

aztec barcode with binary data in VB.NET and ByteScout Barcode SDK

Write code in VB.NET to make aztec barcode with binary data with this How-To tutorial

Today we will explain the steps and algorithm of implementing aztec barcode with binary data and how to make it work in your application. ByteScout Barcode SDK was made to help with aztec barcode with binary data in VB.NET. ByteScout Barcode SDK is the robust library (Software Development Kit) that is designed for automatic generation of high-quality barcodes for printing, electronic documents and pdf. All popular barcode types are supported from Code 39 and Code 129 to QR Code, UPC, GS1, GS-128, Datamatrix, PDF417, Maxicode and many others. Provides support for full customization of fonts, colors, output and printing sizes. Special tools are included to verify output quality and printing quality. Can add generated barcode into new or existing documents, images and PDF.

VB.NET, code samples for VB.NET, developers help to speed up the application development and writing a code when using ByteScout Barcode SDK. Follow the instruction from the scratch to work and copy and paste code for VB.NET into your editor. Test VB.NET sample code examples whether they respond your needs and requirements for the project.

Visit our website provides for free trial version of ByteScout Barcode SDK. Free trial includes lots of source code samples to help you with your VB.NET project.

VB.NET - Program.vb

```
Imports System.Diagnostics
Imports System.Text
Imports Bytescout.BarCode

Class Program
    Friend Shared Sub Main(args As String())
        ' Create Barcode component instance
        Using barcode As New Barcode()
            barcode.RegistrationName = "demo"
            barcode.RegistrationKey = "demo"

            ' Set barcode type
            barcode.Symbology = SymbologyType.Aztec
            ' Force binary mode
            barcode.Options.AztecCompactionMode =
AztecCompactionMode.Binary

            ' Sample byte array to use as value
            Dim byteArray As Byte() = New Byte() {0, 10, 11, 12, 13, 14,
-
                15, &Hff}

            ' Set value by converting byte array to string
```

```
        barcode.Value = Encoding.[Default].GetString(byteArray)

        ' Save barcode image
        barcode.SaveImage("result.png")

        ' Open output image in default associated application
        Process.Start("result.png")
    End Using
End Sub
End Class
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Barcode SDK](#)

[Explore documentation](#)

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

or

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