

QRCode for ASP.NET Web Forms

2018.04.11 更新

グレースィティ株式会社

目次

製品の概要	2
ComponentOne for ASP.NET Web Forms のヘルプ	2
構造	3
クイックスタート:QRCodeの追加と生成	4-5
デザイン時のサポート	6
主な機能	7
リサイズ	7-8
色づけ	8-9
エラー修正レベル	9
コードバージョン	9-10
エンコード	10-11
ユーザーシナリオ	12
フォームに入力したデータの格納	12-14
名刺情報の格納	14-16
ロゴの追加	16-17

製品の概要

QRCode for ASP.NET Web Forms を使用して、関連付けられている項目に関する情報を格納できます。QRCode(Quick Response Code)は、白色背景の四角いグリッド内に並べられた黒色の四角い点で構成される2次元のバーコードです。これは、カメラやスマートフォンなどのイメージングデバイスで読み取ることができる機械読み取り式のコードです。バーコードとは異なり、QRCodeは縦横の両方向にデータを格納するため、同じスペースに従来のバーコードの最大100倍の情報を格納できます。

QRCodeコードは、出荷状態の詳細などを追跡するために製品パッケージやラベルに使用したり、ブランディングの目的やユーザーを他のWebサイトにリダイレクトするために広告、新聞、雑誌、Webサイトなどでも広く使用されています。

主要な機能

- **サイズ**: 必要に応じてQRCodeのサイズおよび解像度を変更します。詳細については、「[リサイズ](#)」を参照してください。
- **色**: 色を変更して、ブランドに合わせてカスタマイズされたQRCodeを作成します。詳細については、「[色づけ](#)」を参照してください。
- **エラー修正**: コードが損傷した場合でもデータを復元できるように、最大30%のエラー修正レベルを設定します。詳細については、「[エラー修正レベル](#)」を参照してください。
- **エンコード**: 4つのエンコードオプション(Automatic、Numeric、Alphanumeric、Byte)から選択します。詳細については、「[エンコード](#)」を参照してください。



ComponentOne for ASP.NET Web Forms のヘルプ

ComponentOne for ASP.NET Web Forms の各コントロールで共通したトピック、アセンブリの追加、テーマの適用、クライアント側情報などについては「[ASP.NET Web Forms ユーザーガイド](#)」を参照してください。

構造

次の図に、QRCodeのさまざまな要素を示します。

- ポジションマーカー
- 書式情報
- アラインメントマーカー
- タイミングコード
- バージョン情報
- デッドスペース
- データ



- **ポジションマーカー**: QRCodeの境界を識別するために使用されます。
- **書式情報**: エラー修正レベルやマスク情報など、失われたデータの復元に使用される情報を保持します。
- **アラインメントマーカー**: さまざまな角度からコードをスキャンできるようにします。
- **タイミングコード**: コードのサイズの決定に使用されます。
- **バージョン情報**: コードリーダーが読み取っているコードのタイプを判定するために使用されます。
- **デッドスペース**: データを周囲から分離します。
- **データ**: QRCodeに関連付けられている項目に関する情報を保持します。


クイックスタート: QRCodeの追加と生成

このトピックでは、ASP.NETアプリケーションでC1QRCodeコントロールを使用および構成する方法について説明します。

手順1/2: アプリケーションの作成

ASP.Netアプリケーションを作成し、WebフォームにQRCodeコントロールを追加するには、次の手順を実行します。

1. **Visual Studio**で、新しいASP.Net Webアプリケーションを作成し、新しい**Webフォーム**を追加します。
2. ツールボックスで **C1QRCode** コントロールを見つけ、それをWebフォームに配置します。

 ツールボックスにこのコントロールが見つからない場合は、右クリックし、**[アイテムの選択]**を選択します。**[ツールボックスアイテムの選択]**ダイアログボックスが表示されます。コントロールを見つけたら、**[OK]**をクリックします。

手順2/2: QRCodeの生成

QRCodeを生成するには、次の手順を実行します。

デザイナの場合

1. **スマートタグ**をクリックします。**[C1QRCodeのタスクメニュー]**が表示されます。
2. QRCodeにエンコードするテキストを**Text**プロパティに追加します。
3. アプリケーションを実行します。QRCodeが生成されます。

ソースビューの場合

QRCodeを生成するには、`<cc1:C1QRCode>`タグ内の**Text**プロパティを設定します。

ソースコードビュー

```
<cc1:C1QRCode ID="C1QRCode1" runat="server" Text="QRCode サンプルテキスト" />
```

コードの場合

次のコードを**Page_Load**イベントに追加して、QRCodeを生成するテキストを追加します。

C#

copyCode

```
C1QRCode1.Text = "QRCode サンプルテキスト";
```

VB

copyCode

```
C1QRCode1.Text = "QRCode サンプルテキスト"
```

ここまでの成果

アプリケーションを実行すると、**Text**プロパティに追加したテキストがQRCodeにエンコードされます。



QRCode for ASP.NET Web Forms

スマートフォンなどのデバイスでQRCodeリーダーを使用してこのQRCodeをスキャンすると、"QRCode サンプルテキスト"というテキストがデコードされます。

デザイン時のサポート

C1QRCodeは、カスタマイズされたコンテキストメニュー、スマートタグ、および充実したデザイン時サポートを提供するデザイナーを備えており、オブジェクトモデルの操作が簡素化されています。

以下のセクションでは、C1QRCodeのデザイン時環境を使用してC1QRCodeコントロールを設定する方法について説明します。

C1QRCode タスクメニュー

C1QRCode タスク	
テキスト	<input type="text"/>
エンコード	Automatic ▼
セルサイズ	3
バージョン	0
エラー訂正レベル	L ▼
<input type="checkbox"/> JavaScript の結合を有効	
バージョン情報...	

テキスト: QRCodeにエンコードするテキストを指定します。

エンコード: 次のエンコードオプションがあります。Automatic、Numeric(3数字ごとに10ビット)、Alphanumeric(2文字ごとに11ビット)およびByte(文字ごとに8ビット)。

セルサイズ: 解像度を改善するには、シンボルのサイズをQRCodeのサイズに応じて増減できます。

バージョン: C1QRCodeコントロールは、1から10までのQRコードバージョンを生成できます。最もコンパクトなバージョンを自動的に使用するには、CodeVersionプロパティを0(デフォルト値)に設定します。

エラー訂正レベル: 以下の4つのオプションがあります。


- Low **L**: 最大7%の損傷を修正できます。
- Medium **M**: 最大15%の損傷を修正できます。
- Quartile **Q**: 最大25%の損傷を修正できます。
- High **H**: 最大30%の損傷を修正できます。

バージョン情報: [バージョン情報]をクリックすると、製品のバージョン情報を確認できるダイアログボックスが表示されます。

主な機能

リサイズ

QRCodeは、容易にスキャンできるくらいに大きくすると同時に、あまりスペースを取らずに1ページに収まるように小さくする必要があります。コントロールおよびコントロール内のシンボルのサイズを変更して、コントロールの解像度を向上することができます。コントロールおよびシンボルのサイズを変更するには、次の手順を実行します。

 正しくスキャンするには、QRCodeを四角形にする必要があります。

デザイナーの場合

1. QRCodeのサイズを変更するには、QRCodeの端をクリックし、目的の高さまたは幅になるようにドラッグします。
2. シンボルのサイズを変更するには、**スマートタグ**をクリックします。[C1QRCodeのタスクメニュー]で、**SymbolSize**プロパティを要件に合わせて設定します。
3. または、QRCodeを右クリックし、[プロパティ]を選択して、プロパティウィンドウを開きます。プロパティウィンドウで、**Height**、**Width**、**SymbolSize**の各プロパティを設定します。

ソースビューの場合

<cc1:C1QRCode>タグ内で**Height**、**Width**、**SymbolSize**の各プロパティを設定して、QRCodeのサイズを変更します。

ソースコードビュー

```
<cc1:C1QRCode ID="C1QRCode1" runat="server" Height="150px" Text="QRCode サンプルテキスト"
Width="150px" SymbolSize="5" />
```

コードの場合

次のコードを**Page_Load**イベントに追加し、コードビューで**Height**、**Width**、**SymbolSize**の各プロパティを設定します。

C#

copyCode

```
C1QRCode1.Height = 150;
C1QRCode1.Width = 150;
C1QRCode1.SymbolSize = 5;
```

VB

copyCode

```
C1QRCode1.Height = 150
C1QRCode1.Width = 150
C1QRCode1.SymbolSize = 5
```


ここまでの成果

アプリケーションを実行し、QRCodeとシンボルのサイズが大きくなっていることを確認します。



色づけ

通常、QRCodeは白黒ですが、QRCodeのスキャン性能を損なうことなくコントロールに色を付けることができます。これは、QRCodeをカスタマイズし、ブランディングを追加する最も簡単な方法です。QRCodeには、色のほかにも、ロゴなどの画像を乗せてカスタマイズすることもできます。詳細については、「[ロゴの追加](#)」を参照してください。

 QRCodeの背景を暗い色にし、シンボルを明るい色にすると、正しくスキャンされない可能性があるため、これはお勧めしません。

デザイナーの場合

1. QRCodeを右クリックし、**[プロパティ]**を選択します。プロパティウィンドウが開きます。
2. プロパティウィンドウで、**BackColor**および**ForeColor**プロパティを設定します。

ソースビューの場合

<cc1:C1QRCode>タグ内で**BackColor**および**ForeColor**プロパティを設定して、QRCodeに色を追加します。

ソースコードビュー

```
<cc1:C1QRCode ID="C1QRCode1" runat="server" BackColor="#FFFFCC" ForeColor="#6600CC"
Text="QRCode サンプルテキスト" />
```

コードの場合

次のコードを**Page_Load**イベントに追加し、コードビューで**BackColor**および**ForeColor**プロパティを設定します。

C#	copyCode
<pre>C1QRCode1.BackColor = System.Drawing.ColorTranslator.FromHtml("#FFFFCC"); C1QRCode1.ForeColor = System.Drawing.ColorTranslator.FromHtml("#6600CC");</pre>	

VB	copyCode
<pre>C1QRCode1.BackColor = System.Drawing.ColorTranslator.FromHtml("#FFFFCC") C1QRCode1.ForeColor = System.Drawing.ColorTranslator.FromHtml("#6600CC")</pre>	

ここまでの成果

アプリケーションを実行し、QRCodeが色付きで表示されることを確認します。



エラー修正レベル

これは、データの一部が欠けている、またはQRCodeが損傷している場合などに、**QRCode**から情報を復元するために役立ちます。この機能は、画像やロゴを乗せてカスタマイズしたQRCodeを作成する場合に便利です。QRCodeに画像やロゴを乗せると、一部のコードが損なわれますが、それでも容易にスキャンすることができます。

QRCodeコントロールには4つのエラー修正レベルがあります。

- Low **L**: 7%のエラーを許容します。
- Medium **M**: 15%のエラーを許容します。
- Quartile **Q**: 25%のエラーを許容します。
- High **H**: 30%のエラーを許容します。

QRCodeコントロールのエラー修正レベルを変更するには、次の手順を実行します。

デザイナの場合

1. **スマートタグ**をクリックします。**[C1QRCodeのタスクメニュー]**が表示されます。
2. 要件に合わせて**ErrorCorrectionLevel**コンボボックスからエラー修正レベルを選択します。
3. または、QRCodeを右クリックし、**[プロパティ]**を選択して、プロパティウィンドウを開きます。プロパティウィンドウで、**ErrorCorrectionLevel**プロパティを設定します。

ソースビューの場合

<cc1:C1QRCode>タグ内の**ErrorCorrectionLevel**プロパティを設定して、QRCodeのエラー修正レベルを変更します。

ソースコードビュー

```
<cc1:C1QRCode ID="C1QRCode1" runat="server" ErrorCorrectionLevel="H" />
```

コードの場合

次のコードを**Page_Load**イベントに追加し、コードビューで**ErrorCorrectionLevel**プロパティを設定します。

C#

copyCode

```
C1QRCode1.ErrorCorrectionLevel = C1.Web.Wijmo.Controls.C1QRCode.ErrorCorrectionLevel.H;
```

VB

copyCode

```
C1QRCode1.ErrorCorrectionLevel = C1.Web.Wijmo.Controls.C1QRCode.ErrorCorrectionLevel.H
```

ここまでの成果

エラー修正レベルが、30%のエラーを許容する**H**(High)に設定されました。

コードバージョン

QRCodeコントロールには、0から10までの11個のコードバージョンがあります。コードバージョンごとに、QRCodeを形成する白黒の四角い点の構成が異なります。構成とは、1つのシンボルに含まれる白黒の点の数を意味します。この白黒の点の配

列を「モジュール」と呼びます。バージョン0は高さ幅がともに21のモジュールで構成されます。バージョンが1つ上がるごとに、モジュールが1辺4つずつ増えます。

QRCodeにエンコードされる情報量が多くなるほど、より多くのモジュールが必要となり、QRCodeシンボルも大きくなります。QRCodeコントロールのコードバージョンを変更するには、次の手順を実行します。

デザイナーの場合

1. **スマートタグ**をクリックします。**[C1QRCodeのタスクメニュー]**が表示されます。
2. 要件に合わせて**CodeVersion**プロパティを設定します。
3. または、QRCodeを右クリックし、**[プロパティ]**を選択して、プロパティウィンドウを開きます。プロパティウィンドウで、**CodeVersion**プロパティを設定します。

ソースビューの場合

<cc1:C1QRCode>タグ内の**CodeVersion**プロパティを設定して、QRCodeのコードバージョンを変更します。

ソースコードビュー

```
<cc1:C1QRCode ID="C1QRCode1" runat="server" CodeVersion="8" Text="QRCode サンプルテキスト" />
```

コードの場合

次のコードを**Page_Load**イベントに追加し、コードビューで**CodeVersion**プロパティを設定します。

C#

copyCode

```
C1QRCode1.CodeVersion = 8;
```

VB

copyCode

```
C1QRCode1.CodeVersion = 8
```

ここまでの成果

QRCodeのコードバージョンが8に変更され、モジュールが増えました。



エンコード

QRCodeは、データを効率よく保存するため、4つのエンコードモードで構成されます。

- **Numeric**: 数字の0~9をエンコードできます。
- **AlphaNumeric**: 数字の0~9、大文字のA~Z、およびその他の文字(スペース \$ % * + - . / : ,)をエンコードできます。

QRCode for ASP.NET Web Forms

- **Byte**: デフォルトでは、ISO/IEC 8859-1をエンコードできます。また、定義されているその他の一部の文字セットをエンコードすることもできます。
- **Automatic**: コントロールに追加されたテキストに基づいてエンコードモードを設定します。

英数字モードでは、バイトモードよりメッセージが圧縮されますが、小文字のアルファベットを格納することができず、いくつかの句読点のみを保存することができます。

QRCodeコントロールのエンコードモードを変更するには、次の手順を実行します。

デザイナの場合

1. **スマートタグ**をクリックします。**[C1QRCodeのタスクメニュー]**が表示されます。
2. 要件に合わせて**Encoding**コンボボックスからエンコードモードを選択します。
3. または、QRCodeを右クリックし、**[プロパティ]**を選択して、プロパティウィンドウを開きます。プロパティウィンドウで、**Encoding**プロパティを設定します。

ソースビューの場合

<cc1:C1QRCode>タグ内の**Encoding**プロパティを設定して、QRCodeのエンコードモードを変更します。

ソースコードビュー

```
<cc1:C1QRCode ID="C1QRCode1" runat="server" Encoding="Byte"
    Text="QRCode サンプルテキスト"/>
```

コードの場合

次のコードを**Page_Load**イベントに追加し、コードビューで**Encoding**プロパティを設定します。

C#

copyCode

```
C1QRCode1.Encoding = C1.Web.Wijmo.Controls.C1QRCode.Encoding.Byte;
```

VB

copyCode

```
C1QRCode1.Encoding = C1.Web.Wijmo.Controls.C1QRCode.Encoding.Byte
```

ここまでの成果

エンコードモードがByteに変更され、小文字、数字、および句読点を含む文字をQRCodeにエンコードすることができます。

ユーザーシナリオ

QRCodeは、次のような多くの目的で使用できます。

- ユーザーをWebサイトにリダイレクトする。
- 問い合わせ電話番号や電子メールアドレス全体を入力することなく、電話したり、電子メールを送信する。
- クーポンなどの特典を取得する。
- 方向を確認する。
- 曲を聴いたり、ビデオを視聴する。
- WIFIネットワークに接続する。
- フィードバックフォームなどのフォームを格納する。
- ブランドなどをプロモートする。

以下のトピックでは、QRCodeの使用例をいくつか示します。以下のトピックは、読者がVisual Studio .NETのプログラミングに精通していることを前提としています。

フォームに入力したデータの格納

QRCodeを使用して、ユーザーがフォームに入力したデータを格納できます。これにより、毎回同じ情報を入力しなくても、QRCodeをスキャンするだけで情報を取得できます。このトピックでは、フォームに追加された情報をQRCodeに格納するアプリケーションを作成する方法を示します。

Visual Studioで、新しいASP.Net Webアプリケーションを作成し、新しいWebフォームを追加します。さらに、次の手順を実行します。

デザイナーの場合

1. Webフォームに3つのラベル、2つのテキストボックス、1つのコンボボックス、1つのボタンを配置し、次のようにプロパティを設定します。

コントロール名	プロパティ	値
Label1	Text	名前:
Label2	Text	電話番号:
Label3	Text	性:
ComboboxItem1	Text、 Value	男性
ComboboxItem1	Text、 Value	女性
Button1	ID、 Text	提出


フォームは次の図のように表示されます。

名前:

電話番号:

性:

2. ツールボックスでC1QRCode コントロールを見つけ、それをWebフォームに配置します。

 ツールボックスにこのコントロールが見つからない場合は、右クリックし、[アイテムの選択]を選択します。[ツールボックス アイテムの選択]ダイアログボックスが表示されます。コントロールを見つけたら、[OK]をクリックします。

QRCode for ASP.NET Web Forms

3. プロパティウィンドウを開き、**DisplayVisible**プロパティをfalseに設定します。これにより、[提出]ボタンがクリックされるまで、コントロールは非表示になります。

ソースビューの場合

すべてのコントロールを追加し、すべてのプロパティを設定すると、ソースビューのフォームのコードは次のようになります。

ソースコードビュー

```
<p>
  <asp:Label ID="Label1" runat="server" Text="名前:"></asp:Label>
  <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
</p>
<p>
  <asp:Label ID="Label2" runat="server" Text="電話番号:"></asp:Label>
  <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
</p>
<p>
  <asp:Label ID="Label3" runat="server" Text="性:"></asp:Label>
  <cc1:C1ComboBox ID="C1ComboBox1" runat="server" Width="160px">
    <Items>
<cc1:C1ComboBoxItem ID="C1ComboBoxItem1" runat="server" Text="男性" Value="男性" />
<cc1:C1ComboBoxItem ID="C1ComboBoxItem2" runat="server" Text="女性" Value="女性" />
    </Items>
  </cc1:C1ComboBox>
</p>
<p>
  <asp:Button ID="Submit" runat="server" Text="提出" /><br/>
  <cc1:C1QRCode ID="C1QRCode1" runat="server" DisplayVisible="false" />
</p>
```

コードの場合

ユーザーが情報を入力して[提出]ボタンをクリックしたらQRCodeが生成されるように、[提出]ボタンのクリックイベントに次のコードを追加します。

C#

copyCode

```
protected void Submit_Click(object sender, EventArgs e)
{
    C1QRCode1.Text = Label1.Text + " " + TextBox1.Text + '\n' + Label2.Text +
    " " + TextBox2.Text + '\n' + Label3.Text + " " + C1ComboBox1.SelectedValue;
    C1QRCode1.DisplayVisible = true;
}
```

VB

copyCode

```
Protected Sub Submit_Click(sender As Object, e As EventArgs) Handles Submit.Click
    C1QRCode1.Text = Label1.Text + " " + TextBox1.Text + ControlChars.Lf + Label2.Text +
    " " + TextBox2.Text + ControlChars.Lf + Label3.Text + " " + C1ComboBox1.SelectedValue
    C1QRCode1.DisplayVisible = True
End Sub
```

ここまでの成果

プロジェクトを実行したら、フォームに詳細を入力し、[提出]ボタンをクリックします。次のように、入力した詳細を格納するQRCodeが生成されます。

名前:

電話番号:

性:



上のQRCodeをリーダーでスキャンすると、次のデータがデコードされます。

名前: 小林直子
 電話番号: 048-2**-****
 性: 男性

名刺情報の格納

QRCodeの使用方法として、名刺の情報を格納することができます。これを利用すると、アプリケーションユーザーが名刺の連絡先詳細を手動で入力したり保存する必要がなくなります。スマートフォンを使用してコードをスキャンするだけで、連絡先詳細を取得できるようになります。

Visual Studioで、新しいASP.Net Webアプリケーションを作成し、新しいWebフォームを追加します。さらに、次の手順を実行します。

デザイナーの場合

1. Webフォームに6つのラベルを配置し、次のようにプロパティを設定します。

コントロール名	プロパティ	値
Label1	Text	小林直子
Label2	Text	GrapeCity Japan
Label3	Text	住所: 〒332-0012 埼玉県川口市本町4-1-8 川口センタービル3F
Label4	Text	電話番号:048-2**-****
Label5	Text	FAX 番号:048-2**-****
Label6	Text	メールアドレス: naoko@grapecity.com

次のようなフォームが表示されます。

小林直子

GrapeCity Japan

住所: 〒332-0012 埼玉県川口市本町4-1-8 川口センタービル 3F

電話番号: 048-2**-****

FAX番号: 048-2**-****

メールアドレス: naoko@grapecity.com

2. ツールボックスでC1QRCode コントロールを見つけ、それをWebフォームに配置します。

ツールボックスにこのコントロールが見つからない場合は、右クリックし、[アイテムの選択]を選択します。[ツールボックスア

QRCode for ASP.NET Web Forms

アイテムの選択ダイアログボックスが表示されます。コントロールを見つけたら、**[OK]**をクリックします。

ソースビューの場合

すべてのコントロールを追加し、プロパティを設定すると、ソースビューのフォームのコードは次のようになります。

ソースコードビュー

```
<b> <asp:Label ID="Label1" runat="server" Text="小林直子"></asp:Label></b>
<p>
  <asp:Label ID="Label2" runat="server" Text="GrapeCity Japan"></asp:Label>
  <asp:Label ID="Label3" runat="server" Text="住所:
    〒332-0012 埼玉県川口市本町4-1-8川口センタービル 3F">
  </asp:Label>
  <asp:Label ID="Label4" runat="server" Text=" 電話番号: 048-2**-****">
  </asp:Label>
  <asp:Label ID="Label5" runat="server" Text=" FAX 番号: 048-2**-****">
  </asp:Label>
  <asp:Label ID="Label6" runat="server" Text="メールアドレス: naoko@grapecity.com">
  </asp:Label>
</p>
<cc1:C1QRCode ID="C1QRCode1" runat="server" />
```

コードの場合

次のコードを**Page_Load**イベントに追加して、フォームに追加されたラベルからQRCodeを生成します。

C#	copyCode
<pre>C1QRCode1.Text = Label1.Text + '\n' + Label2.Text + '\n' + Label3.Text + '\n' + Label4.Text + '\n' + Label5.Text + '\n' + Label6.Text;</pre>	

VB	copyCode
<pre>C1QRCode1.Text = Label1.Text + ControlChars.Lf + Label2.Text + ControlChars.Lf + Label3.Text + ControlChars.Lf + Label4.Text + ControlChars.Lf + Label5.Text + ControlChars.Lf + Label6.Text</pre>	

ここまでの成果

プロジェクトを実行すると、次のように、名刺の詳細を格納するQRCodeが生成されます。

小林直子

GrapeCity Japan

住所: 〒332-0012 埼玉県川口市本町4-1-8 川口センタービル 3F

電話番号: 048-2**-****

FAX番号: 048-2**-****

メールアドレス: naoko@grapecity.com



上のQRCodeをリーダーでスキャンすると、次の情報がデコードされます。

小林直子

GrapeCity Japan

住所: 〒332-0012 埼玉県川口市本町4-1-8 川口センタービル3F

電話番号: 048-2**-****

FAX 番号: 048-2**-****

メールアドレス: naoko@grapecity.com


ロゴの追加

QRCodeでは、エラー修正レベルをH(High)に設定すると、30%のエラーを許容できます。これにより、QRCodeのスキャン性能を損なうことなく、QRCodeの上に画像を乗せてQRCodeをカスタマイズできます。ただし、QRCodeのサイズの1/3を超える大きさの画像を乗せることはお勧めしません。

Visual Studioで、新しいASP.Net Webアプリケーションを作成し、新しいWebフォームを追加します。次に、QRCodeにロゴを乗せるために、次の手順を実行します。

デザイナーの場合

1. ツールボックスで **C1QRCode** および **C1BinaryImage** コントロールを見つけ、それをWebフォームに配置します。

 ツールボックスにこれらのコントロールが見つからない場合は、右クリックし、**[アイテムの選択]**を選択します。**[ツールボックスアイテムの選択]**ダイアログボックスが表示されます。コントロールを見つけたら、**[OK]**をクリックします。
2. BinaryImageコントロールに**ImageURL**を追加して、コントロールに画像を追加します。
3. 次の表に示すように、コントロールのプロパティを設定します。

コントロール名	プロパティ	値
C1QRCode1	Text	QRCode サンプルテキスト
C1QRCode1	ErrorCorrectionLevel	H
C1QRCode1	ForeColor	#CC0000
C1QRCode1	Height、Width	150px
C1BinaryImage1	Height	85px
C1BinaryImage1	Width	100px

ソースビューの場合

1. 両方のコントロールを追加し、すべてのプロパティを設定すると、ソースビューのフォームのコードは次のようになります。

ソースコードビュー

```
<cc1:C1QRCode id="C1QRCode1" runat="server" Text="QRCode サンプルテキスト"
Height="150px" Width="150px" ErrorCorrectionLevel="H" ForeColor="#CC0000" />
<c1:C1BinaryImage id="C1BinaryImage1" runat="server" Height="85px"
ImageUrl="~/C1.png" Width="100px" />
```

2. <head></head> タグ内に次のコードを追加して、QRCodeとBinaryImage1に適用するCSSクラスを作成し、画像とQRCodeを重ねます。

ソースコードビュー

```
<style type="text/css">
```

QRCode for ASP.NET Web Forms

```
.qrcodeA1 { position:absolute; top: -1px; left: 4px; z-index: 1; }  
.imgB1 { position:absolute; top: 35px; left: 25px; z-index: 3; right: 759px; }  
</style>
```

3. **CssClass** それぞれのコントロールのタグ内で、次のようにプロパティを設定します。

ソースコードビュー

```
<c1:C1QRCode id="C1QRCode1" CssClass="qrcodeA1" runat="server" Text="QRCode サンプルテキスト" Height="150px" Width="150px" ErrorCorrectionLevel="H" ForeColor="#CC0000" />  
<c1:C1BinaryImage id="C1BinaryImage1" CssClass="imgB1" runat="server" Height="85px" ImageUrl="~/C1.png" Width="100px" />
```

ここまでの成果

プロジェクトを実行すると、次の図に示すように、ロゴを乗せてカスタマイズされたQRCodeが生成されます。

