

Pager for ASP.NET Web Forms

2018.04.24 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	4-5
デザイン時のサポート	6
スマートタグ	6
コンテキストメニュー	6-7
C1Pager の基礎	8
テーマ	8
ページャモード	8
ツールチップ	8-9
ページインデックス	9
CSS セレクタ	9
タスク別ヘルプ	10
テーマの利用	10
組み込みテーマの使用	10-11
ページャモードの変更	11-12
表示されるページ数の指定	12-13
ページ数の指定	13
選択されるボタンの指定	13-14
CSS セレクタでの作業	14-15
ページ付けされたテーブルの作成	15-16

製品の概要

Pager for ASP.NET Web Forms を使用してアプリケーションにページングを追加します。ページャを使用すると、エンドユーザーは表示される要素をクリックして要素またはページをスキャンできます。ページャを設定して、数字ページングシステムまたは次へ/前へシステムを使用します。

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

Pager for ASP.NET Web Forms は、次のいくつかの機能を備えています。

- **〈次へ〉／〈前へ〉ボタン**
ページを設定して、〈次へ〉ボタンと〈前へ〉ボタンで構成するページネーションコントロールのセットである次へ／前へシステムを使用します。
- **アイコンボタン**
アイコン付きボタンの外観をカスタマイズします。
- **数字ページング**
数字ページングを設定して、ページに直接アクセスするための数字リンクボタンで構成するページネーションコントロールのセットである数字ページングシステムを使用します。

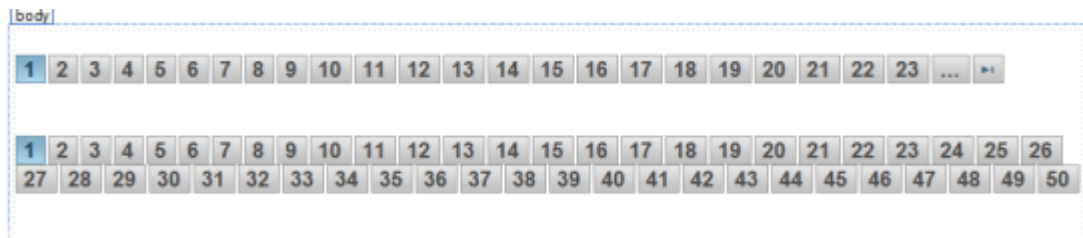
クイックスタート

このクイックスタートでは、**C1Pager** コントロールの機能について学びます。2つのページャコントロールを作成し、それぞれの **PageIndexChanged** イベントを処理して、1つのページャコントロールのページインデックスを選択すると他のコントロールのページインデックスが変わるようにします。

以下の手順を実行します。

1. 新しい ASP.NET Web サイトプロジェクトを作成します。
2. ドキュメントウィンドウのすぐ下の「**デザイン**」タブをクリックして、デザインビューに切り替えます
3. Visual Studio ツールボックスに移動し、**C1Pager** をダブルクリックして、ページにコントロールを追加します。もう一度ダブルクリックして、2番目のコントロールを追加します。
2つの空の **C1Pager** コントロールである **C1Pager1** と **C1Pager2** が Web ページに表示されます。
4. **C1Pager1** のプロパティを次のように設定します。
 - **Mode** プロパティを `NumericFirstLast` に設定します。これにより、数字と先頭ページへのショートカットを提供する〈先頭へ〉ボタンと最後のページへのショートカットを提供する〈最後へ〉ボタンを備えるページャが作成されます。
 - **PageCount** を「50」に設定します。これは、コントロールが保持する総ページ数です。
 - **PageButtonCount** を「23」に設定します。これは、いつでも 50 ページのうち 23 ページのみコントロール上に表示されるように指定します。
 - **AutoPostBack** プロパティを `True` に設定します。後刻クイックスタートでサーバー側イベントを使用する場合があるので、イベントが発生したときにコントロールをサーバーにPostBackする必要があります。
5. **C1Pager2** のプロパティを次のように設定します。
 - **PageCount** プロパティを「50」に設定します。これは、コントロールが保持する総ページ数です。
 - **PageButtonCount** プロパティを「50」に設定します。これは、いつでも 50 ページのうち 50 ページすべてがコントロール上に表示されるように指定します。
 - **AutoPostBack** プロパティを `True` に設定します。後刻クイックスタートでサーバー側イベントを使用する場合があるので、イベントが発生したときにコントロールをサーバーにPostBackする必要があります。

この時点で、プロジェクトは次のように表示されます。



6. **プロパティウィンドウ**で、**C1Pager1** を選択し、〈**イベント**〉ボタンをクリックします。**PageIndexChanged** をダブルクリックし、**Pager1_PageIndexChanged** イベントハンドラをコードビューに追加して、次のコードをイベントハンドラに追加します。

Visual Basic コードの書き方

```
Visual Basic
C1Pager2.PageIndex = C1Pager1.PageIndex
```

C# コードの書き方

```
C#
C1Pager2.PageIndex = C1Pager1.PageIndex;
```

7. 前の **C1Pager2** の手順を繰り返します。ただし、**Pager2_PageIndexChanged** イベントハンドラには次のコードを追加します。

Pager for ASP.NET Web Forms

Visual Basic コードの書き方

Visual Basic

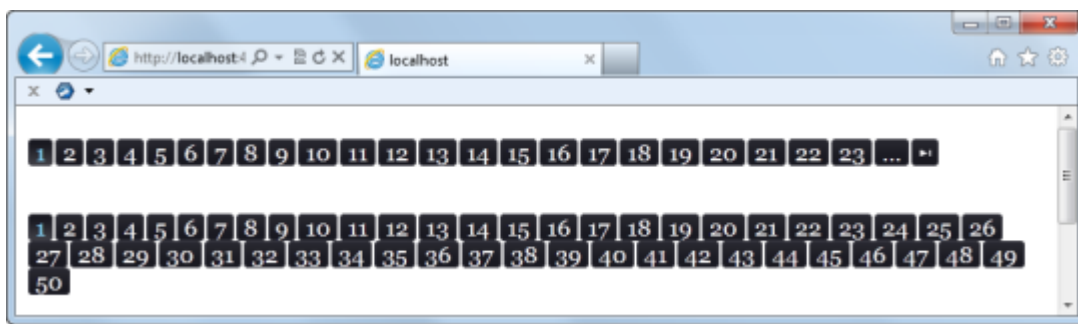
```
C1Pager1.PageIndex = C1Pager2.PageIndex
```

C# コードの書き方

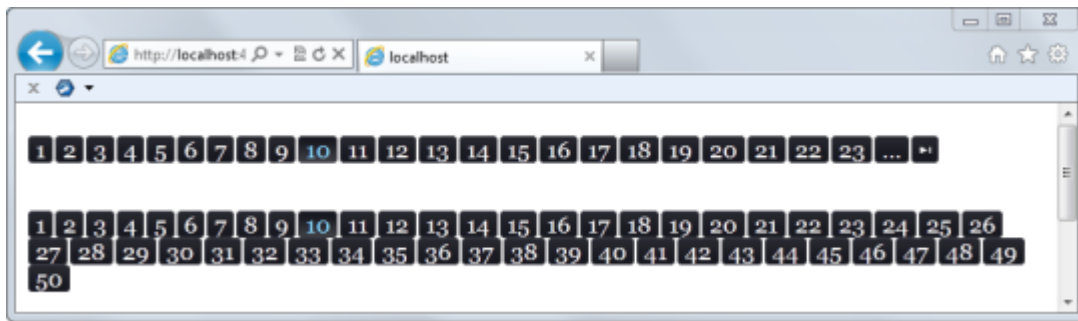
C#

```
C1Pager1.PageIndex = C1Pager2.PageIndex;
```

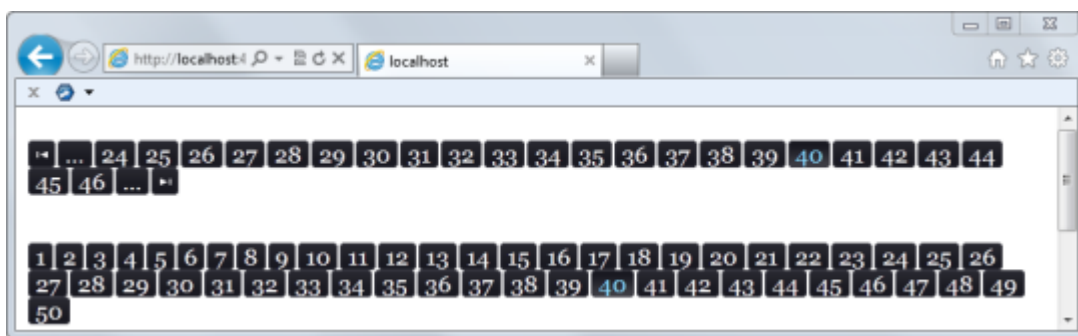
8. デザインビューで、**C1Pager1** のスマートタグをクリックして、[**C1Pager タスク**]メニューを開きます。テーマリストから、**midnight** を選択します。
両方の**C1Pager** コントロールが midnight テーマに変わります。その他の ASP.NET Web Forms コントロールをページに追加すると、これらのコントロールにも midnight テーマが適用されます。
9. [F5]を押して、Web サイトを起動します。実行時に、プロジェクトは次のように表示されます。



10. 上部のページャで **10** をクリックします。両方のページャが 10 に切り替わることを確認してください。



11. 下部のページャで **40** をクリックします。両方のページャが 40 に切り替わることを確認してください。上部のページャが変更されたことにも注意してください。上部のページャは、右側に<...>ボタンと<最後へ>ボタンのみ表示されていましたが、左側に<...>ボタンと<先頭へ>ボタンも表示されています。



おめでとうございます。これで **Pager for ASP.NET Web Forms** 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

C1Pager は、スマートタグ、および充実したデザイン時のサポートを提供するデザイナを備えており、オブジェクトモデルの操作が簡単になっています。

以下のトピックでは、**C1Pager** のデザイン時環境を使用して **C1Pager** コントロールを設定する方法を説明します。

スマートタグ

C1Pager コントロールには、Visual Studio にスマートタグが含まれています。スマートタグは、**C1Pager** 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のタスクメニューを表します。

C1Pager コントロールでは、スマートタグを使用することによって、**C1Pager** デザイナや一般的なプロパティにすばやく簡単にアクセスできます。

[**C1Pager** タスク]メニューにアクセスするには、**C1Pager** 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。

[**C1Pager** タスク]メニューが開きます。



[**C1Pager** タスク]メニューは次のように動作します。

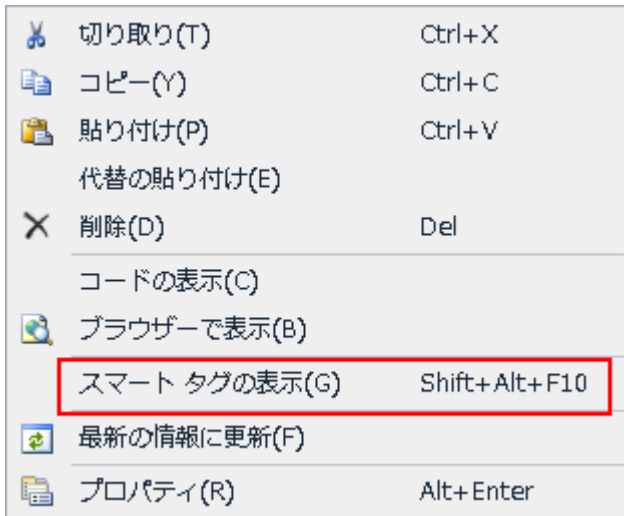
- テーマ**
 [テーマ]ドロップダウンをクリックして提供される6つの組み込み **ComponentOne for ASP.NET Web Forms** テーマから選択してコントロールに適用することができます。
- 新しいテーマの作成**
 [新しいテーマの作成]オプションをクリックすると、**ThemeRoller for Visual Studio** が開きます。したがって、開発環境内でテーマをカスタマイズすることができます。アプリケーションで **ThemeRoller for Visual Studio** を使用する方法については、「[ThemeRoller for Visual 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。

コンテキストメニュー

Pager for ASP.NET Web Forms

C1Pager には、Visual Studio がすべての .NET コントロールや ASP.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利用できる追加的なコマンドがあります。

C1Pager コントロール上の任意の場所を右クリックし、コンテキストメニューを表示します。



C1Pager のコンテキストメニューは次のように動作します。

- **スマート タグの表示**
[スマート タグの表示]をクリックすると、[**C1Pager タスク**]メニューが開きます。

C1Pager の基礎

以下のトピックでは、組み込みテーマはページャモードなど、**C1Pager** コントロールのいくつかの機能の使い方に慣れ親しみます。

テーマ

C1Pager コントロールには、6種類の組み込みテーマが含まれています。これらのテーマの1つを選択すると、ページ上の他のすべての ComponentOne for ASP.NET コントロールはそれに応じてスキンされます。テーマは **C1Pager** コントロールに次のように表示されます。

Aristo	
Cobalt	
Midnight	
Rocket	
Sterling	

C1Pager コントロールのテーマを設定するには、単に Theme プロパティをいずれかの組み込みテーマに設定します。

ページャモード

C1Pager コントロールは、ページに表示するページャのタイプを指定する4種類の異なるモードに設定できます。4種類のモードは、次の **Mode** プロパティによって制御されます。**NextPrevious**、**NextPreviousFirstLast**、**Numeric**、および **NumericFirstLast**。デフォルトでは、**Mode** プロパティは **Numeric** に設定されます。

C1Pager コントロールの4種類のモードは次のように表示されます。

NextPrevious	
NextPreviousFirstLast	
Numeric(デフォルト)	
NumericFirstLast	

ツールチップ

C1PagerStep.ToolTip プロパティを使用して、使いやすいインターフェースを作成できます。ツールチップは、ユーザーインターフェースの要素に関する情報や指示をユーザーに提供する、グラフィックユーザーインターフェースの要素です。ユーザーがページ上にマウスポインタを置くと、追加的な情報が記載されたボックスが表示されます。

ツールチップは、**C1PagerStep.ToolTip** プロパティを文字列に設定することで、**C1Pager** の各ページに適用できます。コントロールの最初のタブにツールチップを設定する場合は、次のコードを使用します。

Visual Basic コードの書き方

```
Visual Basic
```

Pager for ASP.NET Web Forms

```
C1Pager.ToolTip = "Hello World!"
```

C# コードの書き方

```
C#  
C1Pager.ToolTip = "Hello World!";
```

C1PagerStep.ToolTip プロパティは、デザインビューまたはソースビューでも設定できます。

ページインデックス

C1Pager コントロールのタブには、ゼロベースのインデックスが付けられます。つまり、最初のタブのインデックスはゼロです。デフォルトでは、**PageIndex** プロパティはゼロ(「0」)に設定され、実行時には最初のページがフォーカスされます。実行時に選択されるページを変更するには、**PageIndex** プロパティをインデックス内の別の値に設定します。たとえば、4つのページがあり、実行時に最後のページが選択されるようにする場合は、**PageIndex** プロパティを「3」に設定します。

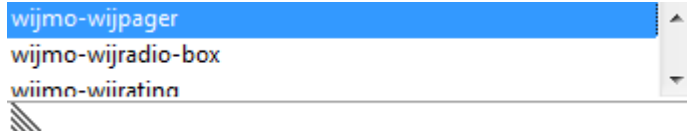
CSS セレクタ

CSS を使用して多くの **C1Pager** 要素にスタイルを適用し、それらの外観を独特のものにすることができます。カスタマイズを簡素化するために、ComponentOne には、その6種類の組み込みテーマごとに CSS セレクタが組み込まれています。

枠、背景、テキスト、フォント、マージン、埋め込み、リスト、輪郭、表などの一般的な CSS プロパティを該当する CSS セレクタに適用できます。

一般に使用される個々の CSS セレクタとグループ化された CSS セレクタのリストについては、プロジェクトの**C1Pager** コントロールを選択し、Visual Studio プロパティウィンドウで **CssClass** プロパティの横にあるドロップダウンリストを表示します。

C1Pager CSS セレクタは以下の wijmo-wijpager で開始します。



タスク別ヘルプ

タスク別ヘルプでは、Visual Studio ASP.NET 環境でのプログラミングに精通し、**Pager for ASP.NET Web Forms** コントロールを全般的に理解しているユーザーを対象としています。

各トピックでは、**C1Pager** 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。各トピックで概説されている手順に従うことによって、さまざまな **C1Pager** 機能を使用したプロジェクトを作成できます。

タスク別ヘルプの各トピックでは、新しい ASP.NET プロジェクトを既に作成していることを前提としています。

テーマの利用


このセクションのトピックは、組み込みテーマとカスタムテーマを利用する方法を示します。

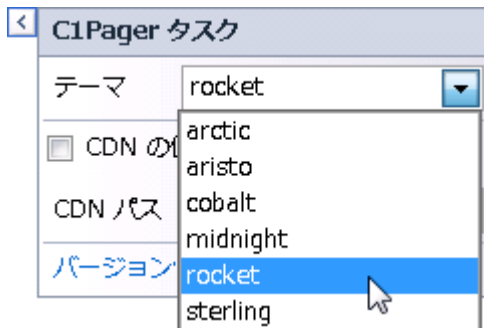
組み込みテーマの使用

C1Pager コントロールには、ほんの数クリックで適用できる6種類の組み込みテーマが用意されています。このトピックでは、デザインビュー、ソースビュー、およびコードでテーマを変更する方法を説明します。テーマについての詳細は、「テーマ」を参照してください。

デザインビューの場合

以下の手順を実行します。

1. **C1Pager** スマートタグ  をクリックして、[**C1Pager タスク**]メニューを開きます。
2. [テーマ]ドロップダウン矢印をクリックして、リストからテーマを選択します。この例では、**rocket** を選択します。



rocket テーマが**C1Pager** コントロールに適用されます。

ソースビューの場合

ソースビューで **C1Pager** のテーマを変更するには、`Theme="rocket"` を `<cc1:C1Pager>` 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:C1Pager runat="server" Theme="rocket"/>
```

コードビューの場合

以下の手順を実行します。

1. 以下の名前空間をプロジェクトにインポートします。

Visual Basic コードの書き方

Visual Basic

```
Imports Cl.Web.Wijmo.Controls
```

C# コードの書き方

C#

```
using Cl.Web.Wijmo.Controls;
```

2. **Theme** プロパティを設定する次のコードを、**Page_Load** イベントに追加します。

Visual Basic コードの書き方

Visual Basic

```
C1Pager1.Theme = "rocket"
```

C# コードの書き方

C#

```
C1Pager1.Theme = "rocket";
```

3. プログラムを実行します。

✔このピックは、次のことを示します。

次の図は、**rocket** テーマが設定された**C1Pager**コントロールを示しています。



ページャモードの変更

Mode プロパティを **NextPrevious**、**NextPreviousFirstLast**、**Numeric**、または **NumericFirstLast** に設定して、表示するページャモードのタイプを変更できます。これらの各種プロパティ設定の概要については、「[ページャモード](#)」を参照してください。

デザインビューの場合

C1Pager コントロールを選択し、プロパティウィンドウに移動して、**Mode** プロパティを **NextPrevious**、**NextPreviousFirstLast**、**Numeric**、または **NumericFirstLast** に設定します。

ソースビューの場合

ソースビューで、`Mode="NextPrevious"`、`Mode="NextPreviousFirstLast"`、`Mode="Numeric"`、または `Mode="NumericFirstLast"` を `<cc1:C1Pager>` タグに追加します。

コードビューの場合

次のコードスニペットは、**Mode** プロパティをコードで設定する方法を示します。

Visual Basic コードの書き方

Visual Basic

「[次へ](#)」および「[前へ](#)」ボタン

```
C1Pager1.Mode = Cl.Web.Wijmo.Controls.C1Pager.PagerMode.NextPrevious
```

'〈次へ〉、〈前へ〉、〈先頭へ〉、および〈最後へ〉ボタン

```
C1Pager1.Mode = C1.Web.Wijmo.Controls.C1Pager.PagerMode.NextPreviousFirstLast
```

'数字ボタン

```
C1Pager1.Mode = C1.Web.Wijmo.Controls.C1Pager.PagerMode.Numeric
```

'〈先頭へ〉および〈最後へ〉ボタン付き数字ボタン

```
C1Pager1.Mode = C1.Web.Wijmo.Controls.C1Pager.PagerMode.NumericFirstLast
```

C# コードの書き方

C#

```
// 〈次へ〉および〈前へ〉ボタン
```

```
C1Pager1.Mode = C1.Web.Wijmo.Controls.C1Pager.PagerMode.NextPrevious;
```

```
// 〈次へ〉、〈前へ〉、〈先頭へ〉、および〈最後へ〉ボタン
```

```
C1Pager1.Mode = C1.Web.Wijmo.Controls.C1Pager.PagerMode.NextPreviousFirstLast;
```

```
// 数字ボタン
```

```
C1Pager1.Mode = C1.Web.Wijmo.Controls.C1Pager.PagerMode.Numeric;
```

```
// 〈先頭へ〉および〈最後へ〉ボタン付き数字ボタン
```

```
C1Pager1.Mode = C1.Web.Wijmo.Controls.C1Pager.PagerMode.NumericFirstLast;
```

表示されるページ数の指定

ページャが一度に表示するページ数を指定できます。デフォルトでは、最大 10 まで表示されます。ただし、これより多くのページまたは少ないページを表示する場合があります。このような場合には、**PageButtonCount** プロパティを希望するページ数に設定できます。

デザインビューの場合

プロパティウィンドウで、**C1Pager** コントロールの **PageButtonCount** プロパティで値を設定します。この例では、「4」に設定します。

ソースビューの場合

ソースビューで、`PageButtonCount = "[Value]"` を `<cc1:C1Pager>` タグに追加します。この例では、次のマークアップに示すように `PageButtonCount="4"` を使用します。

ソースビュー

```
<cc1:C1Pager runat="server" PageButtonCount="4">
```

```
</cc1:C1Pager>
```

コードビューの場合

コードビューで **C1Pager** のページボタン数を指定するには、次のコード (**PageButtonCount** を「4」に設定) を **Page_Load** イベントに追加します。

Visual Basic コードの書き方

Visual Basic

```
C1Pager1.PageButtonCount = 4
```

C# コードの書き方

C#

Pager for ASP.NET Web Forms

```
C1Pager1.PageButtonCount = 4;
```

✔このトピックは、次のことを示します。

次の画像は、一度に 4 ページのみ表示する **C1Pager** コントロールを示します。



ページ数の指定

このトピックでは、**C1Pager** に含まれるページ数を指定する方法について説明します。

デザインビューの場合

デザインビューで **C1Pager** のページ数を指定するには、**C1Pager** コントロールの **PageCount** プロパティで値を設定します。この例では、「12」に設定します。

ソースビューの場合

ソースビューで **C1Pager** のページ数を指定するには、**PageCount** プロパティを `<cc1:C1PagerStep>` タグに追加します。この例では、次のマークアップに示すように `PageCount="12"` を使用します。

ソースビュー

```
<cc1:C1Pager runat="server" PageCount="12">  
</cc1:C1Pager>
```

コードビューの場合

コードビューで **C1Pager** のページ数を指定するには、次のコード (`PageCount` を 12 に設定) を `Page_Load` イベントに追加します。

Visual Basic コードの書き方

Visual Basic

```
C1Pager1.PageCount = 12
```

C# コードの書き方

C#

```
C1Pager1.PageCount=12;
```

✔このトピックは、次のことを示します。

次の画像は、**PageCount** が "12" に設定された **C1Pager** コントロールを示します。



コントロールには 10 ページと <...> ボタンのみ表示されていることを確認してください。<...> ボタンをクリックすると、最後の 2 ページが表示されます。



12 ページをすべて表示する方法については、「[表示されるページ数の指定](#)」を参照してください。

選択されるボタンの指定

C1Pager コントロールの **PageIndex** プロパティを使用して、実行時に選択されるボタンを指定できます。次のトピックでは、デザインビュー、ソースビュー、およびコードで、このプロパティを設定する方法を示します。

デザインビューの場合

プロパティウィンドウで、**PageIndex** プロパティを探し、その値を設定します。この例では、「1」に設定します。

ソースビューの場合

ソースビューで、`PageIndex=" [Value] "` を `<cc1:C1Pager>` タグに追加します。この例では、次のマークアップに示すように `PageIndex="1"` を入力します。

ソースビュー

```
<cc1:C1Pager runat="server" PageIndex="1">
```

コードビューの場合

選択されるインデックスを(たとえば「1」)に変更するには、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

```
C1Pager1.PageIndex = 1
```

C# コードの書き方

C#

```
C1Pager1.Index = 1;
```

✔ このトピックは、次のことを示します。

実行時に、次の図に示されるように、一連のページの2番目が選択されます。



CSS セレクタでの作業

C1Pager は、コントロールの外観を完全にカスタマイズできる CSS スタイル設定をサポートします。このトピックでは、**C1Pager** コントロールに CSS スタイル設定を適用する手順について説明します。

1. デザインビューで、プロパティウィンドウに移動し、**CssClass** プロパティを探します。
2. ドロップダウンリストを使用して適切な CSS セレクタ(このトピックでは `wijmo-wijpager` セレクタ)を選択します。
3. ソースビューに切り替えて、`<asp:Content>` タグの最初のセットを探します。
4. `<asp:Content>` タグの間に `<style type="text/css"> </style>` タグを挿入します。このタグのセットを使って、コントロールに CSS スタイル設定を追加できます。
5. ページャの外観をカスタマイズするには、`<style>` タグの間に次のスクリプトを挿入します。

ソースビュー

```
.wijmo-wijpager
{
    background-color:#002277; font-style:italic; font-family:
```

```
Blackadder ITC; float:right;
}
```

- [F5]を押して、プログラムを実行します。ページャ要素は次の図のように表示されます。



ページ付けされたテーブルの作成

このトピックでは、デザインビュー、ソースビュー、およびコードを使用してページ付けされたテーブルを作成する方法を説明します。このプロジェクトでは、**C1Pager** コントロールと一般的な ASP.NET グリッドコントロールを組み合わせます。このタスク別ヘルプでは、**C:\Users\<ユーザー名>\Documents\ComponentOne Samples\Common** ディレクトリにある Northwind データベース Nwind.mdb を使用します。

以下の手順を実行します。

- ソリューションエクスプローラウィンドウで、**App_Data** フォルダを右クリックして、コンテキストメニューから **[既存項目の追加]** を選択します。
- [既存項目の追加]** ダイアログボックスで、Northwind データベースの格納場所 (デフォルトではサンプルディレクトリ) に移動し、**Nwind.mdb** を選択します。そして **<追加>** をクリックし、ダイアログボックスを閉じて、ファイルをプロジェクトに追加します。
- デザインビューで、**C1Pager** コントロールをプロジェクトに追加します。
- C1Pager** プロパティの **PageCount** を「1」に設定します。
- ソースビューに切り替え、以下のマークアップを `<asp:Content>` タグの2つめのセットに追加して、グリッドコントロールと **C1Pager** コントロールの両方のプロパティを設定します。

ソースビュー

```
<asp:UpdatePanel runat="server" ID="UpdatePanel1">
  <ContentTemplate>
    <cc1:C1Pager runat="server" AutoPostBack="true"
Mode="NumericFirstLast"
    OnPageIndexChanged="C1Pager1_PageIndexChanged" />
    <asp:GridView Width="100%" runat="server"
      DataSourceAllowPaging="True"
      CssClass="ui-widget" ondatabound="GridView1_DataBound">
      <HeaderStyle CssClass="ui-widget-header" />
      <RowStyle CssClass="ui-widget-content" />
      <PagerSettings Visible="false" />
    </asp:GridView>
  </ContentTemplate>
</asp:UpdatePanel>
```

- 次のマークアップを使用して、データソースとグリッドのコンテンツを設定します。

ソースビュー

```
<asp:AccessDataSource runat="server"
  DataFile="~/App_Data/NWind.mdb"
  SelectCommand="SELECT [OrderID] AS 注文コード, [ProductID] AS 商品コード,
[UnitPrice] AS 単価, [Quantity] AS 数量 FROM [Order Details]>
</asp:AccessDataSource>
```

- Default.aspx.cs** ファイルに移動し、次のコードを挿入して **C1Pager** コントロールをグリッドにバインドします。

ソースビュー

```
protected void C1Pager1_PageIndexChanged(object sender, EventArgs e)
{
```



```

        GridView1.PageIndex = C1Pager1.PageIndex;
        GridView1.DataBind();
    }
    protected void GridView1_DataBound(object sender, EventArgs e)
    {
        C1Pager1.PageCount = GridView1.PageCount;
    }
}

```

8. [F5]を押して、プログラムを実行します。ページ付けされたテーブルは、次の図のようになるはずですが。

1	2	3	4	5	6	7	8	9	10	...	▶▶
注文コード		商品コード		単価		数量					
10248	11	952	12								
10248	42	666.4	10								
10248	72	2366.4	5								
10249	14	1264.8	9								
10249	51	2883.2	40								
10249	66	5440	20								
10250	41	523.6	10								
10250	51	2883.2	35								
10250	65	1142.4	15								
10251	22	1142.4	6								

✔ このピックは、次のことを示します。

実行時に、**C1Pager** 要素によって制御されるテーブルが表示されます。