
ComponentOne

BarCode for WinForms

ComponentOne, a division of GrapeCity

201 South Highland Avenue, Third Floor
Pittsburgh, PA 15206 USA

Website: <http://www.componentone.com>

Sales: sales@componentone.com

Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

Trademarks

The ComponentOne product name is a trademark and ComponentOne is a registered trademark of GrapeCity, Inc. All other trademarks used herein are the properties of their respective owners.

Warranty

ComponentOne warrants that the media on which the software is delivered is free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective media to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for the defective media by sending it and a check for \$25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original media on which the software is delivered is set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. ComponentOne is not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

Copying and Distribution

While you are welcome to make backup copies of the software for your own use and protection, you are not permitted to make copies for the use of anyone else. We put a lot of time and effort into creating this product, and we appreciate your support in seeing that it is used by licensed users only.

Table of Contents

BarCode for WinForms Overview	2
Help with WinForms Edition	2
Key Features	3-4
BarCode for WinForms Quick Start	5
Step 1 of 3: Setting Up the Form	5-6
Step 2 of 3: Adding Code to the Project	6-9
Step 3 of 3: Running the Project	9-10
Using BarCode for WinForms	11
Supported Encodings	11
Customizing the C1BarCode Control	11-12
Using the C1BarCode Control in a Document	12-13
Using C1QRCode	14
How to Obtain an Image from C1QRCode	14-15
C1QRCode Encoding Limits	15
Improving the C1QRCode Image Resolution	15-16
BarCode for WinForms Samples	17

BarCode for WinForms Overview

Add barcode images to grid cells, Web pages, or regular .NET PrintDocument objects with **BarCode for WinForms**.

Unlike barcode fonts, **BarCode for WinForms** automatically adds any necessary control symbols and checksums to the value being encoded, depending on the encoding being used, to eliminate reader errors. You can also deploy **BarCode for WinForms** with your applications like any regular assembly. Because it is a royalty-free DLL, you do not have to worry about installing barcode fonts on the client-side and making sure they are royalty free.

And **BarCode for WinForms** is so easy to use – just add the control to your form, set the encoding type, and you're done!

Getting Started

To get started, review the following topics:

- [Key Features](#)
- [Quick Start](#)
- [Samples](#)

Help with WinForms Edition

Getting Started

For information on installing **ComponentOne Studio WinForms Edition**, licensing, technical support, namespaces and creating a project with the control, please visit [Getting Started with WinForms Edition](#).

Key Features

The **C1Barcode** code was originally developed for use in the **C1Report** component and has been a part of **C1Report** since build 151, released in May 2004.

We feel that **BarCode for WinForms** is a valuable addition to the ComponentOne Studios. We decided to package it as a stand-alone component because it can be useful in many applications besides reports. For example, you can use C1Barcode to add barcode images to grid cells, to Web pages, or to regular .NET PrintDocument objects. The image below shows a **C1Report** listing products along with the C1Barcode for each:

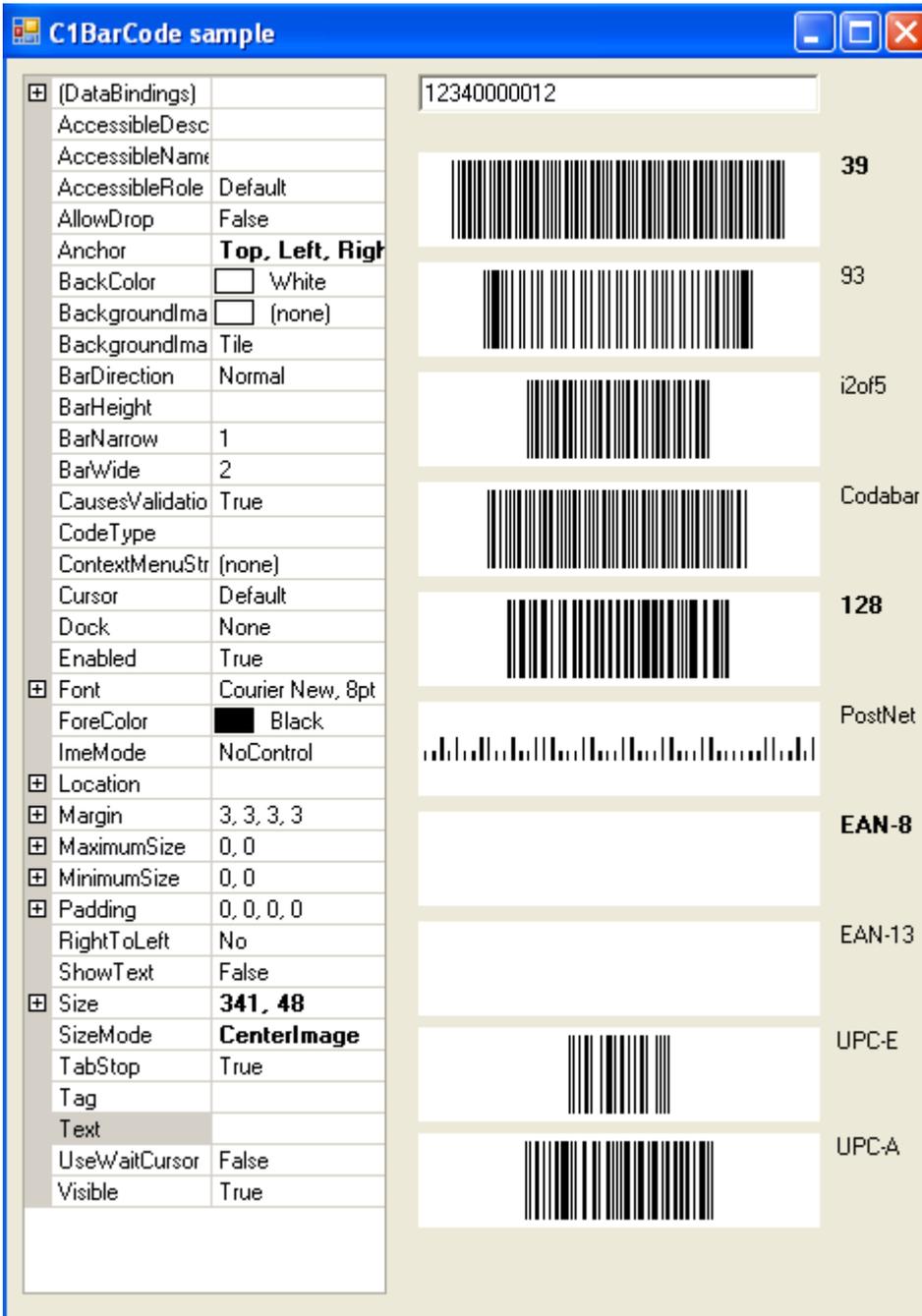
Products Report

<i>ProductID</i>	<i>ProductName</i>	<i>UnitPrice</i>	<i>UnitsInSt</i>	<i>C1Barcode</i>
1	Chai	18	39	
2	Chang	19	17	
3	Aniseed Syrup	10	13	
4	Chef Anton's Cajun Seasoning	22	53	
5	Chef Anton's Gumbo Mix	21.35	0	
6	Grandin's Boysenberry Spread	25	120	
7	Uncle Bob's Organic Dried Pears	30	15	

The main features of **BarCode for WinForms** include:

- Supports 10 different encodings

The C1Barcode control supports 10 encodings, including: Codabar, Cod128, Code39, Code93, Code120%, Ean13, Ean8, PostNet, UpcA, and UpcE.



- Provides the **C1QRCode** format

The **QR code (Quick Response code)** format is one of the most popular 2D barcode formats available today, with free readers available for virtually all smart phones. See [Using C1QRCode](#) for more information.

- Automatically adds checksums

The C1Barcode control automatically adds necessary control symbols and checksums to the value being encoded, depending on the encoding being used, to guarantee a good read on your barcodes.

- Royalty-free DLL for easy deployment

C1Barcode is a royalty-free DLL that can be deployed with your applications like any regular assembly.

BarCode for WinForms Quick Start

This section details some of the features of **BarCode for WinForms**. This quick start will walk through the steps of adding **C1Barcode** to your project and setting C1Barcode appearance and behavior settings. The project that is created in this quick start will demonstrate the various encodings available in **BarCode for WinForms**. For more information about available encodings, see [Supported Encodings](#).

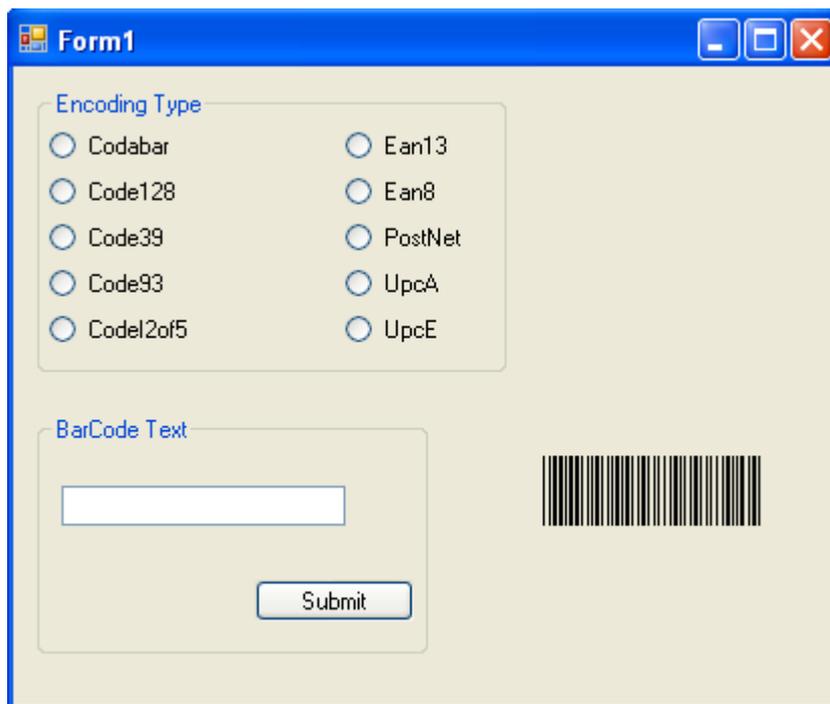
Step 1 of 3: Setting Up the Form

To begin, we will add **C1Barcode** to the form and set up the project.

Complete the following steps:

1. Create a new .NET project.
2. From the Toolbox, add the following controls to the form:
 - Two GroupBox controls
 - Ten RadioButton controls
 - One TextBox control
 - One command Button
 - One C1Barcode control
3. Set the following Text properties for the controls, and arrange the controls as in the image below:

Name	Text Property
GroupBox1	Encoding Type
GroupBox2	Bar Code Text
Button1	Submit
RadioButton1	CodeBar
RadioButton2	Code128
RadioButton3	Code39
RadioButton4	Code93
RadioButton5	Codel2of5
RadioButton6	Ean13
RadioButton7	Ean8
RadioButton8	PostNet
RadioButton9	UpcA
RadioButton10	UpcE



4. In the Properties window, set C1Barcode1's `BackColor` property to **Transparent**.
5. Resize the C1Barcode1 control.

You've just completed the first step in the **BarCode for WinForms** quick start. Continue to the next step to add code to the project.

Step 2 of 3: Adding Code to the Project

In the last step you set up the form and added a `C1Barcode` control to the project, you will continue by adding code to the project.

Complete the following steps:

1. Double-click **Button1** to switch to code view and add the **Button1_Click** event. Add the following code inside the **Button1_Click** event to set C1Barcode1's `Text` property to the text entered in TextBox1:

To write code in Visual Basic

Visual Basic

```
C1Barcode1.Text = TextBox1.Text
```

To write code in C#

C#

```
c1Barcode1.Text = TextBox1.Text;
```

2. Add the following code to handle Radio Button selection:

 **Note:** You must add the **Imports C1.Win.C1Barcode** (Visual Basic projects) or **using C1.Win.C1Barcode;** (C# projects) to the top of your form in order for the following code to work correctly.

To write code in Visual Basic

Visual Basic

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton1.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.Codabar
End Sub

Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton2.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.Code128
End Sub

Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton3.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.Code39
End Sub

Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton4.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.Code93
End Sub

Private Sub RadioButton5_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton5.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.CodeI2of5
End Sub

Private Sub RadioButton6_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton6.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.Ean13
End Sub

Private Sub RadioButton7_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton7.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.Ean8
End Sub

Private Sub RadioButton8_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton8.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.PostNet
End Sub

Private Sub RadioButton9_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton9.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.UpcA
End Sub

Private Sub RadioButton10_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RadioButton10.CheckedChanged
    C1Barcode1.CodeType = CodeTypeEnum.UpcE
End Sub
```

To write code in C#

```
C#
private void radioButton1_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.Codabar;
}

private void radioButton2_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.Code128;
}

private void radioButton3_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.Code39;
}

private void radioButton4_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.Code93;
}

private void radioButton5_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.CodeI2of5;
}

private void radioButton6_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.Ean13;
}

private void radioButton7_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.Ean8;
}

private void radioButton8_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.PostNet;
}

private void radioButton9_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.UpcA;
}

private void radioButton10_CheckedChanged(object sender, System.EventArgs e)
{
    c1Barcode1.CodeType = CodeTypeEnum.UpcE;
}
```

```
}
```

You've just completed the second step in the **BarCode for WinForms** quick start. Continue to the next step to run and view the project.

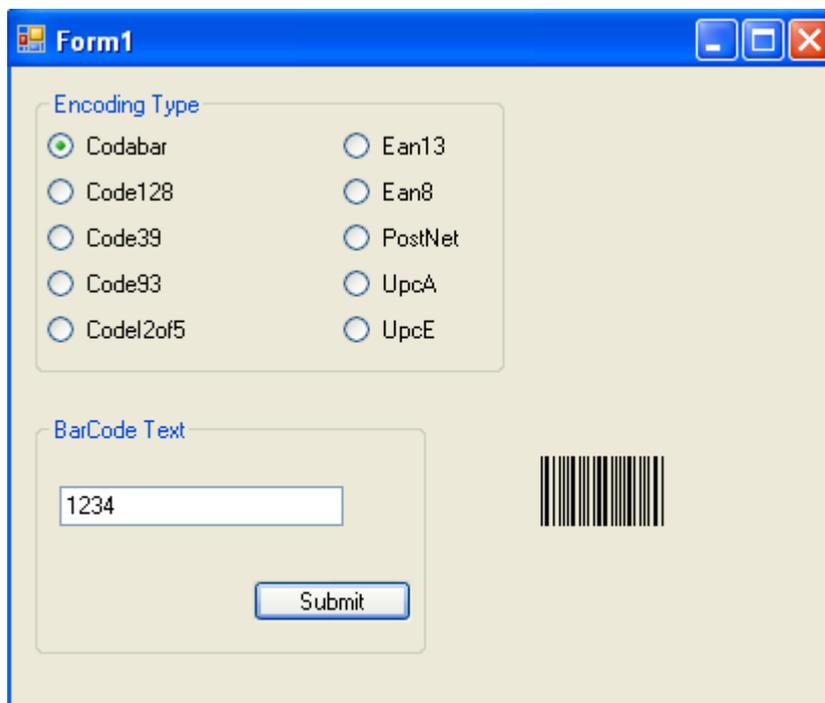
Step 3 of 3: Running the Project

Now that we've set up the project and added code, we'll run the project to view the encodings supported by **BarCode for WinForms**.

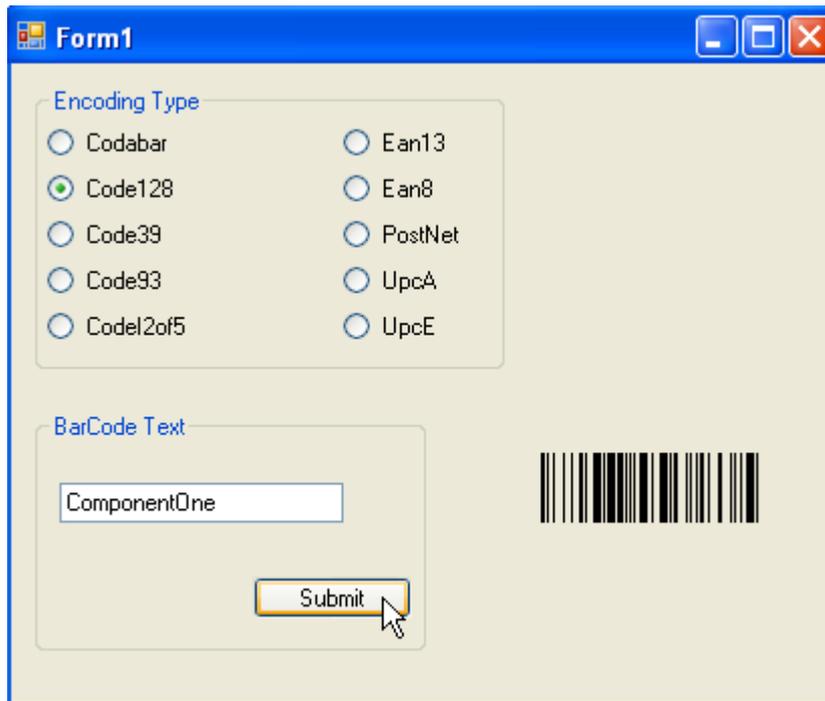
Complete the following steps:

1. Run the project and select the **Code Bar** radio button.
2. Enter a numeric string, for example 1234, in the text box and click the **Submit** button.

Notice that the barcode now appears similar to the following:



3. Select different radio buttons to change the encoding type, and notice that the [C1Barcode](#) changes in appearance.
4. Try entering your name or different strings of alpha numeric characters to see what characters each encoding type will accept:



Note that some encodings have a minimum character requirement, while others will only work with numeric values. For more information about encodings, see [Supported Encodings](#).

This concludes the **BarCode for WinForms** quick start.

Using BarCode for WinForms

The following section details information about using **BarCode for WinForms**, including information about supported encoding formats.

Supported Encodings

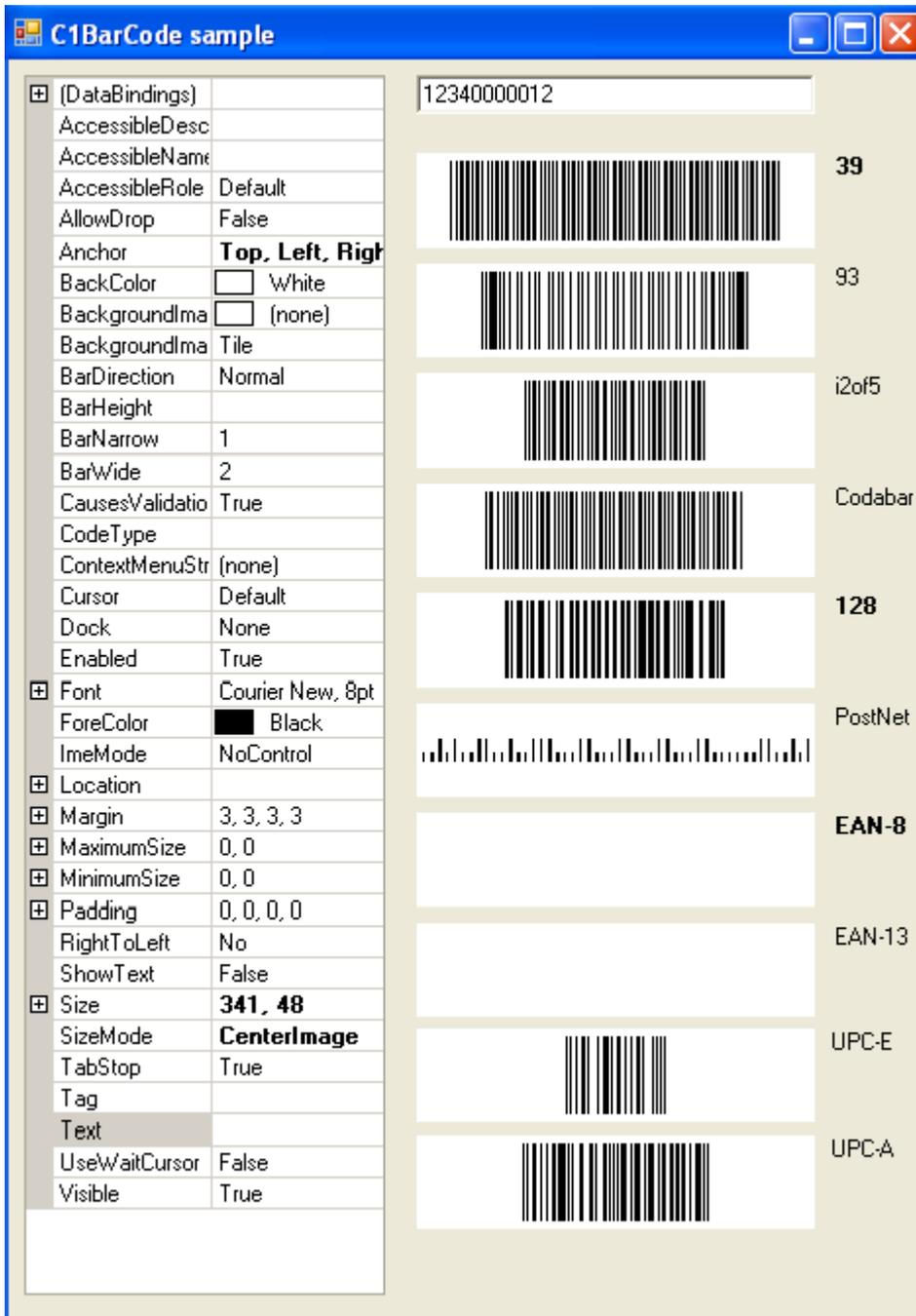
You can change the [C1BarCode](#) encoding type by setting the [CodeType](#) property. The C1BarCode control supports the following encodings:

Encoding	Description
Code Bar	Code Bar may encode 16 different characters (0 through 9 plus -\$.:/.+), plus an additional 4 start/stop characters (A through D). Code Bar is used by some US blood banks, photo labs, and on FedEx airbills.
Code128	Code 128 is a very high density alpha-numeric barcode. It will use the least amount of space of any current 1-D symbology for symbols with 6 characters or more.
Code39	Code 39 is an alpha-numeric encoding also known as 3 of 9 and LOGMARS. This was the first alphanumeric symbology developed, and is one of the most widely used encodings.
Code93	Code 93 is an alpha-numeric encoding that is slightly denser than code 39.
Codel2of5	Code l2of5 is a numeric encoding. The symbol can be as long as necessary to store the encoded data.
Ean13	EAN-13 was implemented by the International Article Numbering Association (EAN) in Europe. EAN-13 encodes a 12-digit code that consists of a 2 digit system code followed by a 5 digit manufacturer code and a 5-digit product code. The 12-digit code is followed by a checksum digit (automatically added by the control).
Ean8	EAN-8 provides a short barcode for small packages. It encodes a 7-digit code that consists of a 2 or 3 digit system code followed by a 4 or 5 digit product code. The 7-digit code is followed by a checksum digit (automatically added by the control).
PostNet	PostNet is a numeric encoding used by the US postal service. It differs from most others in that it is based on the height of the bars rather than on their width.
UpcA	UPC-A is the the common encoding you will find on virtually every consumer good on the shelves of your local supermarket, as well as books, magazines, and newspapers. It is similar to EAN-13, and encodes 11 digits of numeric data along with a trailing check digit.
UpcE	UPC-E is a variation of UPC-A which allows for a more compact barcode by eliminating "extra" zeros. Since the resulting UPC-E barcode is about half the size of a UPC-A barcode, it is generally used on products with very small packaging. When using the UPC-E encoding, set the Text property to an 11-digit string as if you were using the UPC-A encoding. Note that not all UpcA codes can be encoded in UpcE. If the manufacturer code ends with "000", "100", or "200", the product number must be <= 900. If the manufacturer code ends with "00" but not with "100", "200", or "300", then the product number must be <= 90. If the manufacturer code ends with "0" but not with "00", then the product number must be <= 9. If the manufacturer code does not end with "0", then the product number must be between 5 and 9.

Customizing the C1BarCode Control

To use the [C1BarCode](#) control, set the [CodeType](#) property to the type of encoding you want to use, then set the [Text](#)

property to the value you want to encode.



The control will show the barcode image. Note that some encodings have a minimum character requirement, while others will only work with numeric values.

Using the C1Barcode Control in a Document

If you want to include the barcode in a document, use the [Image](#) property to retrieve a resolution-independent image of the barcode.

For example:

To write code in Visual Basic

Visual Basic

```
C1Barcode1.CodeType = C1.Win.C1Barcode.CodeTypeEnum.Code39  
C1Barcode1.Text = "123456"  
PictureBox1.Image = C1Barcode1.Image
```

To write code in C#

C#

```
c1Barcode1.CodeType = CodeTypeEnum.Code39;  
c1Barcode1.Text = "123456";  
pictureBox1.Image = c1Barcode1.Image;
```

Using C1QRCode

The **QR code (Quick Response code)** format is one of the most popular 2D barcode formats available today, with free readers available for virtually all smart phones. Developed by the DENSO-WAVE company, the **QR code** format is efficient and compact, it doesn't require a special scanner to read it, and it is an open and freely available standard (ISO/IEC18004 and others).

The code is simply made up of black and white pixels arranged in patterns. With the **C1QRCode** control, the **QR** patterns are based on the value you want to encode, which you can specify in the **Text** property.



How to Obtain an Image from C1QRCode

The **C1QRCode** control is used to show the QR image. If you want to include the QR image in a document, you can use the **C1QRCode.Image** property to retrieve an image of the barcode.

To use the **C1QRCode** control, follow these steps:

1. Drag a **C1QRCode** control from the Visual Studio Toolbox to your form.
2. In the Properties window, set **C1QRCode's C1QRCode.Text** property to the value you want encoded. The patterns will change based on the value entered in the **C1QRCode.Text** property.

Take a look at the **QRCodeConstructor** sample provided with the product. Samples are installed by default to **Documents\ComponentOne Samples\WinForms**.

This sample demonstrates how formatted information can be entered and converted into a QR code that can be printed or saved.

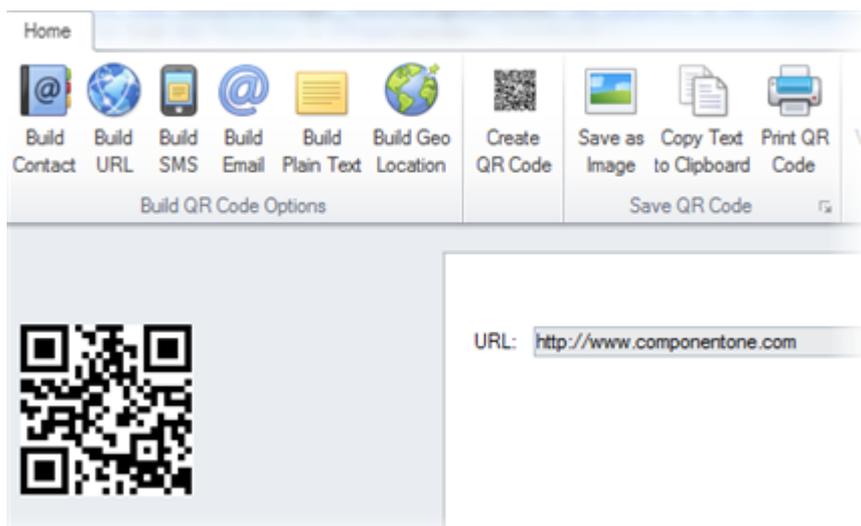
For example, the following code trims the text for a URL that is entered, returning a string with all leading and trailing white-space characters removed; checks the text to make sure it is less than 271 characters; and then creates the QR image based on the text entered.

To write code in Visual Basic

Visual Basic

```
Private Sub CreateURLCode()  
    _QRText = Trim(txtFormWebsiteAddress.Text)  
    If QrCodeSizeCheck() Then  
        _CodeBuilt = True  
        EnableButtons()  
    Else  
        DispalyLengthWarning()  
        Return  
    End If  
End Sub
```

When a URL is entered and **"Create QR Code"** is clicked, this is the resulting **QR** image:



C1QRCode Encoding Limits

Although the QR specification includes 40 levels, or encodings, the [C1QRCode](#) control only implements levels 1 through 10. The table below summarizes the image sizes and content length supported by each level.

Correction Level: L (higher correction levels reduce encoding capacity)

Level	Size	Numeric	Alpha	Bin
1	21x21	41	25	17
2	25x25	77	47	32
3	29x29	127	77	53
4	33x33	187	114	78
5	37x37	255	154	78
6	41x41	322	195	134
7	45x45	370	224	154
8	49x49	461	279	192
9	53x53	552	335	230
10	57x57	652	395	271



NOTE:

The **Level** is the encoding and can be set to *Automatic*.

The **Size** is the image size in pixels.

Alphanumeric characters that are supported include [0-9][A-Z][\$%*+-.:/].

For details on the QR format, please see: <http://www.denso-wave.com/qrcode/qrstandard-e.html> and http://en.wikipedia.org/wiki/QR_code.

Improving the C1QRCode Image Resolution

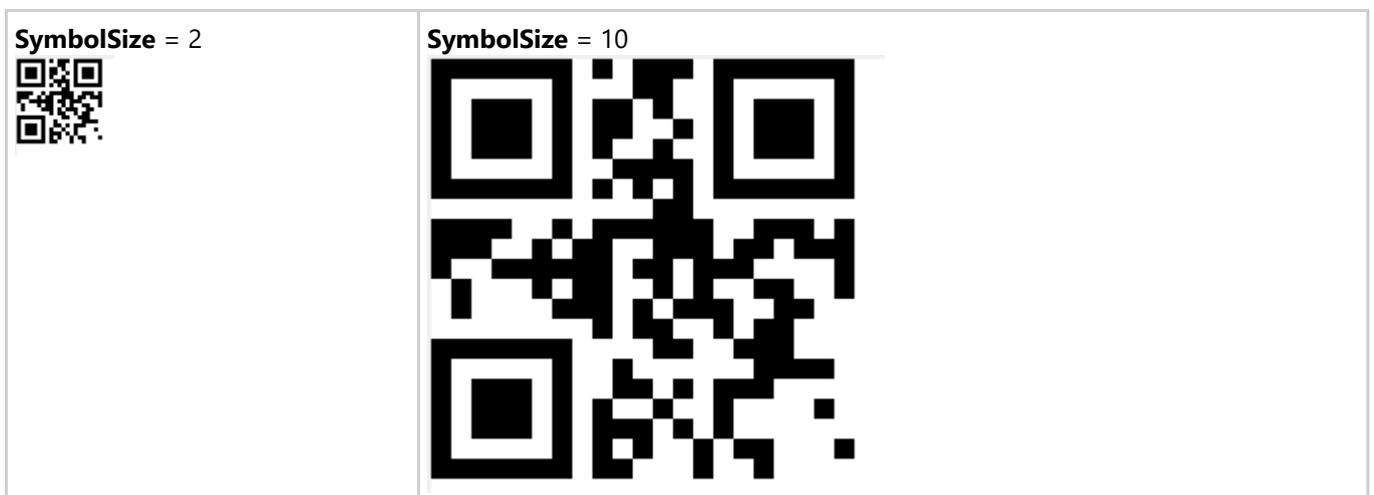
You can change the size of the symbols used to build the **QR** image by setting the [SymbolSize](#) property.

Simply select the [C1QRCode](#) control on the form and set the `SymbolSize` property in the Visual Studio Properties window to a value between **2** and **10**. Larger values will result in larger images which consume more space but may be easier to for some scanners to read.

The default value for the `SymbolSize` property is **3**, which looks like this:



You can set the `SymbolSize` property to any value between **2** and **10**. To give you an example of the difference in sizes, look at the following images:



BarCode for WinForms Samples

Please be advised that this ComponentOne software tool is accompanied by various sample projects and/or demos which may make use of other development tools included with the ComponentOne Studio.

Please refer to the pre-installed product samples through the following path:

Documents\ComponentOne Samples\WinForms

The following tables provide a short description for each sample.

Visual Basic Samples

Sample	Description
PrintBarCodes	Demonstrates how to print barcodes created with the C1BarCode control. This sample uses the C1BarCode control.
QRCodeConstructor	Demonstrates how formatted information can be entered and converted into a QR code that can be saved or printed. This sample uses the C1QRCode control.

C# Samples

Sample	Description
BarCodeSample	Demonstrates the different types of barcode encoding. This sample uses the C1BarCode control.
PrintBarCodes	Demonstrates how to print barcodes created with the C1BarCode control. This sample uses the C1BarCode control.