

Gallery for ASP.NET Web Forms

2018.04.25 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
主な特長	3
クイックスタート	4
手順 1: ページへのコントロールの追加	4
手順 2: コントロールへのコンテンツの追加	4-5
手順 3: コントロールの外観と動作のカスタマイズ	5-6
デザイン時のサポート	7
C1Gallery スマートタグ	7-8
C1Gallery コンテキストメニュー	8
C1Gallery デザイナフォーム	8-9
C1Gallery の外観	10
タスク別ヘルプ	11
ギャラリーの向きの設定	11
ギャラリーへのページの追加	11-12
ギャラリー項目の自動再生	12-13
オンデマンドでのギャラリー項目のロード	13
実行時の遷移アニメーションの変更	13-15
ギャラリーのデータバインディング	15-19

製品の概要

Gallery for ASP.NET Web Forms でウェブサイトにも動的な画像ギャラリーを追加できます。画像とともにサムネイルを表示でき、スライドショー、遷移効果、ページング、テーマ等の機能が対応されています。

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。

主な特長

C1Gallery は、いくつかの独特な特長を備えています。

- **オートプレイ**
選択すると、コントロールは自動的に画像を繰り返し表示します。各画像を表示する時刻と〈再生〉/〈一時停止〉ボタンを備えたタイマーをギャラリー上部に表示するかどうかを設定できます。
- **コントロール**
ナビゲーションボタン、カウンタ、タイマーなど、コントロールに含まれる要素の表示状態を指定できます。
- **サムネイル画像**
ギャラリー内の大きな画像に移動するサムネイル画像を表示し、カスタマイズできます。サムネイル画像をクリックすると、フレームに大きな画像が表示されます。サムネイル画像の表示状態、表示番号、向き、および方向を指定できます。
- **ページング**
サムネイル画像ではなくコントロールにページャ要素を表示するように選択できます。ページャでは、ギャラリー内の画像のナビゲートが可能になり、ページャの位置をカスタマイズできます。
- **ナビゲーション**
コントロールには、ユーザーが実行時に次または前の画像に移動できる2つの組み込みボタン(〈次へ〉/〈前へ〉)が用意されています。
- **動的コンテンツ**
サーバー側または AJAX データからコンテンツデータを非同期にロードし、コントロールのコンテンツをカスタマイズできます。
- **アニメーション**
コントロールがある画像から次の画像に移動するときに表示されるアニメーション効果を選択したり、カスタマイズしたりできます。
- **マルチメディア**
ギャラリーの表示モードを設定し、フレームに表示するメディアのタイプを特定できます。コントロールには、image、SWF、FLV、および iFrame の4つのモードが含まれています。
- **テーマ**
スマートタグをクリックするだけで、6種類のプレミアムテーマ(Arctic、Midnight、Aristo、Rocket、Cobalt、および Sterling)のいずれかを選択して C1Gallery の外観を変更します。オプションとして、jQuery UI からThemeRoller を使用してカスタマイズしたテーマを作成します。テーマの詳細については、[C1Gallery の外観](#)のトピックを参照してください。

クイックスタート

このクイックスタートでは、**C1Gallery** コントロールの機能について学びます。**C1Gallery** は、独自のビューでメディアを追加する機能を提供します。

手順 1: ページへのコントロールの追加

この手順では、Web サイトを作成して設定し、空の **C1Gallery** コントロールを追加します。**C1Gallery** コントロールを Web サイトに追加するには、以下の手順を実行します。

1. まず、ASP.NET Web アプリケーションを作成します。
2. コントロールをツールボックスに追加します。
3. 「**デザイン**」タブを選択します。
4. デザインビューで、Visual Studio ツールボックスに移動し、**C1Gallery** コントロールをダブルクリックしてページのメインコンテンツにコンボボックスを追加します。

この手順では、**C1Gallery** コントロールをフォームに追加しましたが、コントロールは現在空です。クイックスタートの次の手順では、コントロールにコンテンツを追加します。

手順 2: コントロールへのコンテンツの追加

C1Gallery コントロールへのコンテンツの追加は、とても簡単であり、さまざまな方法で行えます。この手順では、**ImageUrl** プロパティを使って画像にリンクします。以下の手順では、「[手順 1: ページへのコントロールの追加](#)」トピックを完了し、**C1Gallery** コントロールがページに追加されていることを前提にしています。

コンテンツを **C1Gallery** コントロールに追加するには、以下の手順を実行します。

1. ソースビューで、`<cc1:C1Gallery></cc1:C1Gallery>` タグの間にマウスを置き、次のマークアップを追加して、項目をコントロールに追加します。

```
ソースビュー
<Items></Items>
```

2. `<Items></Items>` タグの間にマウスを置き、次のマークアップを追加して、**C1GalleryItem** をコントロールに追加します。

```
ソースビュー
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/600/300/sports/1"
ImageUrl="http://lorempixum.com/200/150/sports/1" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/600/300/sports/2"
ImageUrl="http://lorempixum.com/200/150/sports/2" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/600/300/sports/3"
ImageUrl="http://lorempixum.com/200/150/sports/3" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/600/300/sports/4"
ImageUrl="http://lorempixum.com/200/150/sports/4" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/600/300/sports/5"
ImageUrl="http://lorempixum.com/200/150/sports/5" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/600/300/sports/6"
ImageUrl="http://lorempixum.com/200/150/sports/6" />
```

各 **C1GalleryItem** に対して以下のプロパティを設定することに注意してください。

- **ImageUrl**: ギャラリー項目の画像 URL。
 - **LinkUrl**: ギャラリー項目のリンク URL。
3. アプリケーションを実行して、ページが次の画像のように表示されていることを確認します。



- サムネイル画像をクリックして画像をナビゲートします。

この手順では、**C1Gallery** コントロールにコンテンツを追加しました。次の手順では、コントロールの外観と動作をカスタマイズします。

手順 3: コントロールの外観と動作のカスタマイズ

C1Galleryの外観と動作を簡単にカスタマイズできます。次の手順では、コントロールの動作と外観を設定します。以下は、「[手順 2: コントロールへのコンテンツの追加](#)」トピックを完了していることを前提にしています。

以下の手順を実行します。

- ソースビューで、`<cc1:C1Gallery>`タグを編集して、`Width="600px" Height="256px"`:を追加します。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" Width="600px" Height="256px">
```

これにより、コントロールのサイズが設定されます。

- ソースビューで、`<cc1:C1Gallery>`タグを編集して、`ShowPager="False" ShowTimer="True"`:を追加します。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" Width="600px" Height="256px" ShowPager="False" ShowTimer="True">
```

これにより、コントロールの下にページャボタンが表示されなくなり、タイマーが表示されます。

- ソースビューで、`<cc1:C1Gallery>`タグを編集して、`ThumbnailOrientation="Vertical" ThumbsDisplay="3"`:を追加します。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" Width="600px" Height="256px" ShowPager="False" ThumbnailOrientation="Vertical" ThumbsDisplay="3">
```

これにより、サムネイルの向きと数に変更されます。一度に3つのサムネイル画像のみ表示され、サムネイル画像は選択した画像の下に水平方向ではなく垂直方向に表示されます。

- アプリケーションを実行して、コントロールに変更内容が表示されることを確認します。



サムネイル画像の位置と数が変わり、コントロールのサイズが小さくなり、ページャが表示されなくなっていることに注意してください。

この手順では、コントロールの外観と動作をカスタマイズしました。おめでとうございます。このクイックスタートは完了しました。

デザイン時のサポート

以下のセクションでは、**C1Gallery** のデザイン時環境を使用して **C1Gallery** コントロールを構成する方法について説明します。

C1Gallery スマートタグ

Visual Studio では、**C1Gallery** コントロールにスマートタグが用意されています。スマートタグとは、**C1Gallery** で最もよく使用されるプロパティを提供するショートカットタスクメニューです。

C1Gallery コントロールでは、スマートタグによって、よく使用されるプロパティにすばやく簡単にアクセスできます。

[**C1Gallery タスク**]メニューにアクセスするには、**C1Gallery** コントロールの右上端にあるスマートタグをクリックします。これによって、[**C1Gallery タスク**]メニューが開きます。

C1Gallery タスク	
データ ソースの選択:	(なし) ▼
サムネイルの方向	Vertical ▼
ギャラリー項目の追加	
テーマ	aristo ▼
<input checked="" type="checkbox"/> CDN の使用	
CDN パス	http://cdn.wijmo.com/
バージョン情報...	
テンプレートの編集	

[**C1Gallery タスク**]メニューは次のように動作します。

- **データソースの選択**
既存のデータソースを選択するか、[**データソース構成ウィザード**]から新しい接続を作成します。
- **サムネイルの方向**
コントロールのサムネイル画像の向きを **Horizontal** または **Vertical** のいずれかに設定します。デフォルトの向きは **Horizontal** です。
- **ギャラリー項目の追加**
[**ギャラリー項目の追加**]項目をクリックすると、**C1Gallery デザイナフォーム**が開き、C1Gallery コントロールに項目を追加したり、編集したりできるようになります。
- **テーマ**
[**テーマ**]ドロップダウンボックスをクリックすれば、各種の視覚スキームから選択できます。使用可能なテーマの詳細については、「**テーマ**」を参照してください。
- **新しいテーマの作成**
[**新しいテーマの作成**]オプションをクリックすると、**ThemeRoller for Visual Studio** が開きます。したがって、開発環境内でテーマをカスタマイズすることができます。アプリケーションで **ThemeRoller for Visual Studio** を使用方法については、「**ThemeRoller for Visual Studio**」を参照してください。
- **CDN の使用**
[**CDN の使用**]チェックボックスを **ON** にすると、CDN からクライアントリソースがロードされます。これはデフォルトで OFF です。
- **CDN パス**
CDN の URL パスを表示します。

- **Bootstrap の使用**

[**Bootstrap の使用**] オプションを選択すると、コントロールに Bootstrap テーマを適用することができます。アプリケーションで Bootstrap テーマを使用する方法については、「[Bootstrap for ASP.NET Web Forms クイックスタート](#)」を参照してください。

- **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。

- **テンプレートの編集**

このオプションをクリックすると、**テンプレート編集モード**が起動します。

C1Gallery コンテキストメニュー

リスト上の任意の場所を右クリックして、**C1Gallery** のコンテキストメニューを表示します。これは、Visual Studio がすべての .NET コントロールに提供しているコンテキストメニューですが、**C1Gallery** のコンテキストメニューは、機能がいくつか追加されています。

	切り取り(T)	Ctrl+X
	コピー(Y)	Ctrl+C
	貼り付け(P)	Ctrl+V
	代替の貼り付け(E)	
	削除(D)	Del
	コードの表示(C)	
	ブラウザーで表示(B)	
	マスターの編集(M)	Ctrl+M, Ctrl+M
	スマート タグの表示(G)	Shift+Alt+F10
	ギャラリー項目の追加	
	テンプレートの編集(I)	▶
	テンプレート編集の終了(N)	
	最新の情報に更新(F)	
	プロパティ(R)	Alt+Enter

コンテキストメニューのコマンドは、以下のように動作します。

- **スマートタグの表示**

この項目をクリックすると、[**C1Gallery タスク**]メニューが表示されます。スマートタグの使用方法や[**タスク**]メニューで提供される機能の詳細については、「[C1Gallery スマートタグ](#)」を参照してください。

- **ギャラリー項目の追加**

[**ギャラリー項目の追加**]項目をクリックすると、**C1Gallery デザイナフォーム**が開き、C1Gallery コントロールに項目を追加したり、編集したりできるようになります。

- **テンプレートの編集**

このオプションをクリックすると、**テンプレート編集モード**が起動します。

C1Gallery デザイナフォーム

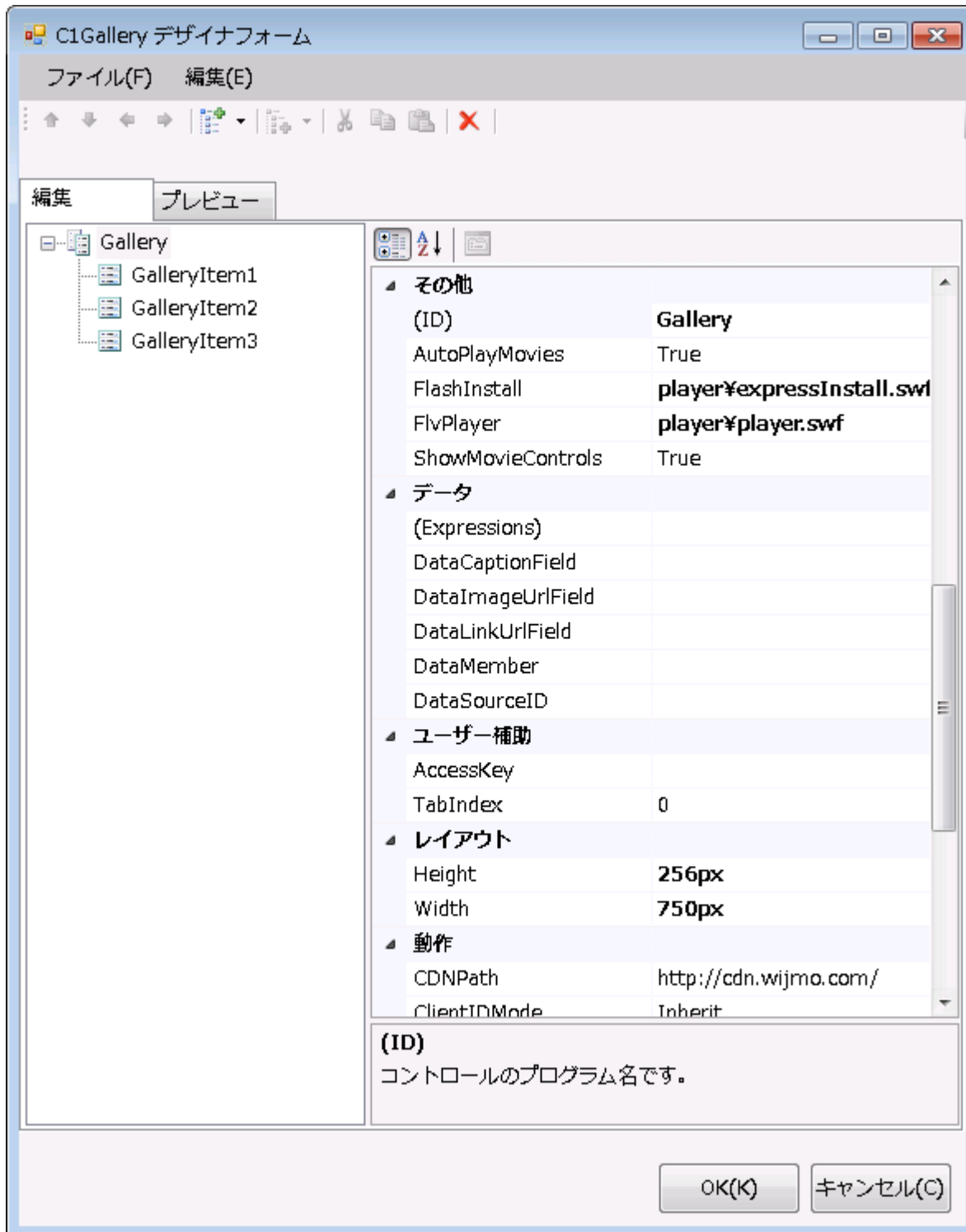
C1Gallery デザイナフォームを使用して、**C1GalleryItem** を追加および編集すると共に、さまざまなプロパティを設定して **C1Gallery** コントロールの外観や動作を変更できます。

[**C1Gallery タスク**]メニュー、コンテキストメニュー、およびプロパティウィンドウからデザイナーフォームにアクセスできます。任

Gallery for ASP.NET Web Forms

意のメニューから[ギャラリー項目の追加]オプションを選択してデザイナフォームを開きます。

デザイナフォームは次のように表示されます。



C1Gallery の外観

C1Gallery コントロールの外観をカスタマイズするには、いくつかのオプションがあります。以下のセクションでは、組み込みのテーマでコントロールの外観を変更する方法や、**C1Gallery** コントロールの他の要素をカスタマイズする方法を説明します。

タスク別ヘルプ

タスク別ヘルプは、ASP.NET のプログラミングに精通し、コントロールの一般的な使用方法を理解しているユーザーを対象としています。ヘルプで概説されている手順を実行すれば、**C1Gallery**のさまざまな機能を実証するプロジェクトを作成し、**C1Gallery** の機能を理解できます。

各トピックでは、**C1Gallery**コントロールを使用した特定のタスクのソリューションを示します。タスク別ヘルプの各トピックでは、新しい ASP.NET プロジェクトを作成し、必要なアセンブリへの参照を追加済みであることも前提となります。

ギャラリーの向きの設定

デフォルトでは、**C1Gallery** コントロールは、水平の向きで表示されます。この例では、**ThumbnailOrientation** プロパティと **ThumbnailDirection** プロパティを使用してギャラリーにサムネイル画像を表示する方法を変更できます。

ソースビューの場合

ソースビューで、`ThumbnailOrientation="Vertical" ThumbnailDirection="Before"` を `<cc1:C1Gallery>` タグに追加します。その結果、次のようになります。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" ThumbnailOrientation="Vertical" ThumbnailDirection="Before">
```

デザインビューの場合

デザインビューで、**C1Gallery**コントロールを選択し、プロパティウィンドウで次のプロパティを設定します。

- **ThumbnailOrientation** プロパティを **Vertical**
- **ThumbnailDirection** プロパティを **Before**

C1Gallery タスクメニューで、サムネイルの方向 プロパティを **Vertical** に設定することもできます。詳細については、「[C1Gallery スマートタグ](#)」を参照してください。

コードの場合

次のコードを `Page_Load` イベントに追加します。

Visual Basic コードの書き方

Visual Basic

```
Me.C1Gallery1.ThumbnailOrientation = "Vertical"
```

```
Me.C1Gallery1.ThumbnailDirection = "Before"
```

C# コードの書き方

C#

```
this.C1Gallery1.ThumbnailOrientation = "Vertical";
```

```
this.C1Gallery1.ThumbnailDirection = "Before";
```

ギャラリーへのページの追加

C1Galleryコントロールには、各種のナビゲーションオプションが含まれており、そのオプションの1つを使用してギャラリーにページを追加することで、ユーザーがより簡単にギャラリー項目をナビゲートできるようになります。**ShowPager** プロパティ

を True に設定すると、ギャラリーの下にページャが表示され、ユーザーがギャラリー内のすべての項目をナビゲートできるようになります。以下の例では、数値ページャを追加します。

ソースビューの場合

ソースビューで、`ShowPager="True"` を `<cc1:C1Gallery>` タグに追加します。その結果、次のようになります。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" ShowPager="True">
```

デザインビューの場合

デザインビューで、**C1Gallery** コントロールを選択し、プロパティウィンドウで **ShowPager** プロパティを **True** に設定します。

コードの場合

以下のコードを **Page_Load** イベントに追加し、ページャをギャラリーに追加します。

Visual Basic コードの書き方

Visual Basic

```
Me.C1Gallery1.ShowPager = True
```

C# コードの書き方

C#

```
this.C1Gallery1.ShowPager = true;
```

ギャラリー項目の自動再生

選択すると、ギャラリー内の項目をスライドショーとして自動的に再生できます。**AutoPlay** が True に設定されている場合、**C1Gallery** は自動的にコンテンツ項目を繰り返し表示します。**ShowTimer** が True に設定されている場合、ユーザーが実行時にスライドショーを開始および停止できる〈再生〉／〈一時停止〉ボタンを備えるプログレスバーが表示されます。

以下の例では、ギャラリーにオートプレイ処理とタイマーを追加します。

ソースビューの場合

ソースビューで、`AutoPlay="True" ShowTimer="True"` を `<cc1:C1Gallery>` タグに追加します。その結果、次のようになります。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" AutoPlay="True" ShowTimer="True">
```

デザインビューの場合

デザインビューで、**C1Gallery** コントロールを選択し、プロパティウィンドウで次のように設定します。

- **AutoPlay** プロパティを **True**.
- **ShowTimer** プロパティを **True**.

コードの場合

次のコードを **Page_Load** イベントに追加します。

Visual Basic コードの書き方

Visual Basic

```
Me.C1Gallery1.AutoPlay = True  
  
Me.C1Gallery1.ShowTimer = True
```

C# コードの書き方

C#

```
this.C1Gallery1.AutoPlay = true;  
  
this.C1Gallery1.ShowTimer = true;
```

オンデマンドでのギャラリー項目のロード

C1Gallery は、データ(ノード)の動的取得をサポートします。子項目はスクロールインされるときに AJAX を使用してロードされます。以下の例では、**LoadOnDemand** プロパティを **True** に設定することにより、オンデマンドでのデータのロードを可能にします。

ソースビューの場合

ソースビューで、**LoadOnDemand="True"** を **<cc1:C1Gallery>** タグに追加します。その結果、次のようになります。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" LoadOnDemand="True">
```

デザインビューの場合

デザインビューで、**C1Gallery** コントロールを選択し、プロパティウィンドウで **LoadOnDemand** プロパティを **True** に設定します。

コードの場合

次のコードを **Page_Load** イベントに追加します。

Visual Basic コードの書き方

Visual Basic

```
Me.C1Gallery1.LoadOnDemand = True
```

C# コードの書き方

C#

```
this.C1Gallery1.LoadOnDemand = true;
```

実行時の遷移アニメーションの変更

C1Gallery コントロールに遷移アニメーション効果を追加することによって、アプリケーションをカスタマイズできます。これらのアニメーションは、ギャラリー画像を移動する際に、ある画像が次の画像に遷移するときに表示されます。この例では、実行時

にこれらのアニメーション効果を設定および変更する方法を示します。

以下の手順を実行します。

1. ソースビューで、次のように `<cc1:C1Gallery>` マークアップを編集します。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" ShowTimer="True" Width="750px" Height="256px"
ThumbnailOrientation="Vertical" ThumbsDisplay="3" ShowPager="false"> <Transitions> <Animated
Disabled="false" Effect="slide" /> </Transitions>
<Items>
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/750/300/sports/1"
ImageUrl="http://lorempixum.com/200/150/sports/1" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/750/300/sports/2"
ImageUrl="http://lorempixum.com/200/150/sports/2" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/750/300/sports/3"
ImageUrl="http://lorempixum.com/200/150/sports/3" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/750/300/sports/4"
ImageUrl="http://lorempixum.com/200/150/sports/4" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/750/300/sports/5"
ImageUrl="http://lorempixum.com/200/150/sports/5" />
<cc1:C1GalleryItem LinkUrl="http://lorempixum.com/750/300/sports/6"
ImageUrl="http://lorempixum.com/200/150/sports/6" />
</Items>
</cc1:C1Gallery>
```

これは、ギャラリーコントロールをカスタマイズし、コントロールに画像を追加します。

2. 以下のマークアップをギャラリーマークアップの直下に追加します。

ソースビュー

```
<label for="showingEffectTypes"> アニメーション</label>
<asp:DropDownList ID="showEffectTypes" runat="server">
<asp:ListItem Value="blind" Selected="True">Blind</asp:ListItem>
<asp:ListItem Value="clip">Clip</asp:ListItem>
<asp:ListItem Value="drop">Drop</asp:ListItem>
<asp:ListItem Value="explode">Explode</asp:ListItem>
<asp:ListItem Value="fade">Fade</asp:ListItem>
<asp:ListItem Value="fold">Fold</asp:ListItem>
<asp:ListItem Value="highlight">Highlight</asp:ListItem>
<asp:ListItem Value="puff">Puff</asp:ListItem>
<asp:ListItem Value="pulsate">Pulsate</asp:ListItem>
<asp:ListItem Value="scale">Scale</asp:ListItem>
<asp:ListItem Value="size">Size</asp:ListItem>
<asp:ListItem Value="slide">Slide</asp:ListItem>
</asp:DropDownList>
```

これは、アニメーション遷移効果のタイプを指定するドロップダウンリストを追加します。

3. 以下のマークアップを前のマークアップの直下に追加します。

ソースビュー

```
<script type="text/javascript">
$(document).ready(function () {
var transitions = {
```

Gallery for ASP.NET Web Forms

```
animated: "fade",
duration: 1000,
easing: null
}
$('#<%=showEffectTypes.ClientID%>').change(function () {
var ee = $('#<%=showEffectTypes.ClientID%> option:selected').val();
$.extend(transitions, { animated: ee });
$('#<%=C1Gallery1.ClientID%>').c1gallery("option", "transitions", transitions);
});
});
</script>
```

このスクリプトは実行時にアニメーション効果を設定します。

✔このトピックの作業結果

アプリケーションを実行し、実行時にドロップダウンボックスからアニメーション効果を選択します。ギャラリーで〈前へ〉または〈次へ〉ボタンをクリックして、選択した遷移アニメーション効果を表示します。別の効果を選択して、ギャラリーで画像を移動する際にどのように表示されるかを確認できます。

ギャラリーのデータバインディング

C1Galleryは、画像 URL、キャプション、およびリンクフィールドをバインドしたり、任意のデータをテンプレートにバインドしたりできるデータバインディングをサポートします。データバインディングでは、以下のプロパティを任意に設定できます。

- **DataSourceID** - データソースの ID を設定します。
- **DataCaptionField** - 画像のキャプションのロード元のデータソース内にフィールドを設定します。
- **DataLinkUrlField** - リンクフィールドのロード元のデータソース内にフィールドを設定します。
- **DataImageUrlField** - URL フィールドのロード元のデータソース内にフィールドを設定します。

この例では、ギャラリーにデータをバインドします。以下の手順を実行します。

1. ソースビューで、次のように `<cc1:C1Gallery>` マークアップを編集します。

ソースビュー

```
<cc1:C1Gallery ID="C1Gallery1" runat="server" ShowTimer="True" Width="750px" Height="416px"
ThumbsDisplay="4">
</cc1:C1Gallery>
```

これはギャラリーコントロールをカスタマイズします。

2. ページを右クリックして[コードの表示]を選択し、切り替えてコードエディタを開きます。
3. 次のコードを初期クラスに追加します。

Visual Basic コードの書き方

Visual Basic

```
Private Const TEXT As String = "{0} Vestibulum venenatis faucibus eros, vitae vulputate ipsum tempor ut. Donec ut ligula a metus volutpat sagittis. Duis sodales, lorem nec suscipit imperdiet, sapien metus tempor nibh, dapibus pulvinar lorem lacus molestie lacus. "
```

C# コードの書き方

C#

```
private const string TEXT = "{0} Vestibulum venenatis faucibus eros, vitae vulputate ipsum tempor ut. Donec ut ligula a metus volutpat sagittis. Duis sodales, lorem nec suscipit imperdiet, sapien metus tempor nibh, dapibus pulvinar lorem lacus molestie lacus. ";
```

4. 次のコードを **Page_Load** イベントに追加します。

Visual Basic コードの書き方

Visual Basic

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles Me.Load
    If Not IsPostBack OrElse IsCallback Then
        bind()
    End If
End Sub
```

C# コードの書き方

C#

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack || IsCallback)
    {
        bind();
    }
}
```

5. メインクラス内の **Page_Load** イベントの直下に次のコードを追加します。

Visual Basic コードの書き方

Visual Basic

```
Private Sub bind()
    Dim list1 As List(Of ContentGallery) =
    GetDataSource("http://loempixum.com/200/150/sports/{0}",
    "http://loempixum.com/750/300/sports/{0}")
    C1Gallery1.DataSource = list1
    C1Gallery1.DataImageUrlField = "ImageUrl"
    C1Gallery1.DataLinkUrlField = "LinkUrl"
    C1Gallery1.DataCaptionField = "Caption"
    C1Gallery1.DataBind()
End Sub
```

C# コードの書き方

C#

```
private void bind()
{
```

```
List list1 = GetDataSource("http://loempixum.com/200/150/sports/{0}",
"http://loempixum.com/750/300/sports/{0}");
C1Gallery1.DataSource = list1;
C1Gallery1.DataImageUrlField = "ImageUrl";
C1Gallery1.DataLinkUrlField = "LinkUrl";
C1Gallery1.DataCaptionField = "Caption";
C1Gallery1.DataBind();
}
```

このコードはギャラリーのデータバインディングを設定します。

6. 次のコードをメインクラスに追加します。

Visual Basic コードの書き方

Visual Basic

```
Private Function GetDataSource(urlFormatStr As String, linkFormatStr As String)
As List(Of ContentGallery)
    Dim list As New List(Of ContentGallery)()
    For i As Integer = 1 To 10
        list.Add(New ContentGallery() With { _
            .Content = String.Format(TEXT, String.Format("{0}.The picture one, ",
i.ToString())), _
            .LinkUrl = String.Format(linkFormatStr, i.ToString()), _
            .ImageUrl = String.Format(urlFormatStr, i.ToString()), _
            .Caption = String.Format("Pic {0}", i.ToString()) _
        })
    Next
    Return list
End Function
```

C# コードの書き方

C#

```
private List GetDataSource(string urlFormatStr, string linkFormatStr)
{
    List list = new List();
    for (int i = 1; i < 11; i++)
    {
        list.Add(new ContentGallery()
        {
            Content = string.Format(TEXT, string.Format("{0}.The picture one, ",
i.ToString())),
            LinkUrl = string.Format(linkFormatStr, i.ToString()),
            ImageUrl = string.Format(urlFormatStr, i.ToString()),
            Caption = string.Format("Pic {0}", i.ToString())
        });
    }
    return list;
}
```

このコードはギャラリーにコンテンツを追加します。

7. 次のクラスコードをメインクラスの直下に追加します。

Visual Basic コードの書き方

```

Visual Basic
Public Class ContentGallery
Public Property Content() As String
    Get
        Return m_Content
    End Get
    Set(value As String)
        m_Content = Value
    End Set
End Property
Private m_Content As String
    Public Property ImgUrl() As String
        Get
            Return m_ImgUrl
        End Get
        Set(value As String)
            m_ImgUrl = value
        End Set
    End Property
Private m_ImgUrl As String
Public Property LinkUrl() As String
    Get
        Return m_LinkUrl
    End Get
    Set(value As String)
        m_LinkUrl = value
    End Set
End Property
Private m_LinkUrl As String
Public Property Caption() As String
    Get
        Return m_Caption
    End Get
    Set(value As String)
        m_Caption = value
    End Set
End Property
Private m_Caption As String
End Class

```

C# コードの書き方

```

C#
public class ContentGallery
{

```

Gallery for ASP.NET Web Forms

```
public string Content
{ get; set; }
public string ImgUrl
{ get; set; }
public string LinkUrl
{ get; set; }
public string Caption
{ get; set; }
}
```

✔このトピックの作業結果

アプリケーションを実行し、**C1Gallery** コントロールのデータがバインドされ、ギャラリーコントロールにいくつかの画像が表示されます。