

## WPF barcode control for desktop in C# with ByteScout Barcode SDK

Tutorial: how to do WPF barcode control for desktop in C#

ByteScout tutorials explain the material for programmers who use C#. WPF barcode control for desktop in C# can be implemented with ByteScout Barcode SDK. 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.

You will save a lot of time on writing and testing code as you may just take the code below and use it in your application. In order to implement this functionality, you should copy and paste code below into your app using code editor. Then compile and run your application. Test C# sample code examples whether they respond your needs and requirements for the project.

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

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Barcode SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Barcode SDK](#)

[Get Free API key for Web API](#)

[visit www.Bytescout.com](http://www.Bytescout.com)

Source Code Files:

## App.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Windows;

namespace Bytescout.BarCode.WPFDemo
{
    /// <summary>
    /// Interaction logic for App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

## MainWindow.xaml.cs

```
using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media.Imaging;

namespace Bytescout.BarCode.WPFDemo
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        #region Constants

        private const int BarHeight = 50;
        private const int PdfBarHeight = 6;

        #endregion

        #region Constructor
        /// <summary>
        /// Initializes a new instance of the <see cref="MainWindow"/> class.
        /// </summary>
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}
```

```

}
#endregion

#region Controls event handlers
/// <summary>
/// Handles the SelectionChanged event of the cmbSymbologyType control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.Controls.SelectionChangedEventArgs"> instance of
private void cmbSymbologyType_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    UpdateBarcode();
}

/// <summary>
/// Handles the Click event of the btnGenerate control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.RoutedEventArgs"/> instance of
private void btnGenerate_Click(object sender, RoutedEventArgs e)
{
    UpdateBarcode();
}

/// <summary>
/// Handles the Click event of the btnSaveToFile control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.RoutedEventArgs"/> instance of
private void btnSaveToFile_Click(object sender, RoutedEventArgs e)
{
    SaveToFile();
}

#endregion

#region Menu items event handlers
/// <summary>
/// Handles the Click event of the mnuSaveToFile control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.RoutedEventArgs"/> instance of
private void mnuSaveToFile_Click(object sender, RoutedEventArgs e)
{
    SaveToFile();
}

/// <summary>
/// Handles the Click event of the mnuExit control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.RoutedEventArgs"/> instance of
private void mnuExit_Click(object sender, RoutedEventArgs e)
{
    Close();
}

/// <summary>
/// Handles the Click event of the mnuCopy control.
/// </summary>

```

```

/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.RoutedEventArgs"/> instance of
private void mnuCopy_Click(object sender, RoutedEventArgs e)
{
    BitmapSource barcode = ctrlBarcodeControl.GetImage();
    Clipboard.SetImage(barcode);
    barcode = null;
}

/// <summary>
/// Handles the Click event of the mnuBarcodeHome control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.RoutedEventArgs"/> instance of
private void mnuBarcodeHome_Click(object sender, RoutedEventArgs e)
{
    System.Diagnostics.Process.Start("http://bytescout.com/bytescoutbarcodesdk
}

/// <summary>
/// Handles the Click event of the mnuHelp control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.RoutedEventArgs"/> instance of
private void mnuHelp_Click(object sender, RoutedEventArgs e)
{
}

}

/// <summary>
/// Handles the TextChanged event of the txtValueToEncode control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.Controls.TextChangedEventArgs"/>
private void txtValueToEncode_TextChanged(object sender, TextChangedEventArgs e)
{
    UpdateBarcode();
}

/// <summary>
/// Handles the TextChanged event of the txtSupplementalValue control.
/// </summary>
/// <param name="sender">The source of the event.</param>
/// <param name="e">The <see cref="System.Windows.Controls.TextChangedEventArgs"/>
private void txtSupplementalValue_TextChanged(object sender, TextChangedEventArgs e)
{
    UpdateBarcode();
}
}
#endregion

#region Private implementation

public object[] GetObjectsFromEnum()
{
    object[] objArray = new object[Enum.GetValues(typeof(SymbologyType)).Length];
    for (int i = 0; i < objArray.Length; i++)
    {
        objArray[i] = ((SymbologyType)Enum.GetValues(typeof(SymbologyType)).Get
    }
    return objArray;
}

```

```

}

private void UpdateBarcode()
{
    SymbologyType symbology = (SymbologyType)Enum.GetValues(typeof(SymbologyType));
    txtSymbologyDescription.Text = ctrlBarcodeControl.GetValueRestrictions(symbology);

    try
    {
        if (symbology == SymbologyType.EAN13 || symbology == SymbologyType.ISBT)
        {
            txtSupplementalValue.IsEnabled = true;
            lblSupplementalValue.IsEnabled = true;
            txtSymbologyDescription.Text += " " + ctrlBarcodeControl.GetSupplementalValue();
        }
        else
        {
            txtSupplementalValue.IsEnabled = false;
            lblSupplementalValue.IsEnabled = false;
        }

        lblErrorMessage.Content = "";
        ctrlBarcodeControl.RegistrationKey = "XXXXXXXXXXXXXXXXXXXX";
        ctrlBarcodeControl.RegistrationName = "YYYYYYYYYYYYYYYYYYYY";
        ctrlBarcodeControl.Symbology = symbology;
        ctrlBarcodeControl.SupplementValue = txtSupplementalValue.Text;
        ctrlBarcodeControl.Value = txtValueToEncode.Text;
        ctrlBarcodeControl.DrawCaptionFor2DBarcodes = chkDrawCaptionFor2D.IsChecked;
        ctrlBarcodeControl.AutoFitToControlSize = chkAutoFitToContainer.IsChecked;
        ctrlBarcodeControl.Caption = "";

        if (symbology == SymbologyType.PDF417 || symbology == SymbologyType.PDF417 ||
            symbology == SymbologyType.MacroPDF417 || symbology == SymbologyType.MacroPDF417 ||
            symbology == SymbologyType.GS1_DataMatrix)
        {
            ctrlBarcodeControl.BarHeight = PdfBarHeight;
        }
        else if (symbology == SymbologyType.MicroPDF417)
        {
            ctrlBarcodeControl.BarHeight = PdfBarHeight / 2;
        }
        else
        {
            ctrlBarcodeControl.BarHeight = BarHeight;
        }
    }
    catch (Exception)
    {
        lblErrorMessage.Content = "Value is invalid for current symbology. Please try another value."
    }
}

private void SaveToFile()
{
    Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();
    dlg.Filter = "PNG Image|*.png|TIFF Image|*.tif;*.tiff|JPEG image|*.jpg;*.jpeg";
    dlg.ValidateNames = true;
    dlg.FilterIndex = 1;
    dlg.OverwritePrompt = true;
    dlg.CheckPathExists = true;
}

```

```

        dlg.AddExtension = true;

        bool? result = dlg.ShowDialog(this);
        if (result.HasValue && result.Value)
        {
            try
            {
                if (System.IO.Path.GetExtension(dlg.FileName).ToLowerInvariant() == ".emf")
                    throw new BarcodeException("Saving as EMF is disabled.\nYou should save as a different format.");

                if (chkCutUnusedSpace.IsChecked.Value)
                {
                    bool cut = ctrlBarcodeControl.CutUnusedSpace;
                    ctrlBarcodeControl.CutUnusedSpace = true;
                    ctrlBarcodeControl.SaveImage(dlg.FileName);
                    ctrlBarcodeControl.CutUnusedSpace = cut;
                }
                else
                {
                    ctrlBarcodeControl.SaveImage(dlg.FileName);
                }
            }
            catch (System.Exception e)
            {
                MessageBox.Show(e.Message);
            }
        }
    }

#endregion

#region Main window event handlers
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    foreach (object o in GetObjectsFromEnum())
    {
        this.cboSymbologyType.Items.Add(o);
    }
    this.cboSymbologyType.SelectedIndex = 0;
}
#endregion
}
}
}

```

<https://www.youtube.com/watch?v=REnj3A-oSPI>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Barcode SDK Home Page](#)  
[Explore ByteScout Barcode SDK Documentation](#)  
[Explore Samples](#)  
[Sign Up for ByteScout Barcode SDK Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)  
[Explore Web API Docs](#)  
[Explore Web API Samples](#)

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

[visit www.PDF.co](http://www.PDF.co)

[www.bytescout.com](http://www.bytescout.com)