

# BinaryImage for ASP.NET Web Forms

2018.04.12 更新

グレースィティ株式会社

## 目次

<a href="#">製品の概要</a>	2
<a href="#">ComponentOne for ASP.NET Web Forms のヘルプ</a>	2
<a href="#">クイックスタート: 画像のロードと設定</a>	3-5
<a href="#">主な機能</a>	6
<a href="#">外部画像の追加</a>	6-7
<a href="#">Http ハンドラ</a>	7
<a href="#">画像の設定</a>	7
<a href="#">サイズ変更モード</a>	7-9
<a href="#">画像の配置</a>	9-10
<a href="#">代替テキスト</a>	10-11
<a href="#">ツールチップ</a>	11-12
<a href="#">画像の保存名</a>	12-13
<a href="#">ユーザーシナリオ</a>	14
<a href="#">DataList への BinaryImage の統合</a>	14-15
<a href="#">GridView への BinaryImage の統合</a>	15-16

## 製品の概要

**BinaryImage** for ASP.NET は、データベースにバイナリデータとして格納されている画像を表示する機能を提供します。また、データソースのバイナリ画像フィールドから、データ連結をサポートするコントロール (C1GridView など) に画像を追加するために使用することもできます。

### 主要な機能

- **画像のサイズ変更**: 画像をサイズ変更したり、**サイズ変更モード**を **Fit**(自動調整)、**Fill**(フィル)、または **Crop**(トリミング)に設定できます。
- **代替テキスト**: 画像を使用できない場合に表示される代替テキストを指定できます。
- **画像の配置**: フォームに追加されている他のコントロールを基準にしてバイナリ画像の配置を設定できます。
- **Http ハンドラ**: BinaryImage コントロールが置かれているフォームにソースから画像をストリーム送信する **Http ハンドラ**を作成できます。

## 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。

## クイックスタート: 画像のロードと設定

ここでは、次の手順を実行して、**BinaryImage** コントロールで画像をロードして設定する方法を学びます。

以下の手順は、Visual Studio 2012 で作成されたアプリケーション用です。使用する Visual Studio のバージョンによっては、手順が多少異なる場合があります。

### 手順1/3: Web フォームへの BinaryImage の追加



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 BinaryImage コントロールを見つけ、Web フォームにドラッグします。ツールボックスでコントロールが見つからない場合は、右クリックして **[項目の選択]** を選択し、**[ツールボックス項目の選択]** ダイアログボックスで BinaryImage を見つけます。


### トピックの内容

- 手順1/3: Web フォームへの BinaryImage の追加
- 手順2/3: コントロールへの画像の追加
- 手順3/3: 画像の設定

### 手順2/3: コントロールへの画像の追加

この手順では、ロードする画像が埋め込みリソースとしてプロジェクトに含まれていることを前提としています。

1. BinaryImage コントロールで、スマートタグ  をクリックして **[BinaryImage Tasks]** メニューを開きます。
2. **ImageUrl** プロパティの横にあるボタン  をクリックします。 **[イメージの選択]** ダイアログボックスが表示されます。
3. **[イメージの選択]** ダイアログボックスで、ロードする画像を選択し、**[OK]** をクリックします。

 **ヒント:** ImageUrl フィールドで URL を指定して画像をロードすることもできます。詳細については、「[外部画像の追加](#)」を参照してください。

#### ソースビューの場合

**ImageUrl** プロパティを `<cc1:C1BinaryImage>` タグ内で設定して、BinaryImage コントロールに画像を追加します。

```
<cc1:C1BinaryImage ID="C1BinaryImage1" runat="server" ImageUrl="~/C1.png" />
```

#### コードの場合

次のコードを **Page\_Load** イベントに追加して、BinaryImage コントロールに画像を追加します。

##### C# でコードを書く場合

```
C#  
C1BinaryImage1.ImageUrl = "C1.png";
```

##### Visual Basic でコードを書く場合

```
Visual Basic  
C1BinaryImage1.ImageUrl = "C1.png"
```


#### ここまでの成果

プロジェクトを実行し、画像が元のサイズで表示されることを確認します。



[先頭に戻る](#)

## 手順3/3: 画像の設定

1. BinaryImage コントロールで、スマートタグ  をクリックして **[BinaryImage Tasks]** メニューを開きます。
2. **[高さ]** を "300px" に、**[幅]** を "400px" に設定します。
3. **[サイズ変更モード]** を "Fit" に設定します。

### ソースビューの場合

**Height** (高さ)、**Width** (幅)、および **SizeMode** (サイズ変更モード) を `<cc1:C1BinaryImage>` タグ内で設定して、BinaryImage コントロール内の画像の設定を行います。

```
<cc1:C1BinaryImage ID="C1BinaryImage1" runat="server"
ImageUrl="~/C1.png" Height="100px" ResizeMode="Fit" Width="100px" />
```

### コードの場合

次のコードを **Page\_Load** イベントに追加して、BinaryImage コントロールを設定します。

#### C# でコードを書く場合

```
C#
C1BinaryImage1.Height = 300;
C1BinaryImage1.Width = 400;
C1BinaryImage1.ResizeMode = C1.Web.Wijmo.Controls.C1BinaryImage.ImageSizeMode.Fit;
```

#### Visual Basic でコードを書く場合

```
Visual Basic
C1BinaryImage1.Height = 300
C1BinaryImage1.Width = 400
C1BinaryImage1.ResizeMode = C1.Web.Wijmo.Controls.C1BinaryImage.ImageSizeMode.Fit
```

### ここまでの成果

# BinaryImage for ASP.NET Web Forms

プロジェクトを実行すると、画像が希望の高さと幅にサイズ変更されて、Web ページに収まります。



[先頭に戻る](#)



## 主な機能

このトピックでC1BinaryImageの主な機能について説明します。

## 外部画像の追加

BinaryImage コントロールに外部画像を追加するには、次の手順に従います。

### デザイナーの場合

1. BinaryImage コントロールを選択し、スマートタグ  をクリックして [BinaryImage のタスク] メニューを開きます。
2. ImageUrl の横にあるボタン  をクリックします。
3. ImageUrl プロパティに、ロードする画像の URL を設定し、[OK] をクリックします。

### ソースビューの場合

<cc1:C1BinaryImage> タグ内の ImageUrl プロパティに画像の URL を設定して、BinaryImage コントロールに外部画像を追加します。

```
<cc1:C1BinaryImage ID="C1BinaryImage1" runat="server"
ImageUrl="http://www.componentone.com/newimages/Products/Visuals/se_theming.png"/>
```

### コードの場合

次のコードを Page\_Load イベントに追加して、BinaryImage コントロールに外部画像を追加します。

#### C# でコードを書く場合

C#

```
C1BinaryImage1.ImageUrl = "http://www.componentone.com/newimages/Products/Visuals/se_theming.png";
```

#### Visual Basic でコードを書く場合

Visual Basic

```
C1BinaryImage1.ImageUrl = "http://www.componentone.com/newimages/Products/Visuals/se_theming.png"
```

### ここまでの成果

プロジェクトを実行すると、BinaryImage コントロールに画像が表示されます。



## Http ハンドラ

BinaryImage コントロールが置かれている Web フォームにソースから画像をストリーム送信する **Http ハンドラ**を作成します。

**HttpHandlerUrl** プロパティを使用して、デフォルトの **BinaryImage Http ハンドラ**を拡張したカスタム **Http ハンドラ**の場所を指定します。これを設定しない場合、BinaryImage は独自のハンドラを持ち、これが汎用の **C1.Web.UI.WebResource.axd** ハンドラから呼び出されます。

次のコードが *web.config* ファイルに生成されます。

```
<configuration>
  <system.web>
    <httpHandlers>
      <add path="WijmoControlsResource.axd" verb="*"
        type="C1.Web.Wijmo.Controls.WijmoHttpHandler, C1.Web.Wijmo.Controls.4" validate="false" />
      <add path="C1BinaryImageResource.axd"
        type="C1.Web.Wijmo.Controls.C1BinaryImage.C1BinaryImageHandler" verb="*" validate="false" />
    </httpHandlers>
    <compilation debug="true" targetFramework="4.5.1">
      <assemblies>
        <add assembly="System.Windows.Forms, Version=4.0.0.0, Culture=neutral,
          PublicKeyToken=B77A5C561934E089" />
        <add assembly="System.Design, Version=4.0.0.0, Culture=neutral,
          PublicKeyToken=B03F5F7F11D50A3A" />
      </assemblies>
    </compilation>
    <httpRuntime targetFramework="4.5.1" />
  </system.web>
  <system.webServer>
    <handlers>
      <add name="WijmoControlsHttpHandler" verb="*"
        precondition="integratedMode" path="WijmoControlsResource.axd"
        type="C1.Web.Wijmo.Controls.WijmoHttpHandler, C1.Web.Wijmo.Controls.4" />
      <add name="C1BinaryImageResource" verb="*"
        precondition="integratedMode" path="C1BinaryImageResource.axd"
        type="C1.Web.Wijmo.Controls.C1BinaryImage.C1BinaryImageHandler" />
    </handlers>
    <validation validateIntegratedModeConfiguration="false" />
  </system.webServer>
</configuration>
```

## 画像の設定

## サイズ変更モード




**ResizeMode** プロパティは、**高さ**と**幅**がピクセル単位で設定されている場合に、**バイナリ画像**を自動的にサイズ変更するかどうかを決定します。BinaryImage は次の3つのサイズ変更モードをサポートします。

- **Fit**(自動調整): 画像の元のアスペクト比を変えずに、設定された高さと幅に合わせて画像のサイズを変更します。
- **Fill**(フィル): 画像を設定された高さと幅に引き伸ばします。
- **Crop**(トリミング): 設定された高さと幅を越えた画像領域を切り取ります。

このトピックでは、サイズ変更モードを切り替える方法、および出力結果に各モードがどのように表示されるかについて説明します。

## デザイナーの場合

1. BinaryImage コントロールを選択し、スマートタグ  をクリックして **[BinaryImage のタスク]**メニューを開きます。
2. **[高さ]**を "100px" に、**[幅]**を "100px" に設定します。
3. **[サイズ変更モード]**を "Fit" に設定します。

## ソースビューの場合

**Height**(高さ)、**Width**(幅)、および **ReziseMode**(サイズ変更モード)を <cc1:C1BinaryImage> タグ内で設定して、BinaryImage コントロールのサイズを変更します。

```
<cc1:C1BinaryImage ID="C1BinaryImage1" runat="server"
ImageUrl="~/Koala.png" Height="100px" ResizeMode="Fit" Width="100px" />
```

## コードの場合

次のコードを **Page\_Load** イベントに追加して、BinaryImage コントロールのサイズを変更します。

### C# でコードを書く場合

```
C#
C1BinaryImage1.Height = 100;
C1BinaryImage1.Width = 100;
C1BinaryImage1.ResizeMode = C1.Web.Wijmo.Controls.C1BinaryImage.ImageResizeMode.Fit;
```

### Visual Basic でコードを書く場合

```
Visual Basic
C1BinaryImage1.Height = 100
C1BinaryImage1.Width = 100
C1BinaryImage1.ResizeMode = C1.Web.Wijmo.Controls.C1BinaryImage.ImageResizeMode.Fit
```

## ここまでの成果

プロジェクトを実行すると、**サイズ変更モード**に応じて画像のサイズが変更されます。

### 元の画像



次の図は、**Fit** モードで画像がどのように表示されるかを示しています。



次の図は、**Fill** モードで画像がどのように表示されるかを示しています。



次の図は、**Crop** モードで画像がどのように表示されるかを示しています。



## 画像の配置


**BinaryImage** コントロールの **ImageAlign** プロパティを変更して、Web フォーム上の他の要素との関係に基づいてコントロールの配置を変更できます。次の配置オプションがあります。

配置	説明
<b>Left (左)</b>	バイナリ画像をコンテナの左側に配置し、右側でテキストを折り返します。
<b>Right (右)</b>	バイナリ画像をコンテナの右側に配置し、左側でテキストを折り返します。
<b>Baseline (ベースライン)</b>	バイナリ画像の下端をコンテナ内のテキストの1行目の下端と揃えます。
<b>Top (上)</b>	バイナリ画像の上端を、コンテナ内の同じ行にある最も高い要素の上端に揃えます。
<b>Middle (上下中央)</b>	バイナリ画像の中央をコンテナ内のテキストの1行目の下端と揃えます。
<b>Bottom (下)</b>	バイナリ画像の下端をコンテナ内のテキストの1行目の下端と揃えます。

<b>AbsBottom (全体の下)</b>	バイナリ画像の下端を、コンテナ内の同じ行にある最も大きい要素の下端に揃えます。
<b>AbsMiddle (全体の中央)</b>	バイナリ画像の中心をコンテナ内の同じ行にある最も大きな要素の中心に揃えます。
<b>TextTop (テキストの上)</b>	バイナリ画像の上端を、コンテナ内の同じ行にある最も高いテキストの上端に揃えます。

### デザイナーの場合

BinaryImage コントロールの相対的な配置を変更するには、次の手順に従います。

1. BinaryImage コントロールを選択し、スマートタグ  をクリックして **[BinaryImage のタスク]** メニューを開きます。
2. **ImageAlign** プロパティを上表に示したオプションのいずれかに設定します。

### ソースビューの場合

**ImageAlign** プロパティを `<cc1:C1BinaryImage>` タグ内で設定して、BinaryImage コントロールの相対的な配置を変更します。

```
<cc1:C1BinaryImage ID="C1BinaryImage1" runat="server"
ImageUrl="~/C1.png" ImageAlign="Right" />
```

### コードの場合

次のコードを **Page\_Load** イベントに追加して、BinaryImage コントロールの相対的な配置を変更します。

#### C# でコードを書く場合

```
C#
C1BinaryImage1.ImageAlign=ImageAlign.Right;
```

#### Visual Basic でコードを書く場合


```
Visual Basic
C1BinaryImage1.ImageAlign=ImageAlign.Right
```

## 代替テキスト

このトピックでは、**BinaryImage** コントロールで画像を使用できない場合に表示される代替テキストを追加する方法を示します。

### デザイナーの場合

代替テキストを BinaryImage コントロールに追加するには、次の手順を実行します。

1. BinaryImage コントロールを選択し、BinaryImage スマートタグ  をクリックして **[BinaryImage のタスク]** メニューを開きます。
2. 画像の代わりに表示する **代替テキスト** を設定します。

### ソースビューの場合

**AlternateText** プロパティを `<cc1:C1BinaryImage>` タグ内で設定して、BinaryImage コントロールの代替テキストを設定

# BinaryImage for ASP.NET Web Forms

します。

```
<ccl:C1BinaryImage ID="C1BinaryImage1" runat="server" AlternateText="この画像は表示できません" ImageUrl="~/C1.png" />
```

## コードの場合

次のコードを **Page\_Load** イベントに追加して、BinaryImage コントロールに代替テキストを追加します。

### C# でコードを書く場合

```
C#  
C1BinaryImage1.AlternateText = "この画像は表示できません";
```

### Visual Basic でコードを書く場合

```
Visual Basic  
C1BinaryImage1.AlternateText = "この画像は表示できません"
```

## ここまでの成果

画像を使用できない場合に代替テキストが表示されます。




## ツールチップ

このトピックでは、バイナリ画像の上にマウスポインタを合わせると表示されるツールチップテキストを **BinaryImage** コントロールに追加する方法について説明します。

### デザイナーの場合

ツールチップテキストを BinaryImage コントロールに追加するには、次の手順を実行します。

1. BinaryImage コントロールを選択し、スマートタグ  をクリックして **[BinaryImage のタスク]** メニューを開きます。
2. **[ツールチップ]** フィールドでテキストを設定します。

### ソースビューの場合

**ToolTip** プロパティを <cc1:C1BinaryImage> タグ内で設定して、BinaryImage コントロールにツールチップテキストを追加します。

```
<cc1:C1BinaryImage ID="C1BinaryImage1" runat="server" ImageUrl="~/C1.png"  
ToolTip="BinaryImage" />
```

## コードの場合

次のコードを **Page\_Load** イベントに追加して、BinaryImage コントロールにツールチップテキストを追加します。

### C# でコードを書く場合

```
C#  
C1BinaryImage1.ToolTip = "BinaryImage";
```

### Visual Basic でコードを書く場合

```
Visual Basic  
C1BinaryImage1.ToolTip = "BinaryImage"
```

## ここまでの成果

画像の上にマウスポインタを合わせると、ツールチップテキストが表示されます。



## 画像の保存名

画像がダウンロードされたときに画像の保存に使用される名前を割り当てることができます。

### デザイナーの場合

**BinaryImage** コントロールに名前を割り当てるには、次の手順に従います。

1. BinaryImage を右クリックし、**[プロパティ]**を選択します。
2. 表示される**[プロパティ]**ウィンドウで、**SavedImageName** プロパティを設定します。

### ソースビューの場合

**SavedImageName** プロパティを <cc1:C1BinaryImage> タグ内で設定して、BinaryImage コントロールに画像の名前を

# BinaryImage for ASP.NET Web Forms

追加します。

```
<cc1:C1BinaryImage ID="C1BinaryImage1" runat="server" ImageUrl="~/C1.png"
SavedImageName="ComponentOne" />
```

## コードの場合

次のコードを **Page\_Load** イベントに追加して、BinaryImage コントロールに画像の名前を設定します。

### C# でコードを書く場合

```
C#
C1BinaryImage1.SavedImageName= "ComponentOne";
```

### Visual Basic でコードを書く場合

```
Visual Basic
C1BinaryImage1.SavedImageName= "ComponentOne"
```

## ここまでの成果

プロジェクトを実行し、画像を右クリックして[画像を別名で保存]を選択すると、[画像を保存]ダイアログボックスに画像の名前として「ComponentOne」が表示されます。

## ユーザーシナリオ


このトピックは、C1BinaryImageのユーザーシナリオを説明します。

## DataList への BinaryImage の統合

このトピックでは、BinaryImage コントロールを DataList コントロールに統合して、バイナリ画像として保存されている画像をデータ連結シナリオで表示する方法について説明します。データベースにある画像を BinaryImage コントロールに追加するには、**ImageData** プロパティを使用します。

### デザイナーの場合

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 DataList コントロールを見つけ、Web フォームにドラッグします。
3. DataList スマートタグ  をクリックし、**[データソースの選択]** ドロップダウンリストから [**新しいデータソース...**] を選択します。これで、**データソース構成ウィザード** が開きます。SQL データベースを選択し、**[OK]** をクリックします。これで、**[データソースの構成]** ダイアログボックスが開きます。
4. **[データソースの構成]** ダイアログボックスで、**[新しい接続]** をクリックします。これで、**[接続の追加]** ダイアログボックスが開きます。データソースを Microsoft Access データベースファイル (OLE DB) に変更し、データベースファイル名として C1NWind を選択します。**[OK]** をクリックします。
5. **[次へ]** をクリックします。接続を保存し、**[次へ]** をクリックします。DataList に追加する列を指定し、**[次へ]** をクリックします。クエリーをテストし、**[完了]** をクリックします。

### ソースビューの場合

`<ItemTemplate></ItemTemplate>` タグ内の `<asp:Label ID="PictureLabel" runat="server" Text='<# Eval("Picture") %>' />` タグを次のマークアップに置き換えます。

```
<c1:C1BinaryImage ID="BinaryImage1" runat="server" ImageData='<# Eval("Picture") %>' />
```

`<asp:DataList>` の最終的なマークアップは次のようになります。

```
<asp:DataList ID="DataList1" runat="server" DataKeyField="CategoryID"
  DataSourceID="SqlDataSource1" RepeatDirection="Horizontal" RepeatColumns="4" Width="100%">
  <ItemTemplate>
    Picture:<br />
    <c1:C1BinaryImage ID="BinaryImage1" runat="server" ImageData='<# Eval("Picture") %>' />
    <br />
    Description:
    <asp:Label ID="DescriptionLabel" runat="server" Text='<# Eval("Description") %>' />
    <br />
    CategoryName:
    <asp:Label ID="CategoryNameLabel" runat="server" Text='<# Eval("CategoryName") %>' />
    <br />
    CategoryID:
    <asp:Label ID="CategoryIDLabel" runat="server" Text='<# Eval("CategoryID") %>' />
  </ItemTemplate>
</asp:DataList>
```

### ここまでの成果

次の図に示すように、DataList コントロールに統合された BinaryImage コントロールに画像が表示されます。



# BinaryImage for ASP.NET Web Forms

Picture:



Description: Soft drinks, coffees, teas, beers, and ales

CategoryName: Beverages

CategoryID: 1

Picture:



Description: Breads, crackers, pasta, and cereal

CategoryName: Grains/Cereals

CategoryID: 5

Picture:



Description: Sweet and savory sauces, relishes, spreads, and seasonings

CategoryName: Condiments

CategoryID: 2

Picture:



Description: Prepared meats

CategoryName: Meat/Poultry

CategoryID: 6

Picture:



Description: Desserts, candies, and sweet breads

CategoryName: Confections

CategoryID: 3

Picture:

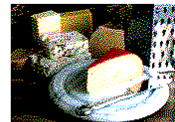


Description: Dried fruit and bean curd

CategoryName: Produce

CategoryID: 7

Picture:



Description: Cheeses

CategoryName: Dairy

Products

CategoryID: 4

Picture:



Description: Seaweed and fish

CategoryName: Seafood


CategoryID: 8

## GridView への BinaryImage の統合

BinaryImage コントロールを GridView コントロール内のテンプレートフィールドに追加することで、バイナリ画像をグリッド内に表示することができます。このトピックでは、BinaryImage コントロールを GridView コントロールに統合して、バイナリ画像として保存されている画像をデータ連結シナリオで表示する方法について説明します。データベースにある画像を BinaryImage コントロールに追加するには、**ImageData** プロパティを使用します。

### デザイナの場合

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 C1GridView コントロールを見つけ、Web フォームにドラッグします。
3. C1GridView スマートタグ  をクリックし、**[データソースの選択]** ドロップダウンリストから [**新しいデータソース...**] を選択します。これで、**データソース構成ウィザード** が開きます。SQL データベースを選択し、**[OK]** をクリックします。これで、**[データソースの構成]** ダイアログボックスが開きます。
4. **[データソースの構成]** ダイアログボックスで、**[新しい接続]** をクリックします。これで、**[接続の追加]** ダイアログボックスが開きます。データソースを Microsoft Access データベースファイル (OLE DB) に変更し、データベースファイル名として C1NWind を選択します。**[OK]** をクリックします。
5. **[次へ]** をクリックします。接続を保存し、**[次へ]** をクリックします。C1GridView に追加する列を指定し、**[次へ]** をクリックします。クエリーをテストし、**[完了]** をクリックします。

### ソースビューの場合

```
<Columns></Columns> タグ内の <cc1:C1BoundField DataField="Picture" HeaderText="Picture" SortExpression="Picture">
```

```
</cc1:C1BoundField> タグを次の <cc1:C1TemplateField> マークアップと置き換えます。
```

```
<cc1:C1TemplateField HeaderText="Picture">
  <ItemTemplate>
    <cc1:C1BinaryImage ID="C1BinaryImage1" runat="server"
      ImageData='<%# Eval("Picture") %>' />
  </ItemTemplate>
</cc1:C1TemplateField>
```

<cc1:C1GridView> の最終的なマークアップは次のようになります。



```
<cc1:C1GridView ID="C1GridView1" runat="server" AutogenerateColumns="False"
  DataKeyNames="CategoryID" DataSourceID="SqlDataSource1">
  <Columns>
    <cc1:C1BoundField DataField="CategoryID" HeaderText="CategoryID"
      ReadOnly="True" SortExpression="CategoryID">
    </cc1:C1BoundField>
```



```
<cc1:C1BoundField DataField="CategoryName" HeaderText="CategoryName"
  SortExpression="CategoryName">
</cc1:C1BoundField>
<cc1:C1BoundField DataField="Description" HeaderText="Description"
  SortExpression="Description">
</cc1:C1BoundField>
<cc1:C1TemplateField HeaderText="Picture">
  <ItemTemplate>
    <cc1:C1BinaryImage ID="C1BinaryImage1" runat="server"
      ImageData='<%# Eval("Picture") %>' />
  </ItemTemplate>
</cc1:C1TemplateField>
</Columns>
</cc1:C1GridView>
```

## ここまでの成果

次の図に示すように、C1GridView コントロールに統合された BinaryImage コントロールに画像が表示されます。

CategoryID	CategoryName	Description	Picture
1	Beverages	Soft drinks, coffees, teas, beers, and ales	
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	
3	Confections	Desserts, candies, and sweet breads	