

How to import from list in VB.NET with ByteScout Spreadsheet SDK

This tutorial will show how to import from list in VB.NET

These sample source codes on this page below are demonstrating how to import from list in VB.NET. ByteScout Spreadsheet SDK is the SDK component for writing, reading, modifying and calculating Excel and CSV spreadsheets. Can calculate and recalculate formulas with Excel installed. You may import or export data to and from CSV, XML, JSON. Supports export to databases, arrays, streams. It can import from list 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 import from list. Follow the instructions from the scratch to work and copy the VB.NET code. 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.

Our website provides trial version of ByteScout Spreadsheet SDK for free. It also includes documentation and 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 System.IO
```

```
Module Module1
```

```
Sub Main()
```

```
    Const fileName As String = "VBNetImportFromList.xls"
```

```
    'Create a new spreadsheet
```

```
    Dim spreadsheet As Bytestout.Spreadsheet.Spreadsheet = New Bytestout.Spreadsheet
```

```
    'Get the data from the jagged array that we want to import
```

```
    Dim list As IList = GetList()
```

```
    'Import data into spreadsheet
```

```
    spreadsheet.ImportFromList(list)
```

```
    'Save the spreadsheet
```

```
    If (File.Exists(fileName)) Then
```

```
        File.Delete(fileName)
```

```
    End If
```

```
    spreadsheet.SaveAs(fileName)
```

```
    'Close spreadsheet
```

```
    spreadsheet.Close()
```

```
    'Open the spreadsheet
```

```
    Process.Start(fileName)
```

```
End Sub
```

```
''' <summary>
```

```
''' Creates a list of planets
```

```
''' </summary>
```

```
''' <returns>A list of planets</returns>
```

```
''' <remarks></remarks>
```

```
Function GetList() As IList
```

```
    Dim planets(9, 1) As String
```

```
    planets(0, 0) = "Mercury"
```

```
    planets(1, 0) = "Venus"
```

```
    planets(2, 0) = "Earth"
```

```
    planets(3, 0) = "Mars"
```

```
    planets(4, 0) = "Jupiter"
```

```
    planets(5, 0) = "Saturn"
```

```
    planets(6, 0) = "Uranus"
```

```
    planets(7, 0) = "Neptune"
```

```
    planets(8, 0) = "Pluto"
```

```
    Return planets
```

```
End Function
```

```
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