

Wizard for ASP.NET Web Forms

2018.04.17 更新

グレースィティ株式会社

目次

製品の概要	3
ComponentOne for ASP.NET Web Forms のヘルプ	3
主な特長	4
クイックスタート	5
手順 1: ページへのコントロールの追加	5
手順 2: C1Wizard デザイナフォームでの作業	5-6
手順 3: コントロールへのコンテンツの追加	6-8
デザイン時のサポート	9
スマートタグ	9-10
コンテキストメニュー	10
デザイナフォーム	10-12
C1Wizard の要素	13-14
C1Wizard の外観および動作	15
テーマ	15-16
ヘッダーテンプレート	16-17
CSS セレクタ	17
ナビゲーションボタン	17-18
オートプレイ機能	18
ツールチップ	18-19
アクティブインデックス	19
タスク別ヘルプ	20
ステップコンテンツの追加と操作	20
ステップの追加	20-21
ステップへのコントロールの追加	21-22
ステップへのキャプションの追加	22-24
ステップへのテキストの追加	24-26
C1Wizard の外観の変更	26
ステップヘッダーの削除	26-27
組み込みテーマの使用	27-28
CSS セレクタの使用	28-29
C1Wizard のステップのオートプレイ	29-30

[コードによるコントロールの作成](#)
[選択されたステップの変更](#)

30-32

32-33

製品の概要

Wizard for ASP.NET Web Forms を使用して、別個のナビゲート可能なページ上にあるコンテンツのセクションを表示します。一連のフォームを作成して、画面の面積を節約したり、入力タスクを簡素化したりします。

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

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

- **リッチコンテンツ**
単にドラッグ & ドロップ操作を行うだけで、画像、テキスト、コントロールを **C1Wizard** コントロール内のページに追加します。コンテンツのカスタマイズは、標準パネルコントロールのデザインと同様に簡単にできます。
- **外部コンテンツ**
プロジェクト内の別の Web ページのコンテンツまたはプロジェクト外の Web サイトのコンテンツを **C1Wizard** コントロール内に表示します。
- **オートプレイ**
オートプレイ機能を使用してページ間を自動的に移動します。ページの遷移間の遅延の設定を含む、オートプレイをカスタマイズします。
- **アニメーション**
直線的なイー징ングまたはしなやかなイー징ングなど、ページ間の遷移に対する組み込み遷移効果から選択します。オプションで、独自のカスタム効果を作成し、持続時間を変更します。
- **オンデマンドでのコンテンツのロード**
クライアント側で現在選択されたページのみ表示するように **C1Wizard** を設定し、最初のページのサイズとロード時間を削減します。
- **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。
- **CSS のサポート**
CSS (Cascading Style Sheet) のスタイルを使用して、カスタムスキンを定義します。

クイックスタート

このクイックスタートでは、コンテンツが埋め込まれた2つのページを作成して、C1Wizard コントロールの機能を探索します。

手順 1: ページへのコントロールの追加

この手順では、まずページに **C1Wizard** コントロールを追加します。

開始するには、以下の手順を実行します。

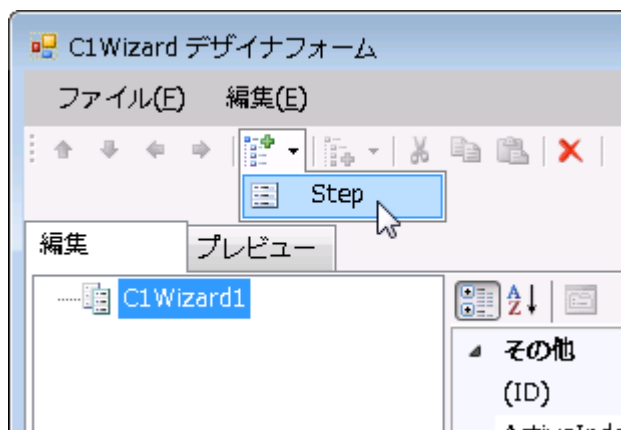
1. 新しい ASP.NET Web サイトプロジェクトを作成します。
2. ドキュメントウィンドウのすぐ下の「**デザイン**」タブをクリックして、デザインビューに切り替えます。
3. Visual Studio ツールボックスに移動し、**C1Wizard** をダブルクリックして、ページにコントロールを追加します。Web ページに空の **C1Wizard** コントロールが組み込まれます。

手順 2: C1Wizard デザイナフォームでの作業

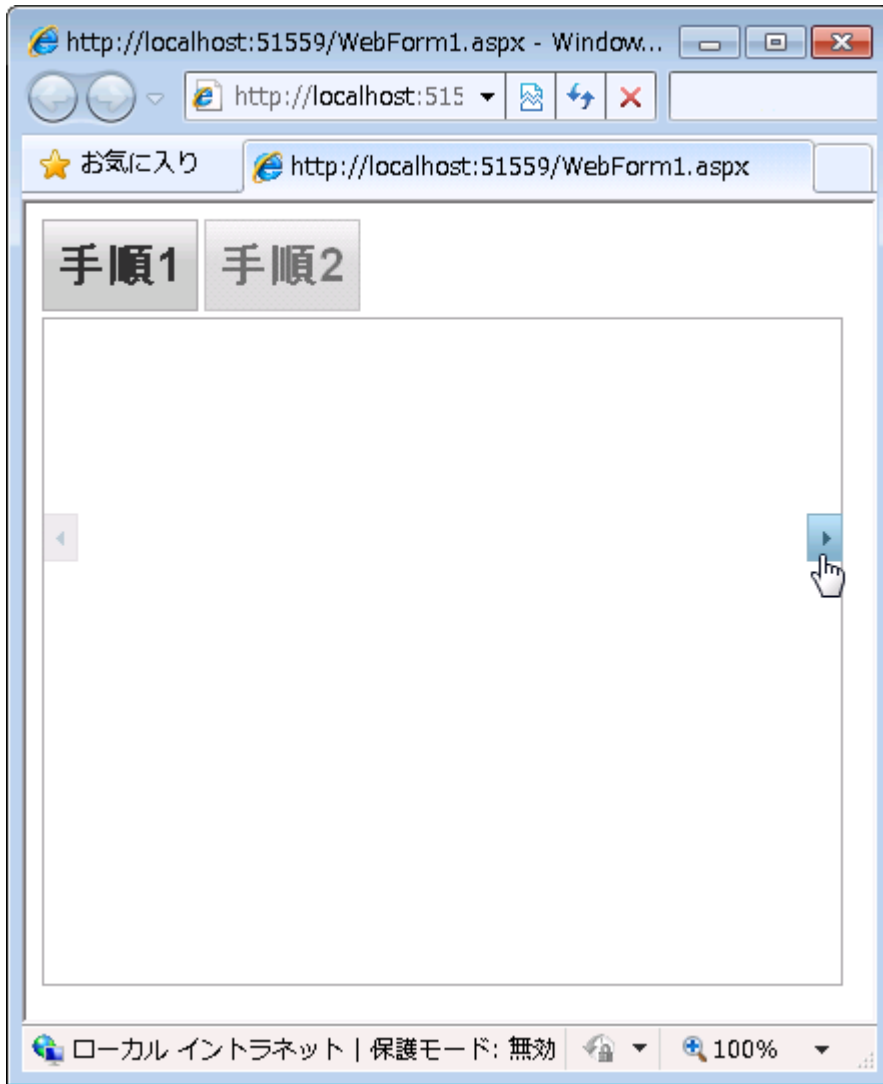
C1Wizard デザイナフォームでは、**C1Wizard** コントロールや選択して取り込む個々の **C1WizardStep** を簡単にカスタマイズできます。この手順では、C1Wizard デザイナフォームを使用して、3つの **C1WizardSteps** を **C1Wizard** コントロールに追加し、その動作を変更します。ここで示す手順を実行する前に、「[手順 1: ページへのコントロールの追加](#)」トピックの手順を実行するか、新しい Web サイト プロジェクトを作成し、ページに C1Wizard コントロールを追加する必要があることに注意してください。

以下の手順を実行します。

1. **C1Wizard1** のスマートタグをクリックして[**C1Wizard タスク**]メニューを開き、**デザイナを開く**を選択します。**C1Wizard デザイナフォーム**が開きます。
2. **C1Wizard デザイナフォーム**で、**子項目の追加**ボタンをクリックしてステップを選択します。



3. 手順 2を繰り返して別の**ステップ**を **Wizard** コントロールに追加します。
4. **C1Wizard1** を選択し、プロパティグリッドで以下のプロパティを設定します。
 - **Height** プロパティを「**300px**」に設定します。
 - **Width** プロパティを「**400px**」に設定します。
 - **NavButtons** プロパティを「**Edge**」に設定します。
5. **OK**をクリックし、C1Wizard デザイナフォームを保存して閉じます。
6. プロジェクトを実行し、追加した2種類のステップをナビゲーションできることを確認します。



次の手順では、**C1Wizard** コントロールにコンテンツを追加し、その外観をカスタマイズします。

手順 3: コントロールへのコンテンツの追加

この手順では、**C1Wizard** コントロールにコンテンツを追加します。コントロールにコンテンツを追加する操作は、コントロールのボディをクリックし、テキストを入力したり、ツールボックスからコントロールを追加する場合と同様に簡単です。ここで示す手順を実行する前に、「[手順 2: C1Wizard デザイナフォームでの作業](#)」に示された手順を完了して **C1Wizard** コントロールに3つのステップを追加します。

以下の手順を実行します。

1. **C1Wizard** コントロールのコンテンツ領域内でクリックし、「これは手順 1です。」を入力します。
C1Wizard コントロールの最初のページにテキストコンテンツが追加されます。
2. ソースビューに切り替えて **C1Wizard** コントロールの構造を確認し、以下に示す タグを2番目のステップの `<cc1:C1WizardStep>` に追加します。これは次のように示されます。

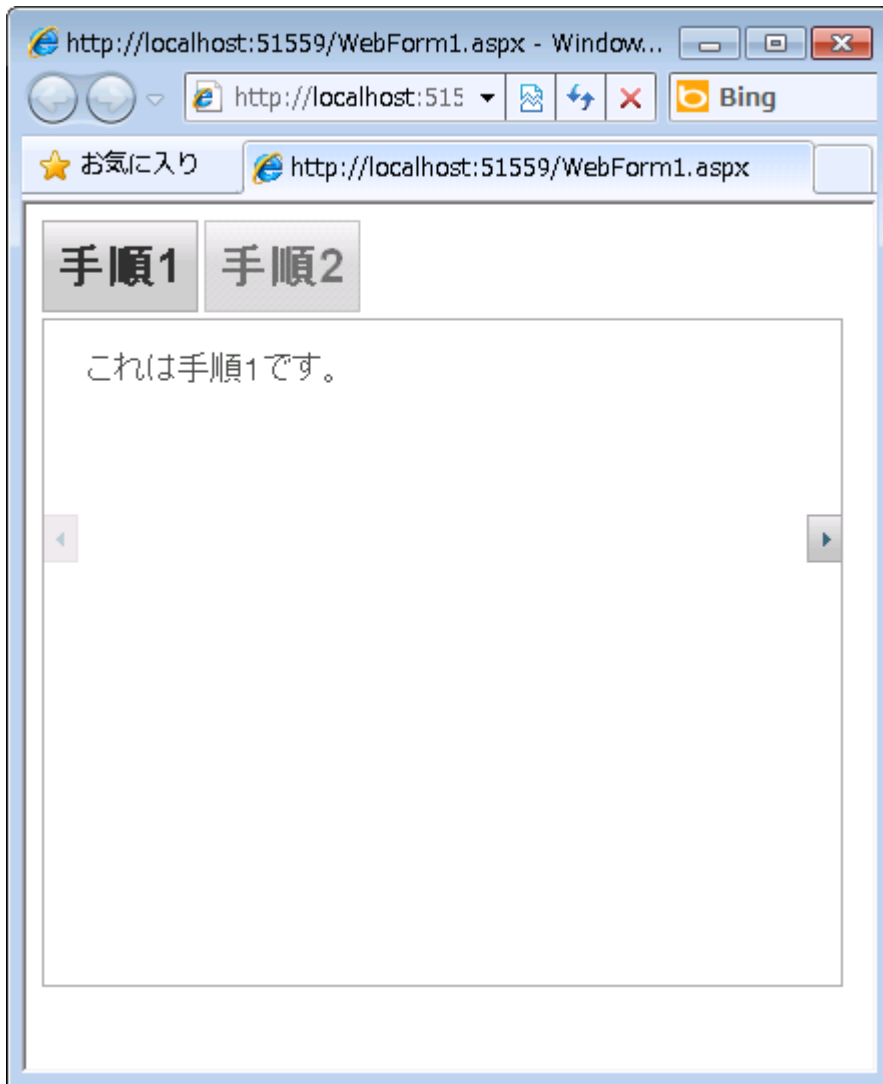
ソースビュー

```
<cc1:C1WizardStepID="C1WizardStep03" runat="server">
  <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
</cc1:C1WizardStep>
```

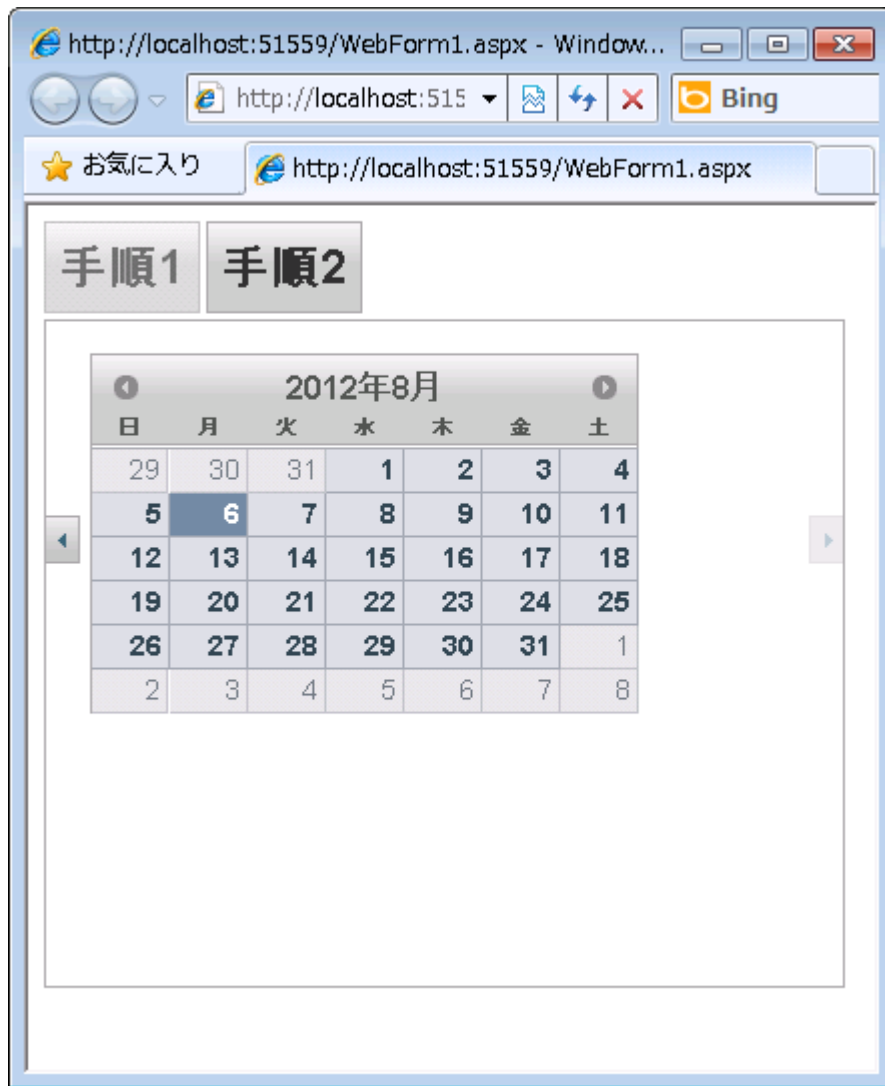
これにより、ウィザードの2ページ目に **Calendar** コントロールが追加されます。

Wizard for ASP.NET Web Forms

3. [F5]を押してアプリケーションを実行し、最初のステップがフォーカスされた状態でウィザードコントロールが表示されることを確認します。



4. 右矢印をクリックし、2番目のステップがフォーカスされることを確認します。



デザイン時のサポート

C1Wizard は、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、C1Wizard のデザイン時環境を使用して C1Wizard 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

C1Wizard 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、**C1Wizard** 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。

C1Wizard 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、**C1Wizard** 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。

[**C1Wizard** タスク]メニューにアクセスするには、**C1Wizard** 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。

[**C1Wizard** タスク]メニューが開きます。



[**C1Wizard** タスク]メニューは、以下のように操作します。

- **デザイナを開く**
デザイナを開くをクリックすると、プロパティウィンドウをスクロールしなくても **C1Wizard** の要素を手軽に設定できる **C1Wizard デザイナフォーム**が開きます。コントロールのコンテンツをロードし、保存して、追加の **C1WizardStep** を追加することができます。**C1Wizard デザイナフォーム**の詳細については、「[C1Wizard デザイナフォーム](#)」を参照してください。
- **テーマ**
テーマを組み込みスキンの1つに設定します。
- **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

C1Wizard 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。

C1Wizard 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。



C1Wizard 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、[C1Wizard タスク]メニューが開きます。

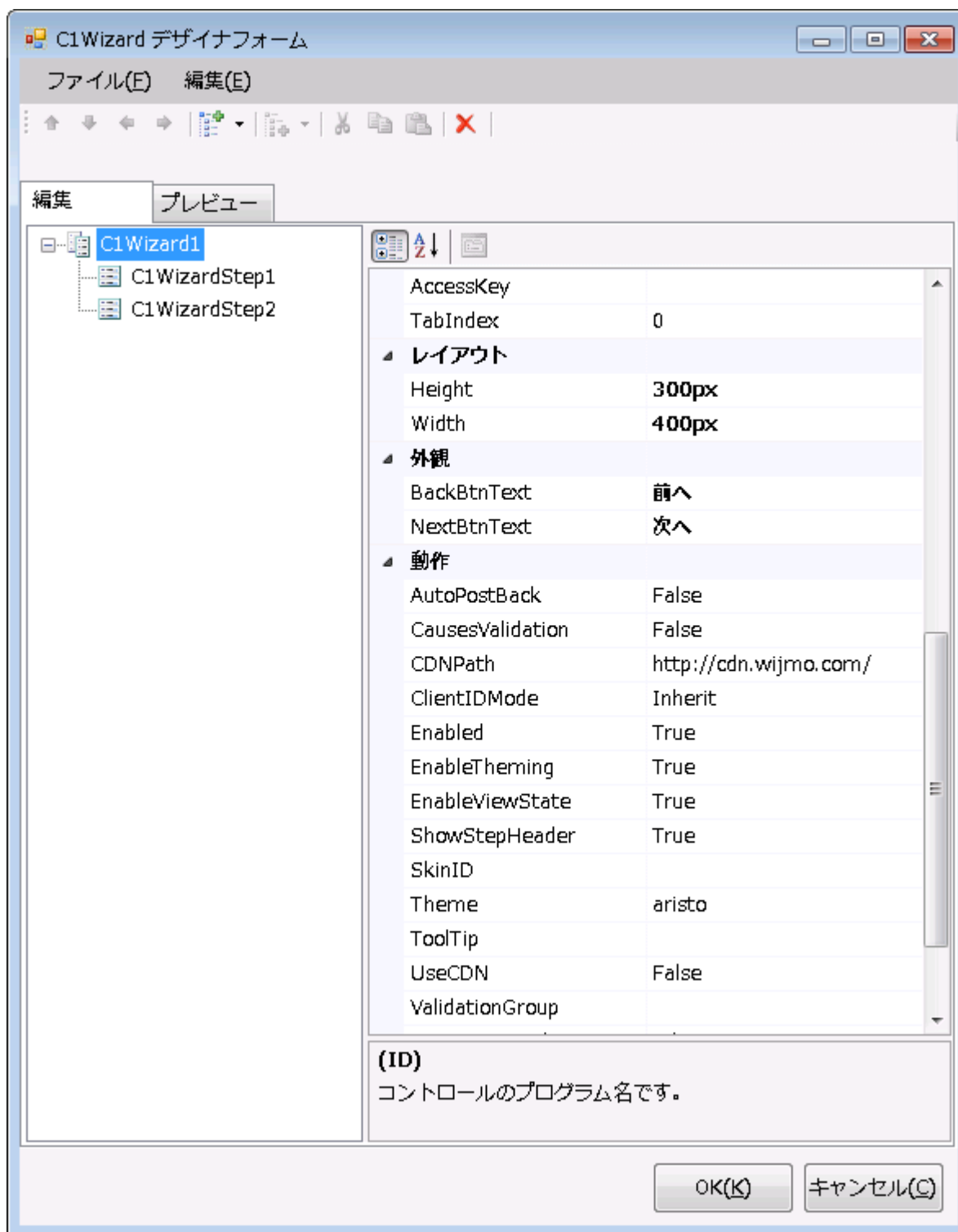
- デザインを開く

項目 [デザインを開く]をクリックすると、プロパティウィンドウをスクロールしなくても **C1Wizard** の要素を手軽に設定できる **C1Wizard デザイナフォーム**が開きます。コントロールのコンテンツをロードし、保存して、追加の **C1WizardStep** を追加することができます。**C1Wizard デザイナフォーム**の詳細については、「[C1Wizard デザイナフォーム](#)」を参照してください。

デザイナフォーム

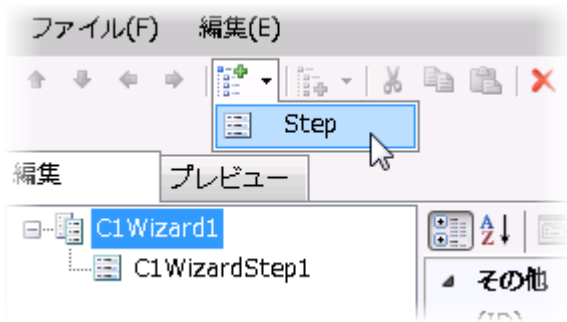
C1Wizard デザイナフォームでは、そのプロパティウィンドウをスクロールしなくても**C1Wizard** コントロールの要素を手軽に設定できます。**C1Wizard デザイナフォーム**を使用すると、コントロールのコンテンツをロードし、保存して、追加のページを追加することができます。

C1Wizard デザイナフォームにアクセスするには、[C1Wizard タスク]メニュー(詳細については、「[C1Wizard スマートタグ](#)」を参照)から項目 [デザインを開く]を選択するか、デザイン時に C1Wizard コントロールを右クリックして[デザインを開く]を選択します。デザイナは次のように表示されます。



C1WizardStep の追加

C1WizardStep を C1Wizard に追加するには、[C1Wizard] をクリックし、ツールバーで「子項目の追加」ボタンをクリックして [ステップ] を選択します。



C1WizardStep の削除

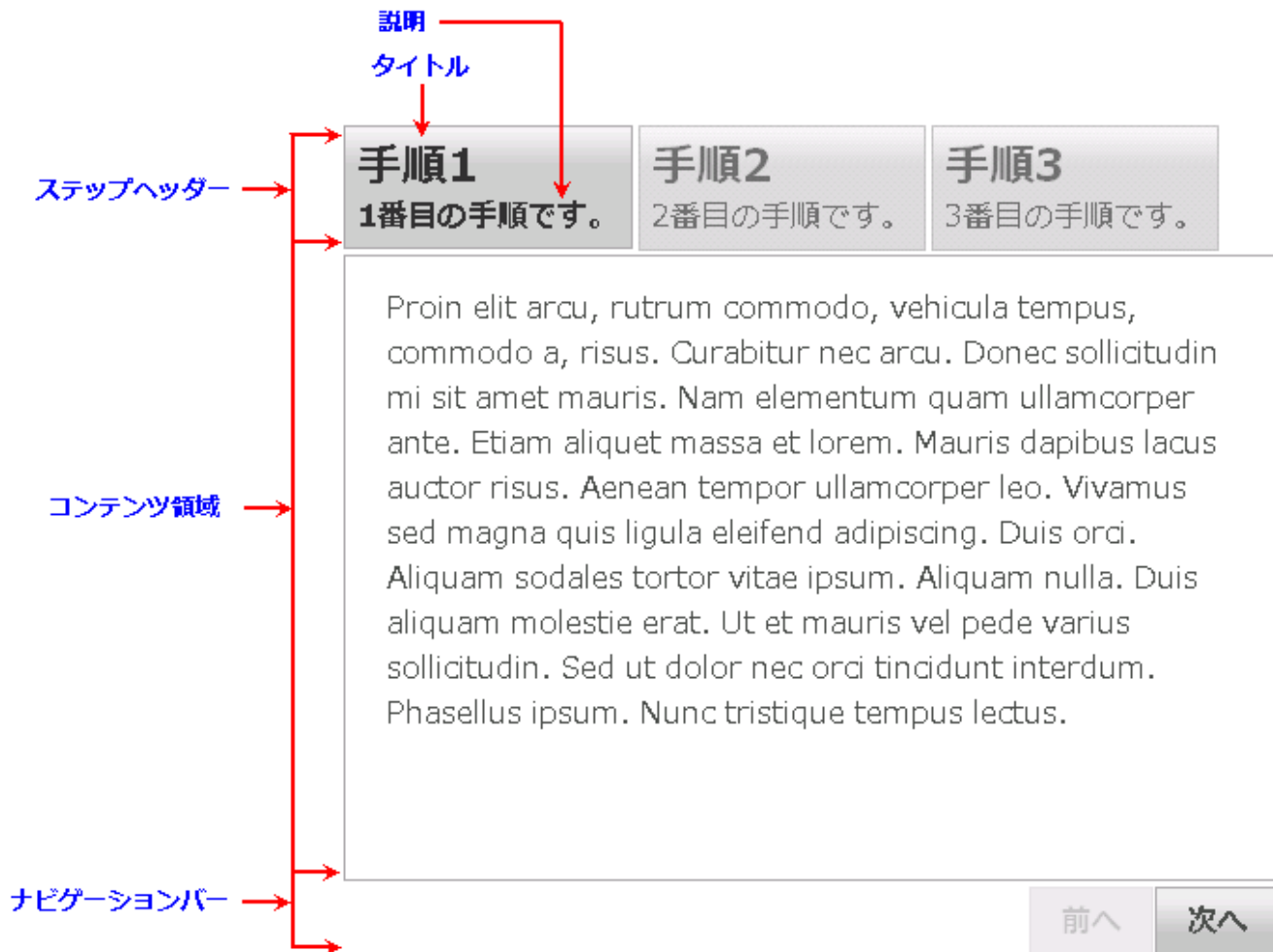
C1WizardStep を削除するには、既存の **C1WizardStep** を選択し、ツールバーから「**削除**」ボタンを選択するか、[編集]メニューまたは項目を右クリックし、表示されるコンテキストメニューから「**削除**」を選択します。

項目の名前変更

C1WizardStep の名前を変更するには、既存の **C1WizardStep** を選択し、ツールバーから「**名前の変更**」ボタンを選択するか、[編集]メニューまたは項目を右クリックし、表示されるコンテキストメニューから「**名前の変更**」を選択します。テキストを入力し、その ID を変更します。

C1Wizard の要素

C1Wizard コントロールはページのコレクションを表示するために使用されます。ページは一度に1ページずつ表示され、ページには大量のコンテンツを保持できるので画面のスペースを無駄にしなくて済みます。ページは**C1WizardStep** クラスによって表されます。各ページは、オプションのナビゲーションボタンを使用して移動できます。



ステップヘッダー

ステップヘッダーセクションには、ウィザードのステップごとのヘッダーが含まれています。ヘッダーを表示しない場合は、ウィザードの **ShowStepHeader** プロパティを **False** にします。

ステップヘッダーには以下の項目が含まれています。

- **タイトル**: ステップヘッダーのタイトルは **C1WizardStep** の **Title** プロパティを使用して設定されます。
- **説明**: ステップヘッダーの説明は **C1WizardStep** の **Description** プロパティを使用して設定されます。

ステップヘッダーのスタイルは **StepHeaderTemplate** を使用して書式設定できます。詳細については、「[ヘッダーテンプレート](#)」を参照してください。

コンテンツ領域

C1Wizard コントロールは、書式設定済みテキスト、任意のコントロール、およびテーブルなどのコンテンツを保持することができます。**C1Wizard** には、**C1WizardStep** タグ内でサーバーコントロールを宣言するだけで、任意のコントロールを追加できます。たとえば、次のマークアップでは、**Button** サーバーコントロールが **C1Wizard** の最初のページ内に追加されます。

ソースビュー

```
<cc1:C1WizardStep ID="C1WizardStep01" runat="server">  
<asp:Button ID="Button1" runat="server" Text="ボタン" />  
</cc1:C1WizardStep>
```

コンテンツ領域

C1Wizard コンテンツ領域は、テキスト、画像、およびコントロールを保持することができます。詳細については、「[ステップコンテンツの追加と操作](#)」を参照してください。

PanelTemplate プロパティには、**C1Wizard** コントロールに追加されるパネルの HTML テンプレートが用意されています。

ナビゲーションバー

デフォルトの**C1Wizard** コントロールでは、ナビゲーションバーにユーザーが1つのステップから次のステップに移動できるボタンが含まれています。ただし、ナビゲーションをコントロールの側面に表示したり、コントロールから完全に削除したりすることもできます。詳細については、「[ナビゲーションボタン](#)」を参照してください。

C1Wizard の外観および動作

以下のトピックでは、C1Wizard の外観および動作に関連する機能について説明します。これらの機能の中には、ロード時のコントロールの動作に影響を与えるものや、ユーザーによるコントロールのインタラクティブ操作に影響を与えるものがあります。

テーマ

C1Wizard 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 ASP.NET Web Forms スタジオコントロールはそれに応じてスキンされます。テーマは**C1Wizard** コントロールに次のように表示されます。

Arctic	<table border="1"><tr><td data-bbox="284 667 592 801">手順1 1番目の手順です。</td><td data-bbox="592 667 900 801">手順2 2番目の手順です。</td><td data-bbox="900 667 1177 801">手順3 3番目の手順です。</td></tr><tr><td colspan="3" data-bbox="284 801 1294 1037">Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.</td></tr><tr><td colspan="3" data-bbox="1023 1037 1294 1122">前へ 次へ</td></tr></table>	手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。	Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.			前へ 次へ		
手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。								
Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.										
前へ 次へ										
Aristo	<table border="1"><tr><td data-bbox="284 1133 592 1267">手順1 1番目の手順です。</td><td data-bbox="592 1133 900 1267">手順2 2番目の手順です。</td><td data-bbox="900 1133 1224 1267">手順3 3番目の手順です。</td></tr><tr><td colspan="3" data-bbox="284 1267 1294 1503">Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.</td></tr><tr><td colspan="3" data-bbox="1023 1503 1294 1592">前へ 次へ</td></tr></table>	手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。	Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.			前へ 次へ		
手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。								
Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.										
前へ 次へ										
Cobalt	<table border="1"><tr><td data-bbox="284 1603 592 1738">手順1 1番目の手順です。</td><td data-bbox="592 1603 900 1738">手順2 2番目の手順です。</td><td data-bbox="900 1603 1224 1738">手順3 3番目の手順です。</td></tr><tr><td colspan="3" data-bbox="284 1738 1294 1973">Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.</td></tr><tr><td colspan="3" data-bbox="1023 1973 1294 2072">前へ 次へ</td></tr></table>	手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。	Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.			前へ 次へ		
手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。								
Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.										
前へ 次へ										

Midnight	手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。
Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.			
			<input type="button" value="前へ"/> <input type="button" value="次へ"/>
Rocket	手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。
Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.			
			<input type="button" value="前へ"/> <input type="button" value="次へ"/>
Sterling	手順1 1番目の手順です。	手順2 2番目の手順です。	手順3 3番目の手順です。
Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.			
			<input type="button" value="前へ"/> <input type="button" value="次へ"/>

C1Wizard 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。テーマに関する詳細については、「[組み込みテーマの使用](#)」を参照してください。

ヘッダーテンプレート

ヘッダーテンプレート

StepHeaderTemplate プロパティには、**C1Wizard** コントロールの各ステップのヘッダーに使用する HTML テンプレートが用意されています。デフォルトでは、**C1Wizard** の **StepHeaderTemplate** プロパティは「<h2>#{title}</h2>#{desc}」に設定され、ステップに **Title** と **Description** プロパティを設定すると、ヘッダーが次のように表示されます。

Wizard for ASP.NET Web Forms

手順1

1番目の手順です。

手順2

2番目の手順です。

ただし、テンプレートにいくつかの変更を加えるだけで、ヘッダーを簡単に変更することができます。たとえば、**StepHeaderTemplate** プロパティの現在の設定を「`{desc}<h5>#{title} 追加したテキスト </h5>`」に置き換えるとヘッダーは次のようになります。

1番目の手順です。

手順1追加したテキスト

2番目の手順です。

手順2追加したテキスト

上記の例では、テンプレートの **Description** ("`{desc}`") と **Title** ("`{title}`") の位置を切り替えました。また、タイトルが見出し5のスタイルで表示され、タイトルの後に文字列「追加したテキスト」が表示されています。コントロールに追加する各ステップはこのヘッダーテンプレートの書式を採用します。

コンテンツテンプレート

PanelTemplate プロパティには、**C1Wizard** コントロールの各ステップのコンテンツ領域に使用する HTML テンプレートが用意されています。デフォルトでは、「`<div></div>`」に設定され、すべてのコンテンツが `<div>` ブロック内に置かれますが、各ステップのコンテンツ領域を特定のスタイルのリストに適合される場合はこのテンプレートを変更することができます。

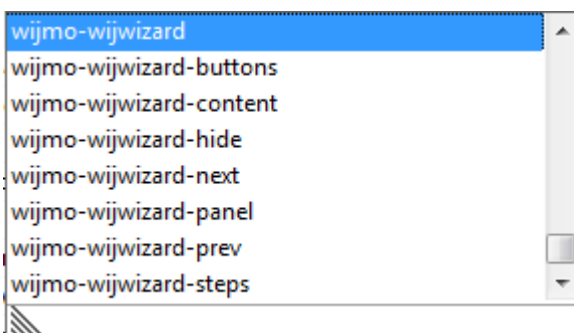
CSS セレクタ

CSS を使用して多くの **C1Wizard** 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 **C1Wizard** 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。

C1Wizard CSS セレクタは以下の `wijmo-wijwizard` で開始します。



ナビゲーションボタン

C1Wizard には、**Common** と **Edge** という2つのナビゲーションタイプが用意されています。**Common** スタイルは〈戻る〉および〈次へ〉ボタンで表示され、**Edge** スタイルはコントロールの左側の戻る矢印およびコントロールの右側の次へ矢印で表示されます。

下表は使用可能なナビゲーションスタイルを比較しています。



ナビゲーションタイプは **NavButtons** プロパティによって設定されます。**NavButtons** プロパティを上述の **Common** または **Edge** スタイルに設定したり、プロパティを **Auto** または **None** 設定するように選択したりできます。**Auto** は、コントロールの高さと幅に応じて適切なスタイルが自動的に選択されるようにし、**None** は、ナビゲーションを完全に削除し、カスタムフォームのナビゲーションを使用できるようにします。このコントロールは **ComponentOne for ASP.NET Web Forms** の **C1Pager** コントロールと組み合わせた場合にうまく機能します。

オートプレイ機能

AutoPlay プロパティを オートプレイ機能 に設定することで、ページビューが自動的に移行するように **C1Wizard** コントロールを設定できます。デフォルトで、自動ページ移行の間に 3,000 ミリ秒 (3秒) の遅延が存在しますが、**Delay** プロパティを設定し、これを調整することができます。**Loop** プロパティを **True** に設定しない場合、ページは **C1Wizard** の最後のページで自動的に停止します。**Loop** を **True** に設定すると、コントロールは最初のページに戻り、オートプレイ処理を繰り返します。

ツールチップ

C1WizardStep.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。ToolTip 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、**C1WizardStep.ToolTip** プロパティを文字列に設定することで、**C1Wizard** 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 ToolTip 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
C1WizardStep.ToolTip = "Hello World!"
```

C# コードの書き方

C#

```
C1WizardStep.ToolTip = "Hello World!";
```

C1WizardStep.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。

アクティブインデックス

C1Wizard 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、**ActiveIndex** 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ステップを変更するには、**ActiveIndex** 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つのステップがあり、実行時に最後のページが選択されるようにする場合は、**ActiveIndex** プロパティを **3** に設定します。

タスク別ヘルプ

タスク別ヘルプセクションでは、Visual Studio ASP.NET 環境でのプログラミングに精通し、**Wizard 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、**C1Wizard** 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 **C1Wizard** 機能を使用したプロジェクトを作成できます。

ステップコンテンツの追加と操作


C1Wizard コントロールのステップでは、任意のコントロールの組み込み、テキストの表示、および外部コンテンツの表示が可能です。次の3つのトピックでは、**C1Wizard** コントロールのステップへのコンテンツの追加について説明します。

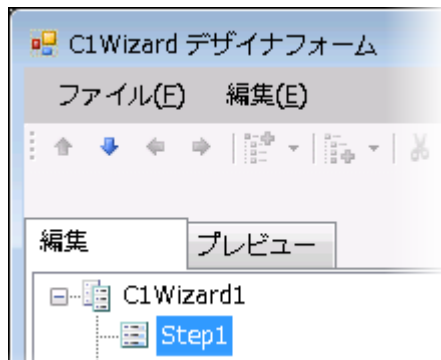
ステップの追加

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、**C1Wizard** コントロールにページを追加する方法を説明します。

デザインビューの場合

以下の手順を実行します。

1. スマートタグをクリックして、[**C1Wizard タスク**]メニューを開きます。デザイナを開くを選択します。
[**C1Wizard デザイナフォーム**]ダイアログボックスが開きます。
2. <子項目の追加>ボタン  をクリックして、**C1WizardStep** を **C1Wizard** コントロールに追加します。ツリービューに「手順1」と表示されます。



3. <OK>をクリックして、[**C1Wizard デザイナフォーム**]ダイアログボックスを閉じます。

ソースビューの場合

<cc1:C1Wizard> タグの間に次のマークアップを追加します。

ソースビュー

```
<Nodes>
<cc1:C1WizardStep ID="Step1" runat="server" Text="手順1">
</cc1:C1WizardStep>
</Nodes>
```

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.C1Wizard
```

C# コードの書き方

```
C#
using Cl.Web.Wijmo.Controls.C1Wizard;
```

2. 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
Dim WizardStep1 As New C1WizardStep()
WizardStep1.Text = "Wizard手順1"
C1Wizard1.Controls.Add(WizardStep1)
```

C# コードの書き方

```
C#
C1WizardStep WizardStep1 = new C1WizardStep();
WizardStep1.Text = "Wizard手順1";
C1Wizard1.Controls.Add(WizardStep1);
```

3. プログラムを実行します。

ステップへのコントロールの追加

ドラッグ&ドロップ操作のみで、XHTML、またはコードを使用して、**C1Wizard** の個々のステップに任意のコントロールを追加できます。このトピックでは、標準 **Button** コントロールをステップに追加する方法について説明します。このトピックは、コントロールに少なくとも1つのステップを追加済みであることが前提となります(「[ステップの追加](#)」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. デザイナーで、コントロールを追加するステップを選択します。ステップを選択するには、**C1Wizard** コントロールを選択し、ステップをクリックします。
2. Visual Studio ツールボックスから **Button** コントロールを選択し、それを **C1WizardStep** にドラッグします。

ソースビューの場合

以下の手順を実行します。

1. コントロールを追加するステップの `<cc1:C1WizardStep>` タグを確認し、それらの間に次のタグを置きます。

ソースビュー

```
<asp:Button ID="Button1" runat="server" Text="ボタン" />
```

2. プログラムを実行し、**Button** コントロールがステップに表示されていることを確認します。

コードビューの場合

以下の手順を実行します。

1. **Button** コントロールを作成し、次のコードを **Page_Load** イベントに入力することによって、そのコントロールにテキストを追加します。

Visual Basic コードの書き方

Visual Basic

```
Dim nuButton As Button = New Button()  
nuButton.Text = "Hello World!"
```

C# コードの書き方

C#

```
Button nuButton = new Button();  
nuButton.Text = "Hello World!";
```

2. **Button** コントロールをステップに追加します。

Visual Basic コードの書き方

Visual Basic

```
C1WizardStep1.Controls.Add(nuButton)
```

C# コードの書き方

C#

```
C1WizardStep1.Controls.Add(nuButton);
```

3. プログラムを実行し、**Button** コントロールがステップに表示されていることを確認します。

✔このピックは、次のことを示します。

下図はボタンコントロールを備えた **C1WizardStep** を示します。



ステップへのキャプションの追加

ステップにキャプションを追加する場合は、**GroupingText** プロパティを文字列に設定します。**GroupingText** プロパティが設定されると、ステップのコンテンツの周囲に枠が表示されます。

デザインビューの場合

以下の手順を実行します。

1. スマートタグをクリックして、[C1Wizard タスク]メニューを開きます。**デザイナを開く**を選択します。
[C1Wizard デザイナフォーム]ダイアログボックスが開きます。
2. ツリービューで、キャプションを追加するステップを選択し、そのプロパティのリストがプロパティグリッドに表示されるようにします。
3. プロパティグリッドで、**GroupingText** プロパティを文字列に設定します。この例では、"グループ化のテキスト" に設定します。
4. [F5]を押してプログラムを実行し、コンテンツに "グループ化のテキスト" というキャプションが付けられていることを確認します。

ソースビューの場合

以下の手順を実行します。

1. `GroupingText="グループ化のテキスト"` を`<cc1:C1WizardStep>` タグに追加して、マークアップを次のように記述します。

ソースビュー

```
<cc1:C1WizardStep ID="C1WizardStep1" runat="server" Title="手順1"  
GroupingText="グループ化のテキスト">
```

2. [F5]を押してプログラムを実行し、コンテンツに "グループ化のテキスト" というキャプションが付けられていることを確認します。

コードビューの場合

以下の手順を実行します。

1. 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


```
C1Wizard1.GroupingText = "グループ化のテキスト"
```

C# コードの書き方

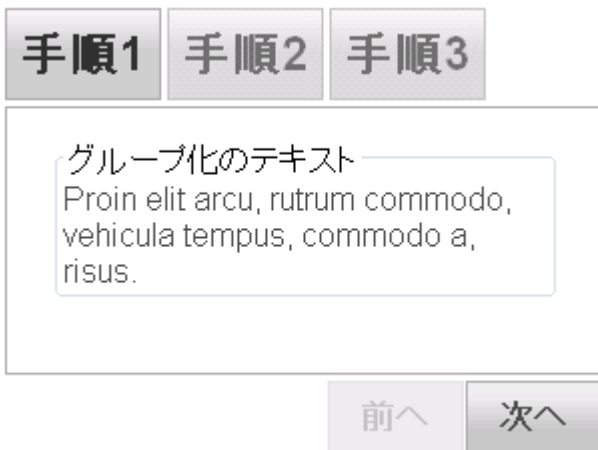
C#

```
C1Wizard1.GroupingText = "グループ化のテキスト";
```

2. [F5]を押してプログラムを実行し、コンテンツに "グループ化のテキスト" というキャプションが付けられていることを確認します。

 このトピックは、次のことを示します。

下図は `GroupingText` プロパティが設定された `C1WizardStep` を示します。

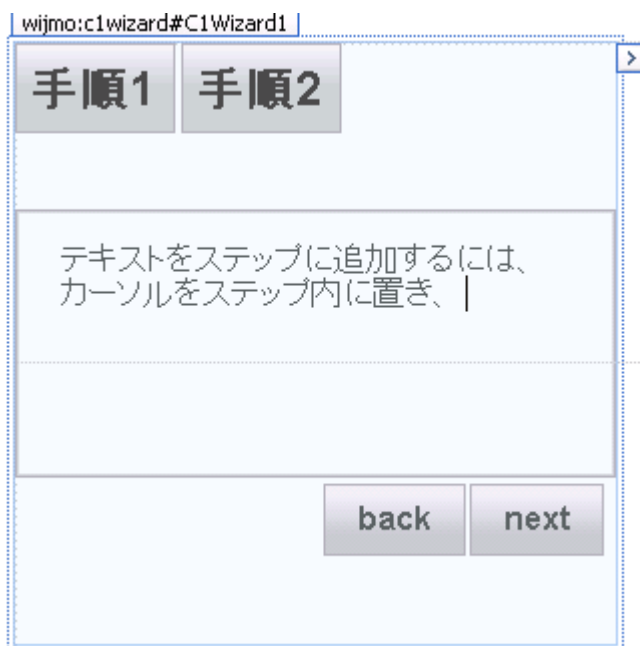


ステップへのテキストの追加

このトピックでは、デザイナーとマークアップを使用して、テキストを **C1Wizard** コントロールに追加する方法を学びます。

デザインビューの場合

テキストをステップに追加するには、カーソルをステップ内に置き、テキストをそのステップに入力(またはコピー)するだけです。



テキストをステップに追加したら、Visual Studio の書式設定ツールバー(このツールバーを表示するには、[表示]→[ツールバー]→[書式設定]を選択)を使用して、テキストを書式設定できます。下図は、実行時に表示されるように書式設定済みテキストを含む **C1WizardStep** を示します。

手順1

手順2

グループ化のテキスト

テキストを**ステップ**に追加するには、カーソルをステップ内に置き、テキストをその**ステップ**に入力(またはコピー)するだけです。

テキストを**ステップ**に追加したら、Visual Studioの書式設定ツールバー(このツールバーを表示するには、[表示] → [ツールバー] → [書式設定]を選択)を使用して、**テキストを書式設定できます。**

前へ

次へ

ソースビューの場合

ソースビューでは、`<cc1:C1WizardStep>` タグの間にテキストを置くことによって、**C1WizardStep** にテキストを追加できます。テキストを書式設定するには、マークアップを使用します。

ソースビューでテキストを **C1WizardStep** に追加するには、以下の手順を実行します。

1. ソースビューに切り替えて、ページに以下の CSS スタイルを追加します。

ソースビュー

```
<style type="text/css">
    .style1
    {
        font-family:"Comic Sans MS";
        font-size:large;
        color:orange;
    }
    .style2
    {
        font-family:Papyrus;
        font-size:large;
        color:#0066FF;
    }
</style>
```

2. `<cc1: C1WizardStep>` タグの間に次のマークアップを追加します。

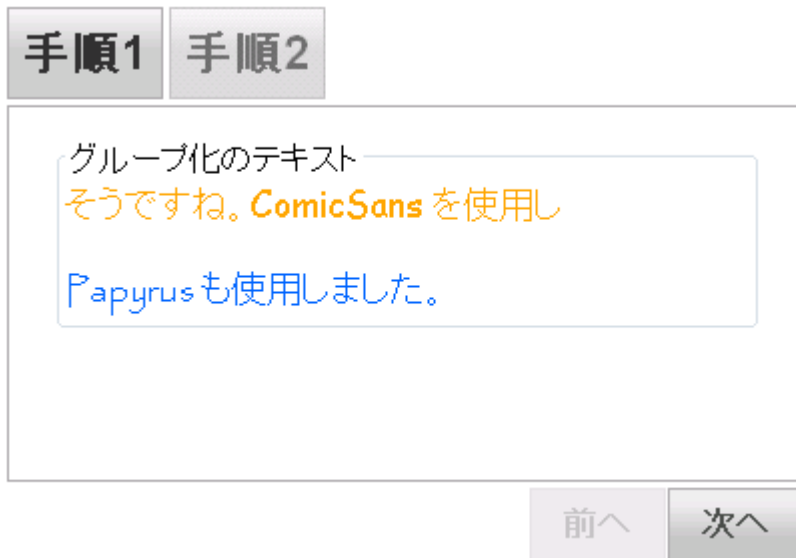
ソースビュー

```
<span class="style1">そうですね。ComicSans を使用し</span>

<span class="style2">Papyrus も使用しました。</span>
```

style1 が最初の文に適用され、**style2** が2番目の文に適用されることに注意してください。

3. [F5]を押してプロジェクトを実行し、**C1Wizard** コントロールの **C1WizardStep** にテキストが追加されていることを確認します。結果は、次の図のようになります。



C1Wizard の外観の変更


このセクションのトピックは、カスタマイズしてテーマを利用し、ヘッダーテンプレートをカスタマイズする方法を説明します。

ステップヘッダーの削除

ShowStepHeader プロパティを **False** に設定してステップヘッダーを削除できます。

デザインビューの場合

以下の手順を実行します。

1. スマートタグ  をクリックして、[**C1Wizard タスク**]メニューを開きます。**デザイナを開く**を選択します。**[C1Wizard デザイナフォーム]**ダイアログボックスが開きます。
2. ツリービューで、階層の最上位にある**C1Wizard** コントロール(デフォルトでは "C1Wizard1")を選択し、そのプロパティのリストにアクセスします。
3. プロパティグリッドで、**ShowStepHeader** プロパティを **False** 文字列に設定します。

ソースビューの場合

`ShowStepHeader="False"` を `<cc1:C1Wizard>` タグに追加して、マークアップを次のように記述します。

ソースビュー

```
<cc1:C1Wizard ID="C1Wizard1" runat="server" ShowStepHeader="False">
```

コードビューの場合

コードビューで、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

Wizard for ASP.NET Web Forms

Visual Basic

```
C1Wizard1.ShowStepHeader = False
```

C# コードの書き方

C#


```
C1Wizard1.ShowStepHeader = false;
```

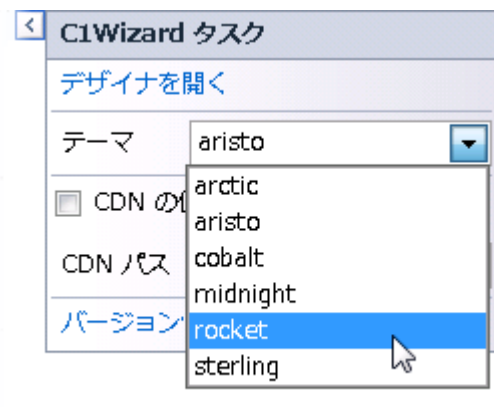
組み込みテーマの使用

C1Wizard 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. **C1Wizard** のスマートタグ  をクリックして、**[C1Wizard 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**C1Wizard** 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 **C1Wizard** のテーマを変更するには、`VisualStyle="rocket"` を、次のように `<cc1:C1Wizard>` タグに追加します。\\

ソースビュー

```
<cc1:C1Wizard ID="C1Wizard1" 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
C1Wizard1.Theme = "rocket"
```

C# コードの書き方

```
C#
C1Wizard1.Theme = "rocket";
```

3. プログラムを実行します。

✔このトピックは、次のことを示します。

次の図は、**rocket** テーマが設定された **C1Wizard** コントロールを示しています。

手順1 手順2

Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.

前へ

次へ



CSS セレクタの使用

コントロールで使用可能な CSS セレクタを使用して、**C1Wizard** コントロールの外観を変更できます。これはアプリケーションのソースビューで実行されます。

1. プロジェクトのソースビューに移動し、`<cc1:C1Wizard>` タグを探します。
2. `CssClass="wijmo-c1wizard"` をタグに追加します。HTML が次のようになります。

ソースビュー

```
<cc1:C1Wizard ID="C1Wizard1" runat="server" CssClass="wijmo-c1wizard"
Width="400px">
```

3. ページ上の `<asp:Content>` タグの最初のセットを探して、以下のタグセットを挿入します。

ソースビュー

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

4. <style> タグの間に以下の CSS スタイルを追加します。

ソースビュー

```
.wijmo-clwizard .ui-button
{
background:#B0C4DE;
}
.wijmo-clwizard .ui-button-text
{
color:#236b8e; font-family:Algerian;
}
.wijmo-clwizard .ui-widget-header
{
background:#B0C4DE; font-family:Algerian;
}
.wijmo-clwizard-content
{
background-color:#c3e4ed; font-family:Algerian; font-size:large; color: #000000;
}
```

5. [F5]を押して、アプリケーションを実行します。**C1Wizard** コントロールは次の図のように表示されます。

手順1 手順2

Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus.

前へ

次へ



C1Wizard のステップのオートプレイ

1つのプロパティを設定することで、**C1Wizard** コントロールがステップを自動的に移動するかどうかを選択することができます。移行間に遅延を設定し、オートプレイをカスタマイズすることもできます。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、オートプレイを設定し、遅延時間をカスタマイズする方法を示します。


デザインビューの場合

オートプレイを設定するには、以下の手順を実行します。

1. **C1Wizard** のスマートタグ  をクリックして、[**C1Wizard** タスク]メニューを開き、[**ウィザードデザイナー**]を選択します。

C1Wizard デザイナーフォームが開きます。

- ツリービューで、**C1Wizard1** のプロパティのリストを選択し、次のように設定します。
 - AutoPlay** プロパティを True に設定します。
 - Delay** プロパティを設定します。この例では、ページが切り替えられる前に 1.5 秒経過するように、このプロパティを "1500" ミリ秒に設定します。

 **注意:** デフォルトでは、プレイスルーはコントロールの最後のステップで終了します。**Loop** プロパティを **True** に設定して、1回のランスルー終了後、**C1Wizard** コントロールがページを自動的に循環するように設定することもできます。

- <OK>を押してデザイナを閉じ、[F5]を押してプロジェクトを実行します。コントロールのステップが自動的に再生されることを確認します。


ソースビューの場合

以下の手順を実行します。

- ソースビューで**AutoPlay** を設定するには、AutoPlay="True" と "Delay="1500" を <cc1:C1Wizard> タグに追加します。HTML が次のようになります。

ソースビュー

```
<cc1:C1Wizard ID="C1Wizard1" runat="server" Height="250px" Width="300px"
AutoPlay="True" Delay="1500">
```

 **注意:** デフォルトでは、プレイスルーはコントロールの最後のステップで終了します。コントロールのタグに Loop=True を追加して、1回のランスルー終了後、C1Wizard コントロールがページを自動的に循環するように設定することもできます。

- プログラムを実行し、コントロールのステップが自動的に再生されることを確認します。

コードビューの場合

以下の手順を実行します。

- 次のコードを **Page_Load** イベントに追加することによって、**AutoPlay**と **Delay** プロパティを設定します。

Visual Basic コードの書き方


Visual Basic

```
C1Wizard1.AutoPlay = True
C1Wizard1.Delay = 1500
```

C# コードの書き方

C#

```
C1Wizard1.AutoPlay = true;
C1Wizard1.Delay = 1500;
```

 **注意:** デフォルトでは、プレイスルーはコントロールの最後のステップで終了します。**Loop** プロパティを **True** に設定して、1回のランスルー終了後、**C1Wizard** コントロールがページを自動的に循環するように設定することもできます。

- プログラムを実行し、コントロールのステップが自動的に再生されることを確認します。

コードによるコントロールの作成

場合によっては、コードで **C1Wizard** コントロールをプロジェクトに追加する必要があるかもしれません。このトピックでは、C# と Visual Basic のコードを使用して、2つの**C1WizardStep** オブジェクトを持つ **C1Wizard** コントロールを作成する方法を学びます。

以下の手順を実行します。

1. **PlaceHolder** コントロールをページに追加します。
2. デザインビューで、ページをダブルクリックして **Page_Load** イベントをプロジェクトに追加し、コードエディタに切り替えます。
3. 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.C1Wizard
```

C# コードの書き方

```
C#
using Cl.Web.Wijmo.Controls.C1Wizard;
```

4. **C1Wizard** オブジェクトを作成し、その **Width** プロパティと **Height** プロパティを設定してから、**Page_Load** イベントに次のコードを配置することによって、そのオブジェクトをプロジェクトに追加します。

Visual Basic コードの書き方

```
Visual Basic
Dim NewWizard As C1Wizard = New C1Wizard()
NewWizard.Width = 300
NewWizard.Height = 200
PlaceHolder1.Controls.Add(NewWizard)
```

C# コードの書き方

```
C#
C1Wizard NewWizard = new C1Wizard();
NewWizard.Width = 300;
NewWizard.Height = 200;
PlaceHolder1.Controls.Add(NewWizard);
```

5. 3つの **C1WizardStep** オブジェクトを作成し、それらを **C1Wizard** に追加します。このコードは、**Page_Load** イベントにも追加する必要があります。

Visual Basic コードの書き方

```
Visual Basic
'3つの C1WizardStep オブジェクトを作成します。
Dim C1WizardStep1 As C1WizardStep = New C1WizardStep()
Dim C1WizardStep2 As C1WizardStep = New C1WizardStep()
'WizardStep の 'Text' プロパティを設定します。
C1WizardStep1.Title = "C1Wizard手順1"
C1WizardStep2.Title = "C1Wizard手順2"
```


3つの C1WizardStep オブジェクトを C1Wizard に追加します。

```
NewWizard.Controls.Add(C1WizardStep1)
NewWizard.Controls.Add(C1WizardStep2)
```

C# コードの書き方

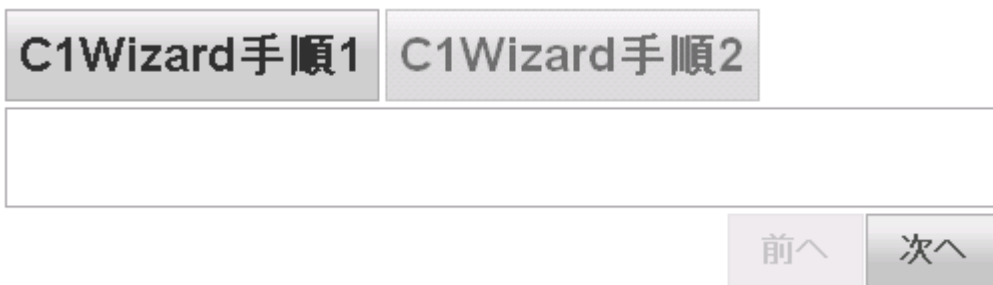
C#

```
3つの C1WizardStep オブジェクトを作成します。
C1WizardStep C1WizardStep1 = new C1WizardStep();
C1WizardStep C1WizardStep2 = new C1WizardStep();
WizardStep の「Text」プロパティを設定します。
C1WizardStep1.Title = "C1Wizard手順1";
C1WizardStep2.Title = "C1Wizard手順2";
3つの C1WizardStep オブジェクトを C1Wizard に追加します。
NewWizard.Controls.Add(C1WizardStep1);
NewWizard.Controls.Add(C1WizardStep2);
```

6. プログラムを実行します。

✔このトピックは、次のことを示します。

プロジェクトが実行されると、C1Wizard コントロールは次の図のようになります。





選択されたステップの変更

C1Wizard コントロールの **ActiveIndex** プロパティを使用すれば、プロジェクト実行時に選択されるステップを決定できます。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

以下の手順を実行します。

1. C1Wizard のスマートタグ  をクリックして、[C1Wizard タスク]メニューを開き、Wizard.SmartTag.Designer を選択します。
C1Wizard デザイナーフォームが開きます。
2. ツリービューからC1Wizard を選択し、そのプロパティリストを表示します。
3. **ActiveIndex** プロパティを指定し、その値を "1" に設定します。

 **注意:**ActiveIndex プロパティのデフォルト値が 0 であることを確認してください。この設定のままにした場合、C1Wizard は最初のタブ C1WizardStep1 が選択された状態でロードされます。

4. <OK>を押してから、プロジェクトを実行します。実行時に2番目のステップ C1WizardStep2 が選択されることを確認してください。

ソースビューの場合

Wizard for ASP.NET Web Forms

ActiveIndex="1" を<cc1:C1Wizard> タグに追加します。結果のマークアップは次のようになります。

ソースビュー

```
<cc1:C1Wizard ID="C1Wizard1" runat="server" ActiveIndex="1">
```

コードビューの場合

選択インデックスを変更するには、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

```
C1Wizard1.ActiveIndex = 1
```

C# コードの書き方

C#

```
C1Wizard1.ActiveIndex = 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、一連のステップの2番目が選択されます。

手順1

手順2

手順3

Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa metus id nunc. Duis scelerisque molestie turpis. Sed fringilla, massa eget luctus malesuada, metus eros molestie lectus, ut tempus eros massa ut dolor. Aenean aliquet fringilla sem.

前へ

次へ