

# How to add stock chart in C# and ByteScout Spreadsheet SDK

The tutorial below will demonstrate how to add stock chart in C#

These source code samples are listed and grouped by their programming language and functions they use. What is ByteScout Spreadsheet SDK? It 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. It can help you to add stock chart in your C# application.

You will save a lot of time on writing and testing code as you may just take the C# code from ByteScout Spreadsheet SDK for add stock chart below and use it in your application. Just copy and paste the code into your C# application's code and follow the instruction. Further enhancement of the code will make it more vigorous.

Free trial version of ByteScout Spreadsheet SDK is available for download from our website. Get it to try other source code samples for C#.

C# - Program.cs

```
using System.Diagnostics;
using Bytescout.Spreadsheet;
using Bytescout.Spreadsheet.Charts;

namespace CSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create new Spreadsheet object
            Spreadsheet spreadsheet = new Spreadsheet();
            spreadsheet.RegistrationName = "demo";
            spreadsheet.RegistrationKey = "demo";

            // Add new worksheet
            Worksheet sheet = spreadsheet.Workbook.Worksheets.Add("Sample");

            // Add few random numbers
            for (int i = 1; i < 6; i++)
            {
                sheet[i, 0].Value = 39082 + i;
                sheet[i, 0].ValueDataTypeByNumberFormatString =
                    Bytescout.Spreadsheet.Constants.NumberFormatType.DateTime;
            }
        }
    }
}
```

```

        sheet[1, 1].Value = 41301; sheet[1, 2].Value = 24.3; sheet[1, 3].Value =
27.2; sheet[1, 4].Value = 23.49; sheet[1, 5].Value = 25.45;
        sheet[2, 1].Value = 35203; sheet[2, 2].Value = 25.4; sheet[2, 3].Value =
25.03; sheet[2, 4].Value = 19.55; sheet[2, 5].Value = 23.05;
        sheet[3, 1].Value = 27908; sheet[3, 2].Value = 23; sheet[3, 3].Value =
19.05; sheet[3, 4].Value = 15.12; sheet[3, 5].Value = 17.32;
        sheet[4, 1].Value = 29567; sheet[4, 2].Value = 17.3; sheet[4, 3].Value =
20.33; sheet[4, 4].Value = 17.84; sheet[4, 5].Value = 20.45;
        sheet[5, 1].Value = 25895; sheet[5, 2].Value = 20.4; sheet[5, 3].Value =
18.56; sheet[5, 4].Value = 16.33; sheet[5, 5].Value = 17.35;

        // Add charts to worksheet
        Chart stockChart = sheet.Charts.AddChartAndFitInto(7, 1, 26, 9,
ChartType.StockHLC);
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));

        stockChart = sheet.Charts.AddChartAndFitInto(7, 10, 26, 18,
ChartType.StockOHLC);
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 2, 5, 2)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));

        stockChart = sheet.Charts.AddChartAndFitInto(28, 1, 46, 9,
ChartType.StockVHLC);
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 1, 5, 1)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));

        stockChart = sheet.Charts.AddChartAndFitInto(28, 10, 46, 18,
ChartType.StockVOHLC);
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 1, 5, 1)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 2, 5, 2)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 3, 5, 3)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 4, 5, 4)));
        stockChart.SeriesCollection.Add(new Series(sheet.Range(1, 5, 5, 5)));

        // 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");
    }
}
}

```

---

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 \[www.PDF.co\]\(http://www.PDF.co\) Web API](#)