

fonts available for cells in VB.NET using ByteScout Spreadsheet SDK

fonts available for cells in VB.NET

Writing of the code to fonts available for cells in VB.NET can be done by developers of any level using ByteScout Spreadsheet SDK. ByteScout Spreadsheet SDK helps with fonts available for cells in VB.NET. ByteScout Spreadsheet SDK is the SDK to create, read, modify and calculate spreadsheets. Formula calculations are supported, import and export to and from JSON, CSV, XML, databases, arrays.

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 implement fonts available for cells. Follow the instruction from the scratch to work and copy and paste code for VB.NET into your editor. You can use these VB.NET sample examples in one or many applications.

ByteScout Spreadsheet SDK is available as free trial. You may get it from our website along with all other source code samples for VB.NET applications.

VB.NET - Module1.vb

```
Imports Bytescout.Spreadsheet
Imports System.IO

Module Module1

    Sub Main()
        ' Create new Spreadsheet
        Dim document As New Spreadsheet()

        ' Add new worksheet
        Dim worksheet As Worksheet = document.Workbook.Worksheets.Add("HelloWorld")

        ' Create array with font names
        Dim fontNames() As String = {"Helvetica", "Times New Roman", "Verdana",
"Times New Roman"}

        ' Use all fonts in fontNames array
        For i As Integer = 0 To fontNames.Length - 1
            ' Set font size based on loop counter
            Dim fontSize As Double = 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)
    Next

    ' remove output file if already exists
    If File.Exists("Output.xls") Then
        File.Delete("Output.xls")
    End If

    ' Save document
    document.SaveAs("Output.xls")

    ' Close Spreadsheet
    document.Close()

    ' open in default spreadsheets viewer/editor
    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](#)