in your VB.NET application.

The SDK samples like this one below explain how to quickly make your application do convert XLS to XML and XML to XLS in VB.NET with the help of ByteScout Spreadsheet SDK. Follow the instructions from the scratch to work and copy the VB.NET code. Code testing will allow the function to be tested and work properly with your data.

Download free trial version of ByteScout Spreadsheet SDK from our website with this and other source code samples for VB.NET.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Spreadsheet SDK

Explore API Documentation

Get Free Training for ByteScout Spreadsheet SDK

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Module1.vb

```
Imports Bytescout.Spreadsheet
Imports System.IO

Module Module1
    Sub Main()
        ' sample XLS to XML conversion
        SampleXLStoXMLConversion()
        ' sample XML to XLS conversion
        SampleXMLtoXLSConversion()
    End Sub

    ''' <summary>
    ''' shows how to convert existing XLS (Excel) _document into XML using Bytescout.Sp
    ''' </summary>
    Sub SampleXLStoXMLConversion()
        Dim document As New Spreadsheet()
document.LoadFromFile("AdvancedReport.xls")

        ' read XLS and save as XML
        Dim tools As New SimpleXMLConverter(document)
        tools.SaveXML("AdvancedReport.xml")

        document.Close()
    End Sub

    ''' <summary>
    ''' shows how to convert XML data into XLS excel using Bytescout.Spreadsheet and By
    ''' </summary>
    Sub SampleXMLtoXLSConversion()
        ' read XML and convert into XLS (Excel) and save

        Dim document As New Spreadsheet()

        Dim tools As New SimpleXMLConverter(document)

        tools.LoadXML("AdvancedReport.xml")

        ' remove output file if already exists
        If File.Exists("AdvancedReportFromXML.xls") Then
            File.Delete("AdvancedReportFromXML.xls")
        End If

        document.SaveAs("AdvancedReportFromXML.xls")

        document.Close()

        ' open in default spreadsheets viewer/editor
        Process.Start("AdvancedReportFromXML.xls")
    End Sub

End Module
```

SimpleXMLConverter.vb

```
Imports Bytescout.Spreadsheet

Public Class SimpleXMLConverter

    Dim _document As Spreadsheet

    Sub New(ByRef document As Spreadsheet)
        _document = document
    End Sub

    Sub LoadXML(ByVal path As String)
        Dim dataSet As New DataSet()
        dataSet.ReadXml(path)

        For Each table As DataTable In dataSet.Tables

            Dim worksheet As Worksheet = _document.Workbook.Worksheets.Add(table.TableN

            ' Add columns
            worksheet.Columns.Insert(0, table.Columns.Count)

            ' Add rows for data
            worksheet.Rows.Insert(0, table.Rows.Count)

            ' Fill data
            For i As Integer = 0 To table.Rows.Count - 1
                For j As Integer = 0 To table.Columns.Count - 1
                    worksheet.Cell(i + 1, j).Value = table.Rows(i)(j).ToString()
                Next
            Next
        Next
    End Sub

    Sub SaveXML(ByVal path As String)

        Dim dataSet As New DataSet()

        For i As Integer = 0 To _document.Workbook.Worksheets.Count - 1

            Dim worksheet As Worksheet = _document.Workbook.Worksheets(i)

            Dim table As DataTable = dataSet.Tables.Add(worksheet.Name)

            For column As Integer = 0 To worksheet.UsedRangeColumnMax
                table.Columns.Add(String.Format("Column_{0}", column))
            Next


            For row As Integer = 0 To worksheet.UsedRangeRowMax
                Dim data() As Object
                Array.Resize(data, worksheet.UsedRangeColumnMax + 1)
                For column As Integer = 0 To worksheet.UsedRangeColumnMax
                    data(column) = worksheet.Cell(row, column).Value
                Next

                table.Rows.Add(data)
            Next
        Next

        dataSet.WriteXml(path)
    End Sub

    Sub PrintDataSet(ByRef ds As DataSet)

        Console.WriteLine("DataSet name: {0}", ds.DataSetName)
        For Each table As DataTable In ds.Tables

            Dim rowCount As Integer = table.Rows.Count
            Dim columnCount As Integer = table.Columns.Count

            Console.WriteLine("\nTable: {0} ({1} rows)", table.TableName, rowCount)

            For Each column As DataColumn In table.Columns
                Console.Write("{0}\t", column.ColumnName)
            Next

            Console.WriteLine()

            For i As Integer = 0 To rowCount - 1
                For column As Integer = 0 To columnCount - 1
                    Console.Write("{0}\t", table.Rows(i)(column))
                Next

                Console.WriteLine()
            Next
        Next
    End Sub

End Class
```

VIDEO

https://www.youtube.com/watch?v=nm_7I0PN1TY

ON-PREMISE OFFLINE SDK