

# BarCode for WinForms

2018.07.20 更新

グレースィティ株式会社

## 目次

<a href="#">BarCode for WinForms の概要</a>	2
<a href="#">主な特長</a>	3
<a href="#">BarCode for WinForms クイックスタート</a>	4
<a href="#">手順 1: プロジェクトの設定</a>	4
<a href="#">手順 2: プロジェクトへのコードの追加</a>	4-6
<a href="#">手順 3: プロジェクトの実行</a>	6-7
<a href="#">BarCode for WinForms を使用する</a>	8
<a href="#">BarCode シンボル</a>	8-15
<a href="#">BarCode のオプション</a>	15-18
<a href="#">C1BarCode コントロールをカスタマイズする</a>	19-20

## BarCode for WinForms の概要

新しいバーコードエンジンは、スタンドアロンコントロール **C1BarCode** を提供します。これは、これまでのバージョンより多くの標準バーコードタイプを実装しています。新しい **C1.Win.Barcode** アセンブリは、**C1BarCode** コントロールと **C1QRCode** コントロールを、異なるコードベースを使用する 1 つの **C1BarCode** コントロールに置き換えます。

バーコードフォントとは異なり、**BarCode for WinForms** は、使用しているエンコーディングに応じて、自動的に必要な制御シンボルとチェックサムを追加しながら値をエンコードすることで、読み取りエラーをなくします。いくつかの標準的なバーコードを使用して、グリッドセル、Web ページ、または通常の .NET PrintDocument オブジェクトにバーコード画像を追加できます。また、通常のアセンブリと同様に、アプリケーションと共に **BarCode for WinForms** を配布できます。これは無償の DLL なので、クライアント側にバーコードフォントをインストールしたり、それが無償であることを確認するための手間は不要です。さらに、**BarCode for WinForms** は、たいへん使いやすい製品です。フォームにコントロールを追加し、エンコーディングタイプを設定するだけで使用できます。

## 主な特長

- **38 種類の BarCode シンボルのサポート**

**C1Barcode** コントロールは、**Windows フォームアプリケーション**のために 38 種類の標準的なバーコードをサポートします。バーコードのレンダリングやカスタマイズのために設定できるプロパティもいくつか提供しています。一部のバーコードでは、FNC1 文字もサポートされます。詳細については、「[BarCode for WinForms を使用する](#)」を参照してください。

- **QRCode 形式の統合**

QR コード (Quick Response コード) 形式は、今日最もよく使用されている 2 次元バーコード形式の 1 つで、ほとんどすべてのスマートフォンの無料リーダーが対応しています。**C1Barcode** コントロールは QR コードの機能を提供しているため、他のコントロールを追加する必要はありません。

- **チェックサムを自動的に追加**

**C1Barcode** コントロールは、バーコードが正しく読み取られるように、使用されているシンボルに応じて、必要な制御シンボルとチェックサムを自動的に追加しながら値をエンコードします。

- **配布が容易な無償の DLL**

**C1Barcode** は、通常のアセンブリと同様に、無償の DLL としてアプリケーションと共に配布できます。

## BarCode for WinForms クイックスタート

このクイックスタートセクションでは、プロジェクトに **C1BarCode** を追加して、簡単なバーコードアプリケーションを作成する手順について説明します。このクイックスタートは、テキストやバーコードタイプを変更したときに、**C1BarCode** で使用できるさまざまなシンボルがどのようにレンダリングされるかを示します。利用できるシンボルの詳細については、「[BarCode シンボル](#)」を参照してください。

### 手順 1: プロジェクトの設定

プロジェクトを設定し、フォームに **C1BarCode** コントロールを追加するには、次の手順を実行します。

1. 新しい .NET プロジェクトを作成します。
2. ツールボックスから、次のコントロールを追加します。
  - C1BarCode
  - ComboBox (C1BarCode コントロールで利用できるバーコードのリストを表示する)
  - TextBox (バーコードを表示するためのテキストを入力する)
  - Label (上のコントロールの名前を表示する 3 つのラベルを追加する)

**C1BarCode** コントロールを追加すると、**C1.Win.BarCode** アセンブリと **C1.Win** アセンブリがプロジェクトの **References** フォルダに追加されます。ツールボックスに **C1BarCode** コントロールがない場合は、次の手順を実行します。

1. **ツールボックス**で、タブを右クリックし、**[アイテムの選択]**オプションを選択して、**[ツールボックス項目の選択]**ウィザードを開きます。
2. **[参照]**をクリックします。デフォルトのインストールパス C:\Program Files (x86)\ComponentOne\WinForms\Bin\v4 に移動し、**C1.Win.BarCode.4 dll** を選択してから、**[OK]**をクリックします。
3. **ComboBox** をクリックし、**[プロパティ]**ウィンドウで、その名前を cbCodeType に変更します。
4. 3 つの **Label** コントロールの **Text** プロパティをそれぞれ「Code Type」、「Text」、および「BarCode」に設定します。以下のように、フォームにコントロールを配置します。



**BarCode for WinForms** を使用した簡単なアプリケーションの作成に向けて最初の手順を完了しました。

### 手順 2: プロジェクトへのコードの追加

プロジェクトにコードを追加するには、次の手順に従います。

# BarCode for WinForms

1. 名前空間 **Imports C1.BarCode** (Visual Basic プロジェクト) または **using C1.BarCode** (C# プロジェクト) を追加します。
2. **Form1** をダブルクリックして **Form1\_Load** イベントを作成し、コードビューに切り替えます。 **TextBox** のデフォルトのテキストを設定し、 **ComboBox** コントロールに **C1BarCode** で使用できるコードタイプを設定するために、次のコードを追加します。

Visual Basic	copyCode
<pre>Dim types As Array = [Enum].GetValues(GetType(CodeType)) Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load     tbText.Text = "HELLO WORLD!"     cbCodeType.DataSource = types     cbCodeType.SelectedIndex = Array.IndexOf(types, CodeType.Code39)     cbCodeType.Select() End Sub</pre>	

C#	copyCode
<pre>Array types = Enum.GetValues(typeof(CodeType)); private void Form1_Load(object sender, EventArgs e) {     tbText.Text = "HELLO WORLD!";     cbCodeType.DataSource = types;     cbCodeType.SelectedIndex = Array.IndexOf(types, CodeType.Code39);     cbCodeType.Select(); }</pre>	

3. フォームの **ComboBox** をクリックします。プロパティウィンドウのイベントから、 **SelectedIndexChanged** をダブルクリックして **cbCodeType\_SelectedIndexChanged** イベントを作成し、次のコードを追加します。

Visual Basic	copyCode
<pre>Private Sub cbCodeType_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cbCodeType.SelectedIndexChanged     C1Barcode1.CodeType = DirectCast(cbCodeType.SelectedValue, CodeType) End Sub</pre>	

C#	copyCode
<pre>private void cbCodeType_SelectedIndexChanged(object sender, EventArgs e) {     c1Barcode1.CodeType = (CodeType)cbCodeType.SelectedValue; }</pre>	

4. フォームの **TextBox** コントロールをクリックします。プロパティウィンドウのイベントから、 **TextChanged** をダブルクリックして **tbText\_TextChanged** イベントを作成し、そのイベントに次のコードを追加します。

Visual Basic	copyCode
<pre>Private Sub tbText_TextChanged(sender As Object, e As EventArgs) Handles tbText.TextChanged     C1Barcode1.Text = tbText.Text End Sub</pre>	

C#	copyCode
<pre>private void tbText_TextChanged(object sender, EventArgs e) {</pre>	

```
c1Barcode1.Text = tbText.Text;  
}
```

この手順では、コントロールに機能を追加しました。次の手順では、プロジェクトを実行し、実行時の動作を確認します。

## 手順 3: プロジェクトの実行

プロジェクトを設定し、コードを追加できました。ここでプロジェクトを実行して、**BarCode for WinForms** でサポートされるコードタイプを表示しましょう。

次の手順を実行します。

1. プロジェクトを実行し、**ComboBox** からコードタイプを選択します。

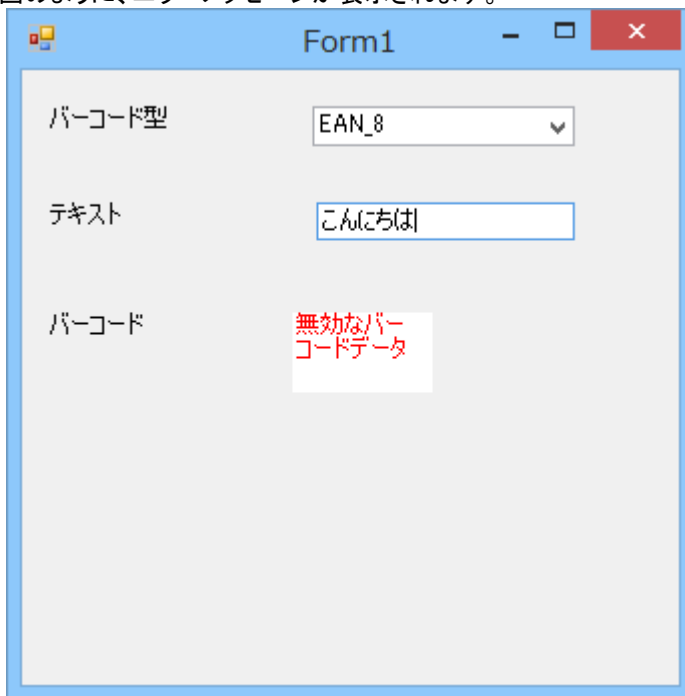


2. さまざまなコードタイプを選択し、対応するバーコードが表示されることを確認します。




# BarCode for WinForms

3. さまざまな英数字テキスト文字列を入力してみます。テキストがバーコードによってサポートされていない場合は、次の図のように、エラーメッセージが表示されます。



おめでとうございます。クイックスタートが完了しました。**BarCode for WinForms** でサポートされるさまざまなバーコードを表示する簡単なアプリケーションを作成しました。

 文字要件が極めて少ないエンコーディングもあれば、数値しか使用できないエンコーディングもあります。サポートされるバーコードシボルの詳細については、「[BarCode シンボル](#)」を参照してください。








## BarCode for WinForms を使用する

新しいバーコードエンジンを使用すると、さまざまな業界標準のバーコードを **Windows フォーム** アプリケーションに導入できます。バーコードの機能を理解するには、バーコードシンボルとプロパティ、またはこれらに関連するオプションについて知っておくことが重要です。

### BarCode シンボル

バーコードシンボルは、文字データをバーコードの太いバー、細いバー、およびスペースから成るパターンに変換するために使用されるエンコードスキームを指定します。次の表は、**BarCode for WinForms** の **CodeType** プロパティに用意されているバーコードシンボルを示します。




コードタイプ	例	説明
Ansi39		ANSI 3 of 9 (Code 39) は、大文字、数字、-、*、\$、/、+、% を使用します。これがデフォルトのバーコードスタイルです。
Ansi39x		ANSI Extended 3 of 9 (Extended Code 39) は、ASCII 文字セット全体を使用します。
BC412		The BC412 barcode was invented by IBM to meet the needs of the semiconductor wafer identification application.
Codabar		Codabar は、A B C D + - : . / \$ および数字を使用します。
Code11		Code11, also known as USD-8, is a high-density barcode symbology developed by Intermec in 1977. It is primarily used to label telecommunication equipments. This symbology is discrete and is able to encode numeric digits through 0-9, dash (-), and start/stop characters.
Code_128_A		Code 128 A は、制御文字、数字、句読点、大文字を使用します。 このバーコードタイプに対する <b>Checksum</b> を有効にする必要があります。
Code_128_B		Code 128 B は、数字、句読点、大文字、小文字を使用します。 このバーコードタイプに対する <b>Checksum</b> を有効にする必要があります。

# BarCode for WinForms

Code_128_C	 01143493	Code 128 C は数字だけを使用します。 このバーコードタイプに対する <b>Checksum</b> を有効にする必要があります。
Code_128auto	 1143493	Code 128 Auto は、ASCII 文字セット全体を使用します。最小のバーコードになるように、Code 128 A、B、および C から 1 つが自動的に選択されます。 このバーコードタイプに対する <b>Checksum</b> を有効にする必要があります。
Code_2_of_5	 3661239	Code 2 of 5 は数字だけを使用します。
Code93	 MSU 09382	Code 93 は、大文字、数字、% \$ * / , + - を使用します。 このバーコードタイプに対する <b>Checksum</b> を有効にする必要があります。
Code25intlv	 1023392210	Interleaved 2 of 5 は数字だけを使用します。
Code39	 AUI%89032	Code 39 は、大文字、数字、% * \$ / , - + を使用します。
Code39x	 BAR92112234	Extended Code 39 は、ASCII 文字セット全体を使用します。 このバーコードタイプに対する <b>Checksum</b> を有効にする必要があります。
Code49	 1293829ABJISSH92K234	Code 49 は、各段 8 文字の 2 ~ 8 段から成る 2 次元高密度スタック型バーコードです。各段には、スタートコードとストップコードが含まれます。ASCII 文字セット全体をエンコードします。
Code93x	 CODE349101%	Extended Code 93 は、ASCII 文字セット全体を使用します。
DataMatrix		Data Matrix は、正方形または長方形のマトリックスパターンに並べられた正方形モジュールから成る 2 次元高密度バーコードです。








EAN_13		<p>EAN-13 は、数字(12 個の数字とチェックデジット)だけを使用します。12 個の数字だけを文字列として受け取ってチェックデジット(CheckSum)を計算し、それを 13 番目の位置に追加します。チェックデジットは、バーコードが正しくスキャンされたかどうかを検証するために使用される追加の桁です。チェックデジットは、Checksum プロパティが true に設定されると自動的に追加されます。</p>
EAN_8		<p>EAN-8 は、数字(7 個の数字とチェックデジット)だけを使用します。 このバーコードタイプに対する<b>Checksum</b>を有効にする必要があります。</p>
HIBCode39		<p>HIBCode39 is a Health Industry Bar Code 39 implementation.</p>
HIBCode128		<p>HIBCode128 is a Health Industry Bar Code 128 implementation.</p>
EAN128FNC1		<p>EAN-128 は、運送業で梱包物を識別するためのアプリケーション識別子(AI)データの 1 次元英数字表現です。</p> <p>このタイプのバーコードには、以下のセクションが含まれています。</p> <ul style="list-style-type: none"> <li>● 先頭のクワイエットゾーン(空白領域)</li> <li>● Code 128 のスタート文字</li> <li>● スキャナに EAN-128 バーコードであることを識別させる FNC(ファンクション)1 文字</li> <li>● データ(AI + データフィールド)</li> <li>● シンボルチェック文字(「スタートコード値 + 各文字位置と各文字値の積」を 103 で除算。この剰余値がチェックサム)。</li> <li>● ストップ文字</li> <li>● 末尾のクワイエットゾーン(空白領域)</li> </ul> <p>データセクションの AI は、後に続くデータのタイプ(ID、日付、数量、計量単位など)を設定します。データのタイプごとに固有のデータ構造があります。この AI があることが、EAN-128 コードが Code 128 と異なる点です。</p> <p>1 つのバーコードで複数の AI(およびそのデータ)を組み合わせたことができます。</p> <p>EAN128FNC1 は UCC/EAN-128 (EAN128)バーコードタイプの 1 つで、任意の場所に FNC1 文字を挿入したり、</p>

# BarCode for WinForms

		<p>バーサイズを調整することができるなど、UCC/EAN-128 にはない機能をサポートしています。</p> <p>FNC1 文字を挿入するには、実行時に Text プロパティに "\n" (C# の場合) または "\vbLf" (VB の場合) を設定します。</p>
lata25		Represents an IATA 2 of 5 barcode.
IntelligentMail		Intelligent Mail (以前の 4-State Customer Barcode) は、米国内の郵便に使用される 65 本のバーコードです。
IntelligentMailPackage		Intelligent Mail Package Barcode.
ISBN		The International Standard Book Number (ISBN) is special commercial book identifier which encodes 9 numeric digits apart from the start number "978", "979".
ISMN		The International Standard Music Number or ISMN (ISO 10957) is a thirteen-character alphanumeric identifier for printed music developed by ISO.
ITF14		ITF14 barcode is the GS1 implementation of an Interleaved 2 of 5 bar code to encode a Global Trade Item Number. It is continuous, self-checking, bidirectionally decodable and it will always encode 14 digits. ITF14 is used on packaging levels of a product in general.
ISSN		The International Standard Serial Number (ISSN) is an eight-digit number used for printed or electronic periodical publications like magazines, etc. This ISSN system was drafted as an International Standard in 1971 and published as ISO 3297 in 1975.
JapanesePostal		これは、日本の郵便制度で使用されるバーコードです。20 桁の英数字をエンコードします。これには、7 桁の郵便番号の後に、必要に応じて番地や部屋番号などの情報が含まれます。エンコードするデータにはハイフンを含めることができます。
Matrix_2_of_5	 790022312	Matrix 2 of 5 は、3 本の黒バーと 2 本の白バーから成る比較的高密度のバーコードです。

MicroPDF417		<p>MicroPDF417 は、PDF417 から派生された 2 次元多段バーコードです。Micro-PDF417 は、データをできる限り小さなサイズの 2 次元シンボル(最大 150 バイト、英数字 250 文字、数字 366 桁)にエンコードする必要があるアプリケーション用に設計されています。</p> <p>MicroPDF417 では、可変長のアプリケーション識別子(AI)のフィールド区切り文字として FNC1 文字を挿入することができます。</p> <p>FNC1 文字を挿入するには、実行時に Text プロパティに "\n"(C# の場合)または "\vLf"(VB の場合)を設定します。</p>
MicroQRCode		<p>MicroQRCode is a variant of QR Code 2005. Compared with other regular QR Codes, it has only one position detection pattern which reduces the barcode size so that it can be used to applications where the space for barcode image is severely restricted.</p>
MSI		<p>MSI Code は数字だけを使用します。</p> <p>このバーコードタイプに対する<b>Checksum</b>を有効にする必要があります。</p>
Pdf417		<p>Pdf417 は、よく使用されている 2 次元高密度バーコードです。最大 1,108 バイトの情報をエンコードできます。このバーコードは、小さなバーコードを積み重ねて構成されます。ASCII 文字セット全体をエンコードします。10 のエラー修正レベルと 3 つのデータコンパクションモード(Text、Byte、および Numeric)があります。このバーコードシンボルは、英数字で最大 1,850 文字、数字で最大 2,710 桁をエンコードできます。</p>
Pharmacode		<p>Pharmacode, also known as Pharmaceutical Binary Code, is a barcode standard, 1D barcode that is used in the pharmaceutical manufacturing industry as a packing control system.</p>
Plessey		<p>MSI barcode, also known as Modified Plessey, is a numeric symbology developed by the MSI Data Corporation, which is used primarily for marking retail shelves for inventory control. Though continuous and self-checking, MSI Plessey provides several module checksum situations.</p>

# BarCode for WinForms

PostNet		PostNet は、数字とチェックデジットを使用します。
PZN		PZN or Pharma-Zentral-Nummer is a barcode standard used in the German pharmaceutical industry for identification of medicines and health-care products.
QRCode		QRCode は、英数字やバイトデータに加えて、日本語の漢字やかな文字も扱うことができる 2 次元バーコードです。このバーコードは最大 7,366 文字をエンコードできます。
RM4SCC		Royal Mail RM4SCC は、文字と数字 (チェックデジット付き) だけを使用します。これは、イギリスのロイヤルメールで 사용되는バーコードです。 このバーコードタイプに対する <b>Checksum</b> を有効にする必要があります。
RSS14	 (01)13393821228905	RSS14 は、全方向型 POS スキャンに EAN.UCC 品目識別を使用する 14 桁の省スペースシンボル (Reduced Space Symbolology: RSS) です。
RSS14Stacked	 (01)03939382122899	RSS14Stacked は、RSS14Truncated と同様にインジケータデジットを含む EAN.UCC 情報を使用しますが、幅が狭い場合、2 段に積み重ねられます。 RSS14Stacked では合成オプションを設定できます。[タイプ]ドロップダウンリストからバーコードのタイプを選択し、[値]フィールドで合成バーコードの値を指定できます。
RSS14StackedOmnidirectional	 (01)01339382122891	RSS14StackedOmnidirectional は、RSS14 と同様に全方向型スキャンを含む EAN.UCC 情報を使用しますが、幅が狭い場合、2 段に積み重ねられます。
RSS14Truncated	 (01)30944382332892	RSS14Truncated は、RSS14 と同様に EAN.UCC 情報を使用しますが、POS でスキャンされない小さな品目に使用するために、0 または 1 のインジケータデジットを含みます。

<p>RSSExpanded</p>	 <p>8110100706401002003100110120</p>	<p>RSSExpanded は、RSS14 と同様に EAN.UCC 情報を使用しますが、重量や賞味期限などの AI 要素を追加します。</p> <p>RSSExpanded では、可変長のアプリケーション識別子(AI)のフィールド区切り文字として FNC1 文字を挿入することができます。</p> <p>FNC1 文字を挿入するには、実行時に Text プロパティに "\n"(C# の場合)または "\vLf"(VB の場合)を設定します。</p>
<p>RSSExpandedStacked</p>	 <p>8110100706401002003100110120</p>	<p>RSSExpandedStacked は、RSSExpanded と同様に AI 要素を含む EAN.UCC 情報を使用しますが、幅が狭い場合、2 段に積み重ねられます。</p> <p>RSSExpandedStacked では、可変長のアプリケーション識別子(AI)のフィールド区切り文字として FNC1 文字を挿入することができます。</p> <p>FNC1 文字を挿入するには、実行時に Text プロパティに "\n"(C# の場合)または "\vLf"(VB の場合)を設定します。</p>
<p>RSSLimited</p>	 <p>(01)00006569232216</p>	<p>RSS Limited は、RSS14 と同様に EAN.UCC 情報を使用しますが、POS でスキャンされない小さな品目に使用するために、0 または 1 のインジケータデジットを含みます。</p> <p>RSSLimited では合成オプションを設定できます。<b>[タイプ]</b>ドロップダウンリストからバーコードのタイプを選択し、<b>[値]</b>フィールドで合成バーコードの値を指定できます。</p>
<p>SSCC 18</p>		<p>Serial Shipping Container Code-18 (SSCC-18) Barcode is a type of barcode that can print in the lower 2-inch (or local equivalent) extended area of the Thermal 4" x 8" or 4" x 8¼" (or local equivalent) label.</p>
<p>Telepen</p>		<p>Telepen is a name of a barcode symbology designed in the UK, in 1972, to directly represent the full ASCII character set without using shift characters for code switching, and use only two different widths for bars and spaces.</p>
<p>UCCEAN128</p>	 <p>BARCODE2312</p>	<p>UCC/EAN -128 は、ASCII 文字セット全体を使用します。HIBC アプリケーションで使用される特殊な Code 128 です。</p> <p>このバーコードタイプに対する<b>Checksum</b>を有効にする必要があります</p>

UPC_A		<p>す。</p> <p>UPC-A は、数字(11 個の数字とチェックデジット)だけを使用します。</p> <p>このバーコードタイプに対する<b>Checksum</b>を有効にする必要があります。</p>
UPC_E0		<p>UPC-E0 は数字だけを使用します。Zero Suppression の UPC シンボルに使用されます。Caption プロパティには、6 桁の UPC-E コードまたは完全な 11 桁(必須のコードタイプ 0 を含む)の UPC-A コードを入力する必要があります。11 桁のコードが入力された場合、バーコードコントロールは、可能であればそれを 6 桁の UPC-E コードに変換します。11 桁のコードを 6 桁のコードに変換できない場合は、何も表示されません。</p>
UPC_E1		<p>UPC-E1 は数字だけを使用します。主に小売業界で在庫ラベルに使用されます。UPC-E1 の入力文字列の長さは数字 6 個です。</p>

以下のバーコードは、FNC1 文字をサポートしています。

- EAN128FNC1
- MicroPDF417
- RSSExpanded
- RSSExpandedStacked

## BarCode のオプション

**C1Barcode** は、すべてのバーコードに共通のオプションと特定のバーコードに固有のオプションを提供しています。これらのオプションは、**C1Barcode** の外観をカスタマイズするために使用されます。

WinForms でバーコードをレンダリングするために、**C1.Win.BarCode** アセンブリで公開されている共通オプションは、次のとおりです。

- **BarDirection**: バーコードの方向を選択できます。次のオプションから選択します。
  - LeftToRight: バーコードシンボルを左から右に印刷します(デフォルト)。
  - RightToLeft: バーコードシンボルを右から左に印刷します。
  - TopToBottom: バーコードシンボルを上から下に印刷します。
  - BottomToTop: バーコードシンボルを下から上に印刷します。
- **BarHeight**: バーコードの高さ(画面ピクセル単位)。バーの高さがコントロールの高さより大きい場合、このプロパティは無視されます。
- **CodeType**: バーコード画像を生成するために Text プロパティに保存された値に適用するエンコーディングを選択できます。
- **CaptionAlignment**: バーコードの値の表示位置を選択できます。使用可能なオプションは、Left、Right、Center です。
- **CaptionGrouping**: 長い番号を読みやすくするために、キャプションの文字グループ間にスペースを入れるかどうかを示す値を指定できます。
- **CaptionPosition**: バーコードシンボルに対するキャプションの相対的な垂直位置を選択します。使用可能なオプションは、None、Above、Below です。
- **Image**: Text プロパティの値を表すバーコードの画像を取得します。画像は、**CodeType** プロパティで指定されたエンコーディングを使用して生成されます。



- **ModuleSize**: バーコードのモジュール(最も狭いバーの幅)(画面ピクセル単位)。広いバーの幅は、バーコードタイプに応じて自動的に計算されます。



- **QuietZone**: バーコードのクワイエットゾーンを指定できます。クワイエットゾーンは、バーコードがどこから始まってどこで終わるかをスキャナが認識できるようにバーコードの左右に置かれた空き領域です。有効なパラメータは、次のとおりです。
  - Left: バーコードの左に残す空き領域のサイズを入力します。
  - Right: バーコードの右に残す空き領域のサイズを入力します。
  - Top: バーコードの上に残す空き領域のサイズを入力します。
  - Bottom: バーコードの下に残す空き領域のサイズを入力します。
 次の図は、左右のクワイエットゾーンを示します。



- **Text**: バーコード画像としてエンコードされる値を指定できます。
- **WholeSize**: バーコード全体のサイズを指定します。WholeWidth はバーコード全体の幅を表し、WholeHeight はバーコード全体の高さを表します。



- **FixLength**: バーコードの値の固定桁数を指定できます。これには、整数値を設定します。
- **AutoSize**: バーコードをコントロールに合わせて拡大するかどうかを指定できます。値は True または False です。
  - AutoSize を True に設定した場合
    - バーコードはコントロールにちょうど収まるように自動的に拡大されます。
    - 読み取り可能なサイズは、バーコード自体によって計算されます。
    - マトリックスバーコードのサイズは、**OnCalculateSize** メソッドによって計算されます。
    - マトリックス以外のバーコードのサイズは、**BarHeight** と **ModuleSize** によって計算されます。

AutoSize を False に設定した場合

- バーコードのサイズは、Width プロパティまたは Height プロパティによって決定されます。
- BarHeight がコントロールの高さより大きい場合は、コントロールがクリップされます。
- BarHeight が高さより小さい場合は、バーコードとコントロールの間に余白が残ります。

# BarCode for WinForms

バーコードのタイプに固有のオプションは、次のとおりです。

**ChecksumEnabled**: チェックデジットを自動的に追加するかどうかを指定できます。連結されているデータに既にチェックデジットが含まれている場合は、チェックデジットが自動的に追加されないようにする必要があります。このプロパティは、Code49、Code128、PostNet5/9/11、JapanesePostal の各バーコードでサポートされます。

## Ean128Fnc1Options:

- **Dpi**: プリンタの解像度を指定します。これには、整数値を設定します。
- **BarAdjust**: ドット単位の調整サイズを指定できます。
- **ModuleSize**: バーコードモジュールの水平方向のサイズを指定できます。これには、整数値を設定します。

## Code25intlvOptions:

- **BearBar**: ITF(Interleaved Two of Five)バーコードにベアラバーを表示するかどうかを選択できます。値は True または False です。
- **LineStroke**: ベアラバーの色を選択できます。
- **LineStrokeThickness**: ベアラバーの線の幅を選択できます。これには、整数値を設定します。

## Code49Options:

- **Grouping**: バーコードでグループ化を使用します。値は True または False です。
- **Group**: バーコードをグループ化する際のグループ番号を取得または設定します。有効な値は 0 ~ 8 です。**Grouping** の値が True の場合、Group の値の範囲は 0 ~ 8 です。Grouping の値が False の場合、Group の値は 0 です。**Grouping** の値が True で、**Group** 値が 0 より小さいか 8 より大きい場合は、BarcodeException.EnumErrorCode.Code49GroupNo が生成されます。

## DataMatrixOptions:

- **EccMode**: ECC モードを選択します。有効な値は、ECC000、ECC050、ECC080、ECC100、ECC140、ECC200 です。
- **Ecc200SymbolSize**: ECC200 シンボルのサイズを選択します。デフォルト値は SquareAuto です。
- **Ecc200EncodingMode**: ECC200 エンコーディングモードを選択します。有効な値は、Auto、ASCII、C40、Text、X12、EDIFACT、Base256 です。
- **Ecc000\_140SymbolSize**: ECC000\_140 シンボルのサイズを選択します。
- **StructuredAppend**: 現在のバーコードシンボルが構造的接続シンボルの一部かどうかを選択します。
- **StructureNumber**: 構造的接続シンボル内の現在のシンボルの構造番号を指定します。この値の範囲は 0 ~ 15 です。
- **FileIdentifier**: 関連する構造的接続シンボルグループのファイル識別子を指定します。有効なファイル識別子値は、[1,254] 以内である必要があります。ファイル識別子を 0 に設定すると、ファイル識別子が自動的に計算されます。

## GS1CompositeOptions:

- **Type**: 合成シンボルタイプを選択します。選択できる値は None または CCA です。CCA(Composite Component - Version A)は、最小タイプの 2 次元コンポジットコンポーネントです。
- **Value**: CCA 文字データを指定できます。

## MicroPDF417Options:

- **CompactionMode**: CompactionMode のタイプを選択します。有効な値は、Auto、TextCompactionMode、NumericCompactionMode、ByteCompactionMode です。
- **FileID**: 構造的接続シンボルのファイル ID を指定します。0 ~ 899 の値を取ります。この値が 0 より小さいか 899 より大きい場合は、BarcodeException.EnumErrorCode.MicroPDF417FileID が生成されます。
- **SegmentCount**: 構造的接続シンボルのセグメント数を指定します。0 ~ 99999 の値を取ります。この値が 0 より小さいか 99999 より大きい場合は、BarcodeException.EnumErrorCode.MicroPDF417SegmentCount が生成されます。
- **SegmentIndex**: 構造的接続シンボルのセグメントインデックスを指定します。0 ~ 99998 の範囲で、セグメント数の値より小さい値を取ります。この値が 0 より小さいか 99998 より大きい場合は、BarcodeException.EnumErrorCode.MicroPDF417SegmentIndex が生成されます。
- **Version**: シンボルサイズを選択します。デフォルト値は ColumnPriorAuto です。

## PDF417Options:

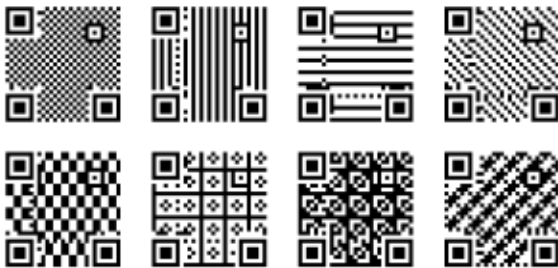
- **Column**: バーコードの列数を指定します。これには、整数値を設定します。デフォルト値は -1 で、値の範囲は 1 ~ 30 です。この値が -1 以外で 1 より小さいか 30 より大きい場合は、BarcodeException.EnumErrorCode.PDF417Col が生成され

ます。

- **Row**: バーコードの行数を指定します。これには、整数値を設定します。デフォルト値は -1 で、値の範囲は 3 ~ 90 です。この値が -1 以外で 3 より小さいか 90 より大きい場合は、BarcodeException.EnumErrorcode.PDF417Row が生成されます。
- **ErrorLevel**: バーコードのエラー修正レベルを指定します。これには、整数値を設定します。値は -1 または値の範囲は 0 ~ 8 です。この値が -1 以外で 0 より小さいか 8 より大きい場合は、BarcodeException.EnumErrorcode.PDF417ErrorLevel が生成されます。
- レベル 0 は強度が低く、デフォルト値は -1 です。
- **Type**: PDF417 バーコードのタイプを選択します。使用可能なタイプは Normal および Simple です。

## QRCodeOptions:

- **Model**: QRCode のモデルを選択します。使用可能なモデルは Model1 および Model2 です。
- **ErrorLevel**: バーコードのエラー修正レベルを選択します。使用可能なオプションは、Low、Medium、Quality、High です。
- **Version**: バーコードのバージョンを指定します。
- **Mask**: バーコードのマスクに使用されるパターンを選択します。QRCode が正しく読み取られるようにするには、マスク処理で明度のバランスを取る必要があります。使用可能なオプションは、Auto、Mask000、Mask001、Mask010、Mask011、Mask100、Mask101、Mask110、Mask111 です。次の図は、QRCode のマスクを示します。



- **Connection**: バーコードにコネクションを使用するかどうかを選択します。値は True または False です。
- **ConnectionNumber**: バーコードのコネクション番号を指定します。0 ~ 15 の範囲の整数値を取ります。この値が 0 より小さいか 15 より大きい場合は、EnumErrorcode.QRCodeConnectionNo 例外が生成されます。
- **Encoding**: バーコードのエンコーディングを選択できます。これには、整数値を設定します。Model プロパティが Model1 に設定されている場合、値は -1 または 1 ~ 14 の範囲です。Model プロパティが Model2 に設定されている場合、値は -1 または 1 ~ 40 の範囲です。

## RssExpandedStackedOptions:

- **RowCount**: 積み重ねる段数を指定します。これには、整数値を設定します。範囲は 1 ~ 11 です。この値が 1 より小さいか 11 より大きい場合は、BarcodeException.EnumErrorcode.RSSExpandedStackedCount が生成されます。

## C1Barcode コントロールをカスタマイズする

プロパティウィンドウは、**C1Barcode** コントロールの外観をカスタマイズするためのさまざまな設計時オプションを提供します。提供されるオプションは、アプリケーションで使用するバーコードのタイプによって異なります。また、PropertyGrid コントロールを使用して、C1Barcode を実行時にカスタマイズすることもできます。実行時に **C1Barcode** コントロールをカスタマイズするには、以下の手順を参照します。このコードでは、[BarCode for WinForms クイックスタート](#) セクションで作成したサンプルを使用します。

1. フォームに **PropertyGrid** コントロールを追加します。
2. 次のコードを **Form\_Load** イベントに追加します。

Visual Basic

```
PropertyGrid1.SelectedObject = C1Barcode1
```

C#

```
propertyGrid1.SelectedObject = c1Barcode1;
```

上のコードは、**C1Barcode** のオプションを表示するための **PropertyGrid** を設定します。

3. プロジェクトを実行します。次の図では、**CodeType** に **QRCode** が設定され、**Text** に URL が設定されています。

Form1

バーコード型: QRCode

テキスト: HELLO WORLD!

バーコード



QuietZone	0;0;0;0
<b>動作</b>	
AllowDrop	False
CaptionGrouping	False
ChecksumEnabled	True
<b>Code Type</b>	QRCode
ContextMenuStrip	(なし)
Enabled	True
ImeMode	NoControl
TabIndex	0
TabStop	True
Visible	True
<b>配置</b>	
Anchor	Top, Left
AutoSize	True
Dock	None
Location	135, 121
Margin	3, 3, 3, 3
MaximumSize	0, 0
MinimumSize	0, 0
Padding	0, 0, 0, 0
Size	151, 150
<b>表示</b>	
BackColor	<input type="checkbox"/> White
BackgroundImage	<input type="checkbox"/> (なし)
BackgroundImageLayout	Tile
Cursor	Default
Font	Microsoft Sans Serif, 12p
ForeColor	<input checked="" type="checkbox"/> Black
RightToLeft	No
Text	HELLO WORLD!
UseWaitCursor	False

**Code Type**  
 バーコード画像を生成するときに使用するエンコードの種類を取得または設定します。