

# How to add combination chart in VB.NET and ByteScout Spreadsheet SDK

## Tutorial on how to add combination chart in VB.NET

We made thousands of pre-made source code pieces for easy implementation in your own programming projects. ByteScout Spreadsheet SDK can add combination chart. It can be used from VB.NET. 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.

The SDK samples like this one below explain how to quickly make your application do add combination chart in VB.NET with the help of ByteScout Spreadsheet SDK. Follow the instructions from the scratch to work and copy the VB.NET code. Enjoy writing a code with ready-to-use sample VB.NET codes.

Our website provides trial version of ByteScout Spreadsheet SDK for free. It also includes documentation and source code samples.

VB.NET - Module1.vb

```
Imports Bytescout.Spreadsheet
Imports Bytescout.Spreadsheet.Charts

Module Module1

    Sub Main()

        ' Create new Spreadsheet object
        Dim spreadsheet As New Spreadsheet()
        spreadsheet.RegistrationName = "demo"
        spreadsheet.RegistrationKey = "demo"

        ' Add new worksheet
        Dim sheet As Worksheet = spreadsheet.Workbook.Worksheets.Add("Sample")

        ' Add few random numbers
        Dim length As Integer = 10
        Dim rnd As New Random()
        For i As Integer = 0 To length - 1
            sheet.Cell(i, 0).Value = rnd.[Next](10)
            sheet.Cell(i, 1).Value = rnd.[Next](10)
        Next

        ' Add charts to worksheet
        Dim columnChart As Chart = sheet.Charts.AddChart(5, 4,
```

```
ChartType.ColumnClustered)
    columnChart.SeriesCollection.Add(New Series(sheet.Range(0, 0, 9, 0)))
    Dim series As New Series(sheet.Range(0, 1, 9, 1))
    series.ChartType = ChartType.Line
    columnChart.SeriesCollection.Add(series)

    ' Save it as XLS
    spreadsheet.SaveAs("Output.xls")

    ' Close the document
    spreadsheet.Close()

    ' Cleanup
    spreadsheet.Dispose()

    ' Open generated XLS file in default associated application
    Process.Start("Output.xls")

End Sub

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](#)

or

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