

How to save barcode to PNG images in Visual Basic 6 using ByteScout Barcode SDK

The tutorial below will demonstrate how to save barcode to PNG images in Visual Basic 6

Save barcode to PNG images is easy to implement in Visual Basic 6 if you use these source codes below. What is ByteScout Barcode SDK? It is the robust SDK that generates high quality barcode images and pdf. Can generate all popular types of barcodes from QR Code, Code 39, Code 128, UPC, GS1, GS-128, PDF417, Datamatrix to more exotic barcode types. Fully customizable fonts, colors, print sizes. Includes special functions to ensure output quality, and tools for adding barcodes to new or existing pdf files and images. It can help you to save barcode to PNG images in your Visual Basic 6 application.

You will save a lot of time on writing and testing code as you may just take the Visual Basic 6 code from ByteScout Barcode SDK for save barcode to PNG images below and use it in your application. This Visual Basic 6 sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Implementing Visual Basic 6 application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

Trial version of ByteScout Barcode SDK is available for free. Source code samples are included to help you with your Visual Basic 6 app.

Visual Basic 6 - Form1.frm

```
VERSION 5.00
Begin VB.Form Form1
    Caption           =   "Form1"
    ClientHeight     =   2400
    ClientLeft       =   45
    ClientTop        =   435
    ClientWidth      =   3750
    LinkTopic        =   "Form1"
    ScaleHeight      =   2400
    ScaleWidth       =   3750
    StartupPosition  =   3   'Windows Default
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub Form_Load()

    Set bc = CreateObject("Bytescout.BarCode.Barcode")
```

```

' display information about Code39 symbology
MsgBox "Encoding '012345' using Code39 symbology" & vbCrLf &
bc.GetValueRestrictions(0) ' 0 = Code39 symbology

' set symbology type
bc.Symbology = 1 ' 1 = Code39

' set value to encode
bc.Value = "012345"

MsgBox "Saving Code39 barcode to 'Code39.png'"

bc.SaveImage "Code39.png" ' change to "c:\Code39.png" if you can not locate the
file. The default directory is the folder where Project1.exe is located

MsgBox "Encoding '012345' using Aztec symbology"

' set symbology type
bc.Symbology = 17 ' 17 = Aztec
' set value to encode
bc.Value = "012345"

' display information about Aztec symbology
MsgBox "Encoding '012345' using Aztec symbology" & vbCrLf &
bc.GetValueRestrictions(17) ' 17 = Code39 symbology

bc.SaveImage "Aztec.png" ' change to "c:\Aztec.png" if you can not locate the file.
The default directory is the folder where Project1.exe is located

Set bc = Nothing

End Sub

```

Visual Basic 6 - Project1.vbp

```

Type=Exe
Reference=*G{00020430-0000-0000-C000-
000000000046}#2.0#0#C:\WINXP\system32\stdole2.tlb#OLE Automation
Form=Form1.frm
Startup="Form1"
ExeName32="Project1.exe"
Command32=""
Name="Project1"
HelpContextID="0"
CompatibleMode="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0

```

```
ServerSupportFiles=0
VersionCompanyName="000"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
F1PointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=1

[MS Transaction Server]
AutoRefresh=1
```

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](#)

or

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