How to convert XLS to SQL server (via CSV BULK insert) in VB.NET using ByteScout Spreadsheet SDK

This code in VB.NET shows how to convert XLS to SQL server (via CSV BULK insert) with this how to tutorial

The documentation is designed to help you to implement the features on your side. ByteScout Spreadsheet SDK can convert XLS to SQL server (via CSV BULK insert). It can be used from VB.NET. ByteScout Spreadsheet SDK is the SDK that can write and read, modify and calculate Excel and CSV spreadsheets. Most popular formulas are supported. You may import or export data to and from CSV, XML, JSON as well as to and from databases, arrays.

You will save a lot of time on writing and testing code as you may just take the VB.NET code from ByteScout Spreadsheet SDK for convert XLS to SQL server (via CSV BULK insert) below and use it in your application. In order to implement the functionality, you should copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. Further enhancement of the code will make it more vigorous.

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:

Program.vb

```vbnet
Imports System.IO
Imports Bytescout.Spreadsheet
Imports System.Data.SqlClient

Class Program
    Friend Shared Sub Main(args As String())
        Try
            ' Load XLS document
            Using document As New Spreadsheet()
                document.LoadFromFile("SimpleReport.xls")
                Dim csvFile As String = Path.GetTempPath() & "SimpleReport.csv"

                ' Save the document as CSV file
                document.Workbook.Worksheets(0).SaveAsCSV(csvFile)
                document.Close()

                If File.Exists(csvFile) Then
                    ' MODIFY THE CONNECTION STRING WITH YOUR CREDENTIALS!!!
                    Dim connectionString As String = "Data Source=localhost;Initial Cat

                    Using connection As New SqlConnection(connectionString)
                        connection.Open()

                        ' Drop test database if exists
                        ExecuteQueryWithoutResult(connection, "IF DB_ID ('XlsTests') IS
                        ' Create empty database
                        ExecuteQueryWithoutResult(connection, "CREATE DATABASE XlsTests
                        ' Switch to created database
                        ExecuteQueryWithoutResult(connection, "USE XlsTests")
                        ' Create a table for CSV data
                        ExecuteQueryWithoutResult(connection, "CREATE TABLE CsvTest (N
                        ' Export CSV data from local file
                        ExecuteQueryWithoutResult(connection, "BULK INSERT CsvTest FRON

                        ' Check the data successfully exported
                        Using command As New SqlCommand("SELECT * from CsvTest", conne
                            Dim reader As SqlDataReader = command.ExecuteReader()

                            If reader IsNot Nothing Then
                                Console.WriteLine()
                                Console.WriteLine("Exported CSV data:")
                                Console.WriteLine()

                                While reader.Read()
                                    Console.WriteLine([String].Format("{0}  |  {1}", re
                                End While
                            End If
                        End Using

                        Console.WriteLine()
                        Console.WriteLine("Press any key.")
                        Console.ReadKey()
                    End Using
```

```vb
                End If
            End Using
        Catch ex As Exception
            Console.WriteLine("Error: " & ex.Message)
            Console.ReadKey()
        End Try
    End Sub

    Private Shared Sub ExecuteQueryWithoutResult(connection As SqlConnection, query As
        Using command As New SqlCommand(query, connection)
            command.ExecuteNonQuery()
        End Using
    End Sub
End Class
```

VIDEO

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

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Spreadsheet SDK Home Page
Explore ByteScout Spreadsheet SDK Documentation
Explore Samples
Sign Up for ByteScout Spreadsheet SDK Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co