

How to add columns into existing spreadsheet in VB.NET using ByteScout Spreadsheet SDK

This code in VB.NET shows how to add columns into existing spreadsheet with this how to tutorial

The code below will help you to implement an VB.NET app to add columns into existing spreadsheet. ByteScout Spreadsheet SDK is the library (SDK) that is capable of writing, reading, modifying and calculating Excel and CSV spreadsheets. Most popular formulas can be calculated and recalculated with Excel installed. You may import or export data to and from CSV, XML, JSON as well as to and from databases, arrays. It can add columns into existing spreadsheet in VB.NET.

Fast application programming interfaces of ByteScout Spreadsheet SDK for VB.NET plus the instruction and the code below will help you quickly learn how to add columns into existing spreadsheet. Just copy and paste the code into your VB.NET application's code and follow the instruction. Detailed tutorials and documentation are available along with installed ByteScout Spreadsheet SDK if you'd like to dive deeper into the topic and the details of the API.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with source code samples.

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](http://www.Bytescout.com)

Source Code Files:

Module1.vb

```
Imports Bytescout.Spreadsheet
Imports System.IO

Module Module1

    Sub Main()
        ' Open Spreadsheet
        Dim document As New Spreadsheet()
        document.LoadFromFile("AdvancedReport.xls")

        ' Get Worksheet
        Dim worksheet As Worksheet = document.Workbook.Worksheets(0)

        ' Add new Column
        worksheet.Columns.Insert(3, 1)

        ' Set values
        worksheet.Rows(0)(2).Value = "Age"

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

        ' Save document
        document.SaveAs("Output.xls")

        ' Close Spreadsheet
        document.Close()

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

End Module
```

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](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com