

How to import from 2d array in VB.NET with ByteScout Spreadsheet SDK

This code in VB.NET shows how to import from 2d array with this how to tutorial

Import from 2d array is easy to implement in VB.NET if you use these source codes below. 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 2d array in VB.NET.

This rich sample source code in VB.NET for ByteScout Spreadsheet SDK includes the number of functions and options you should do calling the API to import from 2d array. In your VB.NET project or application you may simply copy & paste the code and then run your app! Enjoy writing a code with ready-to-use sample VB.NET codes.

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

VB.NET - Module1.vb

```
Imports System.IO
Imports System.Diagnostics
Imports ByteScout.Spreadsheet

Module Module1

    Sub Main()

        Const fileName As String = "VBNetImportFrom2DArray.xls"

        'Create a new spreadsheet
        Dim spreadsheet As ByteScout.Spreadsheet.Spreadsheet = New
        ByteScout.Spreadsheet.Spreadsheet

        'Get the data from the 2D array that we want to import
        Dim stockPrices(,) As String = Get2DArray()

        'Import data into spreadsheet
        spreadsheet.ImportFrom2DArray(stockPrices)

        'Save the spreadsheet
        If (File.Exists(fileName)) Then
            File.Delete(fileName)
        End If

    End Sub

End Module
```

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

```
'Close spreadsheet  
spreadsheet.Close()
```

```
'Open the spreadsheet  
Process.Start(fileName)
```

```
End Sub
```

```
'''
```

```
''' Creates a 2D array of stock prices  
'''
```

```
''' A 2D array of stock prices  
Function Get2DArray() As String(,)
```

```
    Dim stockPrices(10, 2) As String
```

```
    stockPrices(0, 0) = "AAPL "  
    stockPrices(0, 1) = "24,89"
```

```
    stockPrices(1, 0) = "AMZN "  
    stockPrices(1, 1) = "27,95"
```

```
    stockPrices(2, 0) = "DELL "  
    stockPrices(2, 1) = "12,68"
```

```
    stockPrices(3, 0) = "EBAY "  
    stockPrices(3, 1) = "57,27"
```

```
    stockPrices(4, 0) = "GOOG "  
    stockPrices(4, 1) = "28,98"
```

```
    stockPrices(5, 0) = "IBM  "  
    stockPrices(5, 1) = "24,00"
```

```
    stockPrices(6, 0) = "INTC "  
    stockPrices(6, 1) = "12,24"
```

```
    stockPrices(7, 0) = "MSFT "  
    stockPrices(7, 1) = "37,85"
```

```
    stockPrices(8, 0) = "SNE  "  
    stockPrices(8, 1) = "39,91"
```

```
    stockPrices(9, 0) = "YH00 "  
    stockPrices(9, 1) = "78,72"
```

```
    Return stockPrices
```

```
End Function
```

```
End Module
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Spreadsheet SDK](#)

[Explore documentation](#)

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

or

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