fonts available for cells in C# with ByteScout Spreadsheet SDK

Make fonts available for cells in C#

:

Tutorial on how to do fonts available for cells in C#

On this page you will learn from code samples for programming in C#. Fonts available for cells in C# can be implemented with ByteScout Spreadsheet SDK. 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 reculculated with Excel installed. You may import or export data to and from CSV, XML, JSON as well as to and from databases, arrays.

C# code snippet like this for ByteScout Spreadsheet SDK works best when you need to quickly implement fonts available for cells in your C# application. Follow the instruction from the scratch to work and copy and paste code for C# into your editor. C# application implementation typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

ByteScout Spreadsheet SDK free trial version is available for download from our website. Free trial also includes programming tutorials along with source code samples.

C# - Program.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Diagnostics;
using Bytescout.Spreadsheet;
using System.IO;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create new Spreadsheet
            Spreadsheet document = new Spreadsheet();

            // Add new worksheet
            Worksheet worksheet = document.Workbook.Worksheets.Add("HelloWorld");

            // Create array with font names
```

```
            string[] fontNames = new string[] {
                                                "Helvetica",
                                                "Times New Roman",
                                                "Verdana",
                                                "Times New Roman"
            };

            // Use all fonts in fontsNames array
            for (int i = 0; i < fontNames.Length; i++)
            {
                // Set font size based on loop counter
                float fontSize = 10 + i * 3;

                //Set cell font type and font size
                worksheet.Cell(i, 0).Font = new System.Drawing.Font(fontNames[i],
fontSize);

                // Set cell value
                worksheet.Cell(i, 0).Value = fontNames[i];
            }

            // delete output file if exists already
            if (File.Exists("Fonts.xls")){
                File.Delete("Fonts.xls");
            }

            // Save document
            document.SaveAs("Fonts.xls");

            // Close document
            document.Close();

            // open generated XLS document in default program
            Process.Start("Fonts.xls");
        }
    }
}
```
