

SiteMap for ASP.NET Web Forms

2018.04.11 更新

グレースィティ株式会社

目次

製品の概要	2
ComponentOne for ASP.NET Web Forms のヘルプ	2
設計時のサポート	3-8
クイックスタート: SiteMap の作成とカスタマイズ	9-13
主な機能	14
データ連結	14-16
カスタム SiteMap プロバイダ	16-18
ツールチップの追加	18-20
レイアウトのカスタマイズ	20-22
ノードテンプレート	22
キーボードサポートの追加	22-24
C1SiteMapDataSourceの機能	25
新しい SiteMap の作成	25
既存の SiteMap の編集	25-26
サイトマップファイルの追加	26-27

製品の概要

SiteMap for ASP.NET Web Forms を使用すると、ノードを作成したり、すべての Web ページのリストを生成できるため、Web アプリケーション内の移動がさらに容易になります。SiteMap は最新のデータ連結技術を導入し、Microsoft .Net Framework とシームレスに統合されています。

SiteMap は、ノードレベルごとに異なるスタイルとレイアウトを作成でき、ノードレベルごとにカスタムテンプレートを作成できます。

SiteMap パッケージには、次の2つのコントロールが含まれます。

- **C1SiteMap**: データ連結モードまたは非連結モードでサイトマップを作成するために使用します。
- **C1SiteMapDataSource**: SiteMap コントロールをサイトマップファイルに連結するために使用します。

主要な機能

- **データ連結**: SiteMap をデータ連結モードおよび非連結モードで使用します。また、SiteMapDataSource を使用して、SiteMap コントロールをサイトマップファイルまたは XML データに連結します。
- **レイアウト**: リストとフローの2種類のノードレイアウトを作成します。
- **テンプレート**: レベルごとにカスタムテンプレートを設定します。
- **テーマ**: 組み込みのテーマを使用したり、独自のテーマを作成することができます。
- **深度**: レベルごとにノードの最大数を定義します。

ComponentOne for ASP.NET Web Forms のヘルプ

ComponentOne for ASP.NET Webforms 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 Webforms ユーザーガイド](#)」を参照してください。

設計時のサポート

SiteMap は、Visual Studio のプロパティグリッド、メニュー、およびデザイナを使用して設計時に設定できます。以下のセクションでは、**SiteMap** の設計時機能を使用して、コントロールを設定する方法について説明します。

[C1SiteMap のタスク]メニュー



データソースの選択

[**データソースの選択**]ボックスのドロップダウン矢印をクリックすると、使用可能なデータソースのリストが開き、新しいデータソースを追加できます。新しいデータソースをプロジェクトに追加するには、<新しいデータソース>をクリックして、**データソース構成ウィザード**を開きます。

新しいデータソースをプロジェクトに追加する方法については、「[データ連結](#)」を参照してください。

ノードの編集

[**ノードの編集**]をクリックすると、さまざまなレベルにノードを追加および編集するための **C1SiteMap デザイナフォーム**が開きます。詳細については、「[C1SiteMap デザイナフォーム](#)」を参照してください。

DataBindings の編集

[**バインドの編集**]をクリックすると、**SiteMap** に追加するノードのデータ連結を管理するための **Bindings コレクションエディタ**が開きます。

トピックの内容

[C1SiteMap のタスク]メ

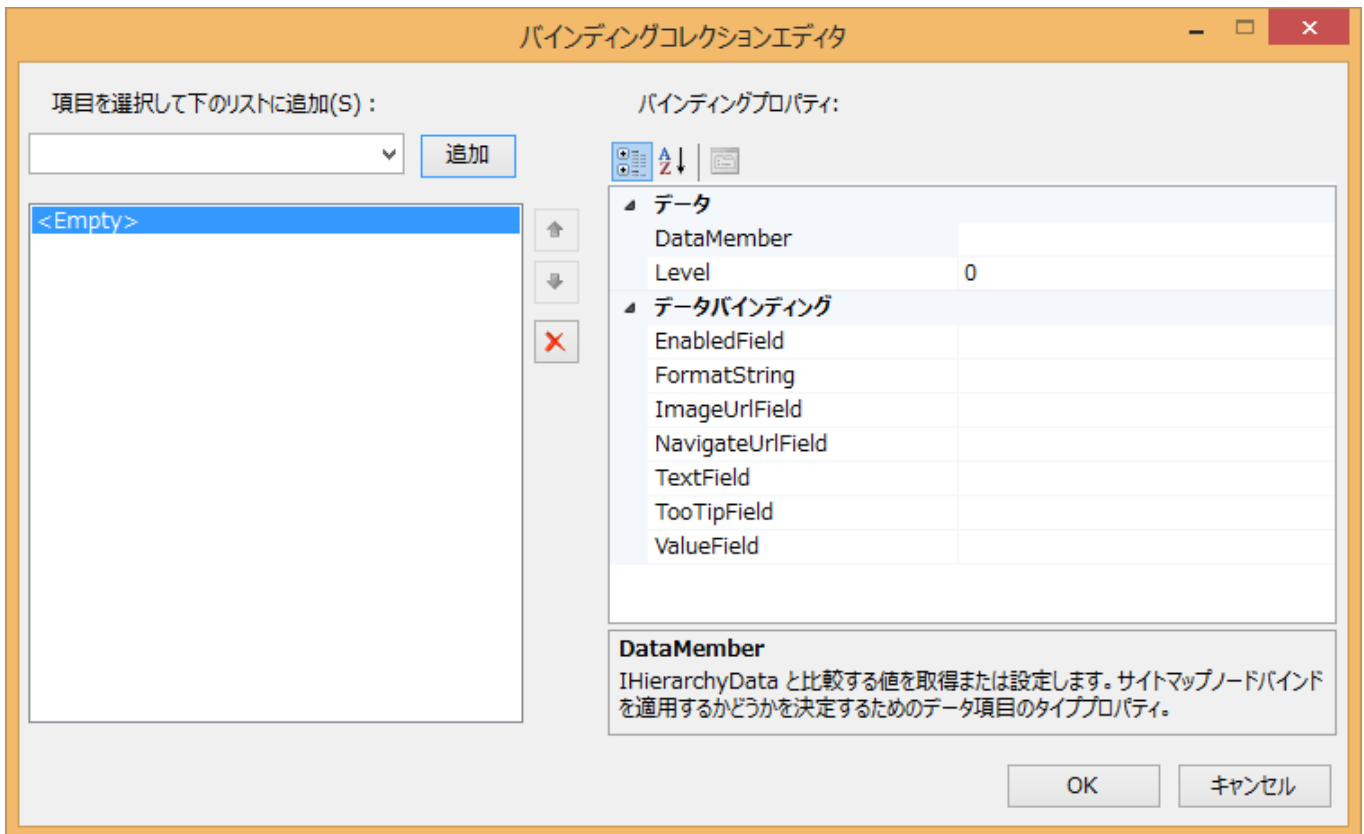
ニュー

C1SiteMap デザイナ

フォーム

C1SiteMapLevelSettings

コレクションエディタ



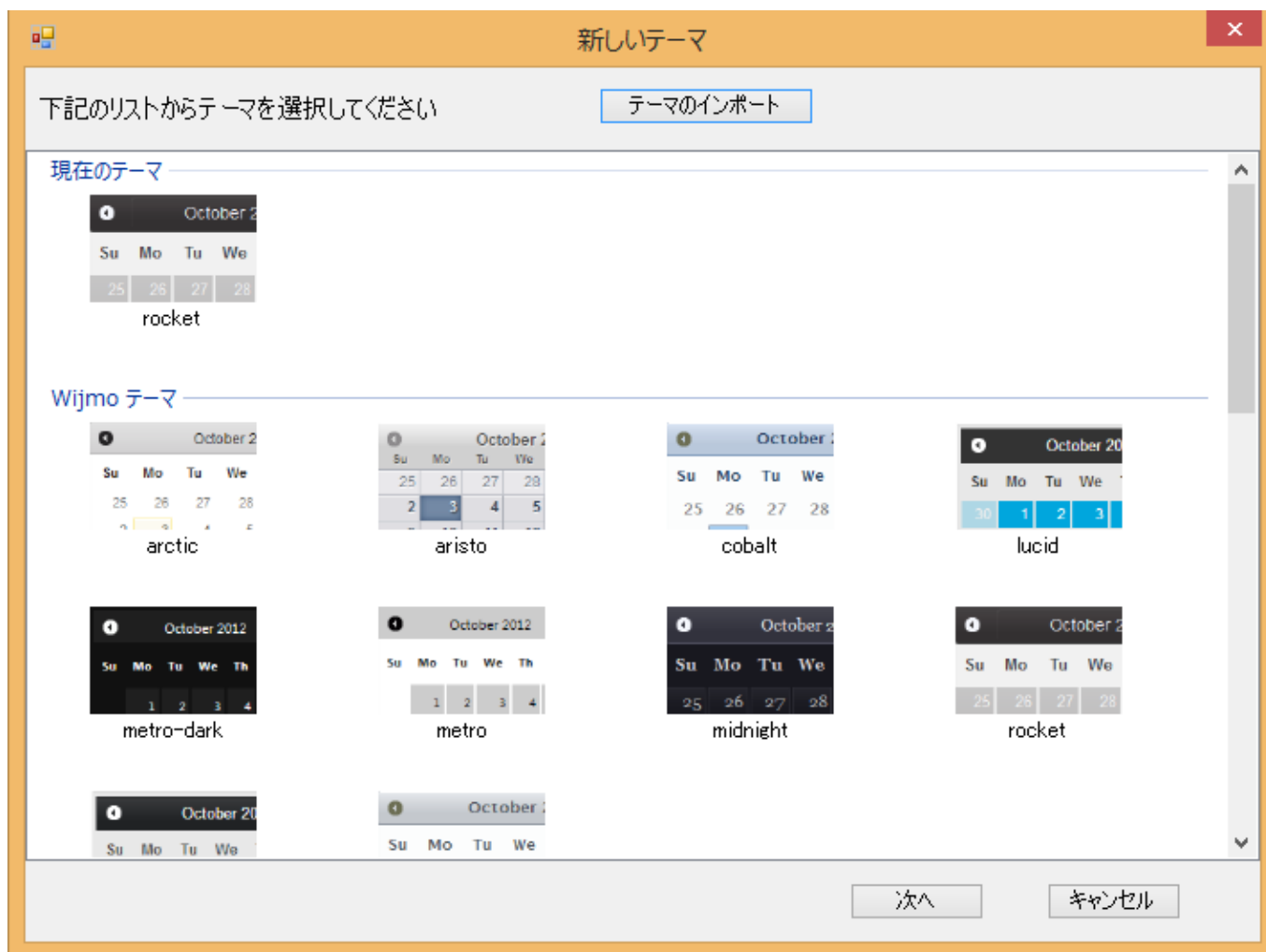
テーマ

[テーマ]ドロップダウンボックスで **Theme** プロパティを設定して、コントロールの外観を定義済みのテーマのいずれかに変更できます。

新しいテーマの作成

このリンクをクリックすると、**新しいテーマフォーム**が開き、既存のテーマを使用して新しいテーマを作成したり、テーマをインポートすることができます。

SiteMap for ASP.NET Web Forms



Themeroller を使用した新しいテーマの作成方法については、「[Themeroller for Visual Studio](#)」を参照してください。

CDN の使用

[**CDN の使用**]チェックボックスをオンにすると、クライアントリソースが CDN からロードされます。デフォルトでは、このチェックボックスはオフです。

Bootstrap の使用

[**Bootstrap の使用**]チェックボックスをオンにすると、コントロールに Bootstrap テーマが適用されます。

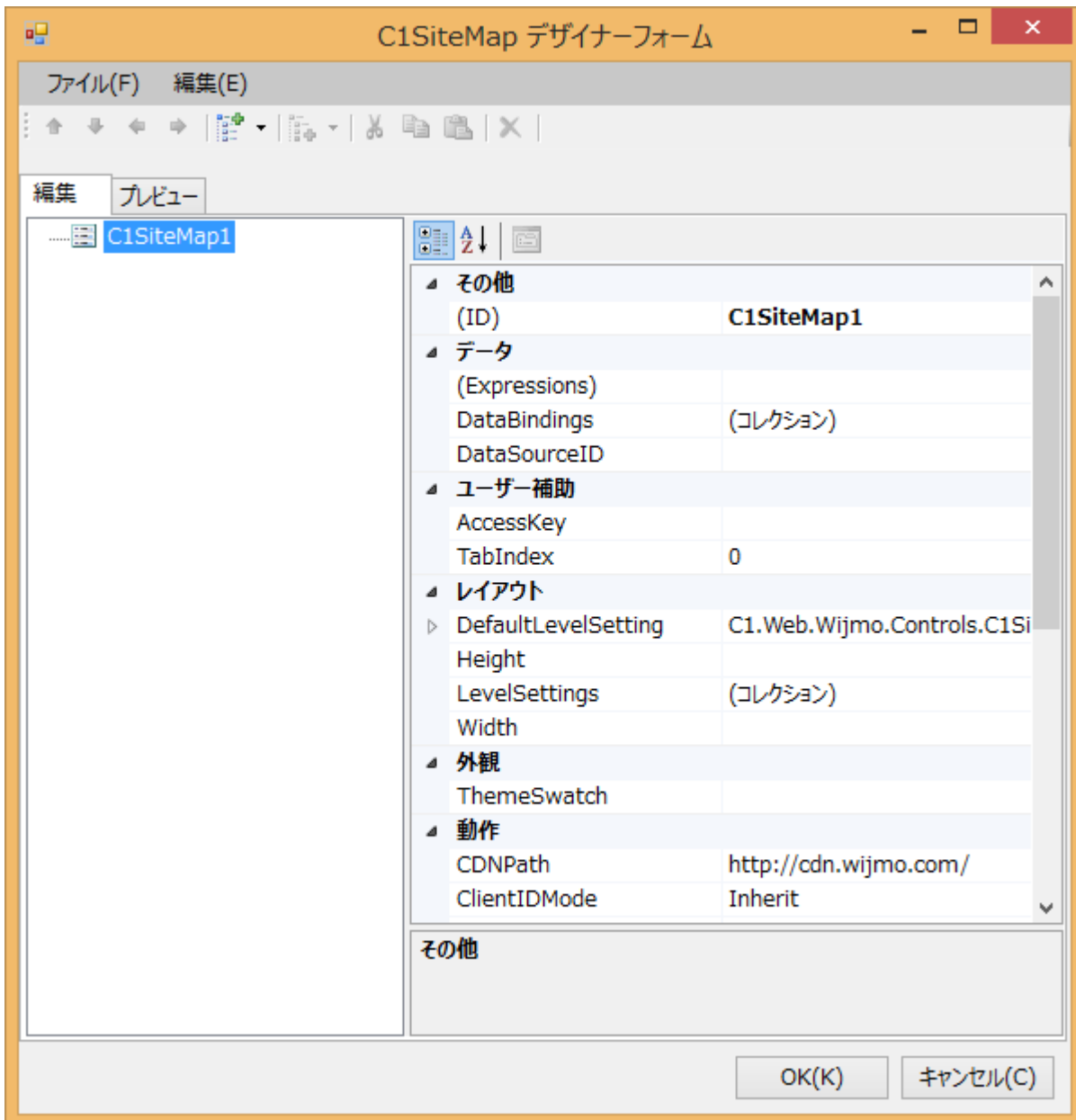
バージョン情報

[**バージョン情報**]項目をクリックすると、[ComponentOne for ASP.NET Web Forms のバージョン情報]ダイアログボックスが表示され、ComponentOne for ASP.NET Web Forms のバージョン番号とオンラインリソースが表示されます。

[先頭に戻る](#)

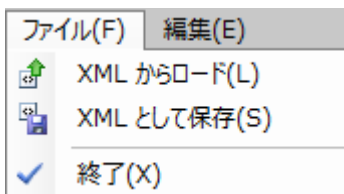
C1SiteMap デザイナフォーム

このフォームでは、**SiteMap** コントロールにノードを追加したり、コントロール内の既存のノードを編集することができます。



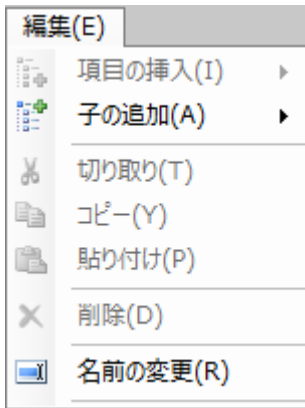
[ファイル]メニュー

[ファイル]メニューでは、既存の XML ファイルからデータをロードしたり、XML ファイルにデータを保存したり、デザイナを終了することができます。



[編集]メニュー

[編集]メニューでは、ノードを追加したり、選択したノードに子ノードを追加することができます。また、切り取り、コピー、貼り付け、削除、名前の変更などの基本編集機能も実行できます。



次の表で、**C1SiteMap デザイナフォーム**の各ボタンの機能について説明します。

アイコン	ボタン	説明
	項目を上に移動	選択した項目を1ノード上に移動します。
	項目を下に移動	選択した項目を1ノード下に移動します。
	項目を左に移動	選択した項目を左に移動します。
	項目を右に移動	選択した項目を右に移動します。
	子項目の追加	選択したノードに子項目を追加します。
	項目の挿入	選択したノードの兄弟ノードを追加します。
	切り取り/コピー/貼り付け/削除	選択したノードの切り取り、コピー、貼り付け、または削除を実行できます。

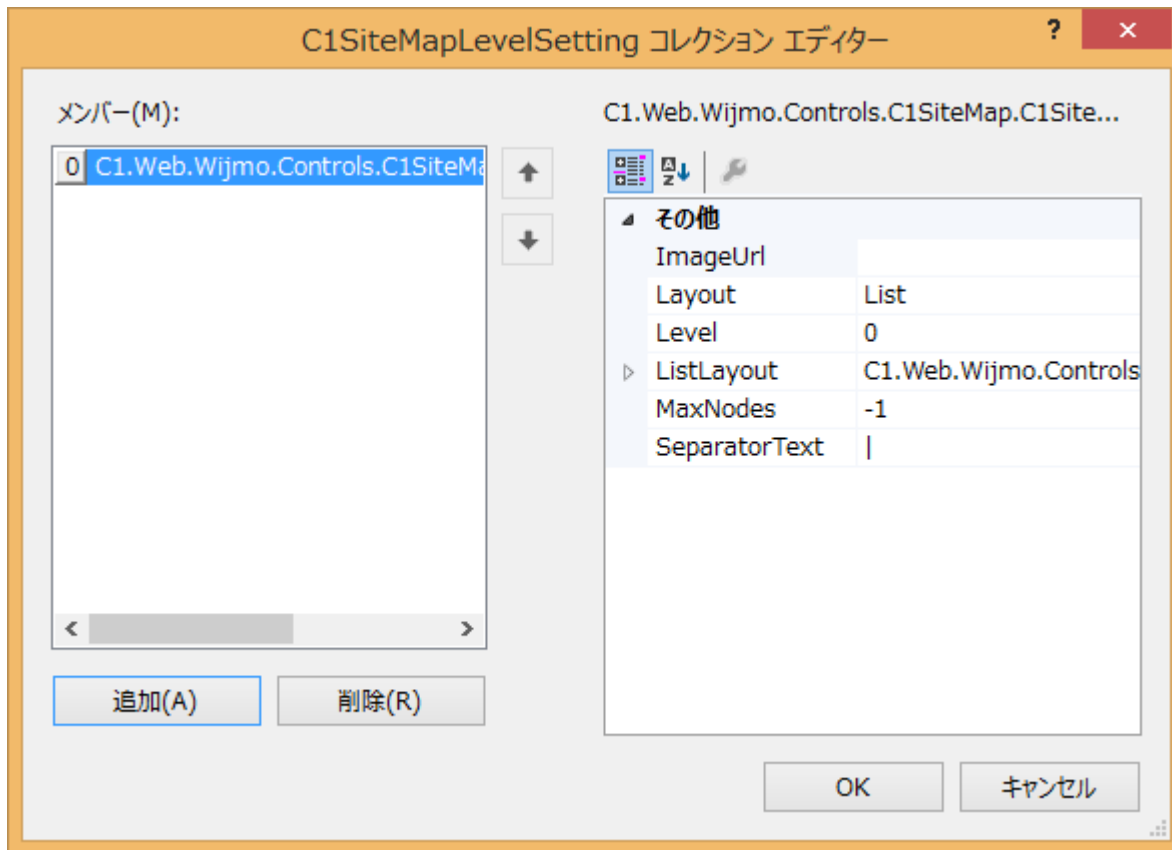
[プレビュー]タブ

[プレビュー]タブでは、ノードが出力にどのように表示されるかを確認できます。

[先頭に戻る](#)

C1SiteMapLevelSettings コレクションエディタ

C1SiteMapLevelSetting コレクションエディタでは、さまざまなレベルのノードの設定をカスタマイズできます。



[追加]ボタンは、新しいレベル設定を追加します。

[削除]ボタンは、選択した設定を削除します。

クイックスタート: SiteMap の作成とカスタマイズ

ここでは、次の手順を実行して、**SiteMap** コントロールを使用する簡単なアプリケーションを作成する方法を学びます。

以下の手順は、Visual Studio 2012 で作成されたアプリケーション用です。使用する Visual Studio のバージョンによっては、手順が多少異なる場合があります。



トピックの内容

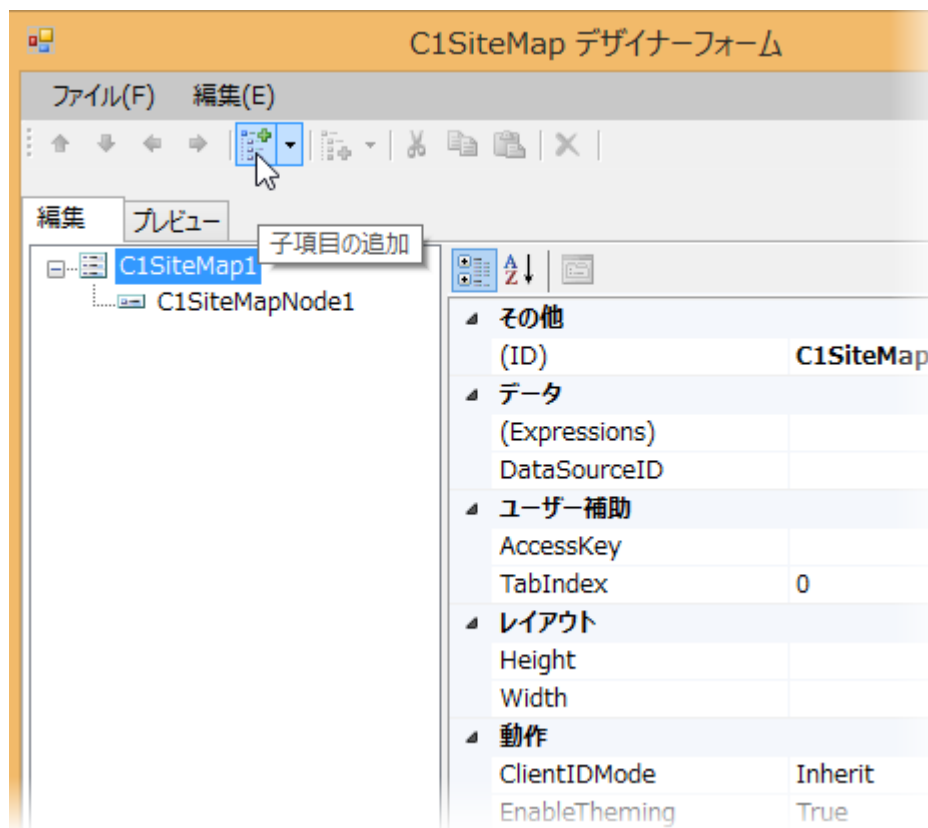
- 手順1/3: Web フォームへの SiteMap の追加
- 手順2/3: SiteMap へのノードの追加
- 手順3/3: 外観とレイアウトのカスタマイズ

手順1/3: Web フォームへの SiteMap の追加

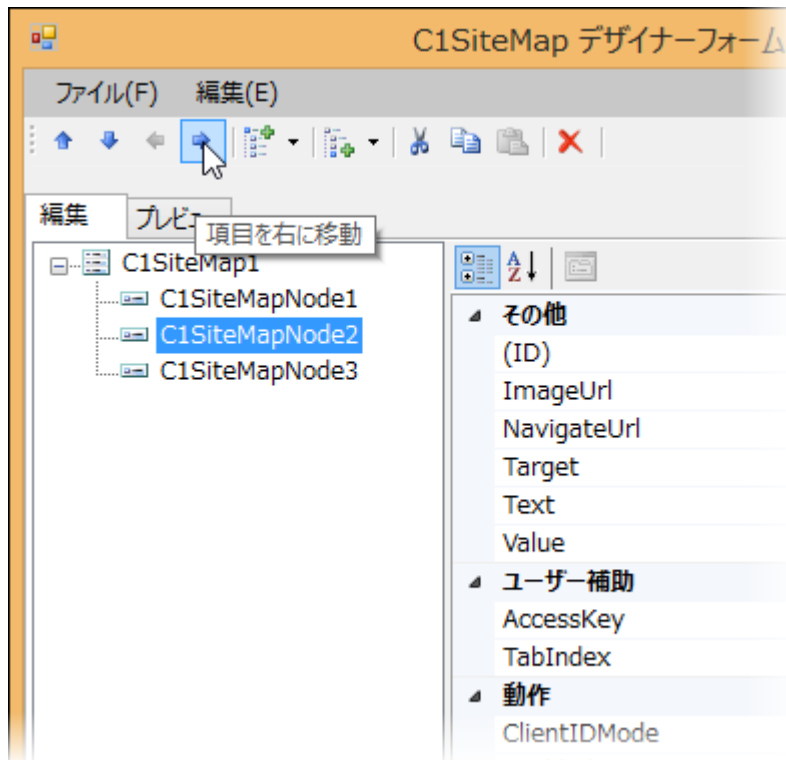
1. Visual Studio で、新しい **ASP.Net Web アプリケーション** を作成し、新しい **Web フォーム** を追加します。
2. ツールボックスで **SiteMap** コントロールを見つけ、Web フォームにドラッグします。ツールボックスでコントロールが見つからない場合は、右クリックして **[項目の選択]** を選択し、**[ツールボックス項目の選択]** ダイアログボックスで **SiteMap** を見つけます。
3. さらに、サイトマップ内でリンクする Web ページごとに1つ、合計3つの Web フォームを追加します。

手順2/3: SiteMap へのノードの追加

1. SiteMap コントロールを選択し、スマートタグ  をクリックして **[SiteMap のタスク]** メニューを開きます。
2. **[ノードの編集]** をクリックします。これにより、**C1SiteMap デザイナーフォーム** が開きます。
3. **[子項目の追加]** ボタン  を3回クリックして、3つの子ノードを追加します。



4. **C1SiteMapNode2** をクリックし、**[項目を右に移動]** ボタンをクリックします。



5. **C1SiteMapNode1** を選択します。プロパティウィンドウの右側で、**NavigateUrl** プロパティを Webform2.aspx に設定します。同様に、**C1SiteMapNode2** と **C1SiteMapNode3** の **NavigateUrl** プロパティをそれぞれ Webform3.aspx と Webform4.aspx に設定します。
6. [F5]キーを押してプロジェクトを実行します。

ソースビューの場合

次のマークアップを `<cc1:C1SiteMap></cc1:C1SiteMap>` タグ内に追加して、**SiteMap** コントロールにいくつかのノードを追加します。

ASP.Net

```
<Nodes>
  <cc1:C1SiteMapNode runat="server" Text="C1SiteMapNode1" NavigateUrl="Web
Form2.aspx">
    <Nodes>
      <cc1:C1SiteMapNode runat="server" NavigateUrl="Web Form3.aspx"
Text="C1SiteMapNode2">
        </cc1:C1SiteMapNode>
      </Nodes>
    </cc1:C1SiteMapNode>
    <cc1:C1SiteMapNode runat="server" Text="C1SiteMapNode3" NavigateUrl="Web
Form4.aspx">
      </cc1:C1SiteMapNode>
    </Nodes>
  </Nodes>
```

コードの場合

次のコードを **Page_Load** イベントに追加して、ノードを SiteMap コントロールに追加します。

C# でコードを書く場合

SiteMap for ASP.NET Web Forms

C#

```
C1SiteMap1.Nodes.Add(new C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapNode());
C1SiteMap1.Nodes.Add(new C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapNode());
C1SiteMap1.Nodes[0].Nodes.Add(new C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapNode());
C1SiteMap1.Nodes[0].Text = "C1SiteMapNode1";
C1SiteMap1.Nodes[1].Text = "C1SiteMapNode1";
C1SiteMap1.Nodes[0].Nodes[0].Text = "C1SiteMapNode1Child1";
```

VB でコードを書く場合

VB

```
C1SiteMap1.Nodes.Add(New C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapNode())
C1SiteMap1.Nodes.Add(New C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapNode())
C1SiteMap1.Nodes(0).Nodes.Add(New C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapNode())
C1SiteMap1.Nodes(0).Text = "C1SiteMapNode1"
C1SiteMap1.Nodes(1).Text = "C1SiteMapNode1"
C1SiteMap1.Nodes(0).Nodes(0).Text = "C1SiteMapNode1Child1"
```

ここまでの成果


プロジェクトを実行すると、C1SiteMapNode が次の図のように表示されます。**SiteMapNode1** と **SiteMapNode3** はレベル 0、**SiteMapNode2** はレベル1のノードです。

```
C1SiteMapNode1
  C1SiteMapNode2
C1SiteMapNode3
```

手順3/3: 外観とレイアウトのカスタマイズ

デザイナーの場合

1. [SiteMap のタスク]メニューを開き、[ノードの編集]を選択します。
2. C1SiteMap デザイナフォームで、C1SiteMap1 を選択し、右側で **CssClass** と **ThemeSwatch** を自由に設定します。
3. **Layout** プロパティグループを展開し、**DefaultLevelSetting** プロパティグループを展開し、**Layout** を List に設定します。

 **メモ:** プロジェクトに追加した画像を **ImageUrl** プロパティで選択することで、選択したレベルに画像を追加することもできます。

4. **ListLayout** プロパティグループを展開し、**RepeatColumns** を2に設定します。

さまざまなレベルのノードのレイアウトを変更する方法として、**C1SiteMapLevelSetting コレクションエディタ**を使用することもできます。**C1SiteMapLevelSetting コレクションエディタ**を使用してレイアウトを変更するには、次の手順に従います。

1. **SiteMap** コントロールを右クリックし、[プロパティ]を選択します。
2. **プロパティウィンドウ**で、**LevelSettings** プロパティの横にある省略符ボタン(...)をクリックします。これで、**C1SiteMapLevelSetting コレクションエディタ**が開きます。
3. **C1SiteMapLevelSetting コレクションエディタ**で、[追加]ボタンをクリックして、新しいレベル設定を追加します。
4. 右側で、**Level** を1、**Layout** を Flow、**SeparatorText** を「,」に設定します。

ソースビューの場合

次のマークアップを <cc1:C1SiteMap></cc1:C1SiteMap>タグ内に追加して、レベル0(デフォルトレベル)ノードとレベル1ノードのレイアウトを変更します。

ASP.Net

```
<LevelSettings>
  <cc1:C1SiteMapLevelSetting Level="1" SeparatorText="," Layout="Flow">
  </cc1:C1SiteMapLevelSetting>
</LevelSettings>
<DefaultLevelSetting SeparatorText=" ||">
  <ListLayout RepeatColumns="2" />
</DefaultLevelSetting>
```

コードの場合

Page_Load イベントに次のコードを追加して、DefaultLevel 設定を変更し、新しいレベル設定を追加します。

C# でコードを書く場合

C#

```
// 新しい LevelSetting を追加します
C1SiteMap1.LevelSettings.Add(new
C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapLevelSetting());

// LevelSetting をカスタマイズします
C1SiteMap1.LevelSettings[0].Level = 1;
C1SiteMap1.LevelSettings[0].Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.Flow;
C1SiteMap1.LevelSettings[0].SeparatorText = ",";

// DefaultLevelSetting をカスタマイズします
C1SiteMap1.DefaultLevelSetting.Level = 0;
C1SiteMap1.DefaultLevelSetting.Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.List;
C1SiteMap1.DefaultLevelSetting.ListLayout.RepeatColumns = 2;
```

Visual Basic でコードを書く場合

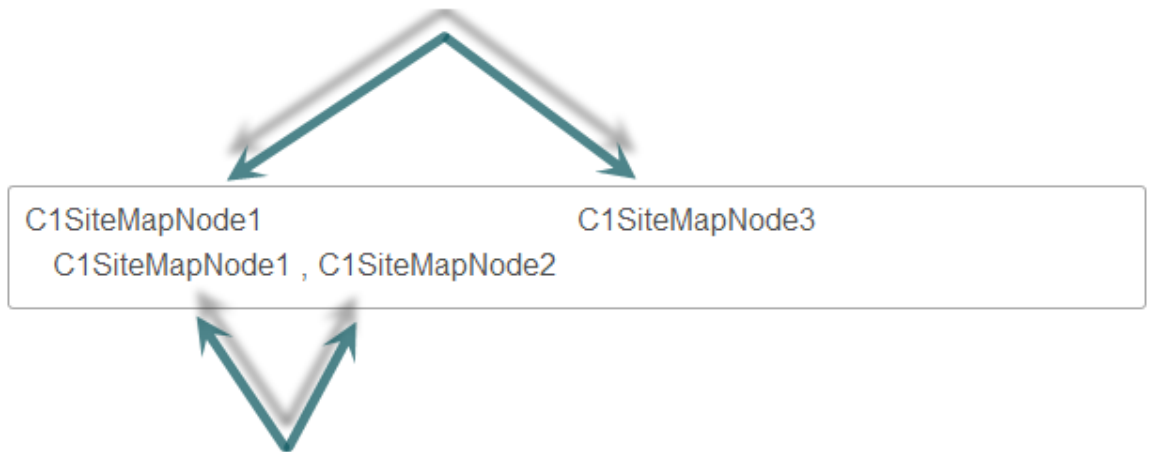
VB

```
' 新しい LevelSetting を追加します
C1SiteMap1.LevelSettings.Add(New
C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapLevelSetting())
' LevelSetting をカスタマイズします
C1SiteMap1.LevelSettings(0).Level = 1
C1SiteMap1.LevelSettings(0).Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.Flow
C1SiteMap1.LevelSettings(0).SeparatorText = ","
' DefaultLevelSetting をカスタマイズします
C1SiteMap1.DefaultLevelSetting.Level = 0
C1SiteMap1.DefaultLevelSetting.Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.List
C1SiteMap1.DefaultLevelSetting.ListLayout.RepeatColumns = 2
```

ここまでの成果

[F5]を押してプロジェクトを実行し、すべてのレベル1ノードがセパレータを使用したフローレイアウトで表示され、すべてのレベル0ノードが2つの列のリストとして表示されることを確認します。

Level 0 のノードが2列で一覧として表示されます。



Level 1 のノードがコンマ区切りで表示されます

主な機能

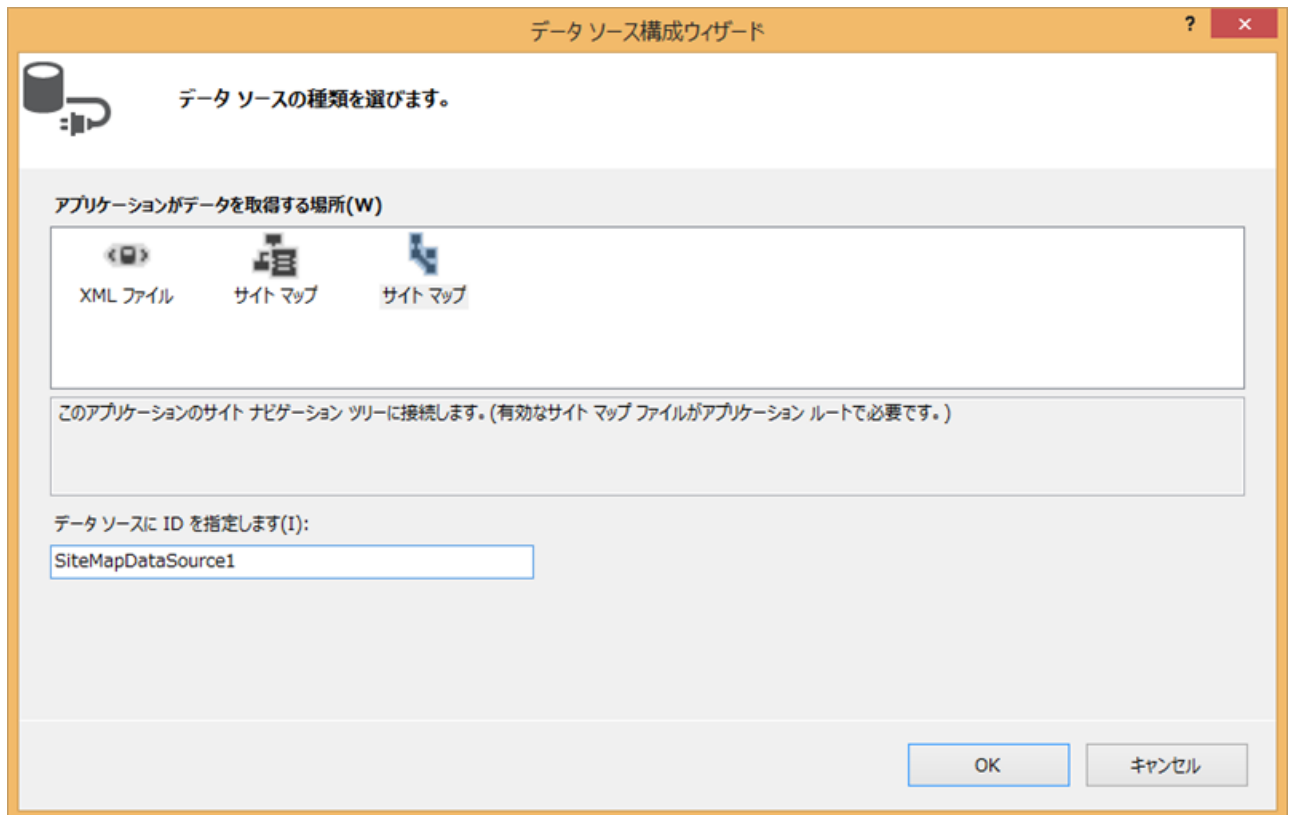
データ連結

このトピックでは、**SiteMap** を **SiteMapDataSource**、**XMLDataSource**、**C1SiteMapDataSource** の各コントロールに連結する方法を示します。

デザイナーの場合

SiteMapDataSource コントロールを使用してコントロールにデータを設定するには、次の手順に従います。

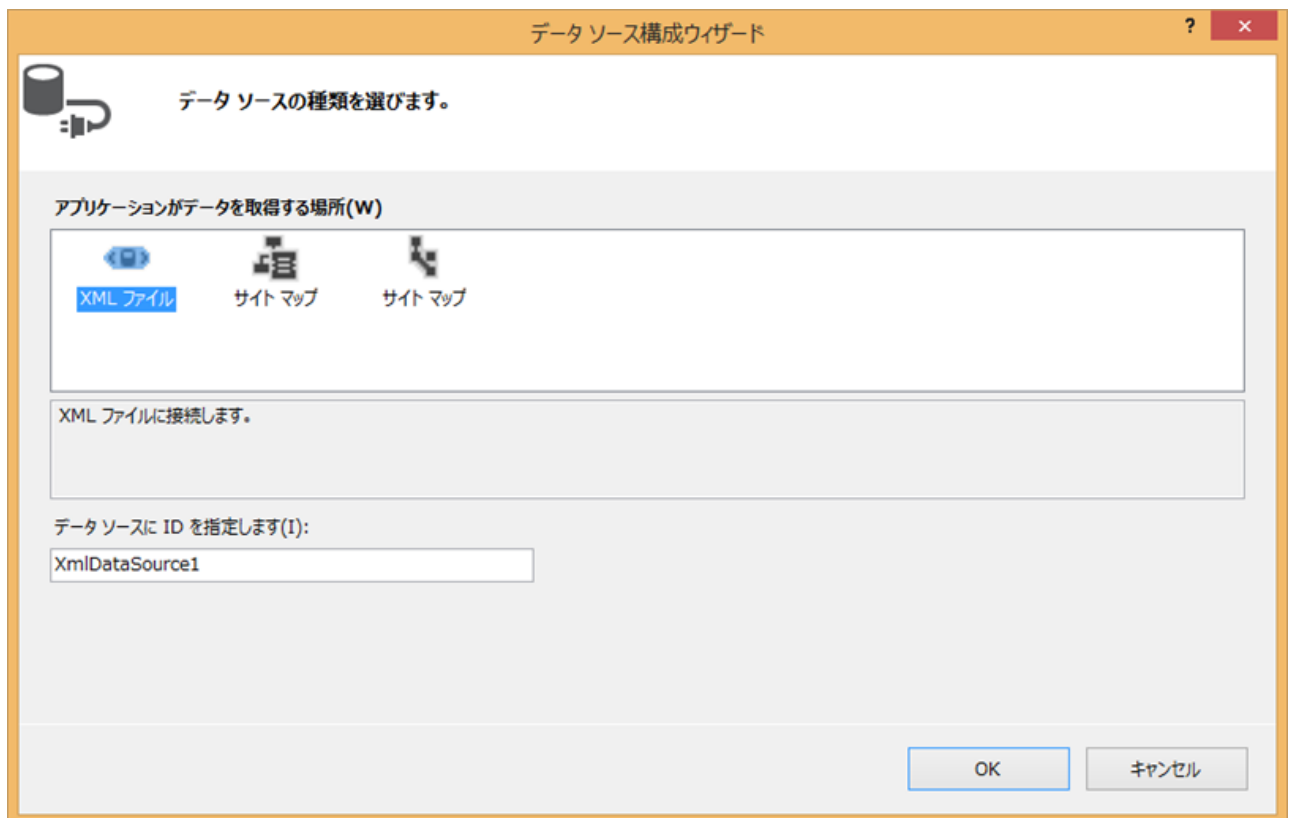
1. SiteMap コントロールを選択し、スマートタグをクリックして **[SiteMap のタスク]メニュー** を開きます。
2. **[データソースの選択]** ドロップダウンを開き、**<新しいデータソース>** を選択します。これにより、ID フィールドに SiteMapDataSource1 が定義された状態で **データソース構成ウィザード** が開きます。



3. **[サイトマップ]** を選択し、**[OK]** をクリックします。これにより、**SiteMapDataSource** コントロールが Web フォームに追加されます。
4. **SiteMapDataSource** コントロールを右クリックし、**[プロパティ]** をクリックします。
5. **プロパティウィンドウ** で、**SiteMapProvider** の名前を追加します。カスタム SiteMapProvider の作成方法については、**「カスタム SiteMap プロバイダ」** を参照してください。

XML データ を使用してコントロールにデータを設定するには、次の手順に従います。

1. SiteMap コントロールを選択し、スマートタグをクリックして **[SiteMap のタスク]メニュー** を開きます。
2. **[データソースの選択]** ドロップダウンを開き、**<新しいデータソース>** を選択します。これにより、**データソース構成ウィザード** が開きます。**[XML ファイル]** を選択します。



3. 表示される[データソースの構成]ダイアログボックスの[データファイル]から XML ファイルを選択し、[OK]をクリックします。

DialogBoxDataSource コントロールを使用してコントロールにデータを設定するには、次の手順に従います。

1. フォームに **C1SiteMapDataSource** コントロールを追加します。
2. **C1SiteMapDataSource** コントロールを右クリックし、[プロパティ]を選択します。
3. **プロパティウィンドウ**で、**SiteMapFile** プロパティに *.sitemap* ファイルを追加します。
4. [SiteMap のタスク]メニューを開きます。
5. [データソースの選択]ドロップダウンを開き、**C1SiteMapDataSource1** を選択します。

ソースビューの場合

SiteMapDataSource コントロールを使用して、**SiteMap** コントロールを連結するには、次の手順に従います。

1. 次のマークアップを `<form></form>` タグ内に追加します。

ソースビュー

```
<asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server"
SiteMapProvider="C1CustomSiteMapProvider" />
```

2. `<c1:C1SiteMap>` タグ内で **DatasourceID** を設定します。

ソースビュー

```
<c1:C1SiteMap ID="C1SiteMap1" runat="server"
DataSourceID="SiteMapDataSource1">
```

SiteMap コントロールを **XML データ** に連結するには、次の手順に従います。

1. 次のマークアップを `<form></form>` タグ内に追加します。

ソースビュー

```
<asp:XmlDataSource ID="XmlDataSource1" runat="server"
```



```
DataFile="~/App_Data/Menu_Structure.xml"></asp:XmlDataSource>
```

2. <cc1:C1SiteMap> タグ内で **DatasourceID** を設定します。

ソースビュー

```
<cc1:C1SiteMap ID="C1SiteMap1" runat="server"
DataSourceID="XmlDataSource1">
```

C1SiteMapDataSource コントロールを使用して、SiteMap コントロールを連結するには、次の手順に従います。

1. 次のマークアップを <form></form> タグ内に追加します。

ソースビュー

```
<cc1:C1SiteMapDataSource ID="C1SiteMapDataSource1" runat="server"
SiteMapFile="~/web1.sitemap" />
```

2. <cc1:C1SiteMap> タグ内で **DatasourceID** を設定します。

ソースビュー

```
<cc1:C1SiteMap ID="C1SiteMap1" runat="server"
DataSourceID="C1SiteMapDataSource1">
```

カスタム SiteMap プロバイダ

カスタム SiteMap プロバイダを作成するには、StaticSiteMapProvider クラスを継承し、要件を実装します。次の2つのメソッドをオーバーライドします。

- BuildSiteMap(): SiteMap を構築します。
- GetRootNodeCore(): 構築したサイトマップのルートノードを返します。

SiteMap プロバイダの作成

次のコードは、単純なサイトマッププロバイダの作成方法を示します。

C# でコードを書く場合

C#

```
public partial class C1CustomSiteMapProvider : StaticSiteMapProvider
{
    private readonly object C1siteMapLock = new object();
    private SiteMapNode C1SiteMapRoot;

    // BuildSiteMap() メソッドをオーバーライドします
    public override SiteMapNode BuildSiteMap()
    {
        // ロックを使用してスレッドセーフを提供します
        lock (C1siteMapLock)
        {
            if (C1SiteMapRoot != null)
            {
                return C1SiteMapRoot;
            }
            base.Clear();
            CreateSiteMapRoot();
        }
    }
}
```

SiteMap for ASP.NET Web Forms

```
        CreateSiteMapNodes();
        return C1SiteMapRoot;
    }
}

// GetRootNodeCore() メソッドをオーバーライドします
protected override SiteMapNode GetRootNodeCore()
{
    return BuildSiteMap();
}

private void CreateSiteMapRoot()
{
    C1SiteMapRoot = new SiteMapNode(this, "C1RootNode", "~/Default.aspx",
"C1RootNode");
    AddNode(C1SiteMapRoot);
}

private void CreateSiteMapNodes()
{
    SiteMapNode node = null;
    for (int num = 1; num <= 3; num++)
    {
        node = new SiteMapNode(this,
            string.Format("C1ChildNode{0}", num),
            string.Format("~/WebForm{0}.aspx", num),
            string.Format("C1ChildNode{0}", num));
        AddNode(node, C1SiteMapRoot);
    }
}
}
```

Visual Basic でコードを書く場合

VB

```
Public Class C1CustomSiteMapProvider

    Inherits StaticSiteMapProvider
    Private ReadOnly C1siteMapLock As New Object()
    Private C1SiteMapRoot As SiteMapNode

    ' BuildSiteMap() メソッドをオーバーライドします
    Public Overrides Function BuildSiteMap() As SiteMapNode
        ' ロックを使用してスレッドセーフを提供します
        SyncLock C1siteMapLock
            If C1SiteMapRoot IsNot Nothing Then
                Return C1SiteMapRoot
            End If
            MyBase.Clear()
            CreateSiteMapRoot()
            CreateSiteMapNodes()
            Return C1SiteMapRoot
        End SyncLock
    End Function
End Class
```

```

End Function

' GetRootNodeCore() メソッドをオーバーライドします
Protected Overrides Function GetRootNodeCore() As SiteMapNode
    Return BuildSiteMap()
End Function

Private Sub CreateSiteMapRoot()
    C1SiteMapRoot = New SiteMapNode(Me, "C1RootNode", "~/Default.aspx",
"C1RootNode")
    AddNode(C1SiteMapRoot)
End Sub
Private Sub CreateSiteMapNodes()
    Dim node As SiteMapNode = Nothing
    For num As Integer = 1 To 3
        node = New SiteMapNode(Me, String.Format("C1ChildNode{0}", num),
String.Format("~/WebForm{0}.aspx", num),
String.Format("C1ChildNode{0}", num))
        AddNode(node, C1SiteMapRoot)
    Next
End Sub
End Class

```

SiteMap プロバイダの配布

カスタム SiteMap プロバイダは、*web.config* ファイルに登録する必要があります。<system.web></system.web> タグ内に次のマークアップを追加します。

マークアップ

```

<siteMap defaultProvider="C1CustomSiteMapProvider" enabled="true">
  <providers>
    <clear/>
    <add name="C1CustomSiteMapProvider"
type="WebApplication1.C1CustomSiteMapProvider"/>
  </providers>
</siteMap>

```

ツールチップの追加

SiteMap では、各ノードにツールチップを追加できます。

デザイナーの場合

1. SiteMap コントロールを右クリックし、[プロパティ]を選択します。
2. SiteMap コントロール全体の ToolTip プロパティにテキストを追加します。
3. [SiteMap のタスク]メニューを開き、[ノードの編集]を選択します。これにより、C1SiteMap デザイナーフォームが開きます。

任意のノードを選択し、各ノードの ToolTip プロパティを設定します。

ソースビューの場合

SiteMap for ASP.NET Web Forms

次のように、<cc1:C1SiteMap> タグ内の **ToolTip** プロパティを設定して、コントロール全体にツールチップを追加します。

ASP.Net

```
<cc1:C1SiteMap ID="C1SiteMap1" runat="server"
ToolTip="This is a SiteMap Control" ShowNodeLines="true"
DataSourceID="C1SiteMapDataSource1">
```

次のように、<cc1:C1SiteMapNode> タグ内の **ToolTip** プロパティを設定して、1つのノードにツールチップを追加します。

ASP.Net

```
<cc1:C1SiteMapNode runat="server" Text="C1SiteMapNode1"
ToolTip="This is a Node Tooltip">
```

コードの場合

コントロール全体にツールチップを追加するには、**Page_Load** イベントに次のコードを追加します。

C# でコードを書く場合

C#

```
C1.SiteMap1.ToolTip = "SiteMap Control";
```

Visual Basic でコードを書く場合

VB

```
C1.SiteMap1.ToolTip = "SiteMap Control"
```

個々のノードにツールチップを追加するには、**Page_Load** イベントに次のコードを追加します。

C# でコードを書く場合

C#

```
C1SiteMap1.Nodes[0].ToolTip = "Parent1";
C1SiteMap1.Nodes[0].Nodes[0].ToolTip = "child1";
C1SiteMap1.Nodes[1].ToolTip = "parent2";
```

Visual Basic でコードを書く場合

VB

```
C1SiteMap1.Nodes(0).ToolTip = "Parent1"
C1SiteMap1.Nodes(0).Nodes(0).ToolTip = "child1"
C1SiteMap1.Nodes(1).ToolTip = "parent2"
```

ここまでの成果

プロジェクトを実行し、**SiteMap** ノードの上にマウスを置きます。ノード上にツールチップが表示されることがわかります。

```
C1SiteMapNode1
  C1SiteMapNode2
    C1SiteMapNode3
      子項目 1
```

レイアウトのカスタマイズ

SiteMap では、ノードレベルごとに異なるレイアウトを追加できます。次の2種類のレイアウトがあります。

- **リスト**: このレイアウトでは、ノードを列に分けて表示できます。**ListLayout.RepeatColumns** プロパティを変更して、レイアウトに表示される列の数を定義できます。
- **フロー**: このレイアウトでは、ノードをセパレータで区切って1行に表示できます。**SeparatorText** プロパティを変更して、セパレータを変更できます。

このトピックでは、レベル0ノードとレベル1ノードのレイアウトを変更する方法を示します。


デザイナーの場合

デザインビューでレベル0(デフォルトレベル)のレイアウトを変更するには、次の手順に従います。

1. [SiteMap のタスク]メニューを開き、[ノードの編集]を選択します。これにより、**C1SiteMap デザイナフォーム**が開きます。
2. **C1SiteMap デザイナフォーム**で、**C1SiteMap1** を選択し、**DefaultLevelSetting.Layout** を List に設定します。
3. **DefaultLevelSetting.ListLayout.RepeatColumns** を2に設定します。

デザインビューでレベル1のレイアウトを変更するには、次の手順に従います。

1. **C1SiteMap デザイナフォーム**で、**LevelSettings** プロパティの横にある省略符ボタン(...)をクリックして、**C1SiteMapLevelSettings コレクションエディタ**を開きます。
2. [追加]ボタンをクリックして、新しいレベル設定を追加します。
3. 右側で、**Level** を1、**Layout** を Flow、**SeparatorText** を「,」に設定します。
4. [OK]をクリックして **C1SiteMapLevelSettings コレクションエディタ**を閉じ、再度[OK]をクリックして **C1SiteMap デザイナフォーム**を閉じます。

 レイアウトを Flow に設定したレベルは、サイトマップの最後のレベルになります。

ソースビューの場合

ソースビューでレベル0(デフォルトレベル)のレイアウトを変更するには、`<cc1:C1SiteMap></cc1:C1SiteMap>` タグ内に次のマークアップを追加します。

```
<DefaultLevelSetting SeparatorText="|" Level="0" Layout="List">
  <ListLayout RepeatColumns="2" />
</DefaultLevelSetting>
```

ソースビューでレベル1のレイアウトを変更するには、`<cc1:C1SiteMap></cc1:C1SiteMap>` タグ内に次のマークアップを追加します。

```
<LevelSettings>
  <cc1:C1SiteMapLevelSetting Layout="Flow" Level="1" SeparatorText=",">
  </cc1:C1SiteMapLevelSetting>
</LevelSettings>
```

SiteMap for ASP.NET Web Forms

コードの場合

コードでデフォルトレベルのレイアウトを変更するには、**Page_Load** イベントに次のコードを追加します。

C# でコードを書く場合

C#

```
C1SiteMap1.DefaultLevelSetting.SeparatorText = ",";
C1SiteMap1.DefaultLevelSetting.Level = 0;
C1SiteMap1.DefaultLevelSetting.Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.List;
C1SiteMap1.DefaultLevelSetting.ListLayout.RepeatColumns = 2;
```

VB でコードを書く場合

VB

```
C1SiteMap1.DefaultLevelSetting.SeparatorText = ","
C1SiteMap1.DefaultLevelSetting.Level = 0
C1SiteMap1.DefaultLevelSetting.Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.List
C1SiteMap1.DefaultLevelSetting.ListLayout.RepeatColumns = 2
```

コードでレベル1のレイアウトを変更するには、**Page_Load** イベントに次のコードを追加します。

C# でコードを書く場合

C#

```
// 新しい LevelSetting を追加します
C1SiteMap1.LevelSettings.Add(new
C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapLevelSetting());
// LevelSetting をカスタマイズします
C1SiteMap1.LevelSettings[0].Level = 1;
C1SiteMap1.LevelSettings[0].Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.Flow;
C1SiteMap1.LevelSettings[0].SeparatorText = ",";
```

VB でコードを書く場合

VB

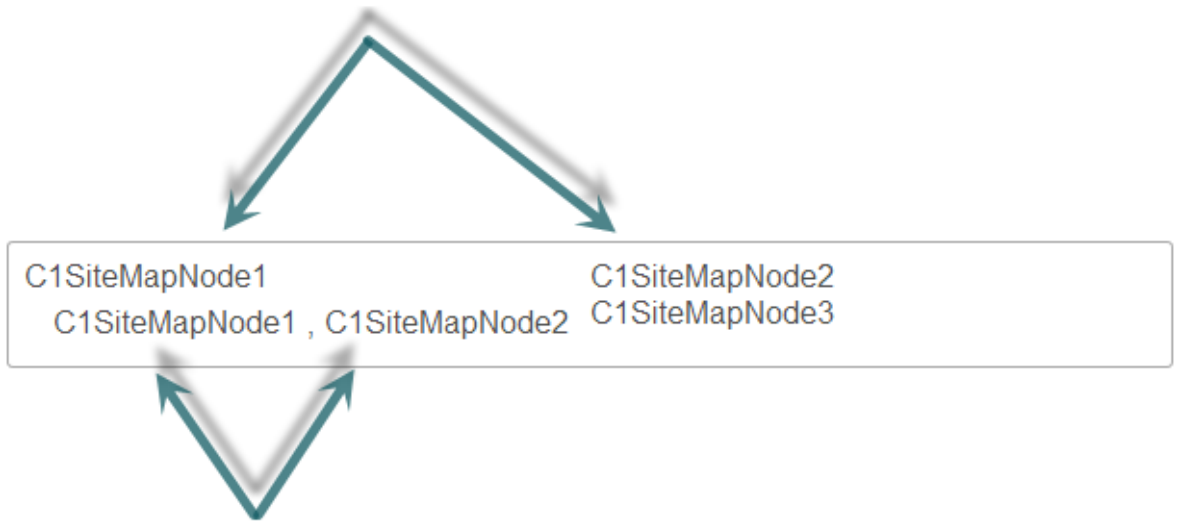
```
' 新しい LevelSetting を追加します
C1SiteMap1.LevelSettings.Add(New
C1.Web.Wijmo.Controls.C1SiteMap.C1SiteMapLevelSetting())
' LevelSetting をカスタマイズします
C1SiteMap1.LevelSettings(0).Level = 1
C1SiteMap1.LevelSettings(0).Layout =
C1.Web.Wijmo.Controls.C1SiteMap.SiteMapLayoutType.Flow
C1SiteMap1.LevelSettings(0).SeparatorText = ","
```

ここまでの成果

プロジェクトを実行すると、次の図のように、レベル0のノードは2列のリストとして表示され、レベル1のノードはカンマ「,」区切

りの1行(フロー)に表示されます。

Level 0 のノードが2列で一覧として表示されます。



Level 1 のノードがコンマ区切りで表示されます。

ノードテンプレート

ノードテンプレートを追加することで、画像を追加したり、ノードレベルのスタイルをカスタマイズすることができます。

ソースビューの場合

次のコードは、`<cc1:C1SiteMapLevelSetting></cc1:C1SiteMapLevelSetting` タグ内でノードテンプレートを作成する方法を示します。

ソースビュー

```

<LevelSettings>
  <cc1:C1SiteMapLevelSetting Level="1">
    <NodeTemplate>
      <asp:Image ID="Image1" runat="server" Width="28px" Height="26px"
        ImageUrl="~/C1SiteMap/Images/Wijmo.png" />
      <a href="#" class="template-text">
        <%# Eval("description") %>
      </a>
    </NodeTemplate>
  </cc1:C1SiteMapLevelSetting>
  <cc1:C1SiteMapLevelSetting Level="2" Layout="Flow">
</cc1:C1SiteMapLevelSetting>
</LevelSettings>
  
```

キーボードサポートの追加

`AccessKey` プロパティを変更して、`SiteMap` コントロールにキーボードショートカットを追加できます。

SiteMap for ASP.NET Web Forms

デザイナーの場合

デザインビューでキーボードサポートを追加するには、次の手順に従います。

1. [SiteMap のタスク]メニューを開き、[ノードの編集]をクリックします。これにより、C1SiteMap デザイナフォームが開きます。
2. C1SiteMap1 を選択し、右側で AccessKey プロパティを「K」などの値に設定します。
3. 左で C1SiteMapNode1 を選択し、右側で AccessKey プロパティを「N」などの値に設定します。

ソースビューの場合

SiteMap コントロールにキーボードショートカットを追加するには、<cc1:C1SiteMap> タグ内で AccessKey プロパティを設定します。

ソースビュー

```
<cc1:C1SiteMap ID="C1SiteMap1" runat="server" AccessKey="K">
```

C1SiteMapNode にキーボードショートカットを追加するには、<cc1:C1SiteMapNode> タグ内で AccessKey プロパティを設定します。

ソースビュー

```
<cc1:C1SiteMapNode runat="server" Text="C1SiteMapNode1" AccessKey="N">
```

コードの場合

コントロール全体にキーボードショートカットを追加するには、Page_Load イベントに次のコードを追加します。

C# でコードを書く場合

C#

```
C1.SiteMap1.AccessKey= "k";
```

Visual Basic でコードを書く場合

VB

```
C1.SiteMap1.AccessKey= "k"
```

個々のノードにキーボードショートカットを追加するには、Page_Load イベントに次のコードを追加します。

C# でコードを書く場合

C#

```
C1SiteMap1.Nodes[0].AccessKey= "a";  
C1SiteMap1.Nodes[0].Nodes[0].AccessKey= "b";  
C1SiteMap1.Nodes[1].AccessKey= "c";
```

Visual Basic でコードを書く場合

VB

```
C1SiteMap1.Nodes(0).AccessKey= "a"  
C1SiteMap1.Nodes(0).Nodes(0).AccessKey= "b"  
C1SiteMap1.Nodes(1).AccessKey= "c"
```


ここまでの成果


プロジェクトを実行したら、[Alt]+[K]キー(AccessKey)を押すと、SiteMap コントロールが強調表示されます。また、[Alt]キーとノードの AccessKey を押すと、特定のノードが強調表示されます。

C1SiteMapDataSourceの機能


このトピックでは、C1SiteMapDataSourceの機能について説明します。

新しい SiteMap の作成

C1SiteMapDataSource コントロールを使用して、新しい SiteMap を作成するには、次の手順に従います。**SiteMap** コントロールを **C1SiteMapDataSource** コントロールに連結して、出力を表示します。

 このトピックでは、4つの Web フォームがプロジェクトに追加されており、それぞれがサイトマップにリンクする Web ページを表していると仮定します。

デザイナーの場合

1. SiteMap コントロールを選択し、スマートタグ  をクリックして **C1SiteMapDataSource** の **タスクメニュー** を開きます。
2. **[ノードの編集]** をクリックします。これにより、**C1SiteMap デザイナーフォーム** が開きます。
3. **[子項目の追加]** ボタンを3回クリックします。
4. **C1SiteMapNode2** をクリックし、**[項目を右に移動]** ボタンをクリックします。
5. **C1SiteMapNode1** を選択します。右側で、**Url** プロパティを `Webform2.aspx` に設定します。同様に、**C1SiteMapNode2** と **C1SiteMapNode3** の **Url** プロパティをそれぞれ `Webform3.aspx` と `Webform4.aspx` に設定します。

`web.sitemap` という名前が付いた `.sitemap` ファイルが作成されて、プロジェクトフォルダに保存されます。

ここまでの成果


SiteMap コントロールを **C1SiteMapDataSource** コントロールに連結し、プロジェクトを実行すると、サイトマップノードが次の図のように表示されます。**C1SiteMapNode1** と **C1SiteMapNode3** はレベル0、**C1SiteMapNode2** はレベル1のノードです。

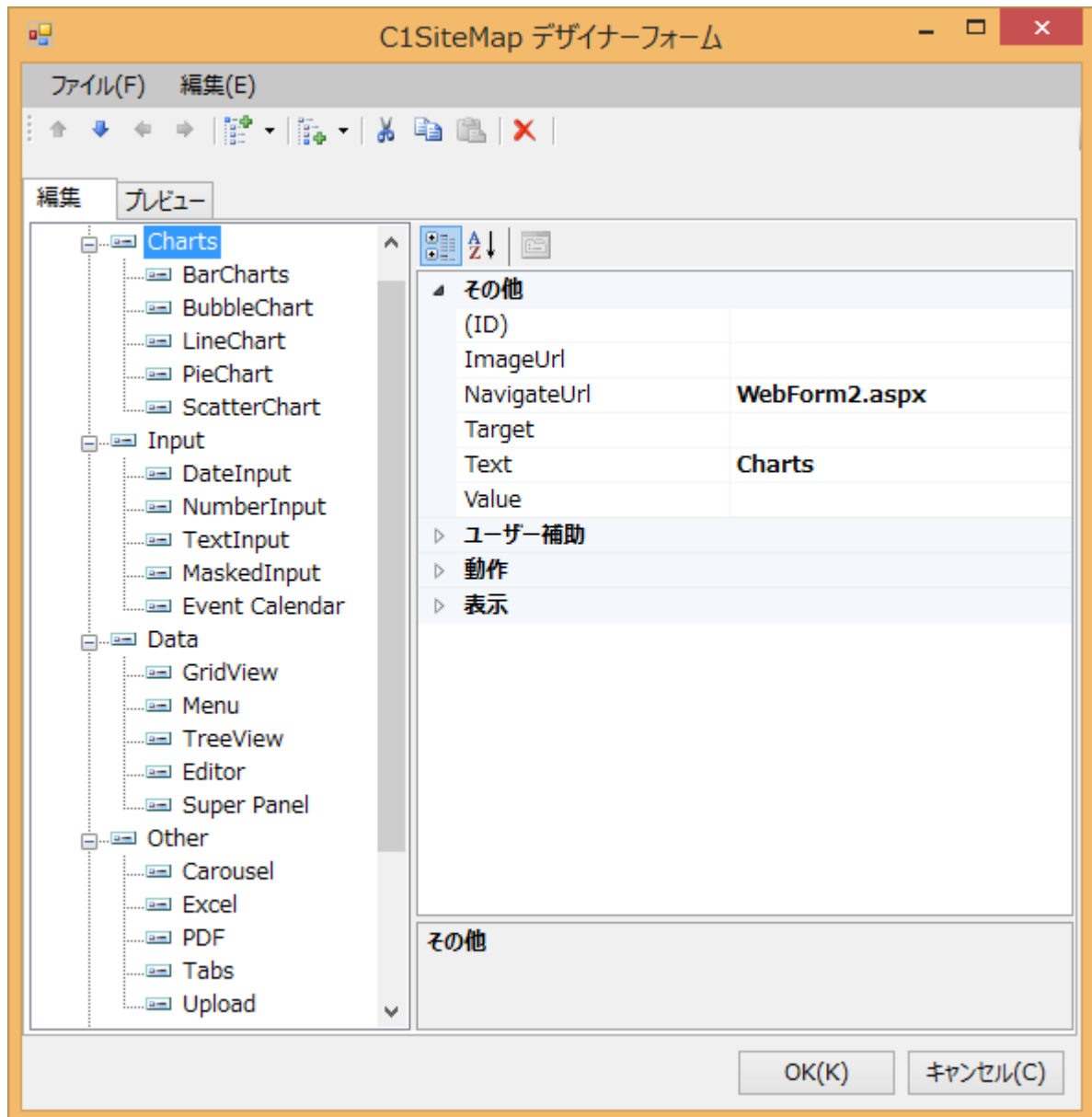
```
C1SiteMapNode1
  C1SiteMapNode2
C1SiteMapNode3
```

既存の SiteMap の編集

既存のサイトマップノードを編集するには、次の手順に従います。

デザイナーの場合

1. スマートタグ  をクリックして **C1SiteMapDataSource** の **タスクメニュー** を開きます。
2. **[既存の SiteMap の編集]** をクリックします。これにより、**C1SiteMap デザイナーフォーム** が開き、フォームの左側に既存のすべてのノードが表示されます。



3. 任意のノードを選択し、[項目を上に移動]、[項目を下に移動]、[項目を左に移動]、[項目を右に移動]の各ボタンをクリックして、選択したノードをそれぞれ上下左右に移動します。
4. [切り取り]、[コピー]、[貼り付け]、[削除]の各ボタンをクリックして、選択したノードを切り取り、コピー、貼り付け、または削除します。
5. フォームの右側にあるプロパティウィンドウから **Description**、**Title**、および **Url** を変更することもできます。

サイトマップファイルの追加

サイトマップファイルを **C1SiteMapDataSource** コントロールに追加してデータを設定できます。

デザイナの場合

.sitemap ファイルを **C1SiteMapDataSource** コントロールに追加するには、次の手順に従います。

1. プロジェクトに .sitemap ファイルを追加します。
2. **C1SiteMapDataSource** コントロールを右クリックし、[プロパティ]を選択します。
3. プロパティウィンドウで、SiteMapFile プロパティの横にある省略符ボタン(...)をクリックします。
4. [ソースサイトマップファイルの選択]ダイアログボックスで .sitemap ファイルを選択し、[OK]をクリックします。

SiteMap for ASP.NET Web Forms

ソースビューの場合

ソースビューで `.sitemap` ファイルを追加するには、`<cc1:C1SiteMapDataSource />` タグ内で `SiteMapFile` プロパティを設定します。

ソースビュー

```
<cc1:C1SiteMapDataSource ID="C1SiteMapDataSource1" runat="server"
SiteMapFile="~/web1.sitemap" />
```

コードの場合

`.sitemap` ファイルを `C1SiteMapDataSourceControl` に追加するには、**Page_Load** イベントに次のコードを追加します。

C# でコードを書く場合

C#

```
C1SiteMapDataSource1.SiteMapFile = "web1.sitemap";
```

Visual Basic でコードを書く場合

VB

```
C1SiteMapDataSource1.SiteMapFile = "web1.sitemap"
```