

TreeView for ASP.NET Web Forms

2018.04.25 更新

グレースィティ株式会社

目次

製品の概要	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: デザイナを使用したツリービューの作成	5-6
手順 3: アプリケーションの実行	6-7
デザイン時のサポート	8
スマートタグ	8-9
NodeBinding コレクションエディタ	9-10
デザイナフォーム	10
デザイナフォームの機能	10-12
デザイナフォームのメニュー	12-13
デザイナフォームのツールバー	13
デザイナの使用方法	13
C1TreeNode の削除	13-14
デザイナでのツリービューノードの名前変更	14
子ノードの追加	14
ノードの挿入	14-15
ツリービューの構造と要素	16-17
ツリービューの作成	18
静的なツリービューの作成	18-19
動的なツリービューの作成	19-21
C1TreeView の外観	22
テーマ	22-24
CSS セレクタ	24
チェックボックス	24-26
ノードのドラッグ & ドロップ	26
ロードオンデマンド	26
ノード選択	26-27
ノードのナビゲーション	27

タスク別ヘルプ	28
チェックボックスノードの作成および構成	28
ノードチェックボックスを作成する	28
子ノードの自動チェックを制御する	28-29
テーマの利用	29-30
組み込みテーマの使用	30-31
CSS セレクタの使い方	31
最上位のノードを追加する	31-32
C1TreeView のノードに子ノードを追加する	32-34
ButtonClick イベントによる子ノードの追加	34-35
C1TreeView のサイトマップへの連結	35-37
C1TreeView を XML に連結する	37-39
C1TreeView の動的作成	39-41
ツリーを XML として保存する	41-42
AutoCollapse プロパティを設定する	42-43
C1TreeView をホバーで展開する	43-44
C1TreeView ノードアイコンを設定する	44-46
ドラッグアンドドロップ動作を有効にする	46
ツリー構造内のドラッグアンドドロップ動作	46-47
ツリー同士の間でのドラッグアンドドロップ動作	47-51

製品の概要

TreeView for ASP.NET Web Forms は階層型のツリー構成で項目を表示します。**C1TreeView**コントロールは、展開/縮小のアニメーション、スタイリッシュなテーマ、そして、最も人気のあるドラッグアンドドロップ機能をサポートしています。

このセクションでは

このセクションは、クイックスタートガイド、デザイン時のサポート、およびタスク別ヘルプを含むユーザーガイドです。

主な特長

クイックスタート

デザイン時のサポート

タスク別ヘルプ

他のセクションでは

APIリファレンスセクションには、`C1.Web.Wijmo.Controls.C1TreeView`のメンバーのすべてのクラス、プロパティ、メソッド、およびイベントのリストが含まれています。

C1TreeView Class

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

C1TreeView の主な特長は次のとおりです。

- **データ連結のサポート**

C1TreeView コントロールをデータソースに連結します。XMLDataSource または SiteMapDataSource に連結したり、AccessDataSource からデータを読み取って **C1TreeView** の階層構造を動的に作成したりできます。

- **ノードのドラッグ&ドロップ**

C1TreeViewNodes ノード上、ノード間、またはツリー間でドラッグ & ドロップできます。灰色の垂直線などのような視覚的合図は、**C1TreeNode** がドロップされる位置を示します。

- **チェックボックスのサポート**

各ノード項目は、ON/OFF 可能な通常のチェックボックスとして実装できます。エンドユーザーが、対応するノードを選択または選択解除するためにチェックボックスを ON/OFF にすることができます。**C1TreeView** のチェックボックスが有効になっていると、チェックボックスのステータスがポスト間で変化したときのアクションを作成できます。

- **アニメーション効果**

C1TreeView には、C1TreeView のノードの展開/縮小に適用できるアニメーション効果が組み込まれています。通常、ツリービュー項目を縮小するときはスクロールイン、フェードイン、フォールドイン、クローズ、またはドロップインのいずれかのアニメーションが使用され、ツリービュー項目を展開するときはスクロールアウト、フェードアウト、フォールドアウト、オープン、またはドロップアウトのいずれかのアニメーションが使用されます。展開/縮小効果ごとに、アニメーション効果およびアニメーションの持続時間も指定できます。

- **テンプレートのサポート**

組み込みテンプレートの編集を利用して、**C1TreeView** コントロールの表示を変更します。テンプレートを利用すれば、テキスト、画像、コントロール(ボタンなど)といった独自の要素を、さまざまなノードに手軽に追加できます。

- **キーボードのサポート**

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り **C1TreeView** コントロールをフォーカスします。実行時に、ユーザーはキーボードの矢印キーを使用してノード項目間を移動できます。

- **テーマ**

ツリービューのスマートタグをクリックし、TreeView のあらかじめ定義された6種類のテーマの1つを選択することでカレンダーの外観を変更できます。Arctic、Midnight、Aristo、Rocket、Cobalt および Sterling から選択します。任意に、カスタマイズテーマを作成するためにjQuery UIからのThemeRollerを使用してください。

- **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。ツリービューコントロールを組織の基準に合致させます。

クイックスタート

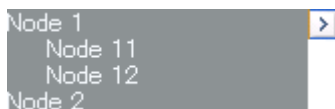
手順 1: ページへのコントロールの追加

このレッスンでは、新しい ASP.NET Web サイトを作成し、**C1TreeView** 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、ASP.NET Web サイトを新に作成します。
2. デザインビューで、Visual Studio ツールボックスに移動し、**C1TreeView** アイコンをダブルクリックして **C1TreeView** icon コントロールをページに追加します。

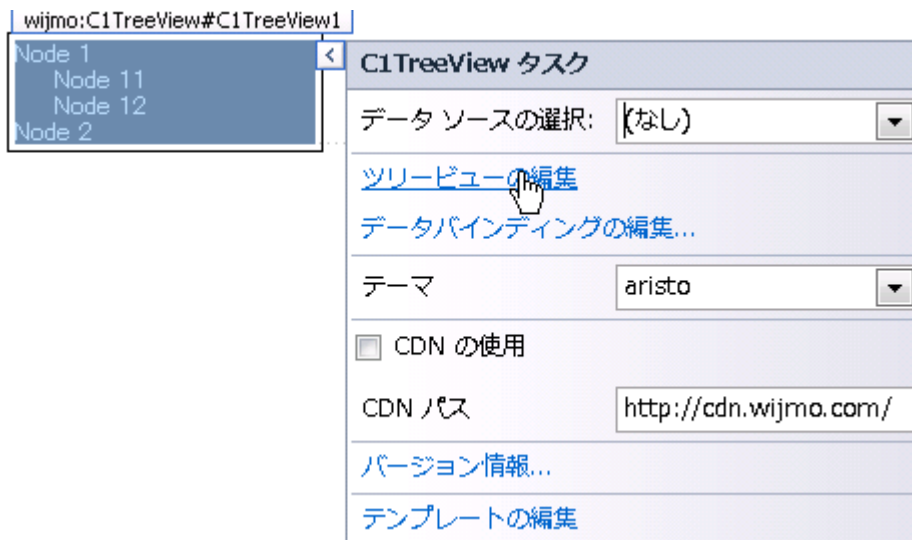
✔ このピックの作業結果



手順 2: デザイナを使用したツリービューの作成

このレッスンでは、ルートノードと子ノードを作成し、テーマを適用し、またノードの隣にチェックボックスを表示する方法を示します。

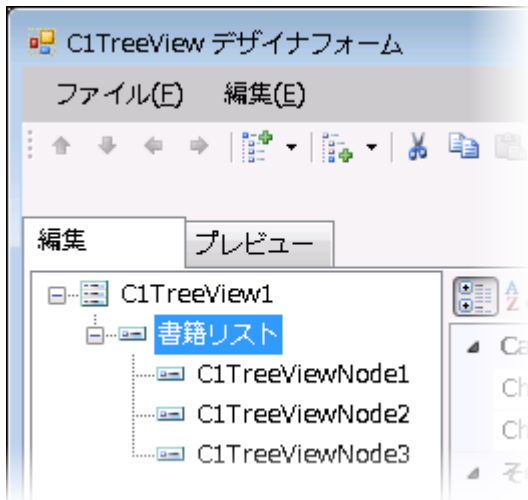
1. **C1TreeView** コントロールを選択し、スマートタグをクリックしてその[**タスク**]メニューを開きます。
2. [**C1TreeView タスク**]メニューから[**ツリービューの編集**]を選択してデザイナを開きます。



3. **C1TreeView** 項目を右クリックして[**子の追加**]→[**C1TreeView ノード**]を選択し、**C1TreeView** コントロールにルートを追加します。C1TreeViewNode1 の **Text** プロパティを「書籍リスト」に設定します。



4. 書籍リストを右クリックし、[子の追加]を選択してルートノードの子を作成します。これをあと2回繰り返します。3つの子ノードが書籍リストの下に追加されます。



5. 書籍リストの下の方の1つ目のノードを選択し、その **Text** プロパティを「言語関連の書籍」に設定します。
6. 書籍リストの下の方の2つ目のノードを選択し、その **Text** プロパティを「セキュリティ関連の書籍」に設定します。
7. 書籍リストの下の方の3つ目のノードを選択し、その **Text** プロパティを「古典書籍」に設定します。
8. 古典書籍ノードを右クリックし、[子の追加]を選択して 古典書籍ノードの子を作成します。これを繰り返して、古典書籍ノードの下に2つの **C1TreeViewNode** を作成します。
9. 古典書籍ノードの下の方の1つ目のノードを選択し、その **Text** プロパティを「The Great Gatsby」に設定します。
10. 古典書籍ノードの下の方の2つ目のノードを選択し、その **Text** プロパティ「Catch-22」に設定します。
11. 書籍リストを右クリックし、[子の追加]を選択して子ノードを追加します。

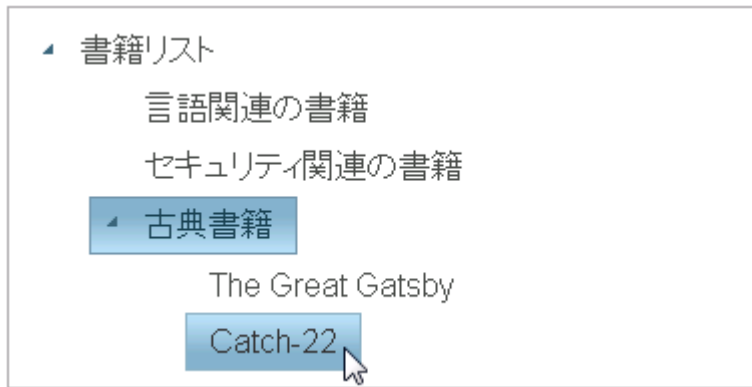
この手順では、**C1TreeView** コントロールにいくつかのノードと子ノードを追加しました。次の手順では、アプリケーションを実行して、このクイックスタートの結果を表示します。

手順 3: アプリケーションの実行

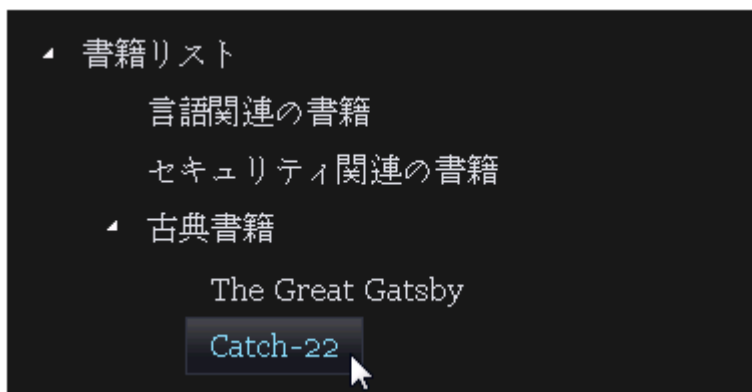
この手順では、アプリケーションを実行して、このクイックスタートの結果を表示します。

1. プロジェクトを保存して実行し、次のことを確認します。
 - **Expanded** プロパティがデフォルトで **False** に設定されているため、ルートノード書籍リストは展開されません。
 - 書籍リストを展開し、作成した子ノードを確認します。
 - テーマがデフォルトのテーマ (Aristo) であることを確認します。

TreeView for ASP.NET Web Forms



2. デザインビューでプロジェクトに戻り、**C1TreeView** のスマートタグをクリックしてその[**タスク**]リストを開きます。[**テーマ**]のドロップダウンの矢印をクリックして、ドロップダウンリストから **Midnight** を選択します。
3. プロジェクトを保存して実行し、テーマMidnight が**C1TreeView** に適用されていることを確認します。



4. デザインビューでプロジェクトに戻り、**C1TreeView デザイナフォーム**を開きます。
5. **C1TreeView** 項目を選択し、プロパティウィンドウでその動作プロパティを次のように設定します。
 - **AllowDrag** は **True**
 - **AllowDrop** は **True**.
 - **ShowCheckBoxes** は **True**
6. **<OK>**をクリックし、デザイナを保存して閉じます。
7. プロジェクトを実行し、次のことを確認します。
 - 書籍リストノードを展開し、各ツリービューノードの隣にあるチェックボックスを確認します。
 - ツリービューノードのいずれかを選択し、それを別の位置にドラッグします。

デザイン時のサポート

スマートタグ

C1TreeView 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。スマートタグとは、**C1TreeView** で最もよく使用されるプロパティを提供するショートカットタスクメニューです。

C1TreeView 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、**C1TreeView デザイナフォーム** 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。

[**C1TreeView タスク**]メニューにアクセスするには、**C1TreeView** 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、[**C1TreeView タスク**]メニューが開きます。

[**C1TreeView 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。

- **ツリービューの編集**

項目 [**ツリービューの編集**] 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 **C1TreeView** の要素を手軽に設定できる **C1TreeView デザイナフォーム** が開きます。このデザイナーでは、**C1TreeNode** の追加、削除、並べ替えや、その外観、動作などを定義するさまざまなプロパティの設定ができます。**C1TreeView デザイナフォーム** の詳細については、「[C1TreeView デザイナフォーム](#)」を参照してください。

- **データバインディングの編集**

項目 [**データバインディングの編集**] をクリックすると、連結の追加や削除、およびプロパティの編集ができる [**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。

- **テーマ**

[**テーマ**] ドロップダウン矢印をクリックすると、組み込みの各種視テーマの中からいずれかを選択できます。詳細については、「[テーマ](#)」を参照してください。

- **新しいテーマの作成**

TreeView for ASP.NET Web Forms

[**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。

- **Use CDN**

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 Path**

CDN の URL パスを表示します。

- **Bootstrap の使用**

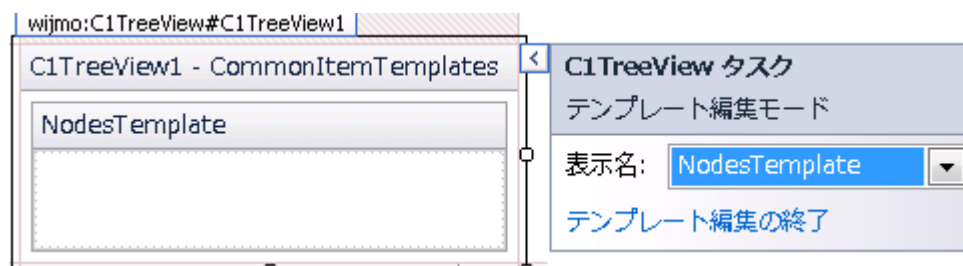
[**Bootstrap の使用**]オプションを選択すると、コントロールに Bootstrap テーマを適用することができます。アプリケーションで Bootstrap テーマを使用する方法については、「[Bootstrap for ASP.NET Web Forms クイックスタート](#)」を参照してください。

- **バージョン情報**

[**バージョン情報**]をクリックすると、製品のバージョン情報を確認できるダイアログボックスが表示されます。

- **テンプレートの編集**

項目 [**テンプレートの編集**]をクリックすると、**C1TreeView** 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、異なったオプションの [**C1TreeView タスク**]メニューが表示されます。

- **表示名**

[**表示名**]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エリア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 [**C1TreeView タスク**]メニューに戻ります。

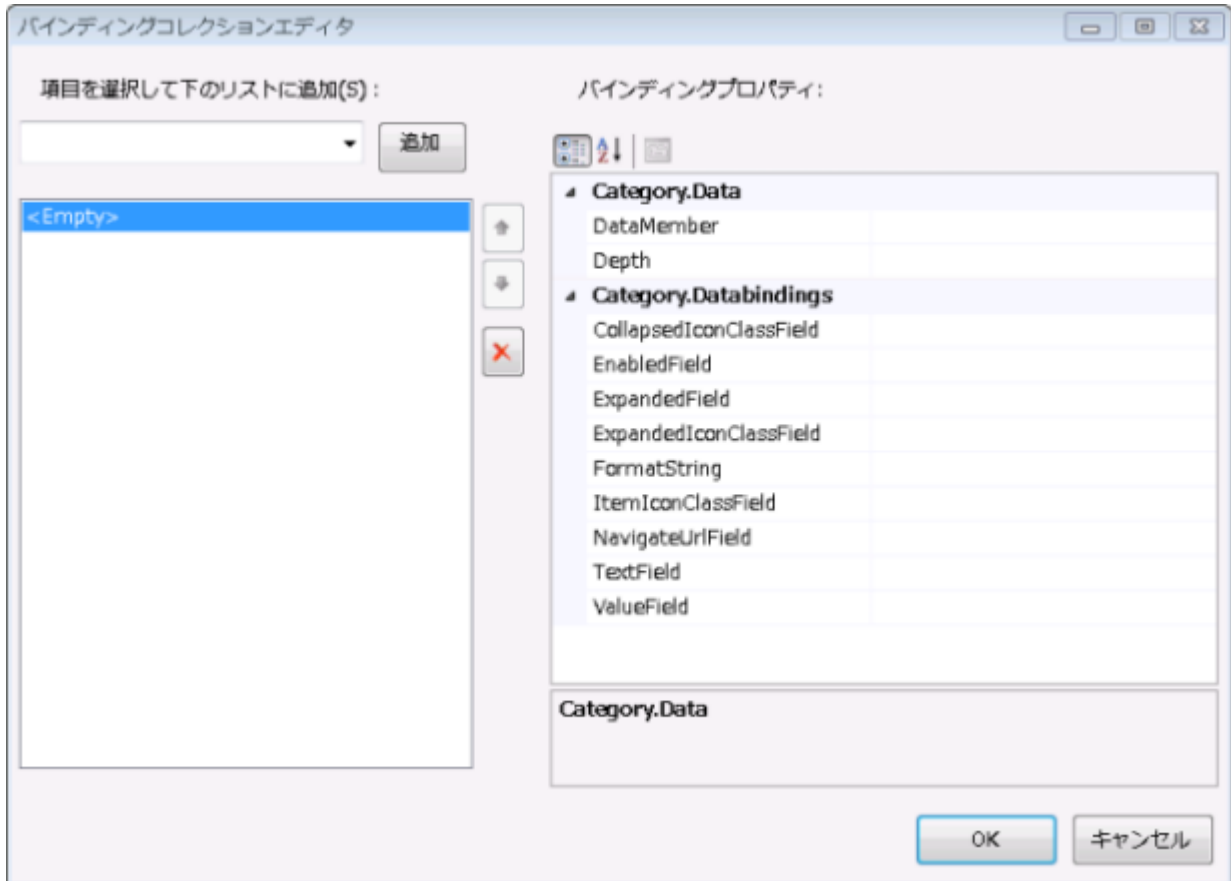
NodeBinding コレクションエディタ

C1TreeView コントロールには、**C1TreeViewNodeCollection**でのデータ連結の追加や削除、および連結プロパティの指定ができるコレクションエディタが含まれています。

C1TreeViewNodeBinding コレクションエディタにアクセスする方法は2つあります。

[C1TreeView タスク]メニューから

1. **C1TreeView** 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、[**C1TreeView タスク**]メニューを開きます。
2. [**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**が表示されます。



C1TreeView デザイナフォーム から

1. **C1TreeView** 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、[**C1TreeView タスク**]メニューを開きます。
2. [**ツリービューの編集**]を選択します。**C1TreeView デザイナフォーム**が表示されます。
3. **C1TreeView** コントロールが選択された状態で、**DataBindings** プロパティの隣にある<...>ボタンをクリックします。**C1TreeViewNodeBinding** コレクションエディタが表示されます。このダイアログボックスは、外観はやや異なりますが、基本的には**バインディングコレクションエディタ**と同じものであり、同じプロパティを含んでいます。

デザイナーフォーム

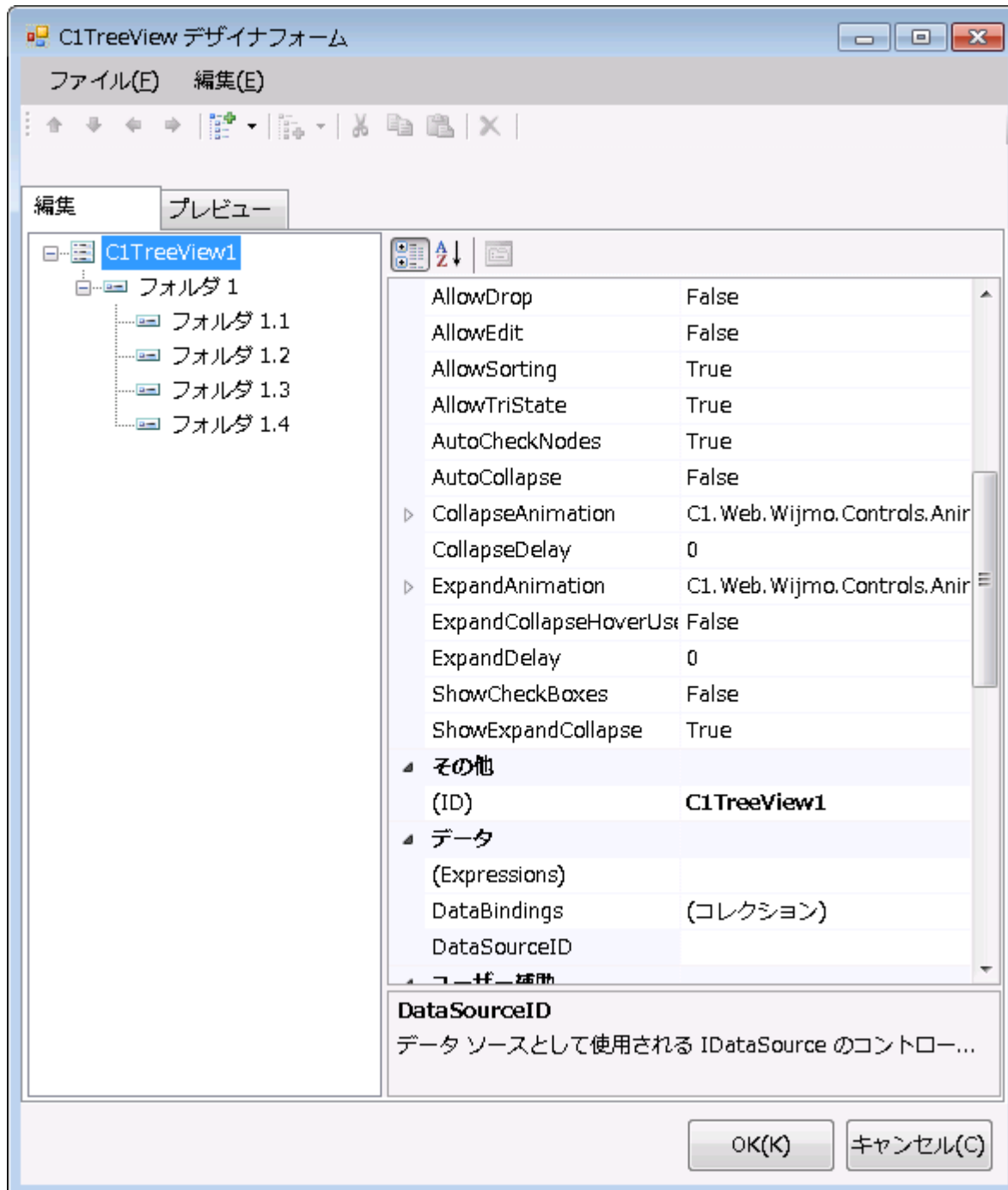
C1TreeView デザイナフォームは、**C1TreeView** のプロパティや **C1TreeViewNode** のプロパティを編集できるデザイナーです。**C1TreeView デザイナフォーム**は、プログラマがコントロールを視覚的に修正できる点でプロパティウィンドウに似ています。しかし、このデザイナーでは、**C1TreeViewNode** の選択、そのプロパティの設定、ノードの操作、さらに **C1TreeView** コントロールの外観のプレビューまで、すべてをフォーム内で行うことができます。

このトピックでは、**C1TreeView デザイナフォーム**のデザインインターフェイスに習熟することにより、デザイナー内でコマンドを使用し、最小限の労力と時間で**C1TreeView** を編集できるようになります。

C1TreeView デザイナフォームを開くには、**C1TreeView** のスマートタグをクリックし、[**C1TreeView タスク**]メニューから[**ツリービューの編集**]リンクを選択します。

デザイナフォームの機能

C1TreeView デザイナフォームには、メニュー、ツールバー、「編集」タブ、「プレビュー」タブ、およびプロパティペインが含まれています。



- 「編集」タブ

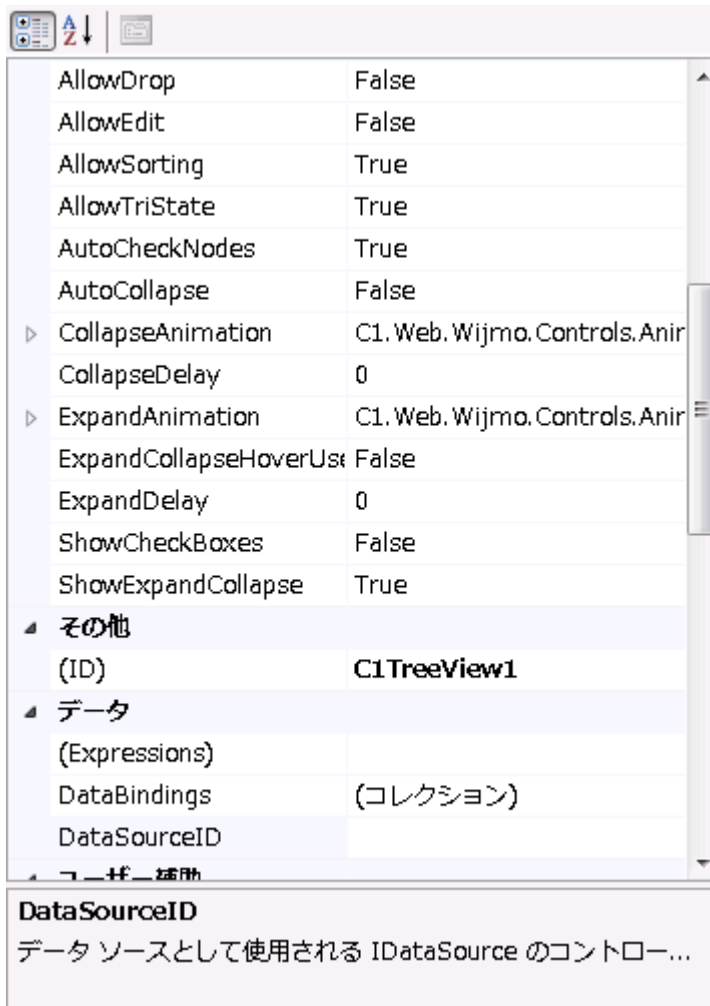
「編集」タブをクリックし、プロパティの操作や調節を行う **C1TreeView** コントロールまたは目的の **C1TreeNode** を選択します。

- 「プレビュー」タブ

C1TreeView コントロールの表示をプレビューする場合は、「プレビュー」タブをクリックします。

● プロパティペイン

C1TreeView デザイナフォームのプロパティペインは、Visual Studio のプロパティウィンドウとほぼ同じです。**C1TreeNode** または **C1TreeView** コントロールを選択し、目的のプロパティを設定するだけです。



● コマンドボタン

次の表は、コマンドボタンの一覧です。

ボタン	説明
OK	<OK>をクリックすると、新しい設定値が C1TreeView コントロールに適用されます。
キャンセル	<キャンセル>をクリックすると、 C1TreeView デザイナフォーム が閉じて新しい設定値は破棄され、デフォルト設定値が C1TreeView コントロールに適用されます。

デザイナフォームのメニュー

C1TreeView デザイナフォームのツールバーは、次のように表示されます。

メニュー項目	サブメニュー項目	説明
ファイル	XML からロード	C1TreeView コントロールのフォーマットを .xml ファイルからロードします。
	XML として保存	C1TreeView コントロールの現在のフォーマットを .xml ファイルに保存します。

TreeView for ASP.NET Web Forms

	終了	C1TreeView デザイナフォーム を閉じます。
編集	項目の挿入	ノードリスト内の指定された場所に新しい C1TreeViewNode を挿入します。
	子の追加	C1TreeView または別の C1TreeViewNode の子として、新しい C1TreeViewNode を追加します。
	切り取り	ノードリスト内で移動するために、選択された C1TreeViewNode を切り取ります。
	コピー	選択された C1TreeViewNode をコピーします。
	貼り付け	ノードリスト内の指定された場所に C1TreeViewNode を貼り付けます。
	削除	選択された C1TreeViewNode を削除します。
	名前の変更	C1TreeViewNode の名前を変更できます。

デザイナフォームのツールバー

C1TreeView デザイナフォームのツールバーは、次のように表示されます。



下の表で、ツールバーの各ボタンについて説明します。

ボタン	名前	説明
	項目を上へ移動	ノードリスト内で、選択された C1TreeViewNode を上へ移動します。
	項目を下へ移動	ノードリスト内で、選択された C1TreeViewNode を下へ移動します。
	項目を左へ移動	階層内で、選択された C1TreeViewNode を左へ移動します。
	項目を右へ移動	階層内で、選択された C1TreeViewNode を右へ移動します。
	子項目の追加	C1TreeView コントロールまたは別の C1TreeViewNode の子として、 C1TreeViewNode を挿入します。
	項目の挿入	ノードリスト内の指定された場所に C1TreeViewNode を挿入します。
	切り取り	ノードリスト内で移動するために、選択された C1TreeViewNode を切り取ります。
	コピー	選択された C1TreeViewNode をコピーします。
	貼り付け	ノードリスト内の指定された場所に C1TreeViewNode を貼り付けます。
	削除	選択された C1TreeViewNode を削除します。

デザイナの使用方法

以下のトピックでは、**C1TreeView デザイナフォーム**を使用した各種の作業について説明します。

C1TreeNode の削除

デザイナーで **C1TreeNode** を削除するときは、次の3つの方法のいずれかを使用できます。

- ショートカットメニューによる子ノードの削除

削除する **C1TreeNode** を右クリックし、[削除]を選択します。

- <削除>ボタン押下による子ノードの削除

削除するノードを選択し、<削除>をクリックします。

- 「編集」メニューによる子ノードの削除

[編集]メニューをクリックし、<削除>を選択します。

デザイナーでのツリービューノードの名前変更

デザイナーで **C1TreeNode** の名前を変更するときは、次の3つの方法のいずれかを使用できます。

- [F2]の押下

- 名前を変更する **C1TreeNode** を選択します。
- [F2]キーを押し、新しい名前を入力します。

- ショートカットメニューからの[名前の変更]の選択

- 名前を変更する **C1TreeNode** を右クリックします。
- コンテキストメニューから[名前の変更]を選択し、新しい名前を入力します。

- [編集]メニューからの[名前の変更]の選択

- 名前を変更する **C1TreeNode** を選択します。
- [編集]メニューをクリックして[名前の変更]を選択し、新しい名前を入力します。

子ノードの追加

デザイナーで子**C1TreeNode**を追加するときは、次の3つの方法のいずれかを使用できます。

- ショートカットメニューによる子ノードの追加

子ノードを追加するノードを右クリックし、[子の追加]→[C1TreeView ノード]を選択します。

- <子の追加>ボタン押下による子ノードの追加

子ノードを追加するノードを選択し、<子の追加>ボタンのドロップダウン矢印をクリックして[C1TreeView ノード]を選択します。

- [編集]メニューによる子ノードの追加

子ノードを追加するノードを選択し、[編集]メニューをクリックして[子の追加]、[C1TreeView ノード]を選択します。

ノードの挿入

デザイナーで **C1TreeNode** を挿入するときは、次の3つの方法のいずれかを使用できます。

- ショートカットメニューによるノードの挿入

子ノードを追加する **C1TreeNode** を右クリックし、[子の挿入]→[C1TreeView ノード]を選択します。

TreeView for ASP.NET Web Forms

- 〈ノードの挿入〉ボタン押下によるノードの挿入

子ノードを追加する **C1TreeNode** を選択し、〈**項目の挿入**〉ボタンのドロップダウン矢印をクリックして [**C1TreeView ノード**] を選択します。

- [編集]メニューを使用したノードの挿入

子ノードを追加する **C1TreeNode** を選択し、[編集]メニューをクリックして[**項目の挿入**]→[**C1TreeView ノード**]を選択します。

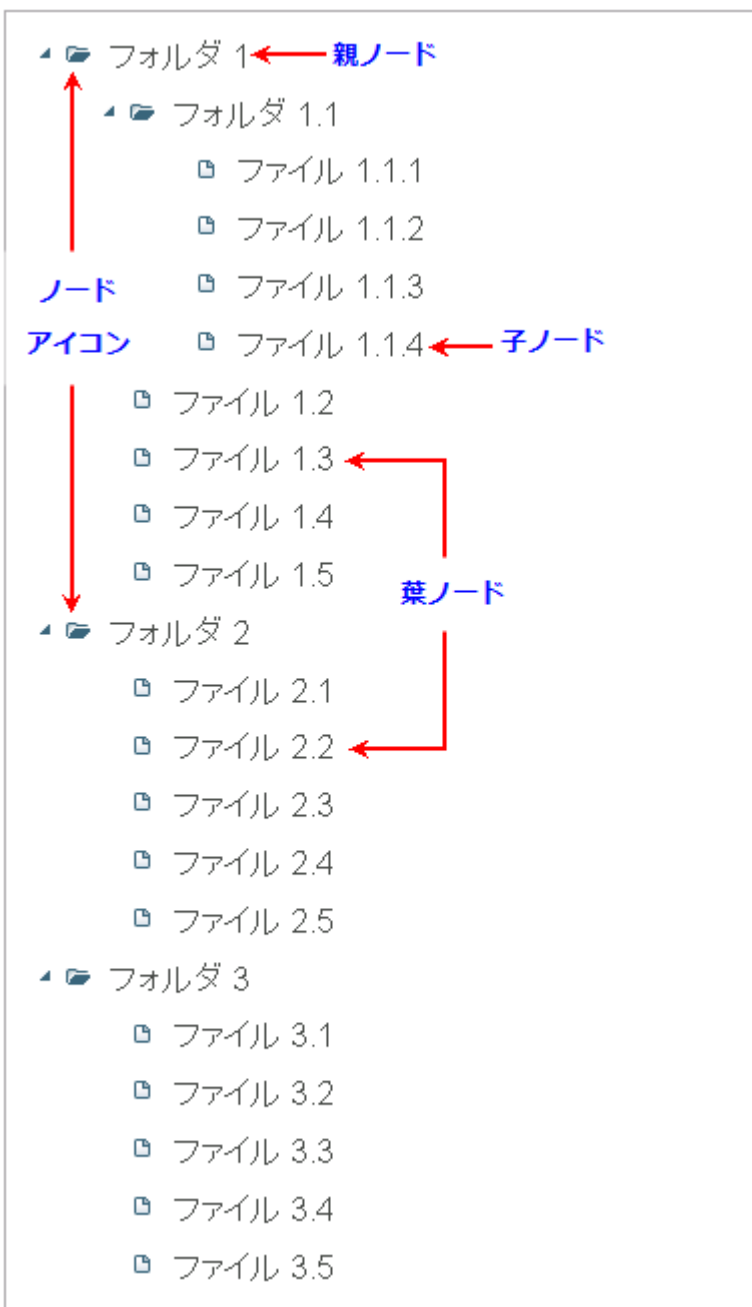
ツリービューの構造と要素

C1TreeViewは、階層型のツリー構造を表示するツリータイプの Web コントロールです。ツリー構造は、階層型のデータをグラフィカルな形式で表現するために使用されます。

ツリーには1つ以上の要素が含まれ、各要素がノードとなります。ノードには、親ノード、子ノード、葉ノードがあります。ノードの各タイプの説明は、次のとおりです。

- 親ノードは、他のノードを含むノードです。
- 子ノードは、別のノードに含まれるノードです。
- 葉ノードは、子ノードを含まないノードです。

典型的なツリー構造と同様、**C1TreeView**には、親ノード、子ノード、および葉ノードから成る1つ以上のノードが含まれます。親ノード、子ノード、および葉ノードは、**C1TreeNode**と呼ばれます。次の図は、C1TreeView コントロールのノードと構造を示します。



C1TreeViewは、ルートが頂点になるように逆にした木のように表示されます。**C1TreeView**には、1つの以上のルートノード

TreeView for ASP.NET Web Forms

を含めることができます。子ノードが含まれる場合、ノードは縮小／展開できます。**ShowExpandCollapse** プロパティを true に設定すれば、ノードを展開／縮小できます。各ノードは、テキストとそれに関連付けられた画像を設定でき、C1TreeView オブジェクトと **C1TreeNode** オブジェクトのプロパティ設定値に応じて、チェックボックスの編集、選択、または表示が可能です。ツリーノードはハイパーリンクとして表示でき、URL をそれに関連付けることができます。

ツリービューの作成

C1TreeNode は、次の方法のいずれかを使用することにより、ページまたはユーザーコントロール上で定義できます。:

- 宣言構文を使用した 静的な作成
- **C1TreeNode** 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
- C1TreeView 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

静的なツリービューの作成

ツリー内の各ノードは、ツリーノードの Text プロパティと Value プロパティでそれぞれ定義される名前と値のペアで表されます。ノードのテキストは表示されますが、ノードの値は表示されず、ポストバックイベントを処理するための追加的なデータとして使用されるのが普通です。

静的なメニューは、ツリービュー構造を作成する最も単純な方法です。

C1TreeView デザインフォーム を使用してツリービ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ノードを指定できます。

デザイナを使用して静的な **C1TreeNode** を表示するには、**C1TreeView デザインフォーム**を開いて**C1TreeNode** を親に追加します。各 **C1TreeNode** のプロパティは、デザイナで直接変更できます。メニューデザイナの詳細については、「[C1TreeView デザインフォーム](#)」を参照してください。

宣言構文を使用して静的な **C1TreeNode** を表示するには、まず、**C1TreeView** コントロールの開始タグと終了タグの間に <Nodes> の開始タグと終了タグをネストします。次に、<Nodes> の開始タグと終了タグの間に <asp:C1TreeNode> 要素をネストすることによって、ツリービュー構造を作成します。各 <asp:C1TreeNode> 要素は、コントロール内のノードを表し、**C1TreeNode** オブジェクトに対応しています。

宣言構文を使用すれば、ページ上で**C1TreeNode** をインラインで定義できます。

たとえば、次のようになります。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" runat="server" AllowSorting="False" AutoCollapse="False"
VisualStyle="Default" VisualStylePath="~/C1WebControls/C1TreeView/VisualStyles">
  <Nodes>
    <cc1:C1TreeNode runat="server" Expanded="False" Text="C1TreeNode">
      <Nodes>
        <cc1:C1TreeNode runat="server" Expanded="False" Text="C1TreeNode">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode runat="server" Expanded="False" Text="C1TreeNode">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode runat="server" Expanded="False" Text="C1TreeNode">
          <Nodes>
            <cc1:C1TreeNode runat="server" Expanded="False" Text="C1TreeNode">
            </cc1:C1TreeNode>
            <cc1:C1TreeNode runat="server" Expanded="False" Text="C1TreeNode">
            </cc1:C1TreeNode>
            <cc1:C1TreeNode runat="server" Expanded="False" Text="C1TreeNode">
            </cc1:C1TreeNode>
          </Nodes>
        </cc1:C1TreeNode>
      </Nodes>
    </cc1:C1TreeNode>
  </Nodes>
</cc1:C1TreeView>
```

```
</Nodes>  
</cc1:C1TreeView>
```

動的なツリービューの作成

動的なツリービューは、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ツリービューを作成するときは、**C1TreeViewNode** 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、**C1TreeView** 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**C1TreeView** コントロールを作成します。

ソースビュー

```
var aTreeView = C1.Web.C1TreeView.createInstance ();  
document.body.appendChild(aTreeView.element);
```

C1TreeView または **C1TreeViewNode** のコンストラクタを使用すれば、**C1TreeView** クラスまたは **C1TreeViewNode** クラスの新しいインスタンスを作成できます。ノードが作成されたら、それらは新しいノードまたはツリービューの **Node** コレクションに追加できます。

たとえば、次のようになります。

Visual Basic コードの書き方

Visual Basic

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs)  
    'クラスのインスタンスを作成する  
    Dim treeView As New C1TreeView()  
    Placeholder1.Controls.Add(treeView)  
    If Not Page.IsPostBack Then  
        Dim P As New C1TreeViewNode()  
        P.Text = "Products"  
        P.Value = "PS"  
        P.Expanded = True  
        treeView.Nodes.Add(P)  
        Dim Pr1 As New C1TreeViewNode()  
        Pr1.Text = "Product 1"  
        Pr1.Value = "Pr1"  
        Pr1.Expanded = True  
        P.Nodes.Add(Pr1)  
        Dim Oview1 As New C1TreeViewNode()  
        Oview1.Text = "Overview"  
        Oview1.Value = "Oview1"  
        Pr1.Nodes.Add(Oview1)  
        Dim Down1 As New C1TreeViewNode()  
        Down1.Text = "Downloads"  
        Down1.Value = "Down1"  
        Pr1.Nodes.Add(Down1)  
        Dim Supp1 As New C1TreeViewNode()  
        Supp1.Text = "Support"  
        Supp1.Value = "Supp1"  
        Pr1.Nodes.Add(Supp1)  
        Dim Pr2 As New C1TreeViewNode()  
        Pr2.Text = "Products 2"
```

```

Pr2.Value = "Pr2"
Pr2.Expanded = True
P.Nodes.Add(Pr2)
Dim Oview2 As New C1TreeNode()
Oview2.Text = "Overview"
Oview2.Value = "Oview2"
Pr2.Nodes.Add(Oview2)
Dim Down2 As New C1TreeNode()
Down2.Text = "Downloads"
Down2.Value = "Down2"
Pr2.Nodes.Add(Down2)
Dim Supp2 As New C1TreeNode()
Supp2.Text = "Support"
Supp2.Value = "Supp2"
Pr2.Nodes.Add(Supp2)
End If
End Sub

```

C# コードの書き方

```

C#
protected void Page_Load(object sender, EventArgs e)
{
    //クラスのインスタンスを作成する
    C1TreeView treeView = new C1TreeView();
    Placeholder1.Controls.Add(treeView);
    if (!Page.IsPostBack)
    {
        C1TreeNode P = new C1TreeNode();
        P.Text = "Products";
        P.Value = "PS";
        P.Expanded = true;
        treeView.Nodes.Add(P);
        C1TreeNode Pr1 = new C1TreeNode();
        Pr1.Text = "Product 1";
        Pr1.Value = "Pr1";
        Pr1.Expanded = true;
        P.Nodes.Add(Pr1);
        C1TreeNode Oview1 = new C1TreeNode();
        Oview1.Text = "Overview";
        Oview1.Value = "Oview1";
        Pr1.Nodes.Add(Oview1);
        C1TreeNode Down1 = new C1TreeNode();
        Down1.Text = "Downloads";
        Down1.Value = "Down";
        Pr1.Nodes.Add(Down1);
        C1TreeNode Supp1 = new C1TreeNode();
        Supp1.Text = "Support";
        Supp1.Value = "Supp1";
        Pr1.Nodes.Add(Supp1);
        C1TreeNode Pr2 = new C1TreeNode();
        Pr2.Text = "Products 2";
    }
}

```

TreeView for ASP.NET Web Forms

```
Pr2.Value = "Pr2";
Pr2.Expanded = true;
P.Nodes.Add(Pr2);
C1TreeNode Oview2 = new C1TreeNode();
Oview2.Text = "Overview";
Oview2.Value = "Oview2";
Pr2.Nodes.Add(Oview2);
C1TreeNode Down2 = new C1TreeNode();
Down2.Text = "Downloads";
Down2.Value = "Down2";
Pr2.Nodes.Add(Down2);
C1TreeNode Supp2 = new C1TreeNode();
Supp2.Text = "Support";
Supp2.Value = "Supp2";
Pr2.Nodes.Add(Supp2);
    }
}
```

C1TreeView の外観

テーマ

C1TreeView には、6つのテーマが組み込まれています。組み込まれたテーマの一つが選択されている場合、ページ上のすべての ComponentOne for ASP.NET Web Forms コントロールに適用されます。テーマ付きの C1TreeView コントロールは、次のように表示されます。

Arctic	 <pre> ▲ フォルダ 1 ▶ フォルダ 1.1 □ ファイル 1.2 □ ファイル 1.3 □ ファイル 1.4 □ ファイル 1.5 ▶ フォルダ 2 ▶ フォルダ 3 </pre>	
Aristo	 <pre> ▲ フォルダ 1 ▶ フォルダ 1.1 □ ファイル 1.2 □ ファイル 1.3 □ ファイル 1.4 □ ファイル 1.5 ▶ フォルダ 2 ▶ フォルダ 3 </pre>	

TreeView for ASP.NET Web Forms

Cobalt

- フォルダ 1
 - フォルダ 1.1
 - ファイル 1.2
 - ファイル 1.3
 - ファイル 1.4
 - ファイル 1.5
- フォルダ 2
- フォルダ 3

Midnight

- フォルダ 1
 - フォルダ 1.1
 - ファイル 1.2
 - ファイル 1.3
 - ファイル 1.4
 - ファイル 1.5
- フォルダ 2
- フォルダ 3

Rocket

- フォルダ 1
 - フォルダ 1.1
 - ファイル 1.2
 - ファイル 1.3
 - ファイル 1.4
 - ファイル 1.5
- フォルダ 2
- フォルダ 3

Sterling

```

└─ フォルダ 1
    └─ フォルダ 1.1
        └─ ファイル 1.2
        └─ ファイル 1.3
        └─ ファイル 1.4
        └─ ファイル 1.5
    └─ フォルダ 2
    └─ フォルダ 3
    
```

C1TreeView コントロールにテーマを適用するには、**Theme** プロパティを組み込まれたテーマの一つに設定します。

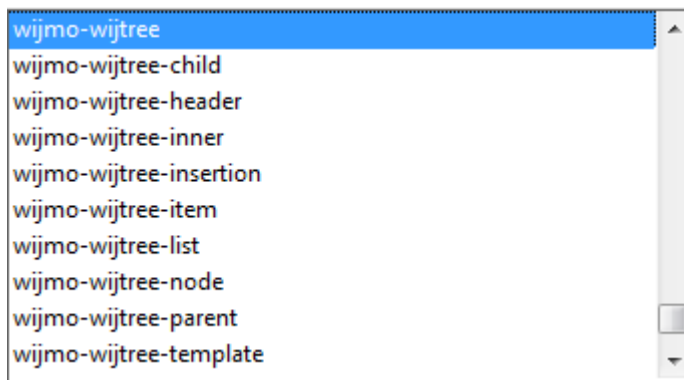
CSS セレクタ

CSS を使用して **C1TreeView** の任意の要素をスタイル設定し、その外観を真に独特のものにすることができます。ツリービュー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、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 セレクタの一覧について、プロジェクト内の **C1TreeView** コントロールを選択し、Visual Studio のプロパティウィンドウの **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。

C1TreeView CSS セレクタは `wijmo-wijtree` から始めます。

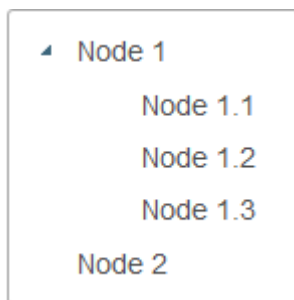


チェックボックス

ShowCheckBoxes が `true` に設定されていると、各 **C1TreeNode** の隣にチェックボックスを表示できます。

C1TreeView のチェックボックスが有効になっていると、**NodeCheckChanged** を使用して、チェックボックスのステータスがポスト間で変化したときのアクションを作成できます。チェックボックスの変化に対し、ポストバックなしでクライアント上すぐに応答するには、サーバー側イベントプロパティ **OnClientNodeCheckChanged** を使用できます。

TreeView for ASP.NET Web Forms



3つのチェック状態

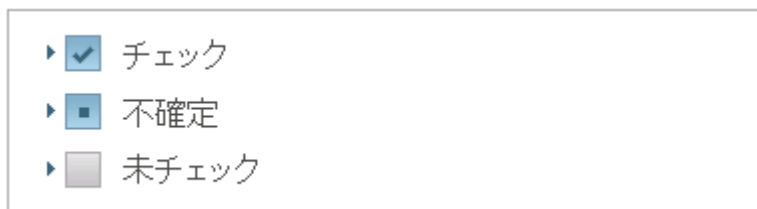
ShowCheckBoxes プロパティと **AllowTriState** プロパティを **true** に設定することで、**C1TreeNode**の横に表示されるチェックボックスで3つのチェック状態を使用することができます。

次の表では、3つのチェック状態を説明すると共に、**C1TreeViewNodes** の横に表示される各チェックボックスに与える影響も説明します。

Checked プロパティは**C1TreeNode**のチェック状態を指定します。

チェックの状態	説明
不確定	いくつかの子ノードだけが選択されている場合、暗い灰色のボックスは親ノードに表示されません。
チェック	すべての子ノードが選択されている場合、親ノードにチェックマークが表示されます。
未チェック	親ノードまたは子ノードのいずれも選択されていない場合、空のチェックボックスが親ノードに表示されます。

次の図は、**C1TreeView** コントロールのためのチェックボックスの各状態(不確定、チェックON、チェックOFF)を示します。



子ノードへの3つのチェック状態の効果

AllowTriStateが **True** に設定されていて、**C1TreeView** のノードが子ノードを含む場合は、その **CheckState** は子の **CheckState** によって決定されます。その3つのケースは以下の通りです。

• ケース 1

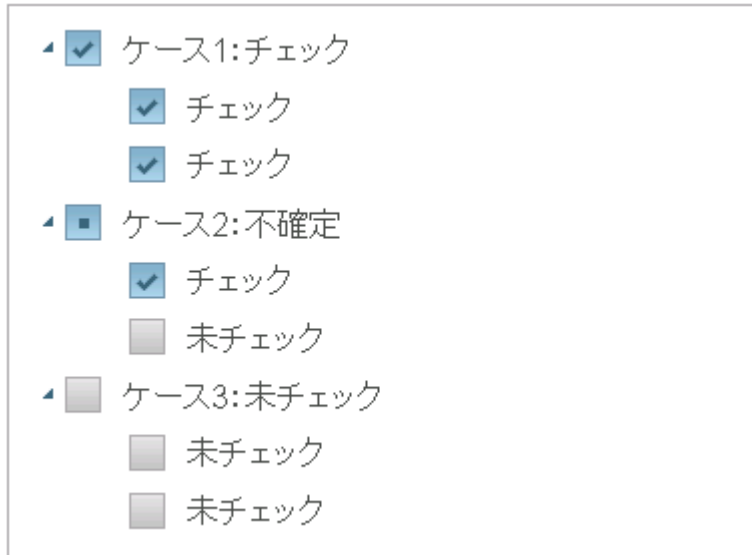
子ノードのすべての **Checked** プロパティが **True**に設定されていると、親ノードの**Checked** プロパティも**True** に設定され、**CheckState** の値は自動的に **Checked** に設定されます。

• ケース 2

いくつかの子ノードの **Checked** プロパティが **True** に設定されていると、親ノードの **Checked** プロパティも**True** に設定され、**CheckState** プロパティの値は **Indeterminate** に設定されます。

• ケース 3

子ノードのすべての **Checked** プロパティが **False** に設定されていると、親ノードの**Checked** プロパティも**False** に設定され、**CheckState** の値は自動的に **Unchecked** に設定されます。

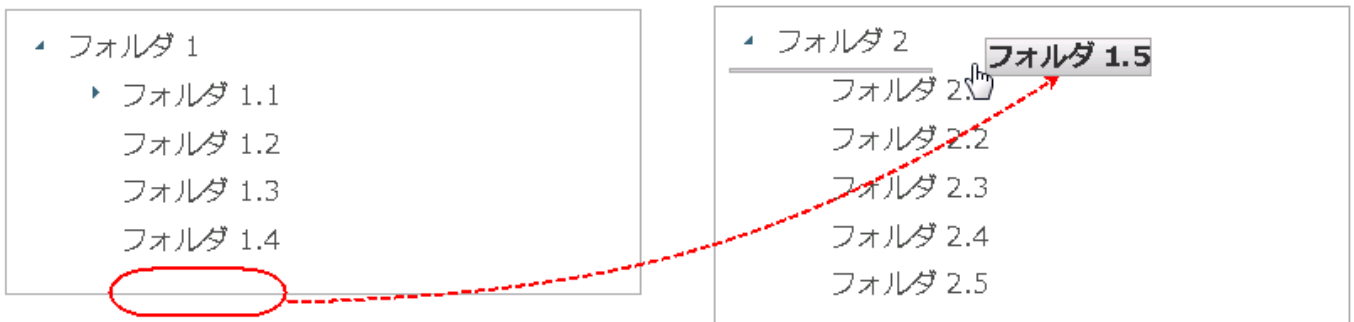


ノードのドラッグ&ドロップ

AllowDrag プロパティと **AllowDrop** プロパティが **true** に設定されていると、**C1TreeNode** はノード上、ノード間、またはツリー間でドラッグ&ドロップできます。**AllowDrag** プロパティと **AllowDrop** プロパティが **C1TreeView** の個別のノードに対しても設定可能です。

注意: **C1TreeView** コントロール全体に対して設定した **AllowDrag** と **AllowDrop** プロパティより個別のノードに設定した **AllowDrag** と **AllowDrop** プロパティが優先されます。

次の図は、ある **C1TreeView** から別の **C1TreeView** に **C1TreeNode** をドラッグする場合を示しています。灰色の垂直線は、**C1TreeNode** がドロップされる位置を示します。



C1TreeNode でドロップされたノードが示されると、サーバー側イベント **NodeDropped** が生成されます。**NodeDropped** のイベントハンドラは、ドロップされたノードを確認して特定のアクションを実行できます。ノードのドラッグ時やドロップ時に、ポストバックなしでクライアント上ですぐに応答するには、**OnClientNodeDragStarted**, **OnClientNodeDragging**, **OnClientNodeDropped** の各サーバー側プロパティイベントを使用できます。

ロードオンデマンド

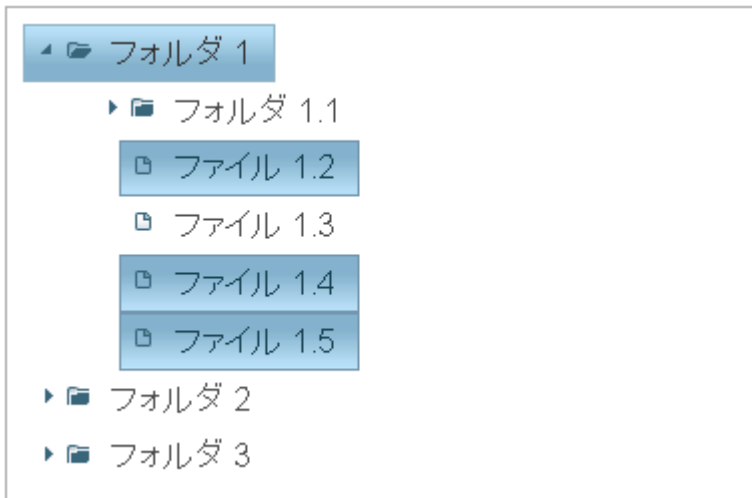
ツリーに多数のノードが含まれており、不可欠な情報のみをサーバーに送信する場合は、**LoadOnDemand** プロパティと **AutoPostBack** プロパティを **True** に設定できます。

ノード選択

実行時にノードをクリックすると、そのノードは自動的に選択状態としてマーキングされます。ノードをクリックする

TreeView for ASP.NET Web Forms

と、**SelectedNodesChanged** イベントが発生し、カスタム機能を提供できます。クリックせずにノードを選択状態にするには、**Selected** プロパティを有効にします。[Ctrl]キーを押しながら複数のノードをクリックしていくと、それら複数のノードを同時に選択できます。ノードを選択解除するには、[Ctrl]キーを押しながらそのノードをもう一度クリックします。次の **C1TreeView** では、複数のノードが選択状態になっています。



ノードのナビゲーション

C1TreeView では、マウスとキーボードによるナビゲーションがサポートされています。

マウスを使用した **C1TreeNode** のナビゲーション

次の表で、**C1TreeNode**間を移動するときのアクションと、対応するマウスコマンドを説明します。

アクション	マウスコマンド
ノードの展開	ノードの名前の左側にあるプラス記号をクリックします。
ノードの縮小	ノードの名前の左側にあるマイナス記号をクリックします。
ノードの選択	ノードの名前をクリックします。

キーボードを使用した **C1TreeNode** のナビゲーション

次の表で、**C1TreeNode**間を移動するときに使用するアクションと、それらに関連付けられたキーを説明します。

アクション	キーボードコマンド
ノードの展開	[+]キー
ノードの縮小	[-]キー
上のノードを選択	上矢印キー
下のノードを選択	下矢印キー
複数ノードの選択	[Ctrl]キー + マウス
フォーカスを次のコントロールに移動	[TAB]キー

タスク別ヘルプ

タスク別ヘルプのセクションは、Visual Studio ASP.NET 環境でのプログラミングに精通し、C1TreeView 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、C1TreeView コントロールを使用した特定のタスクのソリューションを示します。タスク別ヘルプの各トピックでは、新しい ASP.NET プロジェクトを既に作成していることを前提としています。

チェックボックスノードの作成および構成

このセクションでは、**C1TreeNode** 用のチェックボックスの作成と構成を支援するいくつかのタスクが含まれています。チェックボックスの詳細については、「[チェックボックス](#)」を参照してください。

ノードチェックボックスを作成する

チェックボックスノードを持つ **C1TreeView** を作成する場合は、単純に **C1TreeView** の **ShowCheckBoxes** プロパティを True に設定すると、コントロール内のすべてのノードにチェックボックスが追加されます。

デザインビューの場合

次の手順を実行します。

1. スマートタグをクリックして、**C1TreeView** タスクメニューを開きます。[TreeViewの編集]を選択します。**C1TreeView デザインフォーム** ダイアログボックスが開きます。
2. ツリービューのデザイナーでは、C1TreeView コントロール(デフォルトでは**C1TreeView1**)を選択します。
3. プロパティグリッドに移動し、**ShowCheckBoxes** プロパティを True に設定します。

ソースビューの場合

<cc1:C1TreeView> タグに ShowCheckBoxes = True を追加して、マークアップを次のように変更します。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" runat="server" ShowCheckBoxes="True">
```

コードの場合

Page_Load イベントに下記のサンプルコードを追加します。

Visual Basic コードの書き方

Visual Basic

```
C1TreeView1.ShowCheckBoxes="True"
```

C# コードの書き方

C#

```
C1TreeView1.ShowCheckBoxes="True";
```

子ノードの自動チェックを制御する

C1TreeView でチェックボックスが利用されている場合、親ノードをクリックすると、そのノードのすべての子ノードが自動的に

TreeView for ASP.NET Web Forms

チェックされます。この動作を防ぐには、**C1TreeView** の **AutoCheckNodes** プロパティを **False** に設定することができます。

このチュートリアルでは、1つの親ノードと2つの子ノードを持つ C1TreeView コントロールを作成します。ノードをチェックボックスにし、**AutoCheckNodes** プロパティを **False** に設定します。そうすると、子ノードの **Checked** プロパティ設定がその親ノードの **Checked** プロパティに依存しなくなります。

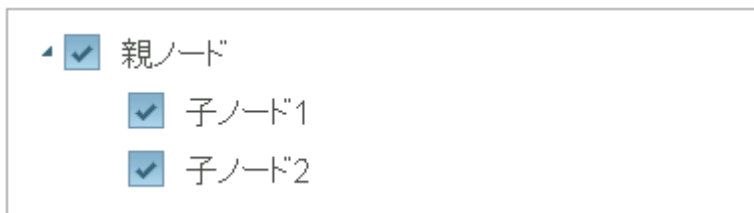
次の手順を実行します

1. [ソース]タブをクリックしてソースビューに切り替えます。
2. <cc1:C1TreeView>タグの間に次のマークアップを追加します。

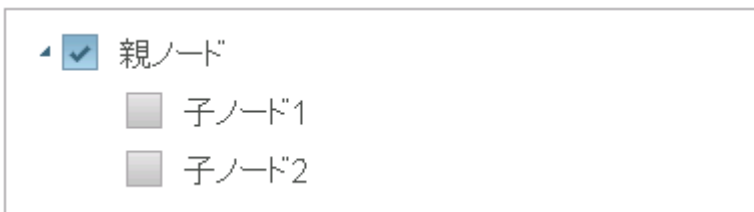
ソースビュー

```
<Nodes>
  <cc1:C1TreeNode runat="server" Text="親ノード">
    <Nodes>
      <cc1:C1TreeNode runat="server" Text="子ノード1">
    </cc1:C1TreeNode>
      <cc1:C1TreeNode runat="server" Text="子ノード2">
    </cc1:C1TreeNode>
    </Nodes>
  </cc1:C1TreeNode>
</Nodes>
```

3. このマークアップは、2つの子ノードを持つ親ノードを作成します。
4. <cc1:C1TreeView> タグの間に **ShowCheckBoxes="True"** を追加してチェックボックスを有効にします。
5. [F5]キーを押してプロジェクトを実行します。下記手順で、チェックボックスのデフォルト動作を確認します。
 - a. ツリービューの階層を表示するには、**親ノード**を展開します。
 - b. 親のチェックボックスをクリックして子ノードが自動的に選択されていることを確認します。



- c. プログラムを終了します。
6. [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。
 7. [プロパティ]ウィンドウにて、ドロップダウンリストから **C1TreeView1** を選択して、**AutoCheckNodes** プロパティを **False** に設定します。
 8. [F5]キーを押してプロジェクトを実行します。下記手順で、ノードが自動チェックに設定されていない場合のチェックボックスの動作を確認します。
 - a. ツリービューの階層を表示するには、**親ノード**を展開します。
 - b. 親のチェックボックスをチェックして子ノードが自動的に選択されないことを確認します。



ヒント: 子ノード1のみをチェックして子ノード2をチェックしないと親チェックボックスに「不定」というマークが表示されますが、**C1TreeView** コントロールの **AllowTriState** プロパティを **False** に設定して本動作を防ぐことができます。

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

C1TreeView 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、デザインビューおよびコードでテーマを変更する方法について説明します。テーマの詳細については、「[C1TreeView の外観](#)」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. **C1TreeView** スマートタグ  をクリックして、[**C1TreeView 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 C1TreeView コントロールに適用されます。

ソースでのテーマの変更

ソースで **C1TreeView** のテーマを変更するには、`<cc1:C1TreeView>` 内に `Theme="rocket"` を追加します。ソースが下記ようになります。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" runat="server" Theme="rocket"/>
```

コードでのテーマの変更

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 C1.Web.Wijmo.Controls
```

TreeView for ASP.NET Web Forms

C# コードの書き方

```
C#  
using C1.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  
C1TreeView1.Theme = "rocket"
```

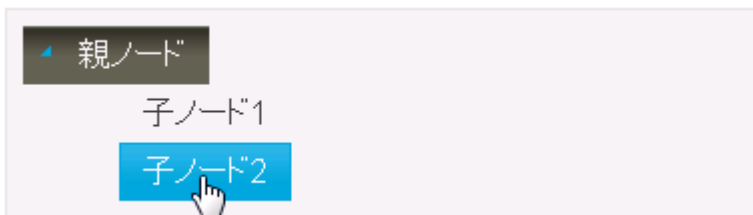
C# コードの書き方

```
C#  
C1TreeView1.Theme = "rocket";
```

3. プログラムを実行します。

✔ このトピックの作業結果

次の図は、**rocket** テーマが設定された**C1TreeView** コントロールを示しています。



CSS セレクタの使い方

C1TreeView で、CSS セレクタを使用してコントロールの外観を完全に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**C1TreeView** の外観をカスタマイズする方法を説明します。

1. デザインビューにて、**C1TreeView** コントロールを追加して、その[プロパティウィンドウ]に移動します。
2. [プロパティ]ウィンドウで、**CssClass** プロパティを「**wijmo-wijtree**」に設定します。
3. ソースビューに切り替えて `<asp:Content>` タグの間に `<<style type="text/css"></style>` タグを追加します。このタグを追加することで CSS スタイリングが有効になります。
4. タグの間に `.wijmo-wijtree { color: #993377; background: #02222; border-color: Black; font-family: Elephant; }` を追加します。
5. プログラムを実行して、**C1TreeView** の下記のようなイメージを確認します。



最上位のノードを追加する

このトピックでは、**C1TreeView** コントロールに最上位のノードを追加する方法を説明します。

デザインビュー

次の手順を実行します。

1. スマートタグをクリックして、**C1TreeViewタスクメニュー**を開きます。[TreeViewの編集]を選択します。
2. **C1TreeView デザイナフォーム**ダイアログボックスが開きます。
3. [子項目の追加]ボタンをクリックして、**C1TreeView** コントロールに **C1TreeNode** を追加します。
4. <OK>をクリックして**C1TreeView 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

<cc1:C1TreeView>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

```
<cc1:C1TreeNode ID="Node1" runat="server" Text="Node1">
</cc1:C1TreeNode>
```

コードビュー

次の手順を実行します。

1. プロジェクトに次の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.C1TreeView
```

C# コードの書き方

```
C#
using C1.Web.Wijmo.Controls.C1TreeView;
```

2. **Page_Load** イベントに次のコードを追加します。

Visual Basic コードの書き方

```
Visual Basic
Dim TreeViewNode1 As New C1TreeNode()
TreeViewNode1.Text = "C1TreeNode1"
TreeView1.Nodes.Add(TreeViewNode1)
```

C# コードの書き方

```
C#
C1TreeNode TreeViewNode1 = new C1TreeNode();
TreeViewNode1.Text = "C1TreeNode1";
TreeView1.Nodes.Add(TreeViewNode1);
```

3. プログラムを実行します。

C1TreeView のノードに子ノードを追加する

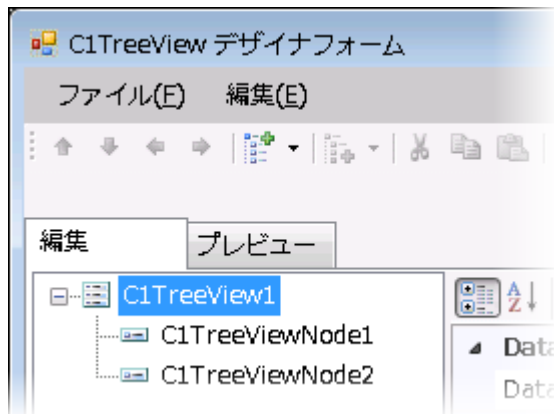
TreeView for ASP.NET Web Forms

このトピックでは、**C1TreeViewNode** コントロールに子ノードを追加する方法を説明します。また、「[C1TreeView コントロールに最上位のノードを追加する](#)」トピックの内容を把握していることを前提します。

デザインビュー

次の手順を実行します。

1. スマートタグをクリックして、**C1TreeView**タスクメニューを開きます。[**TreeViewの編集**]を選択します。**C1TreeView デザイナフォーム**ダイアログボックスが開きます。
2. 子ノードを追加したいノードを選択します。
3. 選択したノードに子ノードを追加するには、[**子項目の追加**]をクリックします。デザイナフォーム上のツリービューが下記のようになります。



4. <OK>をクリックして**C1TreeView 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

子ノードを追加するノードの<cc1:C1TreeViewNode> タグの間に下記のマークアップを追加します。

ソースビュー

```
<Nodes>
  <cc1:C1TreeViewNode ID="Node1" runat="server" Text="Node1">
  </cc1:C1TreeViewNode>
</Nodes>
```

コードビュー

次の手順を実行します。

1. プロジェクトに次の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.C1TreeView
```

C# コードの書き方

```
C#
using C1.Web.Wijmo.Controls.C1TreeView;
```

2. **Page_Load** イベントに次のコードを追加します。

Visual Basic コードの書き方

```

Visual Basic
' 最初のノードを作成し、C1TreeView に追加
Dim C1TreeNode1 As New C1TreeNode()
    C1TreeNode1.Text = "C1TreeNode1"
    C1TreeView1.Nodes.Add(C1TreeNode1)

' 子ノードを作成し、C1TreeNode1に追加
Dim C1TreeNode2 As New C1TreeNode()
    C1TreeNode2.Text = "C1TreeNode1"
    C1TreeNode1.Nodes.Add(C1TreeNode2)

```

C# コードの書き方

```

C#
// 最初のノードを作成し、C1TreeView に追加
C1TreeNode C1TreeNode1 = new C1TreeNode();
C1TreeNode1.Text = "C1TreeNode1";
C1TreeView1.Nodes.Add(C1TreeNode1);

// 子ノードを作成し、C1TreeNode1に追加
C1TreeNode C1TreeNode2 = new C1TreeNode();
C1TreeNode2.Text = "C1TreeNode2";
C1TreeNode1.Nodes.Add(C1TreeNode2);

```

3. プログラムを実行します。

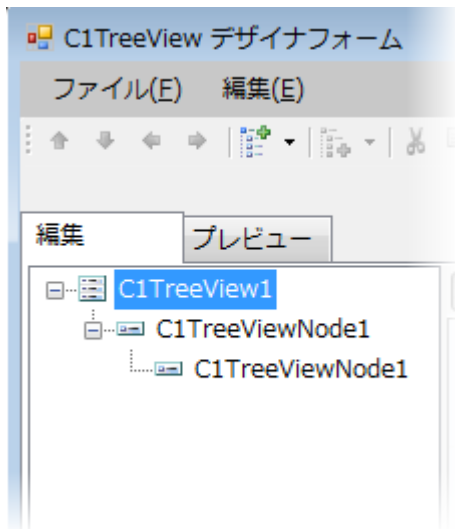
ButtonClick イベントによる子ノードの追加

このトピックでは、**ButtonClick()** イベントによる子ノードの追加について説明します。このヘルプを実行するには、子ノードを含む C1TreeView を作成し、アプリケーションにボタンを追加し、jQuery 構文を使用して **ButtonClick()** イベントを呼び出します。

以下の手順を実行します。

1. デザインビューで開始し、**C1TreeView** コントロールをアプリケーションに追加します。
2. スマートタグをクリックして、**[C1TreeView タスク]**メニューを開きます。**[ツリービューの編集]**を選択します。
3. ツリービュー構造を作成します。次のような表示になります。

TreeView for ASP.NET Web Forms



- 一般的な **Button** コントロールをアプリケーションに追加して、ソースビューに切り替えます。
- `<asp:Button>` タグを探し、このタグに `OnClickClientClick="buttonClick(); return false;"` を追加して、ボタンによって、追加する jQuery スクリプトを呼び出せるようにします。マークアップは次のようになります。

ソースビュー

```
<asp:Button ID="Button1" runat="server" Text="Button" OnClientClick="buttonClick(); return false;" />
```

- 上記の `<asp:Button>` タグの上に以下のスクリプトを追加します。これにより、**ButtonClick()** イベントが呼び出されます。

ソースビュー

```
<script type="text/javascript">
function buttonClick() {
    var nodes = $("#<%= C1TreeView1.ClientID %>").c1treeview("getSelectedNodes");
    //選択されたノードをすべて検出します (ユーザーは[Ctrl]または[Shift]キーを使用して複数選択できます)
    // 選択された最初のノードに新しいノードを追加します
    nodes[0].element.c1treeviewnode("add", { text: "Test User 1", value: "user" });
}
</script>
```

- [F5]を押して、アプリケーションを実行します。ボタンをクリックし、新しいノードをツリーに追加します。

C1TreeView のサイトマップへの連結

このトピックでは、サイトマップデータをC1TreeViewに反映する方法を説明します。

サイトマップを作成して、**C1TreeView** コントロールに連結するには下記の手順を実行します。

- ソリューションエクスプローラで対象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項目をプロジェクトに追加します。

Web.sitemap ファイルには、次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="ルート 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 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>
```

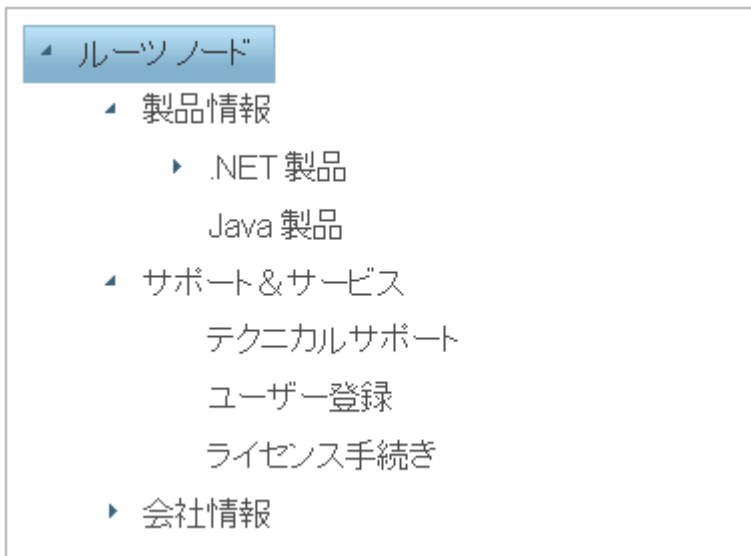
4. **C1TreeView** コントロールの [タスク] メニューを開き、[データソースの選択] ドロップダウンリストボックスから [新しいデータソース...] を選択して [データソース構成ウィザード] を開きます。
5. **サイト マップ** を選択して 〈OK〉 をクリックします。

SiteMapDataSource1 がプロジェクトに追加されます。

6. F5 キーを押してプロジェクトを実行し、次の事項を確認します。

Web.sitemap ファイルのデータが **C1TreeView** コントロールに反映されています。

TreeView for ASP.NET Web Forms



コントロールが ComponentOne という最上位のノードで開きます。次の手順で、最上位ノードを削除して、**C1TreeView** -の第2レベルのノードを公開します。

7. ブラウザを終了して、プロジェクトに切り替えます。
8. デザインビューにて、**SiteMapDataSource** を選択して[プロパティ]ウィンドウで ShowStartingNode を False に設定します。
9. [F5]キーを押してプロジェクトを実行し、最上位ノードが削除されたことを確認します。



C1TreeView を XML に連結する

このチュートリアルでは、Visual Studio のインストール済みテンプレートにより XML ファイルを作成し、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 **C1TreeView** に割り当て、さらに C1TreeView の連結を設定する方法を示します。

次の手順を実行します。

1. ツールボックスにて **C1TreeView** をダブルクリックして、コントロールをプロジェクトに追加します。
2. 下記の手順でXMLファイルを作成します。:
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 TreeView_Hierarchy.xml に変更します。
 - b. XML ビューに切り替えて、次のデータを **TreeView_Hierarchy.xml** に追加します。

```
ソースビュー
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <TreeNode Text="ホーム">
    <TreeNode Text="会社概要"></TreeNode>
    <TreeNode Text="ご挨拶"></TreeNode>
    <TreeNode Text="国内拠点"></TreeNode>
  </TreeNode>
  <TreeNode Text="製品">
```

```
<TreeNode Text="ハードウェア"></TreeNode>
<TreeNode Text="ソフトウェア"></TreeNode>
</TreeNode>
<TreeNode Text="サービス">
  <TreeNode Text="トレーニング"></TreeNode>
  <TreeNode Text="コンサルティング"></TreeNode>
</TreeNode>
</root>
```

4. .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。下記の手順に従い、新しいデータソースを作成します。
 - a. **C1TreeView** のスマートタグをクリックして[**C1TreeView 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**]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。

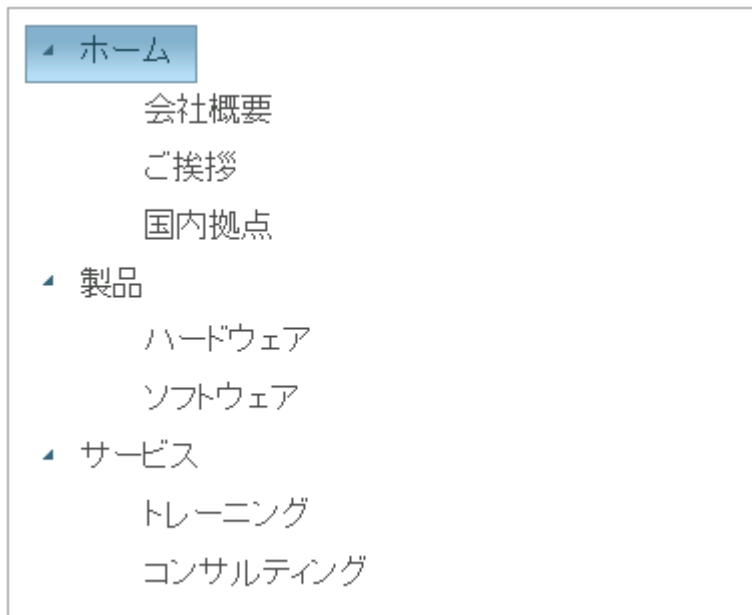
XmlDataSource1 がプロジェクトに追加されます。

5. 下記の手順でデータソースを構成します。
 - a. **C1TreeView** のスマートタグをクリックして[**C1TreeView 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 式のテキストフィールドにて、「root/TreeNode」を入力します。これは、ルートの子となっているすべてのノードを選択しますので、TreeNode がウェブページの最上位ノードとなります。
 - c. プロパティウィンドウで、**DataFile** プロパティの隣にある〈...〉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。
 - d. App_Data プロジェクトフォルダを選択して、[フォルダの内容] ペインから **TreeView_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. **C1TreeViewNodes** に XML タグを連結するには、次の手順を実行します。
 - a. **C1TreeView** のスマートタグをクリックして[**C1TreeView タスク**]メニューを開き、[**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が表示されます。
 - b. プロジェクトに空白の連結を追加するには[追加]をクリックします。
 - c. 連結プロパティを下記のように設定します。
 - **DataMember** プロパティを「TreeNode」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**]ダイアログボックスを閉じます。
7. [F5]キーを押してプロジェクトを実行します。

TreeView_Hierarchy.xml ファイルのデータが**C1TreeView** コントロールに反映されていることを確認してください。



C1TreeView の動的作成

C1TreeView では、コントロールを動的に作成できます。子ノードは、親ノードが展開されるときに AJAX を使用してロードされます。

1. Visual Studio ツールボックスの C1TreeView コントロールをダブルクリックし、アプリケーションに追加します。
2. ソースビューに切り替え、次のマークアップをアプリケーションに追加して、`<asp:SiteMapDataSource>` コントロールを追加します。

ソースビュー

```
<asp:SiteMapDataSource ID="SiteMapDataSource" runat="server" ShowStartingNode="False" />
```

3. C1TreeView コントロールのマークアップを探し、次のように編集します。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" ShowCheckBoxes="true" LoadOnDemand="true"
DataSourceID="SiteMapDataSource" ShowExpandCollapse="true"
DataBindStartLevel="0" Width="350px" runat="server">
</cc1:C1TreeView>
```

4. ソリューションエクスプローラで目的のプロジェクト名を右クリックし、リストから[追加]→[新しい項目]を選択します。
5. [新しい項目の追加]ダイアログで[サイトマップ]を選択します。新しい **Web.sitemap** がアプリケーションに追加されず。 **Web.sitemap** ファイルがすぐに開きます。
6. 次のマークアップを **Web.sitemap** ファイルに追加します。

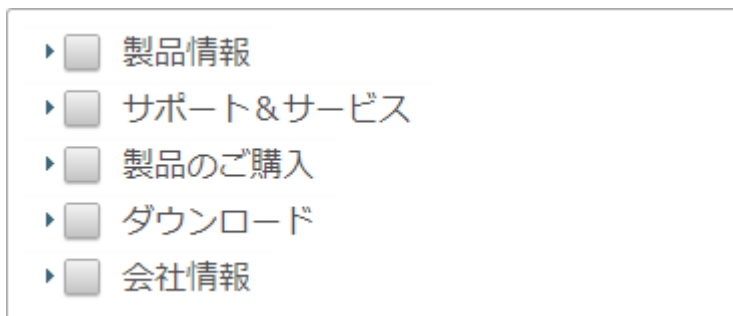
ソースビュー

```
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="RootNodeUrl" title="Root Node" description="Root Node Description" >
    <siteMapNode url="" title="Products" description="Products" >
      <siteMapNode url="" title="ComponentOne Enterprise" description="ComponentOne Enterprise" >
        <siteMapNode url="" title="ComponentOne for WinForms" description="ComponentOne for WinForms" />
      <siteMapNode url="" title="ComponentOne for ASP.NET" description="ComponentOne for ASP.NET" />
    </siteMapNode>
    <siteMapNode url="" title="ComponentOne for WPF" description="ComponentOne for WPF" />
    <siteMapNode url="" title="ComponentOne for Mobile" description="ComponentOne for Mobile" />
  </siteMapNode>
</siteMap>
```

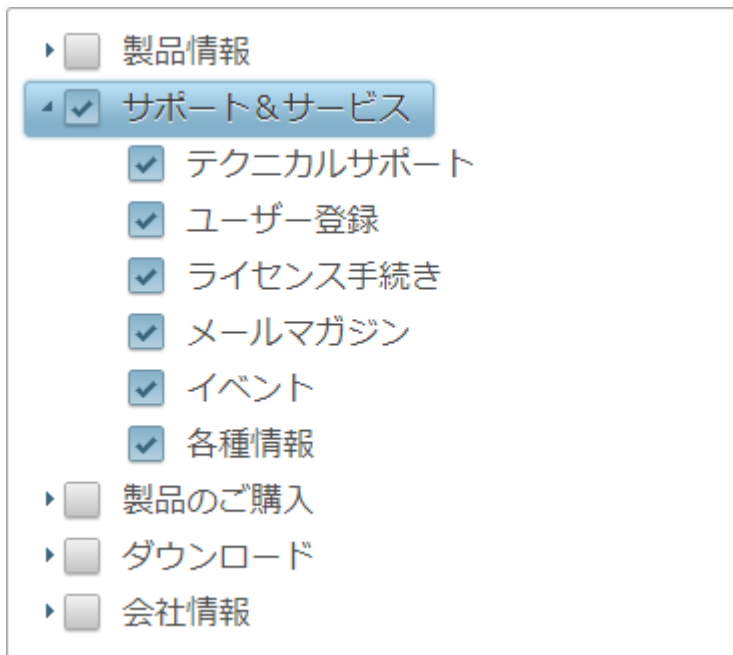


```
<siteMapNode url="" title="ComponentOne for ActiveX" description="ComponentOne for ActiveX" />
<siteMapNode url="" title="ComponentOne for Silverlight" description="ComponentOne for
Silverlight" />
</siteMapNode>
<siteMapNode url="" title="IntelliSpell" description="IntelliSpell" />
<siteMapNode url="" title="Report Designer Edition" description="Report Designer Edition" />
</siteMapNode>
<siteMapNode url="" title="Support" description="Support" >
  <siteMapNode url="" title="Support Services" description="Support Services" />
  <siteMapNode url="" title="HelpCentral" description="HelpCentral" />
  <siteMapNode url="" title="Product Forums" description="Product Forums" />
</siteMapNode>
<siteMapNode url="" title="Company" description="Company" >
  <siteMapNode url="" title="About Us" description="About Us" />
  <siteMapNode url="" title="Partners" description="Partners" />
  <siteMapNode url="" title="Contact Us" description="Contact Us" />
  <siteMapNode url="" title="Join Us" description="Join Us" />
  <siteMapNode url="" title="Press Center" description="Press Center" />
  <siteMapNode url="" title="Governance" description="Governance" />
</siteMapNode>
<siteMapNode url="" title="Store" description="Store">
  <siteMapNode url="" title="Buy Now" description="Buy Now" />
  <siteMapNode url="" title="Resellers" description="Resellers" />
</siteMapNode>
</siteMapNode>
</siteMap>
```

7. アプリケーションを実行します。結果は、次の図のようになるはずです。



ノードのいずれかを開くと、C1TreeView コントロールが次の図のように表示されます。



ツリーを XML として保存する

以下の作業では、C1TreeView コントロールを .xml ファイルとして保存してから、デザイナを使用してそれをプロジェクトにロードする方法を示します。

ツリーを XML として保存

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

1. C1TreeView のスマートタグで[TreeViewを編集する]をクリックして[TreeView デザイナフォーム]を開きます。
2. [ファイル]→[XML として保存]をクリックします。
3. C1TreeView の .xml の名前を入力し、保存先を指定します。
4. <OK>をクリックして[TreeView デザイナフォーム]ダイアログボックスを閉じます。

既存の XML ツリービューをプロジェクトにロード

.xml ファイルとして保存した C1TreeView 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. C1TreeView のスマートタグで[TreeViewを編集する]をクリックして[TreeView デザイナフォーム]を開きます。
2. [ファイル]→[XML からロード]をクリックし、既存の .xml ファイルをクリックして開きます。

コードの使用

.xml ファイルとして保存した C1TreeView 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. C1TreeView 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

```
C1TreeView1.LoadLayout("c:\\Visual Studio  
2005\\WebSites\\LoadLayoutEX\\App_Data\\C1TreeViewControl.xml")
```

C# コードの書き方

C#

```
C1TreeView1.LoadLayout("c:\\Visual Studio
2005\\WebSites\\LoadLayoutEX\\App_Data\\C1TreeViewControl.xml");
```

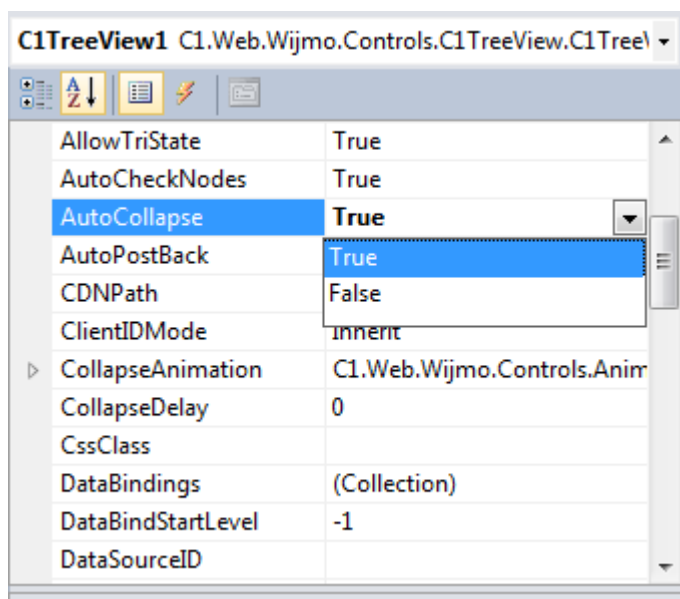
3. [F5] キーを押下して、プログラムを実行します。

AutoCollapse プロパティを設定する

AutoCollapse プロパティは、他のノードが展開されると、展開されていたノードは縮小します。このトピックでは、Autocollapse プロパティの設定方法を説明します。

デザインビュー

1. デザインビューにて、C1TreeView の[プロパティ]ウィンドウに移動します。
2. AutoCollapse プロパティをドロップダウンを使用して「True」に設定します。



3. [F5]キーを押してプログラムを実行し、他のノードが展開されると、展開されていたノードは縮小されることを確認します。

ソースビュー

1. ソースビューにて、下記のマークアップを追加してC1TreeView のノードを作成します。

ソースビュー

```
<Nodes>
  <cc1:C1TreeViewNode Text="フォルダ 1">
    <Nodes>
      <cc1:C1TreeViewNode Text="フォルダ 1.1">
        <Nodes>
          <cc1:C1TreeViewNode Text="フォルダ 1.1.1">
            </cc1:C1TreeViewNode>
          <cc1:C1TreeViewNode Text="フォルダ 1.1.2">
            </cc1:C1TreeViewNode>
          <cc1:C1TreeViewNode Text="フォルダ 1.1.3">
            </cc1:C1TreeViewNode>
          <cc1:C1TreeViewNode Text="フォルダ 1.1.4">
            </cc1:C1TreeViewNode>
        </Nodes>
      </cc1:C1TreeViewNode>
    </Nodes>
  </cc1:C1TreeViewNode>
</Nodes>
```

```
        </cc1:C1TreeNode>
    </Nodes>
</cc1:C1TreeNode>
<cc1:C1TreeNode Text="フォルダ 1.2">
</cc1:C1TreeNode>
<cc1:C1TreeNode Text="フォルダ 1.3">
</cc1:C1TreeNode>
<cc1:C1TreeNode Text="フォルダ 1.4">
</cc1:C1TreeNode>
<cc1:C1TreeNode Text="フォルダ 1.5">
</cc1:C1TreeNode>
</Nodes>
</cc1:C1TreeNode>
<cc1:C1TreeNode Text="フォルダ 2">
    <Nodes>
        <cc1:C1TreeNode Text="フォルダ 2.1">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 2.2">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 2.3">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 2.4">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 2.5">
        </cc1:C1TreeNode>
    </Nodes>
</cc1:C1TreeNode>
<cc1:C1TreeNode Text="フォルダ 3">
    <Nodes>
        <cc1:C1TreeNode Text="フォルダ 3.1">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 3.2">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 3.3">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 3.4">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode Text="フォルダ 3.5">
        </cc1:C1TreeNode>
    </Nodes>
</cc1:C1TreeNode>
</Nodes>
```

2. 下記のように<cc1:C1TreeView>の間に Autocollapse="true"を追加します。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" runat="server" AutoCollapse="true">
```

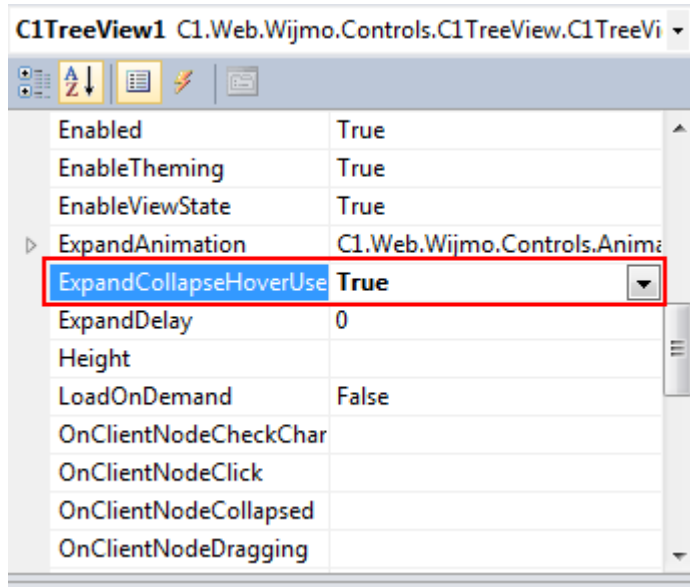
3. [F5]キーを押してプログラムを実行し、他のノードが展開されると、展開されていたノードは縮小されることを確認します。

C1TreeView をホバーで展開する

C1TreeView では、マウスクリックでノードを展開するとともに、マウスホバーで展開する機能もあります。このトピックでは、**ExpandCollapseHoverUsed** プロパティの使用方法について説明します。

デザインビュー

1. デザインビューにて、C1TreeView の[プロパティ]ウィンドウに移動します。



2. **ExpandCollapseHoverUsed** プロパティをドロップダウンを使用して「True」に設定します。
3. プロジェクトを実行して、マウスホバーでノードが展開することを確認します。

ソースビュー

下記コードのように、<cc1:C1TreeView>タグの間にExpandCollapseHoverUse="true" を追加します。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" runat="server" AutoCollapse="True"
ExpandCollapseHoverUsed="True">
```

C1TreeView ノードアイコンを設定する

C1TreeView では、ノードのアイコンを設定することが可能です。さらに、C1Treeview の現在の状態に基づいて、アイコンを切り替えることができます。このトピックでは、ノードアイコンを表示し、C1Treeview の状態に基づいて、ノードアイコンを変更するプロパティを設定する手順を説明します。

デザインビュー

1. C1TreeViewスマートタグをクリックして[C1TreeView タスク]メニューを表示します。
2. [TreeViewの編集]を選択すると、**C1TreeView デザイナフォーム** ダイアログボックスが開きます。

TreeView for ASP.NET Web Forms



C1TreeView タスク

データソースの選択: (なし)

ツリービューの編集
データバインディングの編集...

テーマ: aristo

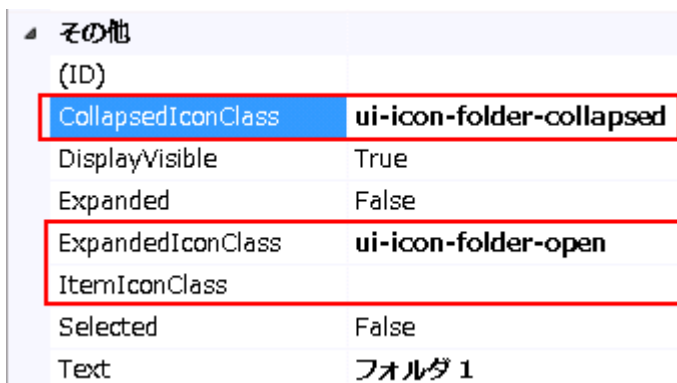
CDN の使用

CDN パス: http://cdn.wijmo.com/

バージョン情報...

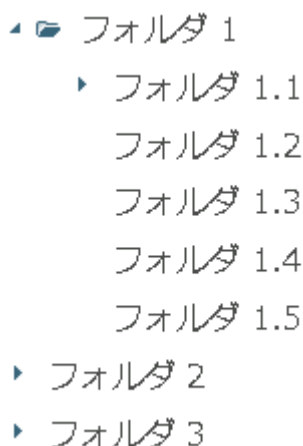
テンプレートの編集

- 最初のノード「フォルダ 1」を選択して、[プロパティ]ウィンドウでそのプロパティを表示します。
- CollapsedIconClass** プロパティを「**ui-icon-folder-collapsed**」に設定します。
- ExpandedIconClass** プロパティを「**ui-icon-folder-open**」に設定します。
- ItemIconClass** プロパティを「**ui-icon-document**」に設定します。デザイナーフォームの[プロパティ]ウィンドウは次のようになります。



その他	
(ID)	
CollapsedIconClass	ui-icon-folder-collapsed
DisplayVisible	True
Expanded	False
ExpandedIconClass	ui-icon-folder-open
ItemIconClass	
Selected	False
Text	フォルダ 1

- [F5]キーを押してプログラムを実行します。C1TreeView コントロールは次のようなイメージとなります。

- 
- フォルダ 1
 - フォルダ 1.1
 - フォルダ 1.2
 - フォルダ 1.3
 - フォルダ 1.4
 - フォルダ 1.5
 - フォルダ 2
 - フォルダ 3

ソースビュー

- ソースビューにて、Text プロパティで「フォルダ 1」と設定されている最初の `<cc1:C1TreeNode>` タグ内に下記のマークアップを追加して、ノードアイコンを設定します。

ソースビュー

```
CollapsedIconClass="ui-icon-folder-collapsed"
ExpandedIconClass="ui-icon-folder-open"
ItemIconClass="ui-icon-document"
```

2. <cc1:C1TreeNode> タグは下記のようにになります。

ソースビュー

```
<cc1:C1TreeNode Text="フォルダ 1"
CollapsedIconClass="ui-icon-folder-collapsed"
ExpandedIconClass="ui-icon-folder-open"
ItemIconClass="ui-icon-document">
```

3. 「フォルダ 1.1」の<cc1:C1TreeNode> タグの間に、下記のマークアップを追加します。

ソースビュー

```
CollapsedIconClass="ui-icon-folder-collapsed"
ExpandedIconClass="ui-icon-folder-open"
ItemIconClass="ui-icon-document"
```

4. <cc1:C1TreeNode> タグは下記のようにになります。

ソースビュー

```
<cc1:C1TreeNode Text="フォルダ 1.1"
CollapsedIconClass="ui-icon-folder-collapsed"
ExpandedIconClass="ui-icon-folder-open"
ItemIconClass="ui-icon-document">
```

5. 「フォルダ 1.1」の <cc1:C1TreeNode>タグの間に、ItemIconClass="ui-icon-document" を追加します。<cc1:C1TreeNode> タグは下記のようにになります。

ソースビュー

```
<cc1:C1TreeNode Text="Folder 1.1.1" ItemIconClass="ui-icon-document">
```

6. [F5]キーを押してプログラムを実行します。C1TreeView コントロールは次のようなイメージとなります。

- ▶ フォルダ 1
 - ▶ フォルダ 1.1
 - フォルダ 1.2
 - フォルダ 1.3
 - フォルダ 1.4
 - フォルダ 1.5
 - ▶ フォルダ 2
 - ▶ フォルダ 3

ドラッグアンドドロップ動作を有効にする

C1TreeView はドラッグアンドドロップ機能があり、ユーザーがノードをドラッグアンドドロップしてツリー構造を変更することができます。ドラッグアンドドロップ動作は一つのツリー構造内または、ツリー同士の間で行うことが可能です。このトピックでは、ドラッグアンドドロップ動作を有効にする基本的なプロパティとツリー同士の間でノードをドラッグアンドドロップできるプロパティを説明します。

ツリー構造内のドラッグアンドドロップ動作

このトピックでは、ツリー構造内のドラッグアンドドロップ動作を有効にする C1TreeView のプロパティ設定について説明します。

デザインビュー

1. C1TreeViewスマートタグをクリックして[C1TreeView タスク]メニューを表示します。
2. [TreeViewの編集]を選択すると、**C1TreeView デザイナフォーム**ダイアログボックスが開きます。



3. [プロパティ]ウィンドウで **AllowDrag** 及び **AllowDrop** プロパティを「True」に設定します。
4. [F5]キーを押してプログラムを実行します。ノードを再配置できることを確認します。

ソースビュー

1. 最初の<cc1:C1TreeView> タグ内に下記のマークアップを追加します。

```
ソースビュー
```

```
AllowDrag="True"  
AllowDrop="True"
```

2. <cc1:C1TreeView> タグは下記コードのようになります。

```
ソースビュー
```

```
<cc1:C1TreeView ID="C1TreeView1" runat="server" AllowDrag="True"  
    AllowDrop="True">
```

3. [F5]キーを押してプログラムを実行します。ノードを再配置できることを確認します。

ツリー同士の間でのドラッグアンドドロップ動作

このトピックでは、2つのツリーの間でドラッグアンドドロップ動作を有効にする C1TreeView のプロパティ設定について説明します。

デザインビュー

1. デザインビューにて、2つの C1TreeView コントロールをプロジェクトに追加して、両方に対していくつかの子ノードを作成します。
2. 最初の C1TreeView コントロールを選択して、[プロパティ]ウィンドウに移動します。
3. [プロパティ]ウィンドウで **AllowDrag** 及び **AllowDrop** プロパティを「True」に設定します。

- 2番目の C1TreeView コントロールを選択して、[プロパティ]ウィンドウに移動します。
- [プロパティ]ウィンドウで **AllowDrop** プロパティを「True」に設定します。これで、実行時にノードを1つ目の C1TreeView コントロールから2番目の C1TreeView コントロールにドロップすることが可能になります。
- [F5]キーを押してプログラムを実行します。ノードを1つ目の C1TreeView コントロールから2番目の C1TreeView コントロールにドロップすることができることを確認します。

ソースビュー

- 2番目の <asp:Content>タグの間に下記のマークアップを挿入して、2つのC1TreeView コントロールを作成します。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" runat="server">
  <Nodes>
    <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
      Owner="C1TreeView1" StaticKey="C1TreeView1_0" Text="C1TreeViewNode6"
      TreeView="C1TreeView1">
    </cc1:C1TreeViewNode>
    <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
      Owner="C1TreeView1" StaticKey="C1TreeView1_1" Text="C1TreeViewNode4"
      TreeView="C1TreeView1">
      <Nodes>
        <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
          Owner="" StaticKey="C1TreeView1_10" Text="C1TreeViewNode1"
          TreeView="C1TreeView1">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
          Owner="" StaticKey="C1TreeView1_11" Text="C1TreeViewNode2"
          TreeView="C1TreeView1">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
          Owner="" StaticKey="C1TreeView1_12" Text="C1TreeViewNode3"
          TreeView="C1TreeView1">
        </cc1:C1TreeViewNode>
      </Nodes>
    </cc1:C1TreeViewNode>
    <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
      Owner="C1TreeView1" StaticKey="C1TreeView1_2" Text="C1TreeViewNode5"
      TreeView="C1TreeView1">
    </cc1:C1TreeViewNode>
    <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
      Owner="C1TreeView1" StaticKey="C1TreeView1_3" Text="C1TreeViewNode1"
      TreeView="C1TreeView1">
      <Nodes>
        <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
          Owner="" StaticKey="C1TreeView1_30" Text="C1TreeViewNode1"
          TreeView="C1TreeView1">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
          Owner="" StaticKey="C1TreeView1_31" Text="C1TreeViewNode2"
          TreeView="C1TreeView1">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="UnChecked" NodeIndex="0"
          Owner="" StaticKey="C1TreeView1_32" Text="C1TreeViewNode3"
          TreeView="C1TreeView1">
        </cc1:C1TreeViewNode>
      </Nodes>
    </cc1:C1TreeViewNode>
  </Nodes>
</cc1:C1TreeView>
```

TreeView for ASP.NET Web Forms

```
        TreeView="C1TreeView1">
    </cc1:C1TreeNode>
</Nodes>
</cc1:C1TreeNode>
<cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
    Owner="C1TreeView1" StaticKey="C1TreeView1_4" Text="C1TreeNode2"
    TreeView="C1TreeView1">
</cc1:C1TreeNode>
<cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
    Owner="C1TreeView1" StaticKey="C1TreeView1_5" Text="C1TreeNode3"
    TreeView="C1TreeView1">
    <Nodes>
        <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
            Owner="" StaticKey="C1TreeView1_50" Text="C1TreeNode1"
            TreeView="C1TreeView1">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
            Owner="" StaticKey="C1TreeView1_51" Text="C1TreeNode2"
            TreeView="C1TreeView1">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
            Owner="" StaticKey="C1TreeView1_52" Text="C1TreeNode3"
            TreeView="C1TreeView1">
        </cc1:C1TreeNode>
    </Nodes>
</cc1:C1TreeNode>
<cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
    Owner="C1TreeView1" StaticKey="C1TreeView1_6" Text="C1TreeNode7"
    TreeView="C1TreeView1">
</cc1:C1TreeNode>
</Nodes>
</cc1:C1TreeView
<cc1:C1TreeView ID="C1TreeView2" runat="server">
    <Nodes>
        <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
            Text="C1TreeNode6">
            <Nodes>
                <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
                    Text="C1TreeNode1">
                </cc1:C1TreeNode>
                <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
                    Text="C1TreeNode2">
                </cc1:C1TreeNode>
            </Nodes>
        </cc1:C1TreeNode>
        <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
            Text="C1TreeNode7">
        </cc1:C1TreeNode>
        <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
            Text="C1TreeNode8">
        <Nodes>
            <cc1:C1TreeNode runat="server" CheckState="UnChecked" NodeIndex="0"
```

```

        Text="C1TreeViewNode1">
    </cc1:C1TreeViewNode>
    <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
        Text="C1TreeViewNode2">
    </cc1:C1TreeViewNode>
    <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
        Text="C1TreeViewNode3">
    </cc1:C1TreeViewNode>
    <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
        Text="C1TreeViewNode4">
    </cc1:C1TreeViewNode>
    </Nodes>
</cc1:C1TreeViewNode>
<cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
    Text="C1TreeViewNode1">
</cc1:C1TreeViewNode>
<cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
    Text="C1TreeViewNode2">
</cc1:C1TreeViewNode>
<cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
    Text="C1TreeViewNode3">
    <Nodes>
        <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
            Text="C1TreeViewNode1">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
            Text="C1TreeViewNode2">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
            Text="C1TreeViewNode3">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
            Text="C1TreeViewNode4">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
            Text="C1TreeViewNode5">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
            Text="C1TreeViewNode6">
        </cc1:C1TreeViewNode>
        <cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
            Text="C1TreeViewNode7">
        </cc1:C1TreeViewNode>
    </Nodes>
</cc1:C1TreeViewNode>
<cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
    Text="C1TreeViewNode4">
</cc1:C1TreeViewNode>
<cc1:C1TreeViewNode runat="server" CheckState="Unchecked" NodeIndex="0"
    Text="C1TreeViewNode5">
</cc1:C1TreeViewNode>
</Nodes>

```

TreeView for ASP.NET Web Forms

```
</cc1:C1TreeView>
```

2. 一番目の <cc1:C1TreeView> タグの間に AllowDrag="True" 及び AllowDrop="True" を追加して、タグを下記例のようにします。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView1" runat="server" AllowDrag="True" AllowDrop="True">
```

3. 一番目の <cc1:C1TreeView> タグの間に AllowDrop="True" を追加して、タグを下記例のようにします。

ソースビュー

```
<cc1:C1TreeView ID="C1TreeView2" runat="server" AllowDrop="True">
```

4. [F5]キーを押してプログラムを実行します。ノードを1つ目の C1TreeView コントロールから2番目の C1TreeView コントロールにドロップすることができることを確認します。