

# Menu for ASP.NET Web Forms

2018.04.25 更新

グレースィティ株式会社

## 目次

<a href="#">製品の概要</a>	3
<a href="#">ComponentOne for ASP.NET Web Forms のヘルプ</a>	3
<a href="#">主な特長</a>	4
<a href="#">クイックスタート</a>	5
<a href="#">手順 1: ページへの C1Menu の追加</a>	5
<a href="#">手順 2: メニューへの SiteMap の移植</a>	5-6
<a href="#">手順 3: プロジェクトの実行</a>	6-7
<a href="#">デザイン時のサポート</a>	8
<a href="#">スマートタグ</a>	8-9
<a href="#">デザイナフォーム</a>	9-10
<a href="#">デザイナフォームのコンテキストメニュー</a>	10-11
<a href="#">バインディングコレクションエディタ</a>	11-12
<a href="#">メニューのタイプ</a>	13
<a href="#">最上位メニュー</a>	13
<a href="#">ドロップダウンメニュー</a>	13
<a href="#">グループメニュー</a>	13
<a href="#">フライアウトメニュー</a>	13-14
<a href="#">スライディングメニュー</a>	14-15
<a href="#">メニューの作成</a>	16
<a href="#">静的なメニューの作成</a>	16
<a href="#">動的なメニューの作成</a>	16-17
<a href="#">データソースメニューの作成</a>	17
<a href="#">C1Menu の外観</a>	18
<a href="#">テーマ</a>	18-19
<a href="#">メニュー項目アイコン</a>	19-20
<a href="#">テンプレート</a>	20-21
<a href="#">メニューのナビゲーションとショートカット</a>	21
<a href="#">ThemeRoller の概要</a>	21-23
<a href="#">C1Menu CSS セレクタ</a>	23
<a href="#">クライアント側での作業</a>	24
<a href="#">クライアント側イベント</a>	24

<a href="#">タスク別ヘルプ</a>	25
<a href="#">コードによる C1Menu コントロールの作成</a>	25
<a href="#">テーマの利用</a>	25
<a href="#">組み込みテーマの使用</a>	25-27
<a href="#">テンプレートでの作業</a>	27
<a href="#">個別項目テンプレートの作成</a>	27-28
<a href="#">ItemsTemplate の作成</a>	28-29
<a href="#">ChildItemsTemplate の作成</a>	29-30
<a href="#">最上位項目テンプレートの作成</a>	30-32
<a href="#">CSS セレクタでの作業</a>	32
<a href="#">CSS セレクタによる C1Menu 外観のカスタマイズ</a>	32-33
<a href="#">CSS セレクタによる C1Menu リンクの外観のカスタマイズ</a>	33-34
<a href="#">コンテキストメニューとしての C1Menu コントロールの表示</a>	34-35
<a href="#">メニューへの最上位項目の追加</a>	35-36
<a href="#">ドロップダウンメニューの作成</a>	36-38
<a href="#">スライディングメニューの作成</a>	38-40
<a href="#">C1Menu のアニメーション化</a>	40-41
<a href="#">メニュー項目のトリガーの変更</a>	41-42
<a href="#">C1Menu 項目へのアイコンの追加</a>	42-43
<a href="#">コンテキストメニューの位置の設定</a>	43-44
<a href="#">C1Menuの関数</a>	44-48
<a href="#">C1Menu への項目の動的な追加</a>	48-50
<a href="#">C1Menu へのサイトマップの移植</a>	50-51
<a href="#">C1Menu への XML の移植</a>	52-53
<a href="#">XML としての C1Menu の保存とロード</a>	53-54

## 製品の概要

**Menu for ASP.NET Web Forms** では、デザイン時サポートとクライアント側 API の使用によって、アニメーション効果、画像、およびチェックボックス項目、インタラクティブ操作による項目スクロール機能などを備えたマルチレベルメニューの作成や、さらにはアプリケーション内部のコンテキストヘルプのポップアップメニューの作成も、簡単に行うことができます。

## 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。

## 主な特長

C1Menu コントロールは、いくつかの独特な機能を備えています。

- **垂直および水平メニュー**  
メニューとそのサブメニューは、水平または垂直方向のどちらにも表示でき、その他のレイアウトオプションも備えています。
- **スクロール**  
C1Menu は、最上位メニュー、サブメニュー、およびサブグループをスクロールできます。スクロールモードオプションをボタンクリックスクロール、ボタンホバースクロール、エッジホバースクロール、またはスクロールバーに設定できます。
- **メニュー項目アイコン**  
メニュー項目は固有のアイコンを持つことができます。多くの組み込みアイコンから選択するか、独自のアイコンをメニューに追加します。
- **アニメーション**  
メニューは各種の拡張/縮小アニメーション効果をサポートします。たとえば、フェードイン、上部からのスクロールイン、水平オープン、バウンドなど
- **Flash、ActiveX、およびウィンドウオブジェクトの併用**  
他のメニューコントロールと異なり、**C1Menu** は任意のウィンドウオブジェクト、Flash、ActiveX、およびその他の標準/サードパーティ製コンポーネントをオーバーレイできるため、Web サイトのデザインで制約されることがありません。
- **データ連結のサポート**  
**C1Menu** コントロールをデータソースに連結します。XML データソースまたは SiteMap データソースに連結できます。また、Access データソースからのデータさえも読み取って **C1Menu** 階層を動的に作成できます。
- **キーボードサポート**  
アクセスキーのサポートを追加して、特定のキーの組み合わせによって **C1Menu** コントロールにフォーカスを付与します。これによって、エンドユーザーはキーボードの矢印キーを使用してメニューやメニュー項目を移動し、[Enter] キーを押してメニュー項目のリンクを開くことができます。
- **テーマ**  
スマートタグを1回クリックするだけで、6つのプレミアムテーマ (Arctic、Midnight、Aristo、Rocket、Cobalt、および Sterling) の1つを選択してメニューの外観を変更します。オプションとして、jQuery UI の ThemeRoller を使用して、カスタマイズしたテーマを作成することもできます。
- **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。CSS サポートによって、メニューコントロールを組織の規格に合致させることができます。

## クイックスタート

**C1Menu** クイックスタートでは、ASP.NET コントロール **C1Menu** の基本的な使用方法を説明します。クイックスタートでは、ASP.NET Web サイトの作成、ページへの **C1Menu** コントロールの追加、各種連結方式の適用、エディタを使用したメニュー項目やサブメニュー項目の追加、メニューの向きの変更などを行います。

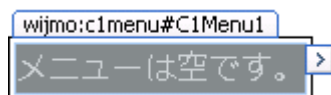
操作は容易であり、各トピックの順序は時系列ではないため、どのトピックから始めてもかまいません。

## 手順 1: ページへの C1Menu の追加

このレッスンでは、新しい ASP.NET Web サイトを作成し、**C1Menu** コントロールをプロジェクトに追加する方法を学びます。クイックスタートを開始するには、以下の手順を実行します。

1. まず、新しい Web サイトを作成します。
2. デザインビューで、Visual Studio ツールボックスに移動し、**C1Menu** アイコンをダブルクリックして **C1Menu** コントロールをページに追加します。

ページは次のように表示されます。



## 手順 2: メニューへの SiteMap の移植

クイックスタートのこのセクションでは、C1Menu を SiteMapDataSource に連結する方法を学びます。

XML ファイルを作成して、それを **C1Menu** に連結するには、**XMLDataSource** コンポーネントを Web サイトに追加し、それを **C1Menu** コントロールに割り当てます。

1. 新しい **AJAX 1.0 対応 ASP.NET 2.0 Web サイト** プロジェクトが開始します。
2. ソリューションエクスプローラで **App\_Data** を右クリックし、[新しい項目の追加]を選択します。[新しい項目の追加]ダイアログボックスが表示されます。
3. XML ファイルを選択し、その名前を「Menu.xml」に変更します。
4. [新しい項目の追加]ダイアログボックスの[追加]をクリックします。
5. XML ビューに切り替えて、次のデータを **Menu.xml** に追加します。

ソースビュー

```
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <menuitem Text="ホーム">
  </menuitem>
  <menuitem Text="製品">
    <menuitem Text="ハードウェア">
    </menuitem>
    <menuitem Text="ソフトウェア">
    </menuitem>
  </menuitem>
  <menuitem Text="サービス">
    <menuitem Text="トレーニング">
    </menuitem>
    <menuitem Text="コンサルティング">
    </menuitem>
  </menuitem>
</root>
```

```
</root>
```

6. .aspx ページに戻り、「**デザイン**」タブを選択してデザインビューに切り替えます。
7. Visual Studio ツールボックスで、**Data** ノードを拡張し、**XmlDataSource** コンポーネントをダブルクリックして、Web ページに追加します。
8. Web ページ上の **XmlDataSource1** を選択し、**プロパティ** ウィンドウに移動します。
9. DataFile プロパティの横の〈...〉ボタンをクリックし、**[XML ファイルの選択]** ダイアログボックスを開きます。**App\_Data** を選択し、**Menu.xml** ファイルをクリックします。〈**OK**〉をクリックして、それを **XmlDataSource1.DataFile** プロパティに追加します。
10. XmlDataSource1.Xpath プロパティを root/menuitem に設定します。
11. スマートタグをクリックして、**[C1Menu タスク]** メニューを開き、**[データソースの選択]** ドロップダウンリストボックスから **XmlDataSource1** を選択します。



次の手順では、プログラムを実行して XML ファイルがどのようにメニューに移植するかを確認します。

### 手順 3: プロジェクトの実行

クイックスタートのこのセクションでは、プロジェクトを実行し、**C1Menu** コントロールに XML ファイルを移植した結果を表示します。

1. プロジェクトを保存して、ビルドします。次のように、3つの最上位メニュー項目が存在することを確認します。

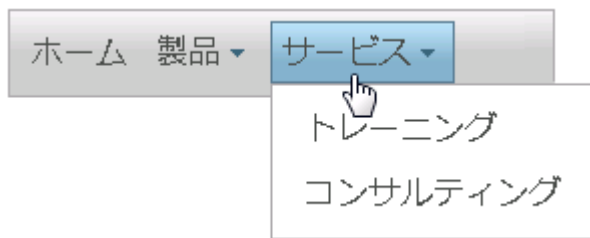


2. カーソルを[製品]にホバーして、2つの項目がサブメニューに表示されることを確認します。



3. カーソルを[サービス]にホバーして、2つの項目がサブメニューに表示されることを確認します。

# Menu for ASP.NET Web Forms



おめでとうございます！ **Menu for ASP.NET Web Forms** クイックスタートが正常に完了しました。



## デザイン時のサポート

**C1Menu** では、スマートタグ、デザイナー、および バインディングコレクションエディタを備えており、充実したデザイン時サポートを提供し、オブジェクトモデルでの作業を簡素化します。

以下のトピックでは、C1Menu のデザイン時環境を使用して **C1Menu** 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

**C1Menu** コントロールには、Visual Studio のスマートタグが含まれています。スマートタグは、**C1Menu** 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。

[**C1Menu タスク**]メニューにアクセスするには、**C1Menu** 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、[**C1Menu タスク**]メニューが開きます。

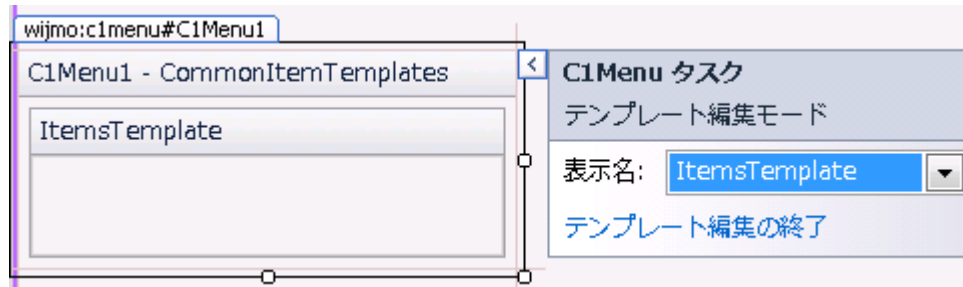
[**C1Menu タスク**]メニューは次のように動作します。

- データソースの選択**  
 項目[**データソースの選択**]をクリックすると、既存のデータソースや連結する新しいデータソースを選択できるドロップダウンリストが開きます。
- メニューの編集**  
 項目[**メニューの編集**]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 C1Menu の要素を手軽に設定できる C1Menu **デザイナーフォーム**が開きます。コントロールのコンテンツをロードおよび保存でき、またリンク項目、ヘッダー、グループ、セパレータの各要素を追加できます。**C1 Menu デザイナーフォーム**の詳細については、「[C1Menu デザイナーフォーム](#)」を参照してください。
- メニューのデータバインディングの編集**  
 項目[**メニューのデータバインディングの編集**]をクリックすると、[**バインディングコレクションエディタ**]ダイアログボックスが開き、連結の追加や削除、およびプロパティの編集ができます。
- テーマ**  
 [**テーマ**]のドロップダウン矢印をクリックすると、各種の組み込みスキンから選択できます。
- 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** を使用方法については、「[ThemeRoller for Visual Studio](#)」を参照してください。
- Bootstrap の使用**

# Menu for ASP.NET Web Forms

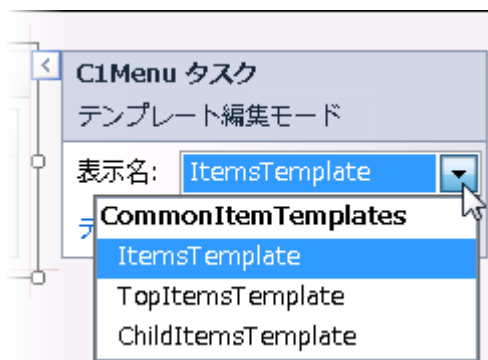
[**Bootstrap の使用**] オプションを選択すると、コントロールに Bootstrap テーマを適用することができます。アプリケーションで Bootstrap テーマを使用する方法については、「[Bootstrap for ASP.NET Web Forms クイックスタート](#)」を参照してください。

- **バージョン情報**  
項目 [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**  
項目 [テンプレートの編集] をクリックすると、**C1Menu** コントロールがテンプレート編集モードに切り替わります。



テンプレート編集モードでは、異なるオプションの [**C1Menu タスク**] メニューが表示されます。

- **表示名**  
[表示名] のドロップダウン矢印を選択すると、カスタマイズ可能なテンプレート領域のリストが開きます。



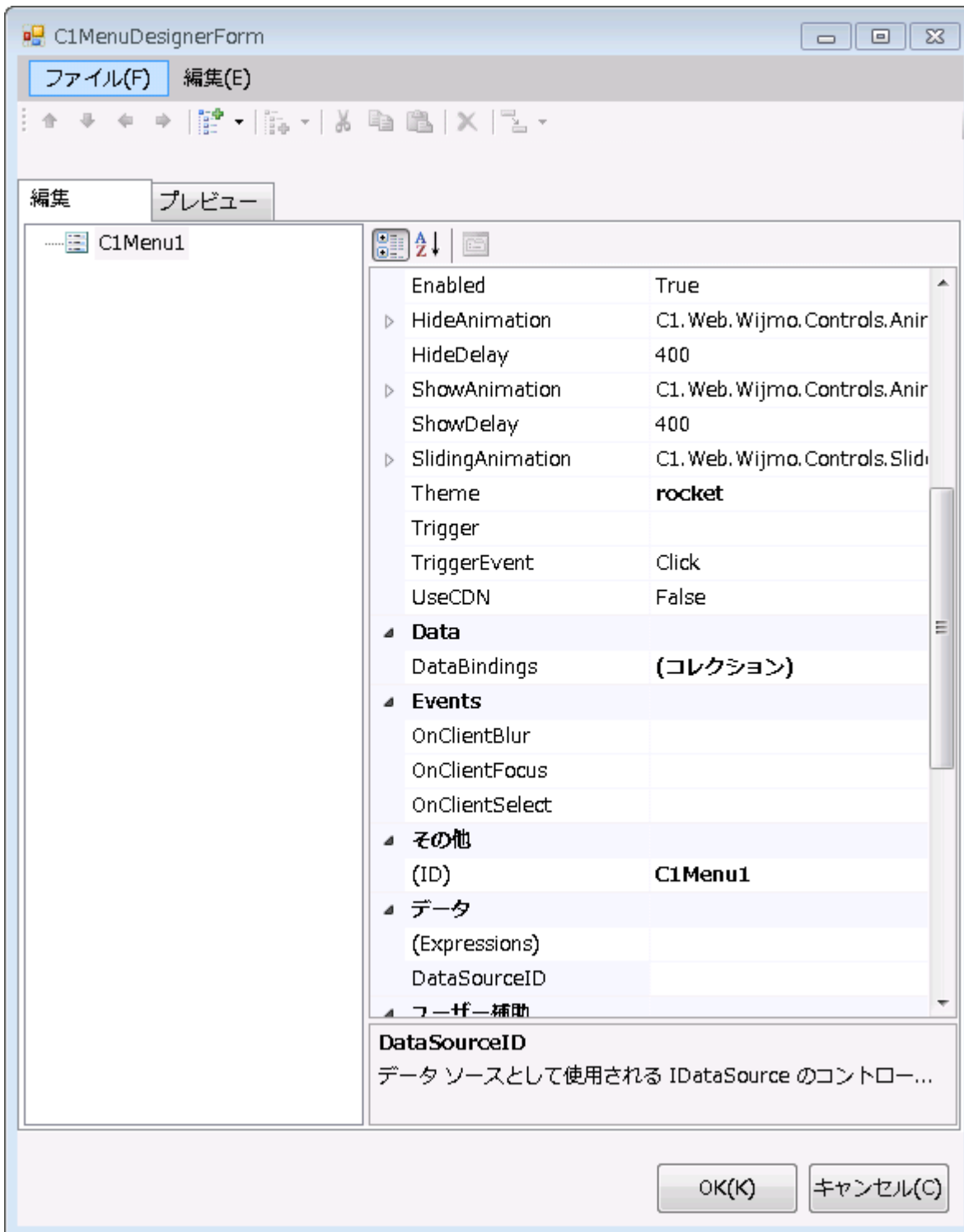
このリストからテンプレートを選択し、その編集対象のテンプレートを開きます。

- **テンプレート編集の終了**  
項目 [テンプレート編集の終了] をクリックすると、テンプレート編集モードが終了し、メインの [**C1Menu タスク**] メニューに戻ります。

## デザイナフォーム

[**C1Menu デザイナフォーム**] ダイアログボックスでは、**C1Menu** コントロールのプロパティウィンドウをスクロールしなくても **C1Menu** の要素を手軽に設定できます。**C1Menu デザイナフォーム** を使用すれば、**C1Menu** コントロールでリンク項目、ヘッダー、グループ、セパレータの各要素の追加、操作、および削除ができ、また **C1Menu** コントロールのコンテンツをロードおよび保存できます。

[**C1Menu デザイナフォーム**] ダイアログボックスにアクセスするには、[**C1Menu タスク**] メニューから項目 [**メニューの編集**] を選択するか (詳細については、「[C1Menu スマートタグ](#)」を参照)、またはデザイン時に **C1Menu** コントロールを右クリックして [**メニューの編集**] を選択します。項目を追加したときのデザイナは次のようになります。



**C1Menu デザイナフォーム**には、「編集」タブと「プレビュー」タブが含まれています。「編集」タブ(上図)には、追加されたメニュー項目の順序を表示する左ペインがあります。「編集」タブの右側には、追加された項目を手軽にカスタマイズできるプロパティグリッドがあります。

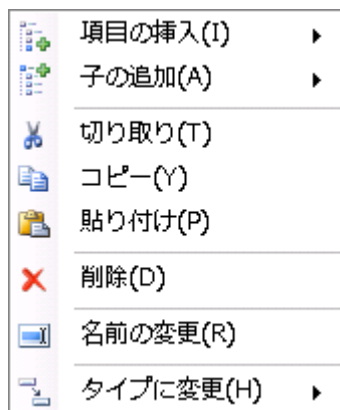
「プレビュー」タブでは、**C1Menu** コントロールを表示し、加えている変更をすばやく確認できます。

## デザイナフォームのコンテキストメニュー









**C1Menu デザイナフォーム**のコンテキストメニューでは、**C1Menu** の要素を手軽に設定できます。**C1Menu デザイナフォーム**を使用すれば、**C1Menu** コントロールのコンテンツをロードおよび保存でき、またリンク項目、ヘッダー、グループ、セパレータの各要素の追加や削除が可能です。

# Menu for ASP.NET Web Forms

**C1Menu デザイナーフォーム**のコンテキストメニューにアクセスするには、「編集」タブの左ペインで右クリックします。コンテキストメニューは次のように表示されます。



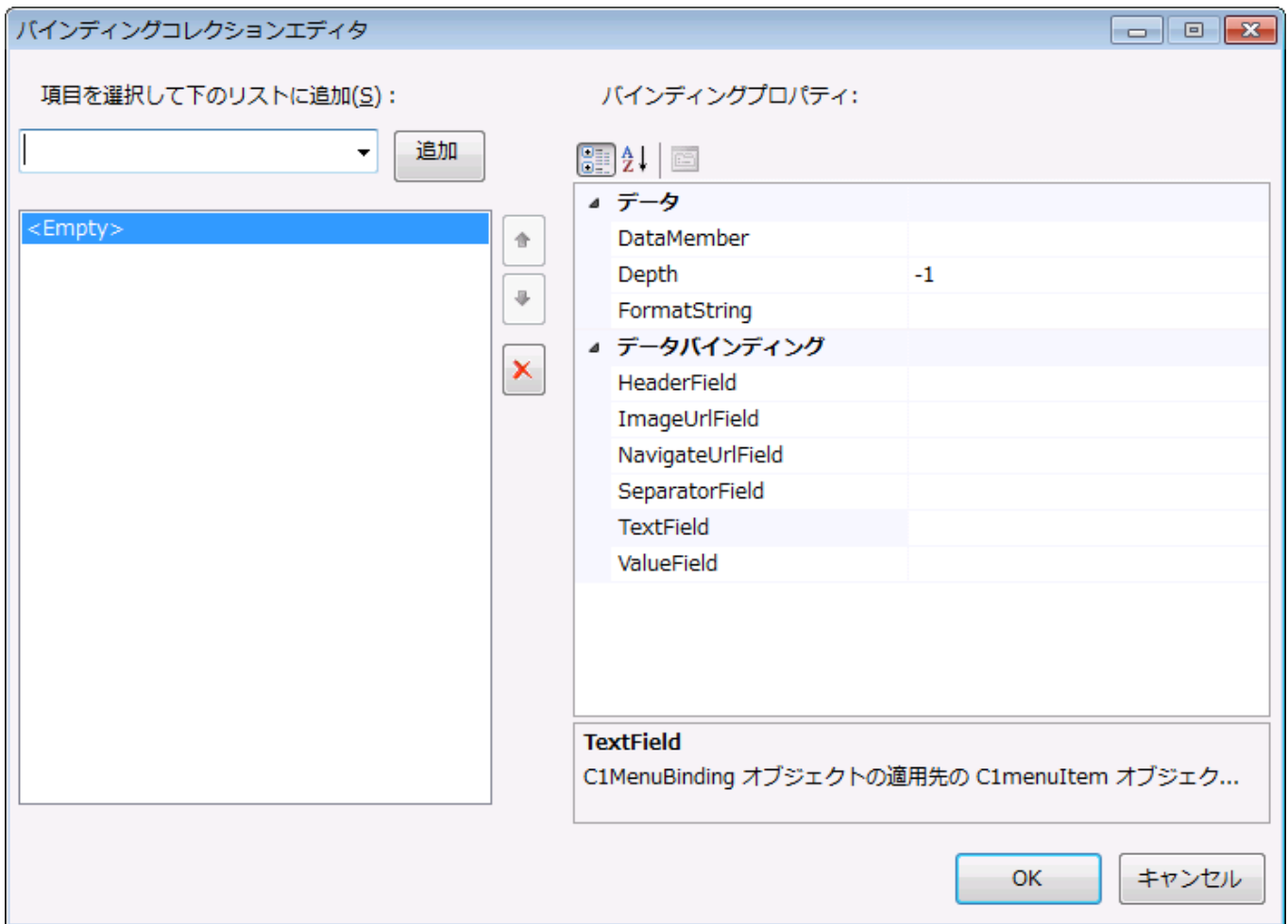
下の表で、**C1Menu デザイナーフォーム**のコンテキストメニューに含まれる各項目の機能を説明します。

ボタン	名前	説明
	項目の挿入	新しい項目をメニューに挿入します。リンク項目(🔗)、ヘッダー(H)、グループ(📁)、セパレータ(⇨)の各要素が選択できます。
	子の追加	現在の項目の下に子項目を挿入します。リンク項目(🔗)、ヘッダー(H)、グループ(📁)、セパレータ(⇨)の各要素が選択できます。
	切り取り	選択されている項目を切り取ります。
	コピー	選択されている項目をコピーします。
	貼り付け	切り取りまたはコピーされた項目を、選択した位置に貼り付けます。
	削除	選択されている項目を削除します。
	名前の変更	選択されている項目の名前を変更します。このオプションを選択して、選択項目の新しい名前を入力します。
	タイプに変更	現在の項目を別のタイプの項目に変更します。リンク項目(🔗)、ヘッダー(H)、グループ(📁)、セパレータ(⇨)の各要素が選択できます。

## バインディングコレクションエディタ

[**バインディングコレクションエディタ**]ダイアログボックスを使用すれば、データ項目とそれが連結されているメニュー項目の間の関係を容易に定義できます。[**バインディングコレクションエディタ**]ダイアログボックスにアクセスするには、プロパティウィンドウで **DataBindings** プロパティの横の<...>ボタンを選択するか、または[**C1Menu タスク**]メニューから項目[**メニューのデータバインディングの編集**]を選択します。

[**バインディングコレクションエディタ**]ダイアログボックスは、次のように表示されます。



**バイディングコレクションエディタ**は、ドロップダウンボックス(追加する既存の項目を選択)、左側のリストボックス(追加された項目を一覧表示)、および右側のプロパティグリッド(当該項目のデータとデータ連結のプロパティを変更)で構成されます。バイディングコレクションエディタで加えた変更は、すべてソースビューの `<DataBindings>` タグに反映されます。たとえば、次のようになります。

#### ソースビュー

```
<cc1:c1menu id="C1Menu1" runat="server" datasourceid="SiteMapDataSource1" visualstyle="Default"
visualstylepath="~/C1WebControls/C1Menu/VisualStyles">

  <DataBindings>

    <cc1:C1MenuItemBinding DataMember="SiteMapNode" Depth="5" NestedGroupHeight=""
NestedGroupWidth="" />
  </DataBindings>
</cc1:c1menu>
```

## メニューのタイプ

通常、最上位メニューやサブメニューはナビゲーションシステムの作成に使用されますが、**C1Menu** コントロールでは、メニューの機能を拡張してグループを含むメニューも作成して、各項目をカテゴリにまとめることができます。

このセクションでは、**C1Menu** コントロールを使用して作成できるメニューの一般的なタイプについて詳しく説明します。

## 最上位メニュー

最上位メニューは、メインのメニューです。このメニューは、水平または垂直メニューバー上に配置されたメニュー項目で構成されます。各メニュー項目には、一連のサブメニュー項目が含まれる場合もあります。最上位メニューは、フォーム上に常に表示されています。通常、最上位メニューの項目数は 15 項目以下です。

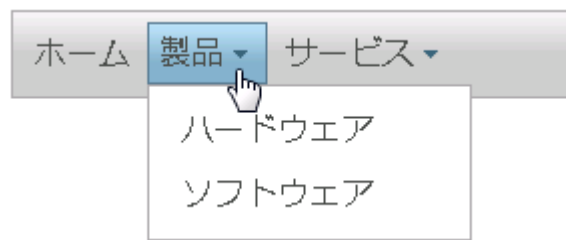
標準的な最上位メニューは次のような表示になります。



## ドロップダウンメニュー

メニューバーには、アプリケーションのドロップダウンメニューとサブメニューがすべて含まれます。メニューバー内の各メニューは、メニュータイトルで示されます。

サブメニューやドロップダウンメニューは、1つのメニュー項目内に表示される追加的なメニュー項目です。サブメニュー項目は、最上位メニューやサブメニューレベルでメニュー項目上にマウスポインタを置いたり、メニュー項目をクリックしたりしたときに表示されます。メニューのスタイルに応じて、矢印やその他のグラフィックスが使用されているため、メニュー項目からサブメニュー項目が開かれるタイミングがわかります。サブメニューの下にサブメニューなどが含まれる場合もあります。

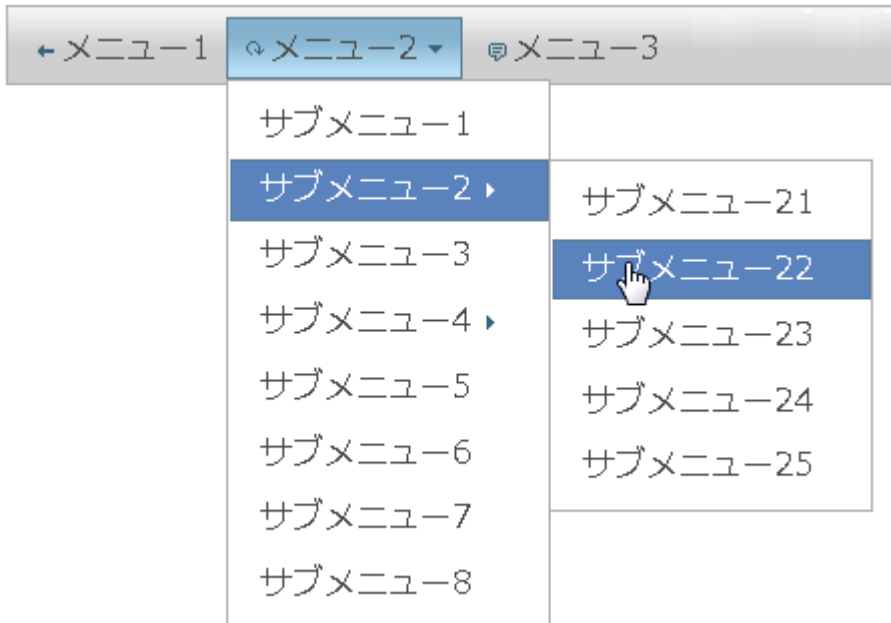


## グループメニュー

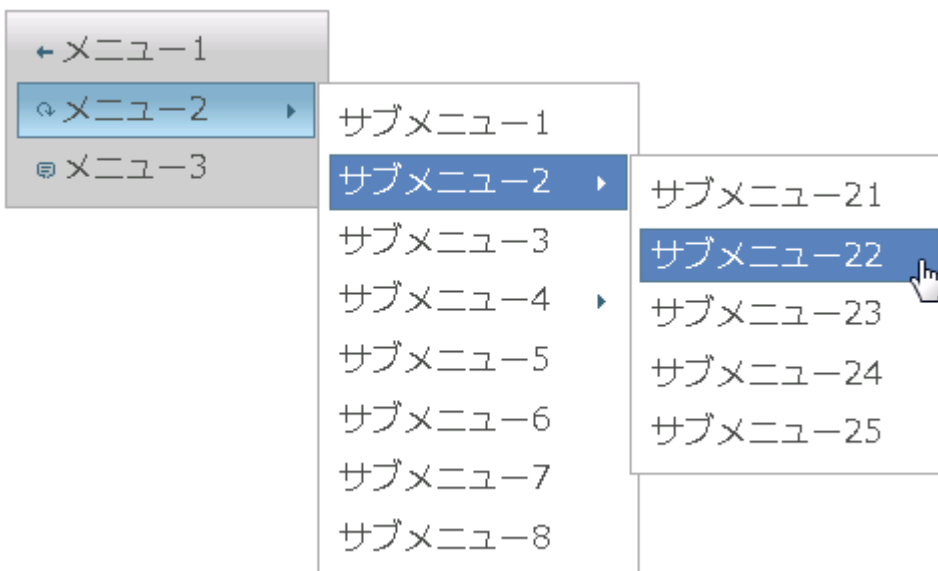
グループメニューは、1つまたは複数のカテゴリにまとめられたメニュー項目のセットです。あるグループ内の各メニュー項目は、共通の機能を共有します。各グループには、グループのカテゴリ名を示すヘッダー項目タイプを含めることができます。通常、グループメニューは、メニュー項目グループのタイトルとなる1つ以上のヘッダー項目を含んだドロップダウンメニューとして表示されます。各グループは、ヘッダー項目やセパレータ項目で区分できます。

## フライアウトメニュー

デフォルトでは、表示されるメニューのタイプは、水平位置のフライアウトメニューです。これは、大部分のアプリケーションで見られる基本的なメニューです。



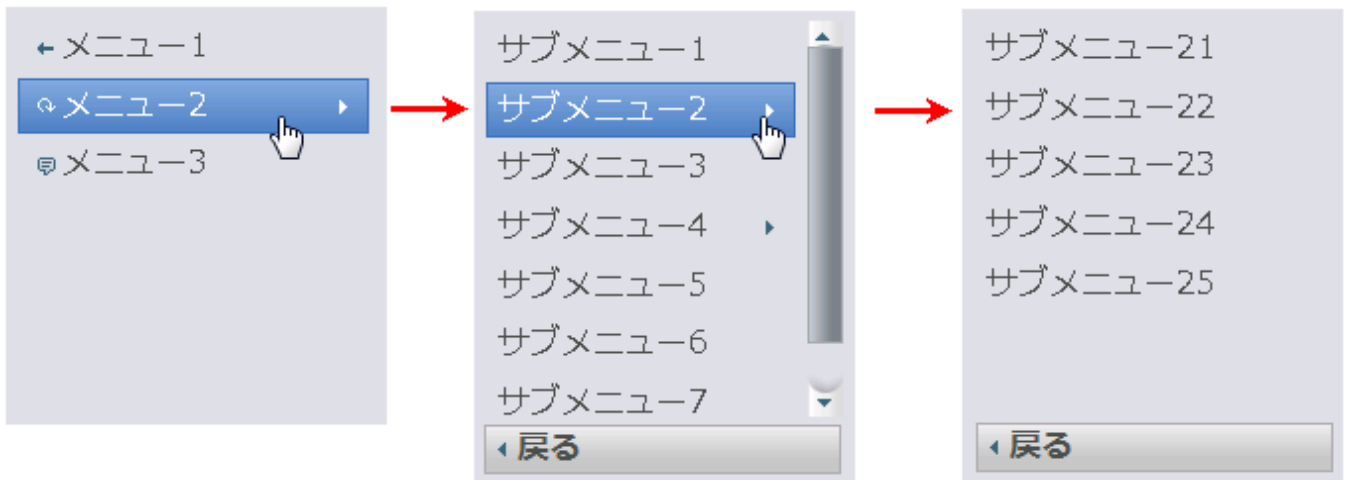
垂直位置のフライアウトメニューを作成するには、**C1Menu** コントロールの **Orientation** プロパティを Vertical に設定します。



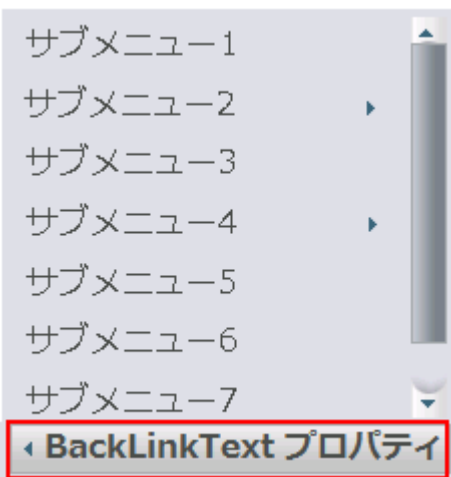
## スライディングメニュー

スライディングメニューは、主としてスマートフォン機器で使用されるメニューのタイプです。ユーザーがメニュー項目をクリックするかホバーしたときにサブメニューをフライアウトさせる代わりに、サブメニューが表示領域にスライドして、直前のメニューを置き換えます。

## Menu for ASP.NET Web Forms



上の図では2つのサブメニュー上に「戻る」ボタンが表示されていることに注意してください。デフォルトでは、スライディングメニューによって「戻る」移動ボタンがサブメニューに適用されます。希望する場合は、**C1Menu** コントロールの **BackLinkText** プロパティを設定して「戻る」ボタンの文字列を変更できます。



また、「戻る」ボタンの代わりに、ブレッダクラムナビゲーションを使用するには、**C1Menu** コントロールの **BackLink** プロパティを **False** に設定します。



## メニューの作成

メニューシステムは、次の方式のいずれかを使用して作成できます。

- 宣言構文を使用した静的な作成
- **C1MenuItem** クラスの新しいインスタンスを作成するコンストラクタを使用した動的な作成
- **C1Menu** を **SiteMapDataSource**、**XMLDataSource**、または **AccessDataSource** に連結することによるデータソースの作成。

## 静的なメニューの作成

静的なメニューは、メニュー構造を作成する最も単純な方法です。

**C1Menu デザイナフォーム**を使用してメニューシステムを構築するか、または .aspx ファイルで宣言構文を使用してメニュー項目を指定できます。宣言構文を使用して静的なメニュー項目を表示するには、まず、メニューコントロールの開始タグと終了タグの間に `<Item>` の開始タグと終了タグをネストします。次に、`<Item>` の開始タグと終了タグの間に `<cc1:C1MenuItem>` 要素をネストすることによって、メニュー構造を作成します。各 `<cc1:C1MenuItem>` 要素は、コントロール内のメニュー項目に相当し、**C1MenuItem** オブジェクトにマッピングされます。

### ソースビュー

```
<cc1:C1Menu ID="C1Menu1" runat="server">
<HideAnimation>
<Animated Effect="fade"></Animated>
</HideAnimation>
  <Items>
    <cc1:C1MenuItem runat="server" ImagePosition="Left" Text="LinkItem1">
      <Items>
        <cc1:C1MenuItem runat="server" ImagePosition="Left" Text="LinkItem1">
        </cc1:C1MenuItem>
      </Items>
    </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" ImagePosition="Left" Text="LinkItem2">
    </cc1:C1MenuItem>
  </Items>
</cc1:C1Menu>
```

## 動的なメニューの作成

動的なメニューは、サーバー側またはクライアント側で作成できます。サーバー側で動的なメニューを作成するときは、**C1MenuItem** クラスの新しいインスタンスを動的に作成するコンストラクタを使用します。クライアント側では、**CreateInstance** コンストラクタを使用すれば、**C1Menu** コントロールの新しいインスタンスを動的に作成できます。たとえば、次のスクリプトは、クライアント側で新しいメニューコントロールを作成します。

## ソースビュー

```
var aMenu = C1.Wijmo.Controls.C1Menu.createInstance ();  
document.body.appendChild(aMenu.element);
```

## データソースメニューの作成

メニュー項目は、**XMLDataSource** または **SiteMapDataSource** のような階層型のデータソースコントロールから作成できます。これを利用すれば、コードを編集しなくてもメニュー項目を更新できます。また、メニュー項目は、**AccessDataSource** コンポーネントのような非階層型のコントロールに連結することもできます。

**C1Menu** コントロールへのデータソースの移植に関するチュートリアルについては、「[C1Menu へのサイトマップの移植](#)」と「[C1Menu への XML の移植](#)」を参照してください。

## C1Menu の外観

以下のトピックでは、**C1Menu** コントロールの外観を変更する機能を説明します。

### テーマ

**C1Menu** コントロールは、6つの組み込みテーマを備えています。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 コントロールもそれに従ってスキン設定されます。これらのテーマは、次のように**C1Menu** コントロール上に表示されます。

Arctic	 <p>The Arctic theme features a light gray menu bar. The 'News' (ニュース) item is highlighted with a darker gray background. The dropdown menu is white with a light gray border and contains the following items: ニュース (News), エンタメ (Entertainment), 政治 (Politics), 国際 (International), スポーツ (Sports), 社会 (Society), and 天気 (Weather).</p>
Aristo	 <p>The Aristo theme features a light gray menu bar. The 'News' (ニュース) item is highlighted with a darker gray background. The dropdown menu is white with a light gray border and contains the following items: ニュース (News), エンタメ (Entertainment), 政治 (Politics), 国際 (International), スポーツ (Sports), 社会 (Society), and 天気 (Weather).</p>
Cobalt	 <p>The Cobalt theme features a light blue menu bar. The 'News' (ニュース) item is highlighted with a darker blue background. The dropdown menu is white with a light blue border and contains the following items: ニュース (News), エンタメ (Entertainment), 政治 (Politics), 国際 (International), スポーツ (Sports), 社会 (Society), and 天気 (Weather).</p>

Midnight	 The Midnight theme features a dark, almost black menu bar. The items are '新着', 'エンタメ', '金融', 'ライフ', '国際', 'ニュース', '政治', and 'スポーツ'. The 'ニュース' item is highlighted with a dark grey background, and its dropdown menu is open, showing 'ニュース', 'エンタメ', '政治', '国際', 'スポーツ', '社会', and '天気'.
Rocket	 The Rocket theme has a dark menu bar with light-colored text. The 'ニュース' item is highlighted with a dark grey background, and its dropdown menu is open, showing 'ニュース', 'エンタメ', '政治', '国際', 'スポーツ', '社会', and '天気'.
Sterling	 The Sterling theme has a light grey menu bar. The '新着' item is highlighted with a yellow background, and its dropdown menu is open, showing 'ニュース', 'エンタメ', '政治', '国際', 'スポーツ', '社会', and '天気'.

C1Menu コントロールのテーマを設定するには、**Theme** プロパティを組み込みテーマの1つに設定するだけです。

## メニュー項目アイコン

アイコンを個別のメニュー項目に簡単に付加するには、そのメニュー項目の **IconClass** プロパティを設定します。jQuery UI の `ui-icon` クラスを宣言してから、2番目のクラスをそれに追加して、使用したいアイコンのタイプを記述します。これらのアイコンクラスは、次のような構文に従います。

ソースビュー

```
ui-icon ui-icon-[icon name]
```

jQuery UI フレームワークアイコンセットに含まれている任意のアイコンを使用できます。jQuery UI フレームワークアイコンの

リストについては、jQuery UI ThemeRoller ページ(<http://jqueryui.com/themeroller/>)を参照してください。

アイコンのデフォルト位置は、メニュー項目テキストの左側です。これを変更したい場合は、メニュー項目の **ImagePosition** プロパティを Left に設定します。

## テンプレート

**C1Menu** は、基本的な項目別テンプレートからグローバルテンプレートまで、いくつかのタイプのテンプレートを備えています。たとえば、**ChildItemsTemplate** はコントロールの全部分のテンプレートを制御するために使用されます。

### 個別テンプレート

テンプレートを使用して、HTML コントロールをメニュー項目に埋め込むことができます。また、書式設定のような単純な設定にも使用できます。ソースビューでこれらのテンプレートを作成するには、タグを個別メニュー項目に追加します。たとえば、次のようになります。

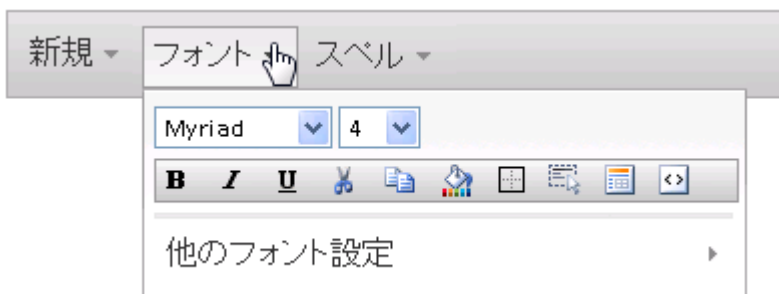
#### ソースビュー

```
<cc1:C1MenuItem ID="C1MenuItem1" runat="server">
  <Template>
    <div>
      <a href="#" class="wijmo-wijmenu-text">New</a> <span class="wijmo-wijmenu-icon-right">Ctrl+N</span>
    </div>
  </Template>
</cc1:C1MenuItem>
```

※グローバルテンプレート(**ItemsTemplate**など)が**C1Menu** コントロールに割り当てられている場合、個別テンプレートはそれを上書きします。

個別テンプレートのチュートリアルについては、「[個別項目テンプレートの作成](#)」を参照してください。

### グローバルテンプレート



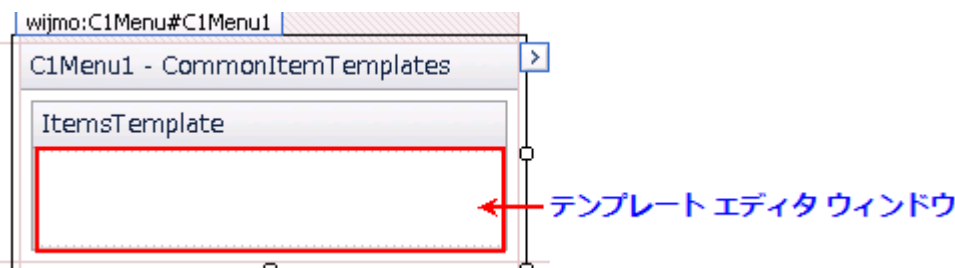
**C1Menu** は、最上位メニュー項目 (**TopItemsTemplate**)、子メニュー項目 (**ChildItemsTemplate**)、またはすべてのメニュー項目 (**ItemsTemplate**) をカスタマイズするための特別なテンプレートデザイナーも備えています。異なるタイプのテンプレートを作成すると、メニュー項目の全体だけでなく特定部分のデザインさえも制御できます。テンプレートは、新しい項目のプロトタイプとしても便利です。すなわち、動的に追加された項目をメニュー内の他の項目のデザインに合わせることもできます。

次のような **C1Menu** コントロールでは、**ChildItemsTemplate** は各メニュー項目に2つの HTML 要素、1つの入力(チェックボックス)コントロール、および1つのラベルコントロールを組み込むようにカスタマイズされています。



## テンプレートにアクセスする

1. **C1Menu** コントロールのスマートタグをクリックして、[**C1Menu タスク**]メニューを開きます。  
**テンプレート編集モード**が始動します。
2. [表示]のドロップダウン矢印をクリックして、ドロップダウンリストから、テンプレートのタイプ (**TopItemsTemplate**、**ChildItemsTemplate**、または **ItemsTemplate**) を選択します。
3. HTML 要素または書式をテンプレートデザイナウィンドウに追加します (チュートリアルについては「[テンプレートでの作業](#)」を参照)。



## メニューのナビゲーションとショートカット

メニューシステムをナビゲートするには、次の2つの方式が一般的です。

- マウスを使用してポイント&クリック
- キーボードの矢印キーを使用

### マウスによるナビゲーション

ユーザーがメニュー項目をクリックすると、Menu コントロールは、リンク先の Web ページに移動するか、またはサーバーへのポストバックだけを行います。メニュー項目の **NavigateUrl** プロパティが設定されている場合、Menu コントロールはリンク先のページに移動します。その他の場合は、ページを処理のためにサーバーにポストバックします。デフォルトでは、リンク先のページは、Menu コントロールと同じウィンドウまたはフレームに表示されます。リンク先のコンテンツを別のウィンドウまたはフレームに表示するには、Menu コントロールの **Target** プロパティを使用します。

### キーボードによるナビゲーション

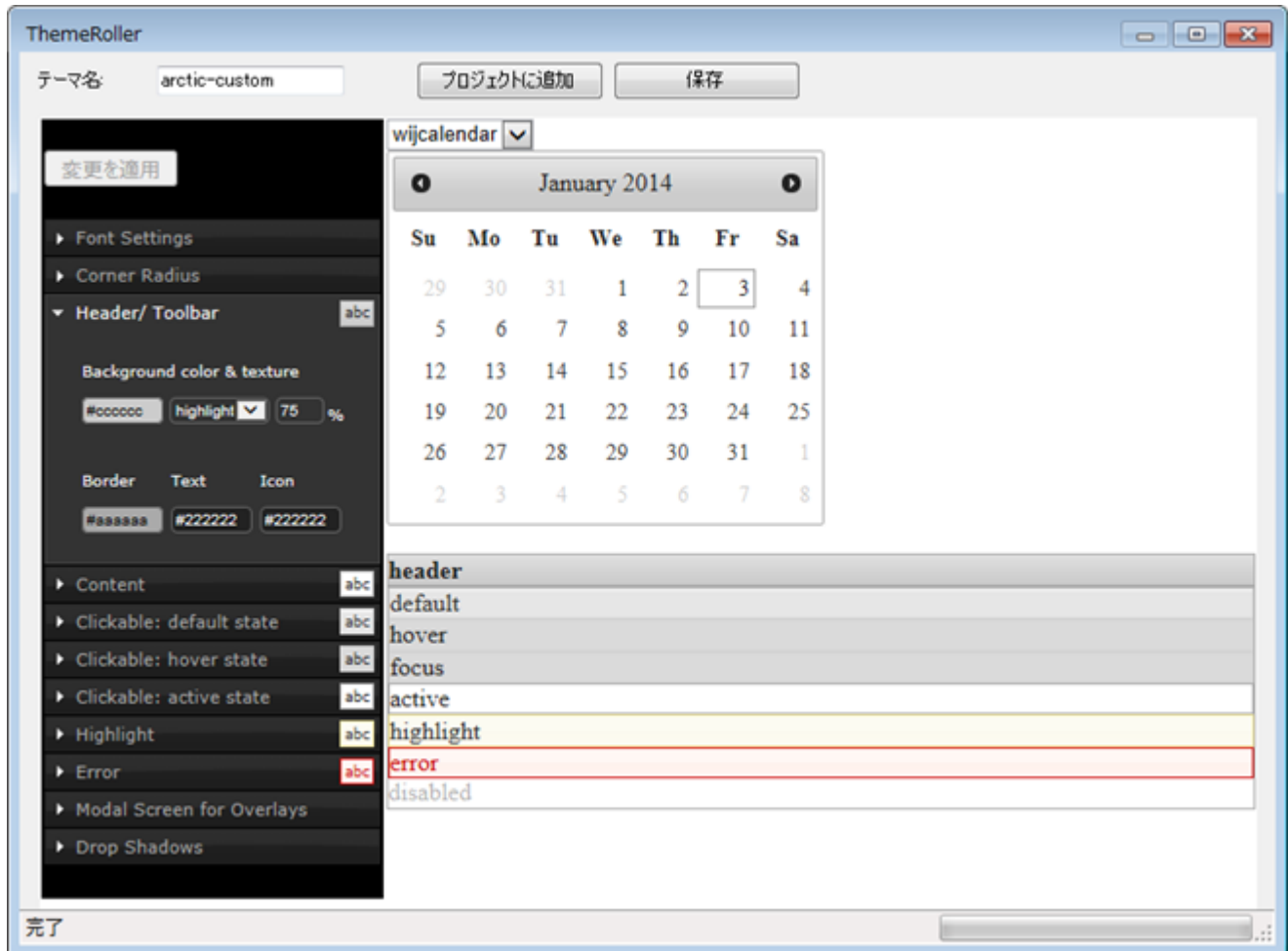
**C1Menu** では、ユーザーによるメニュー項目のナビゲーションを強化するアクセスキーがサポートされています。アクセスキーは、メニューのナビゲートに使用され、[Alt] キーと指定のキーを使用するのが普通です。

アクセスキーは、**AccessKey** プロパティを使用して、任意のメニュー項目やサブメニュー項目に追加できます。アクセスキーをメニュー項目に割り当てるときは、いずれかの文字を使用してください。

## ThemeRoller の概要

Wijmo の最も便利な機能の1つは ThemeRoller との互換性です。ThemeRoller は、独自の jQueryUI テーマを作成して、Web アプリケーションのコントロールのスキンを生成する Web アプリケーションです。その簡単なインタフェースと WYSIWYG プレビューにより、グラフィックエディタを開くためにかかる時間よりも短時間に、すべての Wijmo コントロールおよびその他の ThemeRoller 互換コントロールのスキンを生成できます。

ThemeRoller Web アプリケーションは <http://jqueryui.com/themeroller/> で利用できます。アプリケーションに到達したら、ページの左側の列を見て、「Roll Your Own」、「Gallery」、および「Help」という3つのタブがあることを確認します。



### 「Roll Your Own」タブ

「Roll Your Own」タブは魔法が起こる場所です。このタブから、テーマをきめ細かく調整して完全させるパワーが得られます。要素を変更するためにしなければならないのは、ノードを展開して作業に着手するだけです。

### 「Gallery」タブ

「Gallery」をクリックすると、ThemeRoller のプレミアムテーマのリストとプレビューが表示されます。ここから、次の方法で既製のテーマの1つをプレビュー、ダウンロード、または編集できます。

- テーマのスニペットビューをクリックすると、ギャラリーの右側にインタラクティブなプレビューがロードされます。
- <Download theme> ボタンをクリックすると、**Build Your Download** ページに移動します。既製のテーマの1つをダウンロードして Wijmo コントロールのスキンを生成する場合は、薄オレンジ色のパネルに移動し、[Advanced Theme Settings] と [Version] を選択して <Download> をクリックするだけです。
- <Edit> をクリックすると選択したテーマがロードされ、「Roll Your Own」タブに戻り、そこで ThemeRoller を使用してテーマをきめ細かく調整できます。

## 「Help」タブ

「Help」タブをクリックすると、プラグイン開発者およびブラウザサポートに関する通知など、ThemeRoller に関するクイックリファレンスと特別な情報を参照できます。

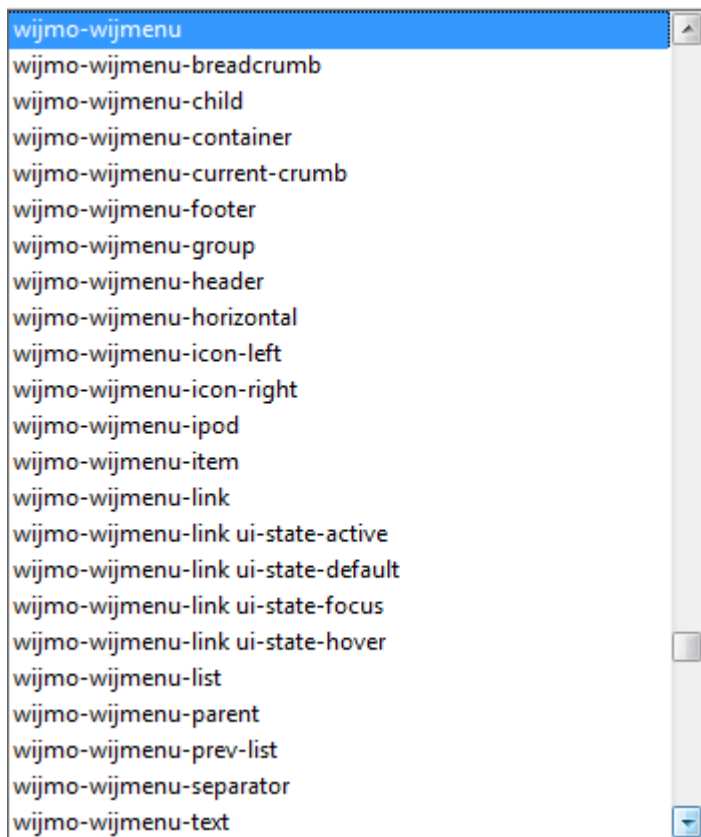
## C1Menu CSS セレクタ

CSS を使用して多くの **C1Menu** 要素をスタイル設定し、その外観を独特なものにできます。このカスタマイズ処理を簡素化するために、ComponentOne for ASP.NET Web Forms には、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。

枠線、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。

一般的な個別 CSS セレクタやグループ化した CSS セレクタのリストについては、プロジェクトの **C1Menu** コントロールを選択して、Visual Studio プロパティウィンドウで **CssClass** プロパティの横のドロップダウンリストを表示します。

**C1Menu CSS セレクタ**は、次のような wijmo-wijmenu で開始します。





## クライアント側での作業

Menu for ASP.NET Web Forms コントロールには、非常に充実したクライアント側オブジェクトモデルがあります。そのメンバは、ほとんどがサーバー側コントロールのメンバと同じです。

**C1Menu** コントロールが表示されると、クライアント側コントロールのインスタンスが自動的に生成されます。これは、サーバーにポストバックしなくても、**C1Menu** コントロールのプロパティやメソッドにアクセスできるということです。

クライアント側コードを使用すれば、時間をかけて Web サーバーに情報を送信しなくても、Web ページに多くの機能を実装できます。クライアント側オブジェクトモデルを使用すると、Web サイトの効率を向上させることができます。

## クライアント側イベント

Menu for ASP.NET Web Forms には、複数のクライアント側イベントが含まれています。それらを利用すれば、ダイアログウィンドウのサイズ変更などの処理が行われたときに、**C1Menu** コントロールを操作できます。

クライアント側イベントの表にリストされたサーバー側プロパティを使用して、特定のクライアント側イベントに反応する JavaScript 関数の名前を指定できます。たとえば、「Focus」という JavaScript 関数を割り当てて、メニューがフォーカスされたときに応答させるには、**OnClientFocus** プロパティを focus に設定します。

下の表に、クライアントスクリプトで使用できるイベントを示します。これらのプロパティはサーバー側で定義されていますが、実際のイベントや各 JavaScript 関数用に宣言する名前はクライアント側で定義されます。

イベントのサーバー側プロパティ名	イベント名	説明
<b>OnClientBlur</b>	blur	メニュー項目がフォーカスを失ったときに発生します。
<b>OnClientFocus</b>	focus	マウスポインタが置かれるか、キーボードのカーソルキーがナビゲーションで使用されたときに発生します。
<b>OnClientHidden</b>	hidden	メニューが閉じられたときに発生します。
<b>OnClientHidding</b>	hidding	メニューが閉じているときに発生します。
<b>OnClientSelect</b>	select	メニュー項目が選択されたときに発生します。
<b>OnClientShowing</b>	showing	メニューが開いているときに発生します。
<b>OnClientShown</b>	shown	メニューが開かれるときに発生します。

## タスク別ヘルプ

タスク別ヘルプは、Visual Studio ASP.NET 環境でのプログラミングに精通し、**C1Menu** 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。各トピックでは、**C1Menu** 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、読者が新しい ASP.NET プロジェクトを作成した経験を持っていることも前提となります。

## コードによる C1Menu コントロールの作成

コードによる **C1Menu** コントロールの作成は容易なプロセスです。このトピックでは、**PlaceHolder** コントロールをページに追加して、インポートステートメントを追加した後、**C1Menu** を追加して、このコントロールを **PlaceHolder** に追加します。

以下の手順を実行します。

1. デザインビューで、Visual Studio ツールボックスに移動し、**PlaceHolder** コントロールをページに追加します。
2. ページをダブルクリックして、**Page\_Load** イベントを追加し、コードビューに切り替えます。
3. 次のステートメントをコードエディタの上部に追加し、適切な名前空間をインポートします。

### Visual Basic コードの書き方

```
Visual Basic
Imports C1.Web.Wijmo.Controls.C1Menu
```

### C# コードの書き方

```
C#
using C1.Web.Wijmo.Controls.C1Menu;
```

4. 次のコードを追加し、**C1Menu** コントロールを作成して、このコントロールを **PlaceHolder** に追加します。

### Visual Basic コードの書き方

```
Visual Basic
新しい C1Menu control を作成します。
Dim C1M As New C1Menu()
' C1Menu を PlaceHolder コントロールに追加します。
PlaceHolder1.Controls.Add(C1M)
```

### C# コードの書き方

```
C#
// 新しい C1Menu control を作成します。
C1Menu C1M = new c1Menu();
// C1Menu を PlaceHolder コントロールに追加します。
PlaceHolder1.Controls.Add(C1M);
```

5. [F5]を押して、プログラムを実行します。コードによって基本的な **C1Menu** コントロールを作成しました。

## テーマの利用

このセクションの各トピックでは、組み込みテーマとカスタムテーマの使用方法を例示します。

## 組み込みテーマの使用

**C1Menu** コントロールは、数回のクリックだけで適用できる6つの組み込みテーマを備えています。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例示します。テーマについての詳細は、「[テーマ](#)」を参照してください。

### デザインビューでのテーマの変更

以下の手順を実行します。

1. C1Menu のスマートタグ (🔗) をクリックして、[**C1Menu タスク**]メニューを開きます。
2. [テーマ]の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 **C1Menu** コントロールに適用されます。

### ソースビューでのテーマの変更

ソースビューで **C1Menu** のテーマを変更するには、`VisualStyle="rocket"` を `<cc1:C1Menu>` タグに追加して、次のような記述にします。

ソースビュー

```
<cc1:C1Menu runat="server" Theme="rocket"/>
```

### コードによるテーマの変更

以下の手順を実行します。

1. 次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 C1.Web.Wijmo.Controls
```

#### C# コードの書き方

```
C#
using C1.Web.Wijmo.Controls;
```

# Menu for ASP.NET Web Forms

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  
C1Menu1.Theme = "rocket"
```

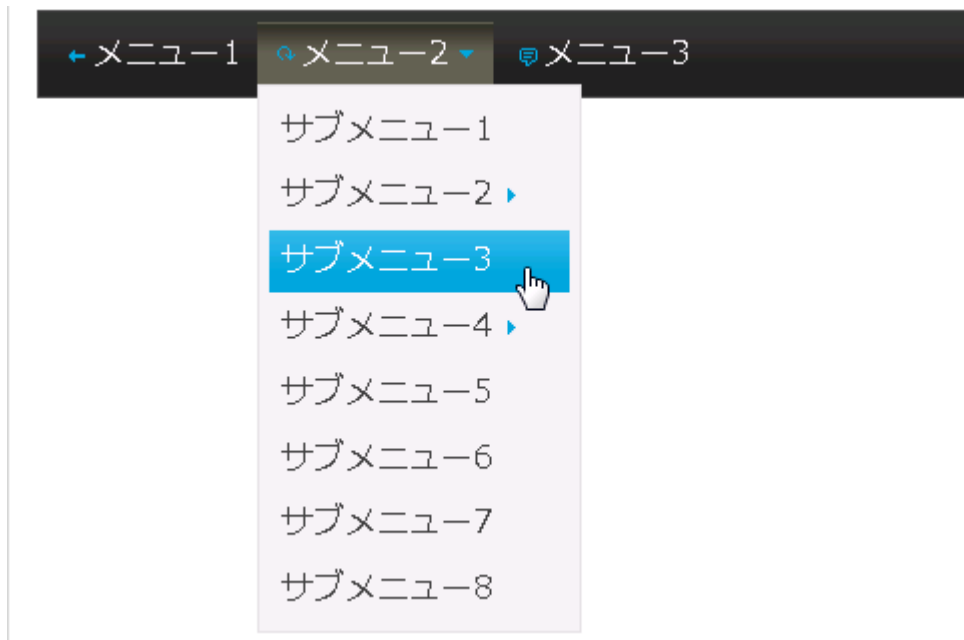
## C# コードの書き方

```
C#  
C1Menu1.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 **C1Menu** コントロールを示しています。



## テンプレートでの作業

このセクションの各チュートリアルでは、**ChildItemsTemplate**、**ItemsTemplate**、および **TopItemsTemplate** の使用方法を学びます。

## 個別項目テンプレートの作成

このチュートリアルでは、個別メニュー項目のテンプレートの作成方法を学びます。

**注意:** 個別テンプレートは、グローバルテンプレートよりも優先され、テンプレートを個別項目に付加すると、**ItemsTemplate**、**ChildItemsTemplate**、および **TopItemsTemplate** で使用された書式が抑制されます。

以下の手順を実行します。

1. 1つの最上位メニュー項目を持つ **C1Menu** コントロールを作成します。2つの子項目をその最上位メニュー項目に追加して、サブメニューを作成します。

2. ソースビューに切り替えて、次のマークアップを `<cc1:C1MenuItem ...>` タグの間に追加します。

```

ソースビュー
<Template>
  <div>
    <a href="#" class="wijmo-wijmenu-text">新規</a> <span class="wijmo-wijmenu-icon-right">Ctrl+N</span>
  </div>
</Template>

```

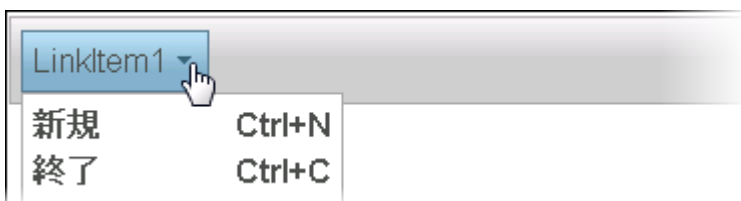
3. 今度は、次のマークアップを `<cc1:C1MenuItem ...>` タグの間に追加します。

```

ソースビュー
<Template>
  <div>
    <a href="#" class="wijmo-wijmenu-text">終了</a> <span class="wijmo-wijmenu-icon-right">Ctrl+C</span>
  </div>
</Template>

```

4. [F5]を押して、プロジェクトを実行します。プロジェクトのロード時に、最上位メニュー項目をクリックして、サブメニューを開きます。2つのメニュー項目が指定したテンプレートを採用していることを確認します。

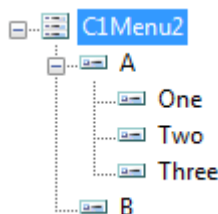


## ItemsTemplate の作成

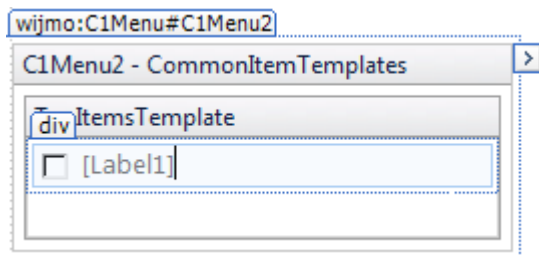
このチュートリアルでは、**C1Menu** コントロールの **ItemsTemplate** テンプレートの作成方法を学びます。このテンプレートは、**C1Menu** コントロールのすべての項目のプロトタイプとなり、**Div** 要素、**入力(チェックボックス)**コントロール、**ラベル**コントロールで構成されます。ラベルコントロールの **Text** プロパティは、個別メニュー項目の **Text** プロパティに連結されます。


以下の手順を実行します。

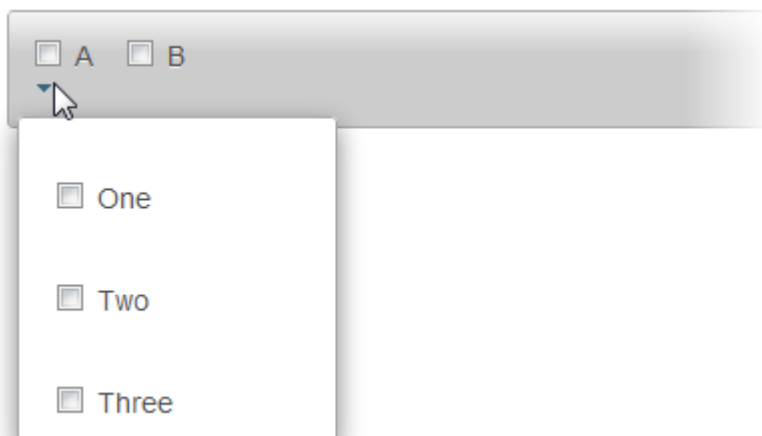
1. A と B という名前の2つの最上位メニュー項目を含む **C1Menu** コントロールを作成します。項目 A に3つの子項目を追加して、それらに One、Two、および Three という名前を付けます。



2. **C1Menu** コントロールのスマートタグ (🔗) をクリックして、[**C1Menu タスク**]メニューから[**テンプレートの編集**]を選択します。[**C1Menu タスク**]メニューがテンプレート編集モードに切り替わります。[**説明**]のドロップダウンリストに示されたデフォルトモードは、**ItemsTemplate** であるため、それを変更する必要はありません。
3. ツールボックスに移動し、ドラッグ & ドロップ操作で、**Div** 要素を **ChildItems** テンプレートに追加します。
4. ツールボックスに移動し、ドラッグ & ドロップ操作で、**入力(チェックボックス)**コントロールと**ラベル**コントロールを **Div** 要素に追加します。テンプレートは次のようになります。




- ラベル要素を選択し、プロパティ ウィンドウに移動して、**Text** プロパティを「<% #DataBinder.Eval(Container.DataItem, "Text") %>」に設定します。これによって、ラベルの **Text** プロパティがデータ連結（後述の手順で指定）に連結されます。
- [**テンプレート編集の終了**] をクリックして、テンプレートを閉じます。
- 次の手順を実行して、データ連結を作成します。
  - C1Menu** コントロールのスマートタグ (  ) をクリックして、[**C1Menu タスク**] メニューから [**メニューのデータバインディングの編集...**] を選択します。[**バインディングコレクションエディタ**] ダイアログボックスで、次の手順を実行します。
    - <追加> をクリックして、<Empty> データ連結をプロジェクトに追加します。
    - <Empty> データ連結の **TextField** プロパティを **Text** に設定します。
    - <OK> をクリックして、[**バインディングコレクションエディタ**] を閉じます。
- プロジェクトを実行して、**A** を選択します。**C1Menu** コントロールの各メニュー項目（最上位と子項目）が1つのチェックボックスと1つのラベルが含むことを確認します。また、ラベルテキストがこのチュートリアルの手順1で指定したテキストと同じであること（このチュートリアルで指定したデータ連結であるため）を確認します。

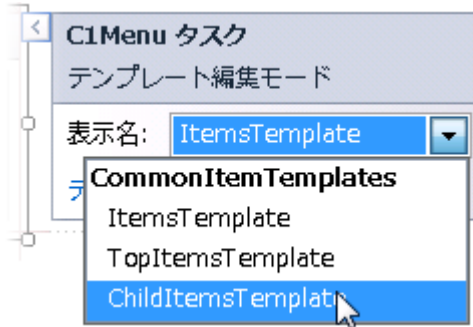


## ChildItemsTemplate の作成

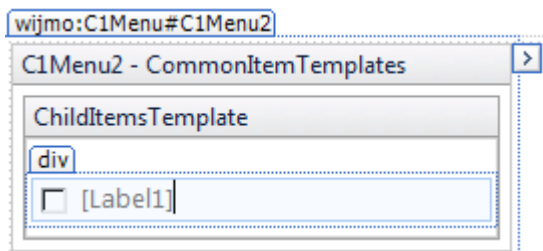
このチュートリアルでは、**C1Menu** コントロールの **ChildItemsTemplate** テンプレートの作成方法を学びます。このテンプレートは、**C1Menu** コントロールのすべての項目のプロトタイプとなり、**Div** 要素、**入力(チェックボックス)** コントロール、ラベルコントロールで構成されます。ラベルコントロールの **Text** プロパティは、個別メニュー項目の **Text** プロパティに連結されません。

以下の手順を実行します。

- 1つの最上位メニュー項目を持つ **C1Menu** コントロールを作成します。子項目に One、Two、および Three という名前を付けます。
- C1Menu** コントロールのスマートタグ (  ) をクリックして、[**C1Menu タスク**] メニューから [**テンプレートの編集**] を選択します。[**C1Menu タスク**] メニューがテンプレート編集モードに切り替わります。
- [**表示名**] のドロップダウンリストボックスから **ChildTopItemsTemplate** を選択します。



4. ツールボックスに移動し、ドラッグ & ドロップ操作で、**Div** 要素を **ChildItems** テンプレートに追加します。
5. ツールボックスに移動し、ドラッグ & ドロップ操作で、**入力(チェックボックス)**コントロールと**Label**コントロールを **Div** 要素に追加します。テンプレートは次のようになります。



6. Label 要素を選択し、プロパティウィンドウに移動して、Text プロパティを「<% #DataBinder.Eval(Container.DataItem, "Text") %>」に設定します。これによって、ラベルの Text プロパティがデータ連結(後述の手順で指定)に連結されます。
7. [テンプレート編集の終了]をクリックして、テンプレートを閉じます。
8. 次の手順を実行して、データ連結を作成します。
  - a. **C1Menu** コントロールのスマートタグ ( ) をクリックして、[**C1Menu タスク**]メニューから[**メニューのデータバインディングの編集...**]を選択します。[**バインディングコレクションエディタ**]ダイアログボックスで、次の手順を実行します。
    - b. <追加>をクリックして、<Empty> データをプロジェクトに追加します。
    - c. <Empty> データ連結の **TextField** プロパティを **Text** に設定します。
    - d. <OK>をクリックして、[**バインディングコレクションエディタ**]を閉じます。
9. プロジェクトを実行して、LinkItem1 を選択します。サブメニューの各メニュー項目が1つのチェックボックスと1つのラベルを含むことを確認します。また、ラベルテキストがこのチュートリアルの手順1で指定したテキストと同じであることを確認します。これは、このチュートリアルで指定したデータ連結であるためです。



## 最上位項目テンプレートの作成

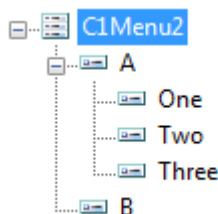
このチュートリアルでは、**C1Menu** コントロールの **TopItemsTemplate** テンプレートの作成方法を学びます。このテンプレー

# Menu for ASP.NET Web Forms

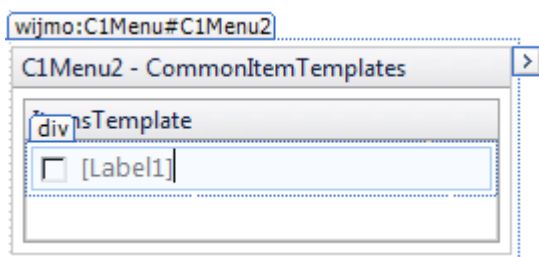
トは、**C1Menu** コントロールのすべての項目のプロトタイプとなり、**Div** 要素、**入力(チェックボックス)**コントロール、ラベルコントロールで構成されます。ラベルコントロールの **Text** プロパティは、個別メニュー項目の **Text** プロパティに連結されます。

以下の手順を実行します。

1. A と B という名前の2つの最上位メニュー項目を含む **C1Menu** コントロールを作成します。項目 A に3つの子項目を追加して、それらに One、Two、および Three という名前を付けます。

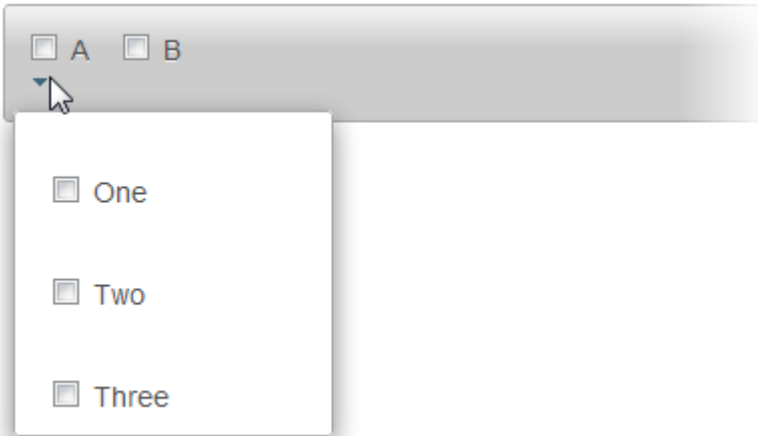


2. **C1Menu** コントロールのスマートタグ ( ) をクリックして、[**C1Menu タスク**]メニューから[**テンプレートの編集**]を選択します。[**C1Menu タスク**]メニューがテンプレート編集モードに切り替わります。[**説明**]のドロップダウンリストに示されたデフォルトモードは、**TopItemsTemplate** であるため、それを変更する必要はありません。
3. ツールボックスに移動し、ドラッグ & ドロップ操作で、**Div** 要素を **ChildItems** テンプレートに追加します。
4. ツールボックスに移動し、ドラッグ & ドロップ操作で、**入力(チェックボックス)**コントロールと**Label**コントロールを **Div** 要素に追加します。テンプレートは次のようになります。



5. **ラベル**要素を選択し、**プロパティ**ウィンドウに移動して、**Text** プロパティを「<% #DataBinder.Eval(Container.DataItem, "Text") %>」に設定します。これによって、ラベルの **Text** プロパティがデータ連結(後述の手順で指定)に連結されます。
6. [**テンプレート編集の終了**]をクリックして、テンプレートを閉じます。
7. 次の手順を実行して、データ連結を作成します。
  - a. **C1Menu** コントロールのスマートタグ ( ) をクリックして、[**C1Menu タスク**]メニューから[**メニューのデータバインディングの編集...**]を選択します。[**バインディングコレクションエディタ**]ダイアログボックスで、次の手順を実行します。
    - a. **<追加>**をクリックして、<Empty> データ連結をプロジェクトに追加します。
    - b. <Empty> データ連結の **TextField** プロパティを **Text** に設定します。
    - c. **<OK>**をクリックして、[**バインディングコレクションエディタ**]を閉じます。
8. プロジェクトを実行して、**A** を選択します。最上位メニュー項目のみが1つのチェックボックスと1つのラベルを含むことを確認します。また、最上位メニュー項目上のラベルテキストがこのチュートリアルの手順1で指定したテキストと同じであることを確認します。これはこのチュートリアルで指定したデータ連結であるためです。





## CSS セレクタでの作業

C1Menu では、CSS サポートによる完全なカスタマイズが可能です。このセクションでは、組み込み CSS セレクタを使用して **C1Menu** コントロールをカスタマイズする方法を学びます。単独のセレクタを使用することも、複数のセレクタを組み合わせることで CSS をより特定のすることもできます。

## CSS セレクタによる C1Menu 外観のカスタマイズ

ソースビューで、CSS セレクタを使用して、**C1Menu** コントロールの外観をカスタマイズできます。

1. ソースビューで、最初の `<asp:Content>` タグセットを探します。次のタグを、CSS スタイルを挿入できる `<asp:Content>` `<asp:Content>` タグの間に追加します。

ソースビュー

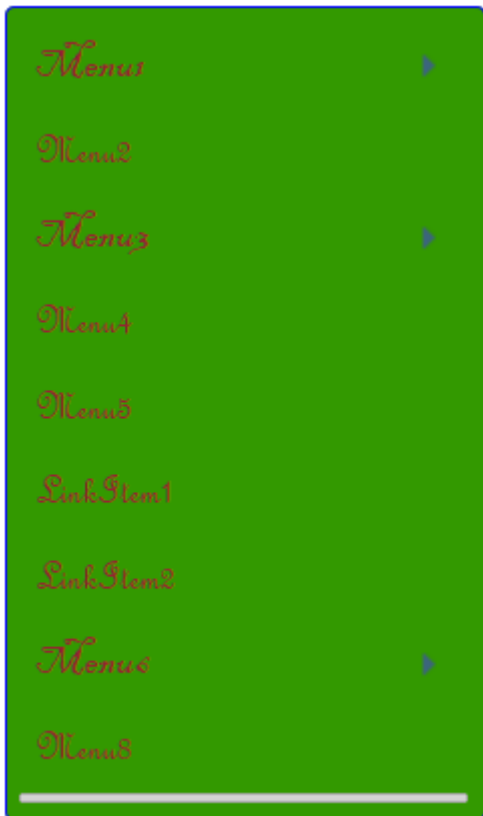
```
<style type="text/css"> </style>
```

2. 次の CSS セレクタを使用して、コントロールの一般的な外観を設定します。

ソースビュー

```
.wijmo-wijmenu
```

3. `{background: #339900; border-color: Blue; }` を CSS セレクタの後に挿入します。これによって、緑色の背景が青色の枠線で設定されます。
4. `.wijmo-wijmenu-text {color: #992233; }` を挿入して、コントロールのテキストの色を設定します。
5. `.wijmo-wijmenu-item {font-family: French Script MT; }` と `.wijmo-wijmenu-parent {font-family: Blackadder ITC; }` を挿入して、メニュー項目と親メニュー項目のフォントを設定します。
6. [F5]を押して、プログラムを実行します。**C1Menu** コントロールの外観に加えた変更にご注意ください。次の図のようになります。親メニュー項目と通常メニュー項目が異なるフォントスタイルで表示されることに注意してください。



## CSS セレクタによる C1Menu リンクの外観のカスタマイズ

C1Menu では、CSS セレクタを使用してリンクの外観をカスタマイズできます。このトピックでは、CSS セレクタを使用してマウスホバー時のリンクをカスタマイズする方法を学びます。

1. ソースビューで、最初の `<asp:Content>` タグセットを探します。次のタグを、CSS スタイルを挿入できる `<asp:Content>` `<asp:Content>` タグの間に追加します。

ソースビュー

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

2. 次の CSS セレクタを使用して、コントロールの一般的な外観を設定します。

ソースビュー

```
.wijmo-wijmenu a.wijmo-wijmenu-link:hover  
{  
  font-family:Calibri; background: Pink; border-color: green; border-style: solid;  
}
```

3. [F5]を押して、プログラムを実行します。メニュー項目の1つにホバーします。次のような図になります。



## コンテキストメニューとしての C1Menu コントロールの表示

C1Menu は、他のほとんどの ComponentOne コントロールのコンテキストメニューとして使用できます。このトピックでは、C1Menu コントロールを C1TreeView コントロールのコンテキストメニューとして表示する手順について説明します。

以下の手順を実行します。

1. **C1Menu** コントロールと **C1TreeView** コントロールをアプリケーションに追加します。
2. **C1Menu** のスマートタグをクリックして[**C1Menu タスク**]メニューを開き、[**メニューの編集**]を選択します。
3. **C1Menu デザイナー**フォームで、**<子項目の追加>**ボタンをクリックし、**C1MenuItem** を **C1Menu** コントロールに追加します。
4. 今追加した **C1MenuItem** を選択し、プロパティウィンドウで **Text** プロパティを探します。**Text** プロパティのテキストボックスに "ノードを追加する" と入力します。
5. **<子項目の追加>**ボタンをクリックして、**C1Menu** に **C1MenuItem** を追加します。
6. 今追加した **C1MenuItem** を選択し、プロパティウィンドウで **Text** プロパティを探します。**Text** プロパティのテキストボックスに "ノードを削除する" と入力します。
7. **<OK>**をクリックして、**C1Menu デザイナー**フォームを閉じ、ソースビューに切り替えます。
8. `<cc1:C1Menu>` タグに次のマークアップを追加します。

ソースビュー

```
onclientselct="C1Menu_OnClientSelect" Orientation="Vertical"
  Trigger="#C1TreeView1" TriggerEvent="Rtclick"
```

完全な開始タグのマークアップは次のようになります。

ソースビュー

```
<cc1:C1Menu ID="C1Menu1" runat="server"
  onclientselct="C1Menu_OnClientSelect" Orientation="Vertical"
  Trigger="#C1TreeView1" TriggerEvent="Rtclick">
```

9. `<cc1:C1TreeView>` を探し、このタグの間に以下のマークアップを追加します。

## ソースビュー

```
<Nodes>
  <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
    Text="Node1">
    <Nodes>
      <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
        Text="SubNode1">
      </cc1:C1TreeNode>
      <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
        Text="SubNode2">
      </cc1:C1TreeNode>
    </Nodes>
  </cc1:C1TreeNode>
  <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
    Text="Node2">
    <Nodes>
      <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
        Text="SubNode1">
      </cc1:C1TreeNode>
      <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
        Text="SubNode2">
      </cc1:C1TreeNode>
    </Nodes>
  </cc1:C1TreeNode>
</Nodes>
```

10. `<cc1:C1Menu>` マークアップの上に以下のスクリプトを追加します。

## ソースビュー

```
<script type="text/javascript">
  $(document).ready(function () {
    $(".wijmo-wijtree-node").mousedown(function (e) {
      if (e.button == 2)
        $(this).parent().c1treeviewnode("option", "selected", "true");
    });
  });
  >

  function C1Menu_OnClientSelect(sender, args) {
    alert(args.item[0].innerText + " が選択されました。");
  }
</script>
```

11. アプリケーションを実行します。 **C1TreeView** ノードのいずれかを右クリックすると、メニューが表示されます。

## メニューへの最上位項目の追加

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、最上位メニュー項目を **C1Menu** コントロールに追加する方法を例示します。

### デザインビューの場合

以下の手順を実行します。

1. スマートタグをクリックして、[C1Menu タスク]メニューを開きます。[メニューの編集]を選択します。  
[C1Menu デザインフォーム]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、C1MenuItem を C1Menu コントロールに追加します。
3. <OK>をクリックして、C1Menu デザインフォームを閉じます。

### ソースビューの場合

次のマークアップを <cc1:C1Menu> タグの間に追加します。

#### ソースビュー

```
<cc1:C1MenuItem runat="server" Text="LinkItem1">
</cc1:C1MenuItem>
```

### コードビューの場合

1. 以下の手順を実行して、次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 C1.Web.Wijmo.Controls.C1Menu
```

#### C# コードの書き方

```
C#
using C1.Web.Wijmo.Controls.C1Menu;
```

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 MenuItem1 As New C1MenuItem()
C1MenuItem1.Text = "LinkItem1"
C1Menu1.Items.Add(C1MenuItem1)
```

#### C# コードの書き方

```
C#
C1MenuItem MenuItem1 = new c1MenuItem();
C1MenuItem1.Text = "LinkItem1";
C1Menu1.Items.Add(c1MenuItem1);
```

3. プログラムを実行します。

## ドロップダウンメニューの作成

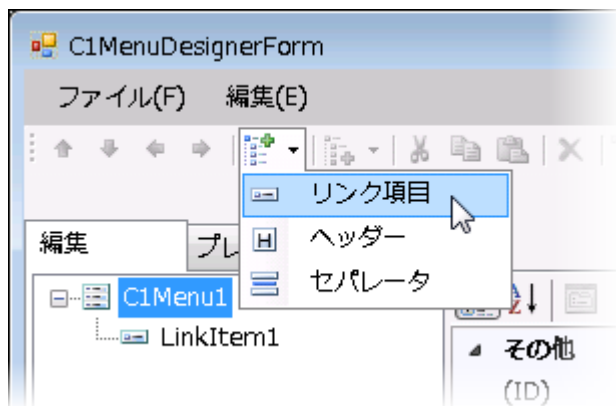
このトピックでは、デザインビュー、ソースビュー、およびコードを使用したドロップダウンメニューの作成を例示します。ドロップダウンメニューを追加するには、必要な作業は1つ(またはそれ以上)の **C1MenuItem** を最上位 C1MenuItem の子として追加するだけです。このトピックでは、「[メニューへの最上位項目の追加](#)」を完了したものと想定しています。

### デザインビューの場合

# Menu for ASP.NET Web Forms

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、[C1Menu タスク]メニューを開きます。[メニューの編集]を選択します。  
[C1Menu デザイナーフォーム]ダイアログボックスが開きます。
2. サブ項目を追加したいメニュー項目を選択します。
3. [子項目の追加]のドロップダウン矢印をクリックして、リストから[リンク項目]を選択して、子項目を選択されたメニュー項目に追加します。



4. <OK>をクリックして、C1Menu デザインフォームを閉じます。

## ソースビューの場合

次のマークアップを、サブメニューを追加したい項目の <cc1:C1MenuItem> タグの間に追加します。

### ソースビュー

```
<Nodes>
  <cc1:C1MenuItem runat="server" Text="LinkItem1">
    </cc1:C1MenuItem>
</Nodes>
```

## コードビューの場合

以下の手順を実行します。

1. 次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 C1.Web.Wijmo.Controls.C1Menu
```

### C# コードの書き方

```
C#
using C1.Web.Wijmo.Controls.C1Menu;
```

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
'最初のノードを作成して、C1Menu に追加します。
```

```
Dim C1MenuItem1 As New C1MenuItem()
C1MenuItem1.Text = "LinkItem1"
C1Menu1.Items.Add(C1MenuItem1)
' 子ノードを作成して、C1MenuItem1 に追加します。
Dim C1MenuItem2 As New C1MenuItem()
C1MenuItem1.Text = "LinkItem1"
C1MenuItem1.Items.Add(C1MenuItem2)
```

### C# コードの書き方

```
C#
// 最初のノードを作成して、c1Menu に追加します。
C1MenuItem c1MenuItem1 = new c1MenuItem();
C1MenuItem1.Text = "LinkItem1";
C1Menu1.Items.Add(c1MenuItem1);
// 子ノードを作成して、c1MenuItem1 に追加します。
C1MenuItem c1MenuItem2 = new c1MenuItem();
C1MenuItem2.Text = "LinkItem1";
C1MenuItem1.Items.Add(c1MenuItem2);
```

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。

サブメニューの作成は、子の **C1MenuItem** を親の **C1MenuItem** に追加するだけで簡単です。

## スライディングメニューの作成

**Menu for ASP.NET Web Forms** は、従来のフライアウトメニューか、またはタッチフォンタイプのスライディングメニューとして表示できます。このチュートリアルでは、ブレッドクラムヘッダーと伸縮スライディングアニメーションを設定したスライディングメニューを作成します。

以下の手順を実行します。

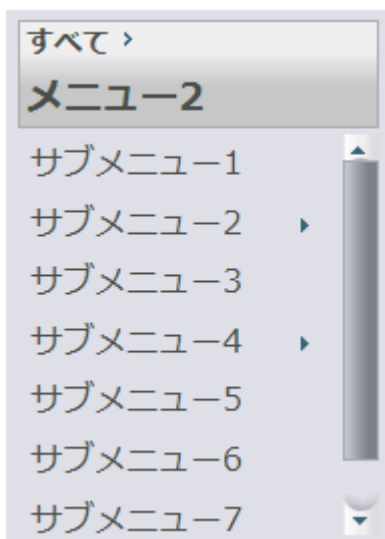
1. ソースビューで、次のマークアップを `<cc1:C1Menu>` と `<cc1:C1Menu>` タグの間に追加して、メニュー項目とサブメニュー項目を **C1Menu** コントロールに追加します。

#### ソースビュー

```
<Items>
<cc1:C1MenuItem runat="server" Text="メニュー1" IconClass="ui-icon ui-icon-arrowthick-1-w"
ID="MenuItem1">
</cc1:C1MenuItem>
<cc1:C1MenuItem runat="server" Text="メニュー2" IconClass="ui-icon ui-icon-arrowrefresh-1-s"
ID="C1MenuItem1">
<Items>
<cc1:C1MenuItem runat="server" Text="サブメニュー1">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem8" runat="server" Text="サブメニュー2">
<Items>
<cc1:C1MenuItem Text="サブメニュー21" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem15" Text="サブメニュー22" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem16" Text="サブメニュー23" runat="server">
```

```
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem17" Text="サブメニュー24" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem18" Text="サブメニュー25" runat="server">
</cc1:C1MenuItem>
</Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem9" runat="server" Text="サブメニュー3">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem10" runat="server" Text="サブメニュー4">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem11" runat="server" Text="サブメニュー5">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem12" runat="server" Text="サブメニュー6">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem13" runat="server" Text="サブメニュー7">
</cc1:C1MenuItem>
</Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem runat="server" Text="メニュー3" IconClass="ui-icon ui-icon-comment"
ID="C1MenuItem2">
</cc1:C1MenuItem>
</Items>
```

2. デザインビューに切り替えます。
3. **C1Menu** コントロールのスマートタグをクリックして、[**C1Menu タスク**]メニューを開きます。[**メニューの編集**]をクリックします。  
[**C1Menu デザイナフォーム**]ダイアログボックスが開きます
4. ツリービューで **C1Menu1** が選択された状態で、次のプロパティを設定します。
  - **Mode** プロパティを **Sliding** に設定します。これによって、**C1Menu** は従来のフライアウトメニューからスライドフオンタイプのスライディングメニューに切り替わります。
  - **BackLink** プロパティを **False** に設定します。これによって、デフォルトのバックリンクがブレッドクラムリンクに置き換わります。
  - **SlidingAnimation** ノードを拡張し、**Easing** プロパティを **EaseInOutElastic** に設定します。これによって、アニメーションがスライディングメニューに設定されます。
5. <OK>をクリックして、**C1Menu デザイナフォーム** を閉じます。
6. [F5]を押して、プロジェクトを実行します。プロジェクトは次のような表示になります。

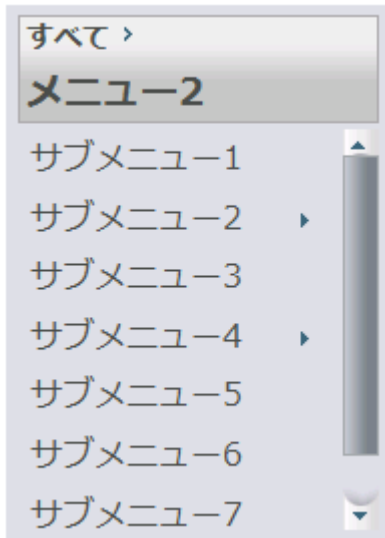


メニューの上部の「オプションの選択」テキストはプレースホルダです。これは、メニューをナビゲートしたときにブレッド

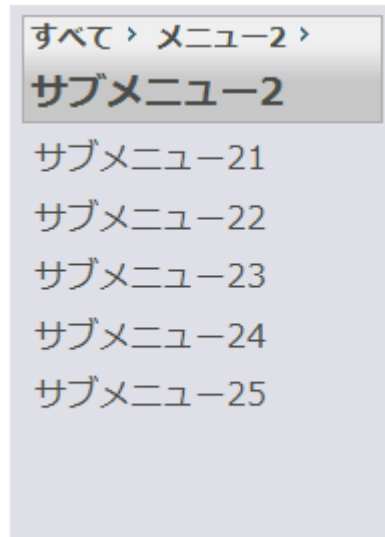


クラムリンクが表示される場所です。

7. [メニュー2]をクリックします。伸縮アニメーションが新しいメニュー画面として表示領域にスライドインするのに気づきましたか？また、ブレッダラムバーが[すべて] > [メニュー2]となっていることも確認します。



8. [サブメニュー2]をクリックして、ブレッダラムが[すべて] > [メニュー2] > [サブメニュー2]となることを確認します。[メニュー2]メニューに戻るには、[メニュー2]をクリックします。最上位メニューに戻るには、[すべて]をクリックします。



## C1Menu のアニメーション化

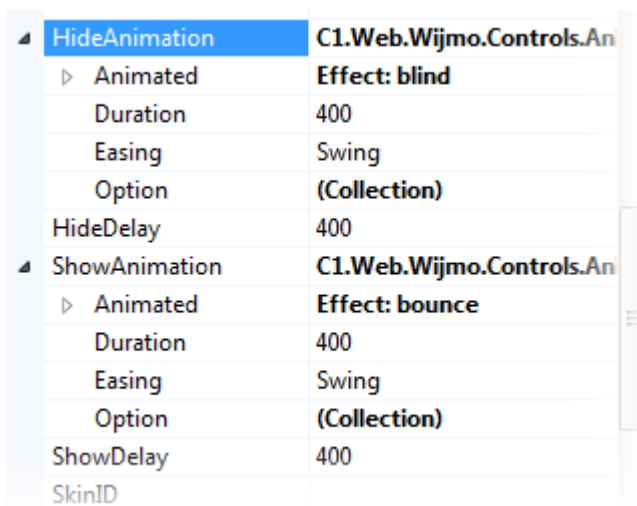
C1Menu コントロールはアニメーション効果をサポートします。このトピックでは、[C1Menu タスク]メニューまたはソースビューのいずれかを使用して、アニメーション効果を変更する方法を例示します。

### デザインビューの場合

1. C1Menu のスマートタグをクリックして、[C1Menu タスク]メニューを開きます。
2. <メニューの編集>リンクをクリックして、[C1Menu デザイナフォーム]を開きます。



3. リストの **Hide Animation** と **Show Animation** プロパティを探します。矢印を使用して、プロパティノードを拡張します。



HideAnimation	C1.Web.Wijmo.Controls.An
▶ Animated	Effect: blind
Duration	400
Easing	Swing
Option	(Collection)
HideDelay	400
ShowAnimation	C1.Web.Wijmo.Controls.An
▶ Animated	Effect: bounce
Duration	400
Easing	Swing
Option	(Collection)
ShowDelay	400
SkinID	

4. **HideAnimation** > **Animation** > **Effect** プロパティを **blind** に設定し、**ShowAnimation** > **Animation** > **Effect** プロパティを **bounce** に設定します。
5. **<OK>** をクリックして、**[F5]** を押してプログラムを実行します。サブメニューの開閉時のアニメーション効果に注意してください。

## ソースビューの場合

1. プロジェクトのソースビューで、次のマークアップを `<cc1:C1Menu>` タグの間に追加します。

```
ソースビュー
<ShowAnimation>
  <Animated Effect="bounce" />
</ShowAnimation>
<HideAnimation>
  <Animated Effect="blind"></Animated>
</HideAnimation>
```

2. **[F5]** を押して、プログラムを実行します。マウスをサブメニューにホバーし、アニメーション効果を表示させます。

## メニュー項目のトリガーの変更

**C1Menu** コントロールは、異なるトリガーイベントに反応してメニュー項目を開くことができます。このトピックでは、デザインビューまたはソースビューで **Trigger** と **TriggerEvent** プロパティを設定し、メニュー項目を開くトリガーを変更する方法を例示します。

### デザインビューの場合

1. C1Menu のスマートタグをクリックして、[C1Menu タスク]メニューを開きます。
2. <メニューの編集>リンクをクリックして、[C1Menu デザイナフォーム]を開きます。



3. リストの **Trigger** と **TriggerEvent** プロパティを探します。
4. **Trigger** プロパティを **.wijmo-wijmenu-item** に設定してから、**TriggerEvent** のドロップダウンリストを使用してトリガーイベントを選択します。このタスクでは、次の図のように、**TriggerEvent** を **Rtclick** に設定します。

SlidingAnimation	C1.Web.Wijmo.Controls.SlideAnimation
Theme	cobalt
ToolTip	
Trigger	.wijmo-wijmenu-item
TriggerEvent	Rtclick
UseCDN	False

5. <OK>をクリックし、[F5]を押してアプリケーションを実行します。メニュー項目を右クリックするとサブメニューが表示されます。

### ソースビューの場合

1. ソースビューで、TriggerEvent="Rtclick" と Trigger=".wijmo-wijmenu-item" を <cc1:C1Menu> タグに追加します。[F5]を押して、プロジェクトを実行します。メニュー項目の右クリックしてサブメニューをドロップダウンさせるようになったことに注意してください。
2. [F5]を押して、プロジェクトを実行します。メニュー項目の右クリックしてサブメニューをドロップダウンさせるようになったことに注意してください。

## C1Menu 項目へのアイコンの追加

iconClass プロパティを使用して、C1Menu コントロールにアイコンを表示できます。このトピックは、コントロールにアイコンを追加する方法について説明します。

以下のアイコンが、このヘルプに表示されます。

アイコン名	アイコン	アイコンのマークアップ
-------	------	-------------

# Menu for ASP.NET Web Forms

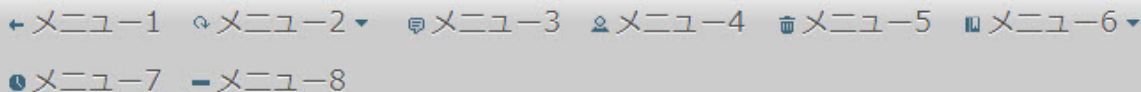
太い矢印		ui-icon ui-icon-arrowthick-1-w
更新矢印		ui-icon ui-icon-arrowrefresh-1-s
コメント		ui-icon ui-icon-comment
人		ui-icon ui-icon-person
ごみ箱		ui-icon ui-icon-trash
ブックマーク		ui-icon ui-icon-bookmark"
時計		ui-icon ui-icon-clock
太いマイナス		ui-icon ui-icon-minusthick

1. ソースビューで、次のマークアップを最初の `<cc1:C1MenuItem>` タグに追加します。

ソースビュー

```
"IconClassui-icon ui--icon-arrowthick-1-w"
```

2. 残りの C1MenuItem へのアイコンのマークアップの追加を続けます。
3. アプリケーションを実行すると、次の図のように表示されます。



## コンテキストメニューの位置の設定

メニューを表示するボタンなどのトリガーと関連させて、コンテキストメニューを表示する位置をカスタマイズできます。

1. C1Menu コントロールと一般的な Button コントロールをプロジェクトに追加します。
2. ソースビューに切り替え、次のような記述になるようにマークアップを編集します。

ソースビュー

```
<input type="button" value="click here" id="btn1" />
  <cc1:C1Menu runat="server" ID="Menu1" Orientation="Vertical" Trigger="#btn1"
  TriggerEvent="Click">
  <Animation Option=""></Animation>

  <ShowAnimation Option=""></ShowAnimation>

  <HideAnimation Option="">
  <Animated Effect="fade"></Animated>
  </HideAnimation>
    <Items>
      <cc1:C1MenuItem ID="C1MenuItem1" runat="server" Text="Breaking News">
      </cc1:C1MenuItem>
      <cc1:C1MenuItem ID="C1MenuItem2" runat="server" Text="Entertainment">
      </cc1:C1MenuItem>
      <cc1:C1MenuItem ID="C1MenuItem3" runat="server" Text="Finance">
      </cc1:C1MenuItem>
```

```

<cc1:C1MenuItem ID="C1MenuItem4" runat="server" Text="Food &#38; Cooking">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem5" runat="server" Text="Lifestyle">
  <Items>
    <cc1:C1MenuItem ID="C1MenuItem8" runat="server"
Text="submenu"></cc1:C1MenuItem>
  </Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem6" runat="server" Text="News">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem7" runat="server" Text="Politics">
</cc1:C1MenuItem>
<cc1:C1MenuItem ID="C1MenuItem9" runat="server" Text="Sports">
</cc1:C1MenuItem>
</Items>
</cc1:C1Menu>

```

C1Menu コントロールのクリックイベントとトリガープロパティが設定されていることに注意してください。

- デザインビューに戻り、C1Menu コントロールをクリックして、コントロールプロパティをプロパティウィンドウに表示します。下方へスクロールして、Position プロパティを見つけ、次の図のようにオプションを設定します。

Position	
▲ At	<b>Center, Top</b>
Left	<b>Center</b>
Top	Top
▲ Collision	<b>Flip, Flip</b>
Left	Flip
Top	Flip
▲ My	<b>Right, Top</b>
Left	<b>Right</b>
Top	Top
▲ <b>Offset</b>	<b>0, 0</b>
Left	0
Top	0

- アプリケーションを実行すると、Button コントロールが表示されます。ボタンをクリックすると、コンテキストメニューがプロパティで指定した位置に表示されます。

## C1Menuの関数

**C1Menu** コントロールは、大規模なクライアント側 API を備えています。このトピックでは、ソースビューで、jQuery UI に見られる同じパターンを使用してクライアント側メソッドを呼び出す方法を例示します。

- ソースビューで、次のマークアップを <cc1:C1Menu> タグの間に追加します。

### 追加するマークアップ

```

ソースビュー
<Items>
  <cc1:C1MenuItem Text="MenuItem" runat="server">

```

# Menu for ASP.NET Web Forms

```
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="新着" runat="server">
  <Items>
    <cc1:C1MenuItem runat="server" Header="true" Text="ヘッダー2">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Separator="true">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="エンタメ">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="政治">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="国際">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="文化">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="社会">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="天気">
      </cc1:C1MenuItem>
  </Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem runat="server" Text="エンタメ">
  <Items>
    <cc1:C1MenuItem runat="server" Text="エンタメトップ">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="舞台">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="映画">
      </cc1:C1MenuItem>
    <cc1:C1MenuItem runat="server" Text="音楽">
      <Items>
        <cc1:C1MenuItem runat="server" Text="オルタナティブ">
          </cc1:C1MenuItem>
        <cc1:C1MenuItem runat="server" Text="カントリー">
          </cc1:C1MenuItem>
        <cc1:C1MenuItem runat="server" Text="ダンス">
          </cc1:C1MenuItem>
        <cc1:C1MenuItem runat="server" Text="エレクトロニカ">
          </cc1:C1MenuItem>
        <cc1:C1MenuItem runat="server" Text="メタル">
          </cc1:C1MenuItem>
        <cc1:C1MenuItem runat="server" Text="ポップ">
          </cc1:C1MenuItem>
        <cc1:C1MenuItem runat="server" Text="ロック">
          <Items>
            <cc1:C1MenuItem runat="server" Text="バンド">
              <Items>
                <cc1:C1MenuItem runat="server" Text="ドッケン">
                  </cc1:C1MenuItem>
              </Items>
            </cc1:C1MenuItem>
          </Items>
        </cc1:C1MenuItem>
      </Items>
    </cc1:C1MenuItem>
  </Items>
</cc1:C1MenuItem>
```

```

        </cc1:C1MenuItem>
        <cc1:C1MenuItem runat="server" Text="歌曲">
        </cc1:C1MenuItem>
    </Items>
</cc1:C1MenuItem>
</Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem runat="server" Text="スライドショー">
</cc1:C1MenuItem>
<cc1:C1MenuItem runat="server" Text="赤い絨毯">
</cc1:C1MenuItem>
</Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="金融" runat="server">
    <Items>
        <cc1:C1MenuItem Text="個人" runat="server">
            <Items>
                <cc1:C1MenuItem Text="ローン" runat="server">
                </cc1:C1MenuItem>
                <cc1:C1MenuItem Text="貯蓄" runat="server">
                </cc1:C1MenuItem>
                <cc1:C1MenuItem Text="貸付金" runat="server">
                </cc1:C1MenuItem>
                <cc1:C1MenuItem Text="借金" runat="server">
                </cc1:C1MenuItem>
            </Items>
        </cc1:C1MenuItem>
        <cc1:C1MenuItem Text="企業" runat="server">
        </cc1:C1MenuItem>
    </Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="スポーツ" runat="server">
    <Items>
        <cc1:C1MenuItem Text="野球" runat="server">
        </cc1:C1MenuItem>
        <cc1:C1MenuItem Text="サッカー" runat="server">
        </cc1:C1MenuItem>
        <cc1:C1MenuItem Text="ゴルフ" runat="server">
        </cc1:C1MenuItem>
        <cc1:C1MenuItem Text="一般" runat="server">
            <Items>
                <cc1:C1MenuItem Text="テニス" runat="server">
                </cc1:C1MenuItem>
                <cc1:C1MenuItem Text="水泳" runat="server">
                </cc1:C1MenuItem>
                <cc1:C1MenuItem Text="ラグビー" runat="server">
                </cc1:C1MenuItem>
            </Items>
        </cc1:C1MenuItem>
    </Items>
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="生活" runat="server">

```

```
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="ニュース" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="政治" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="スポーツ" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="小説" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="マガジン" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="ブック" runat="server">
</cc1:C1MenuItem>
<cc1:C1MenuItem Text="教育" runat="server">
</cc1:C1MenuItem>
</Items>
```

2. 終了タグ <cc1:C1Menu>の後に、次のマークアップを挿入して、ボタンコントロールを作成します。

#### ソースビュー

```
<p>
  <input type="button" value="前へ" />
  <input type="button" value="次へ" />
  <input type="button" value="前のページ" />
  <input type="button" value="次のページ" />
</p>
```

3. 次のスクリプトを使用して、クライアント側関数を呼び出します。

#### ソースビュー

```
<script type="text/javascript">
  var count = 0;
  $(document).ready(function () {
    $("#previous").click(function () {
      $("#<%= Menu1.ClientID %>").focus().c1menu("previous");
      count++;
    });
    $("#next").click(function () {
      $("#<%= Menu1.ClientID %>").focus().c1menu("next");
      count++;
    });
    $("#previousPage").click(function () {
      if (count === 0) {
        $("#<%= Menu1.ClientID %>").find(".wijmo-wijmenu-link:first").click();
      }
      $("#<%= Menu1.ClientID %>").c1menu("previousPage");
      count++;
    });
    $("#nextPage").click(function () {
      if (count === 0) {
        $("#<%= Menu1.ClientID %>").find(".wijmo-wijmenu-link:first").click();
      }
      $("#<%= Menu1.ClientID %>").c1menu("nextPage");
      count++;
    });
  });
</script>
```



```

    });
});

</script>

```

4. [F5]を押して、プログラムを実行します。メニューは、次の図のようになります。



## C1Menu への項目の動的な追加

このトピックでは、クライアント側コントロールを使用して、項目を **C1Menu** に動的に追加する方法を例示します。

1. ソースビューに移動し、次のマークアップを `<cc1:C1Menu>` タグの間に挿入して、メニューに移植します。

ソースビュー

```

<Items>
  <cc1:c1menuitem runat="server" text="メニュー項目">
  </cc1:c1menuitem>
  <cc1:c1menuitem runat="server" separator="true">
  </cc1:c1menuitem>
  <cc1:c1menuitem runat="server" text="垂直" value="DynamicOrientationItem">
    <Items>
      <cc1:C1Menuitem runat="server" Text="メニュー項目">
      </cc1:C1Menuitem>
      <cc1:C1Menuitem runat="server" Text="メニュー項目">
      </cc1:C1Menuitem>
      <cc1:C1Menuitem runat="server" Text="メニュー項目">
      </cc1:C1Menuitem>
      <cc1:C1Menuitem runat="server" Text="メニュー項目">
      </cc1:C1Menuitem>
      <cc1:C1Menuitem runat="server" Text="メニュー項目">
      </cc1:C1Menuitem>
    </Items>
  </cc1:c1menuitem>
  <cc1:c1menuitem runat="server" text="メニュー項目">
  </cc1:c1menuitem>
  <cc1:c1menuitem runat="server" text="メニュー項目">
  </cc1:c1menuitem>
</Items>

```

2. 終了タグ `</Items>` の後に、次のマークアップを挿入して、add と remove 関数を制御するボタンを作成します。

### ソースビュー

```
<fieldset>
  <legend>関数の削除</legend>
  <label for="tbSelector">
    セレクタ: </label>
  <input type="text" />
  <label for="tbIndex">
    インデックス: </label>
  <input type="text" />
  <input type="button" value="削除" onclick="remove()" />
</fieldset>
<fieldset>
  <legend>関数の追加</legend>
  <label for="tbItem">
    項目: </label>
  <input type="text" />
  <label for="tbAddSelector">
    セレクタ: </label>
  <input type="text" />
  <label for="tbAddIndex">
    インデックス: </label>
  <input type="text" />
  <input type="button" value="追加" onclick="add()" /> </fieldset>
```

3. 次のスクリプトを挿入して、ボタンを初期化します。

### ソースビュー

```
<script type="text/javascript">
  function remove() {
    var index, selector;
    if ($("#tbIndex").val() != "" && !isNaN($("#tbIndex").val())) {
      index = parseInt($("#tbIndex").val());
    }
    if ($("#tbSelector").val() != "") {
      selector = ($("#tbSelector").val());
    }
    if (!selector && index != undefined) {
      selector = index;
      index = null;
    }
    $("#<%= Menu1.ClientID %>").c1menu("remove", selector, index);
  }

  function add() {
    var index, selector, item;
    item = ($("#tbItem").val());
    if ($("#tbAddIndex").val() != "" && !isNaN($("#tbAddIndex").val())) {
      index = parseInt($("#tbAddIndex").val());
    }
    if ($("#tbAddSelector").val() != "") {
      selector = ($("#tbAddSelector").val());
    }
  }
</script>
```

```

if (!selectr && index != undefined) {
    selector = index;
    index = null;
}
$("#<%= Menu1.ClientID %>").c1menu("add", item, selector, index);
}
</script>

```

4. [F5]を押して、プログラムを実行します。**C1Menu** コントロールは、次の図のようになります。

## C1Menu へのサイトマップの移植

このレッスンでは、**C1Menu** にサイトマップデータを移植する方法を例示します。

サイトマップを作成して、それを **C1Menu** コントロールに連結するには、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で、プロジェクト名を右クリックし、[新しい項目の追加]を選択します。  
[新しい項目の追加]ダイアログボックスが表示されます。
- テンプレートのリストから[サイトマップ]を選択し、<追加>をクリックして、新しい **Web.sitemap** ページをプロジェクトに追加します。  
次のような **Web.sitemap** ファイルのデフォルトソースコードが表示されます。

ソースビュー

```

<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="" description="">
    <siteMapNode url="" title="" description="" />
    <siteMapNode url="" title="" description="" />
  </siteMapNode>
</siteMap>

```

3. この Web.sitemap ファイルのデフォルトデータを、次のデータで置き換えます。

ソースビュー

```

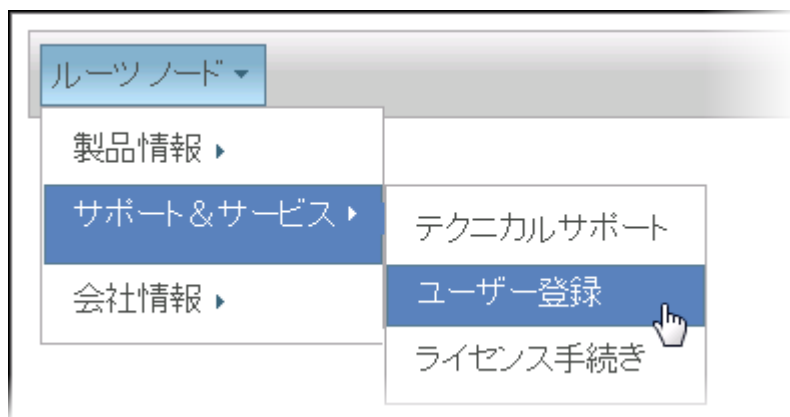
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="RootNodeUrl" title="ルーツ ノード" description="Root Node" >
    <siteMapNode url="" title="製品情報" description="製品情報" >
      <siteMapNode url="" title=".NET 製品" description=".NET 製品">
        <siteMapNode url="" title="Windows Forms" description="Windows Forms" />
        <siteMapNode url="" title="ASP.NET" description="ASP.NET" />
        <siteMapNode url="" title="Silverlight" description="Silverlight" />
        <siteMapNode url="" title="WPF" description="WPF" />
        <siteMapNode url="" title="Windows Phone" description="Windows Phone" />
      </siteMapNode>
    </siteMapNode>
  </siteMapNode>
</siteMap>

```

# Menu for ASP.NET Web Forms

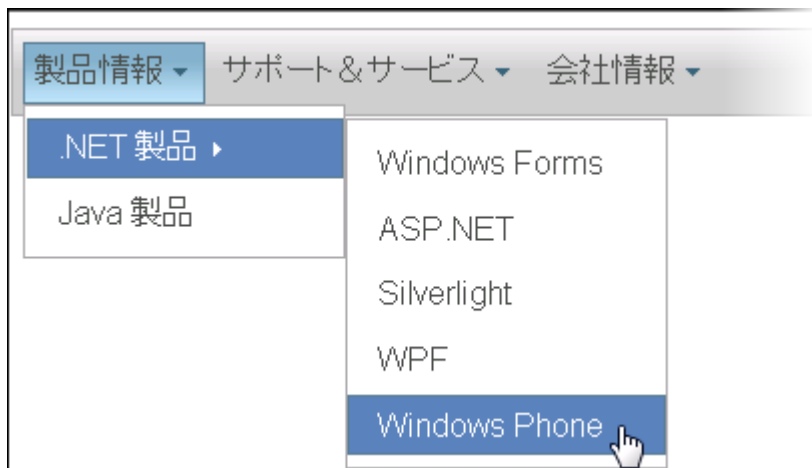
```
<siteMapNode url="" title="Java 製品" description="Java 製品" />
</siteMapNode>
<siteMapNode url="" title="サポート&サービス" description="サポート&サービス" >
  <siteMapNode url="" title="テクニカルサポート" description="テクニカルサポート" />
  <siteMapNode url="" title="ユーザー登録" description="ユーザー登録" />
  <siteMapNode url="" title="ライセンス手続き" description="ライセンス手続き" />
</siteMapNode>
<siteMapNode url="" title="会社情報" description="会社情報">
  <siteMapNode url="" title="概要" description="概要" />
  <siteMapNode url="" title="ニュースリリース" description="ニュースリリース" />
  <siteMapNode url="" title="お問合せ" description="お問合せ" />
</siteMapNode>
</siteMapNode>
</siteMap>
```

4. **C1Menu** コントロールのタスクメニューを開き、[データソースの選択]のドロップダウン矢印をクリックします。[新しいデータソース]を選択して、[データソース構成ウィザード]を開きます。
5. [サイトマップ]を選択して、<OK>をクリックします  
**SiteMapDataSource1** がプロジェクトに追加されます。
6. [F5]を押して、プロジェクトを実行します。次のことを確認します。  
**Web.sitemap** ファイルのデータが **C1Menu** コントロールに反映されています。



コントロールが最上位ノードである ComponentOne で開くことを確認します。次の手順では、**C1Menu** の最上位ノードを削除して、第2階層のノードを表示させる方法を学びます。

7. ブラウザを閉じて、プロジェクトに戻ります。
8. デザインビューで、**SiteMapDataSource** を選択し、プロパティウィンドウで **ShowStartingNode** を **False** に設定します。
9. [F5]を押して、プロジェクトを実行します。最上位ノードが削除されたことを確認します。



## C1Menu への XML の移植

このチュートリアルでは、テンプレートをインストールし、XML データソースコンポーネントを Web サイトに追加して、それを **C1Menu** コントロールに割り当てた後、**C1Menu** の連結を設定する方法を学びます。

以下の手順を実行します。

1. ツールボックスから、**C1Menu** アイコンをダブルクリックして、このコントロールをプロジェクトに追加します。
2. 次の手順を実行して、XML ファイルを作成して準備します。
3. ソリューションエクスプローラで、**App\_Data** を右クリックし、[新しい項目の追加]を選択します。[新しい項目の追加]ダイアログボックスが表示されます。
  - a. XML ファイルを選択し、その名前を Menu\_Hierarchy.xml に変更します。XML ファイルが開きます。
  - b. XML ビューで、次のデータを **Menu\_Hierarchy.xml** ドキュメントに追加します。

ソースビュー

```
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <MenuItem Text="ホーム">
    <MenuItem Text="会社概要"> </MenuItem>
    <MenuItem Text="ご挨拶"> </MenuItem>
    <MenuItem Text="国内拠点"> </MenuItem>
  </MenuItem>
  <MenuItem Text="製品">
    <MenuItem Text="ハードウェア"> </MenuItem>
    <MenuItem Text="ソフトウェア"> </MenuItem>
  </MenuItem>
  <MenuItem Text="サービス">
    <MenuItem Text="トレーニング"> </MenuItem>
    <MenuItem Text="コンサルティング"> </MenuItem>
  </MenuItem>
</root>
```

4. **.aspx** ページのデザインビューに戻り、次の手順を実行して、新しいデータソースを作成します。
  - a. C1Menu のスマートタグをクリックして、[**C1Menu タスク**]メニューを開いた後、[**データソースの選択**]の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。
  - b. [**データソース構成ウィザード**]ダイアログボックスが開きます。
  - c. [**XML ファイル**]を選択して、**<OK>**をクリックします。  
**XmlDataSource1** がプロジェクトに追加されます。
5. 次の手順を実行して、データソースを設定します。
  - a. **C1Menu** のスマートタグをクリックして、[**C1Menu タスク**]メニューを開き、[**データソースの構成**]をクリックします。  
[**データソースの構成**]ダイアログボックスが開きます。
  - b. [XPath 式]テキストフィールドに、root/MenuItem を入力します。これによって、ルートの子であるすべての MenuItem が選択されて、Web ページ上の最上位となります。
  - c. [**データファイル**]テキストフィールドの横の**<参照>**をクリックして、[**XML ファイルの選択**]ダイアログボックスを開きます。
  - d. **App\_Data** プロジェクトフォルダを選択してから、[**フォルダの内容**]ペインから **Menu\_Hierarchy.xml** を選択します。
  - e. **<OK>**をクリックして、[**XML ファイルの選択**]ダイアログボックスを閉じます。
  - f. **<OK>**をクリックして、[**データソースの構成**]ダイアログボックスを閉じます。
6. 以下の手順を実行して、XML タグを **C1MenuItem** に連結します。
  - a. **C1Menu** のスマートタグをクリックして、[**C1Menu タスク**]メニューを開き、[**メニューのデータバインディングの編集**]をクリックします。  
[**バインディングコレクションエディタ**]ダイアログボックスが開きます。
  - b. **<追加>**をクリックして、空の連結をプロジェクトに追加します。

# Menu for ASP.NET Web Forms

- c. 連結のプロパティを次のように設定します。
    - **DataMember** プロパティを MenuItem に設定します。
    - **TextField** プロパティを Text に設定します。
  - d. <OK>をクリックして、**バインディングコレクションエディタ**を閉じます。
7. [F5]を押して、プロジェクトを実行します。

✔ このトピックは、次のことを示します。

プロジェクトの実行中、メニューをクリックして、**Menu\_Hierarchy.xml** ファイルのデータが **C1Menu** コントロールに反映されていることを確認します。



## XML としての C1Menu の保存とロード

以下のタスクでは、**C1Menu** コントロールを .xml ファイルとして保存した後、デザイナを使用してプロジェクトにロードする方法を示します。

### XML としての C1Menu の保存

デザイナを使用してツリーを XML ファイルとして保存するには、以下の手順を実行します。

1. **C1Menu** のスマートタグをクリックして、[メニューの編集]を選択して、**C1Menu** デザイナフォームを開きます。
2. [ファイル]→[XML として保存]に移動します。
3. **C1Menu** に .xml ファイル名を付けて、保存先を指定します。
4. <OK>をクリックして、[**C1Menu** デザイナフォーム]ダイアログボックスを閉じます。

### プロジェクトへの既存 XML C1Menu のロード

.xml ファイルとして保存した **C1Menu** コントロールをプロジェクトにロードするには、以下の手順を実行します。

1. **C1Menu** のスマートタグをクリックして、[メニューの編集]を選択して、**C1Menu** デザイナフォームを開きます。
2. **C1Menu** デザイナフォームを開きます。
3. [ファイル]→[XML からロード]の順に移動して、クリックオープンで既存の .xml ファイルを開きます。

### コードによる既存の XML C1Menu のロード

.xml ファイルとして保存した **C1Menu** コントロールをプロジェクトにロードするには、以下の手順を実行します。

1. **C1Menu** 構造の XML ファイルを作成します。
2. 次のようにファイルパスを指定して **LoadLayout** メソッドを呼び出し、項目をロードします。

#### Visual Basic コードの書き方

```
Visual Basic
```

```
C1Menu1.LoadLayout("c:\\Visual Studio 2010\\WebSites\\LoadLayoutEX\\App_Data\\C1MenuControl.xml")
```

#### C# コードの書き方

```
C#
```

```
C1Menu1.LoadLayout("c:\\Visual Studio 2010\\WebSites\\LoadLayoutEX\\App_Data\\C1MenuControl.xml");
```

3. [F5]を押して、プログラムを実行します。