

ListView 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	4
ComponentOne for ASP.NET Web Forms のヘルプ	4
主な特長	5
クイックスタート	6
手順 1: ページへの C1ListView コントロールの追加	6
手順 2: C1ListViewItem の追加	6-7
手順 3: アプリケーションの実行	7-8
設計時サポート	9
C1ListView のスマートタグとタスクメニュー	9-11
C1ListView のコンテキストメニュー	11
C1ListView デザイナフォーム	11-12
C1ListView デザイナフォームの操作	12-15
C1ListView デザイナフォームメニュー	15-16
C1ListView デザイナフォームタスクバー	16-17
C1ListView デザイナフォームの使用法	17
C1ListViewItem の削除	17
C1ListViewItem の名前の変更	17
子項目の追加	17-18
C1ListViewItem の挿入	18
C1ListView の要素	19
C1ListView の項目	19
C1ListViewItem	19-20
C1ListViewButtonItem	20
C1ListViewControlGroupItem	20
normal select	20-21
radioList	21-22
checkboxList	22
C1ListViewFlipSwitchItem	22
C1ListViewInputItem	22-23
label	23
text	23

textarea	23
search	23-24
slider	24
C1ListViewLinkItem	24
C1ListViewNestedItem	24-25
C1ListViewDividerItem	25
C1ListView でサポートされているリストタイプ	26
読み取り専用リスト	26
ナビゲーションリスト	26
番号付きリスト	27
ネストされたリスト	27-28
インセットスタイルのリスト	28
フォームリスト	28-29
C1ListView の外観	30
テーマスウォッチ	30
スウォッチ a	30-31
スウォッチ b	31
スウォッチ c	31-32
スウォッチ d	32-33
スウォッチ e	33
jQuery Mobile Themeroller の概要	33-35
アイコン	35-36
タスク別ヘルプ	37
ナビゲーションリストの作成	37
フィルタ機能と検索機能の使用	38
フィルタ機能の設定	38-39
FilterText プロパティの使用	39
フィルタ機能のカスタマイズ	39-40
XML テンプレートの作成とロード	40
XML テンプレートの作成	40-41
XML テンプレートのロード	41
テーマ	41
デフォルトのスウォッチの適用	41-42

製品の概要

ListView for ASP.NET Web Forms を使用して、対話式のリストベースのモバイルアプリケーションを作成することができます。この強力なコントロールは、データ表示、ナビゲーション、検索結果などを行うリストをサポートします。最もよく使用されているデザインパターンをさまざまなリストタイプによってカバーしているため、機能豊富でタッチ操作に適したモバイルアプリケーションを作成できます。

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

C1ListView は、次のようなユニークな機能を備えています。

- **モバイルアプリケーションの最適化**
C1ListView は、特にモバイルアプリケーションの作成を目的として設計されています。このコントロールを使用して、機能豊富でタッチ操作に適したアプリケーションを作成することができます。
- **さまざまなタイプのリストを作成可能**
C1ListView コントロールは、データ表示やナビゲーションに使用できる多様なリストをサポートしています。
- **幅広い機能**
C1ListView コントロールには、単純なナビゲーションリストから複雑なフォームに至るまで、さまざまなアプリケーションを開発するための機能が多数含まれています。

クイックスタート

このクイックスタートでは、**Listview for ASP.NET Web Forms** を初めて使用するための手順について説明します。このクイックスタートでは、Visual Studio でプロジェクトを作成し、**C1ListView** コントロールをページに追加して、コントロールの外観と動作を変更します。また、**C1ListView** の実行時の動作を確認します。

手順 1: ページへの C1ListView コントロールの追加

この手順では、Visual Studio で新しいアプリケーションを作成し、**C1ListView** コントロールをページに追加します。

- Visual Studio の[ファイル]メニューから、[新規作成]→[プロジェクト]を選択します。[新しいプロジェクト]ダイアログボックスが表示されます。
 - 左ペインで言語を選択します。
 - 右ペインで[ASP.NET 空の Web アプリケーション]を選択します。
 - アプリケーションの名前を入力し、[OK]を選択します。新しいアプリケーションが作成されます。
- ソリューションエクスプローラで、References フォルダを右クリックし、[参照の追加]を選択します。
- [参照の追加]ダイアログボックスで、[参照]を選択します。**C1.Web.Wijmo.Controls.dll** アセンブリと **C1.Web.Wijmo.Controls.Design.dll** アセンブリを見つけて選択し、[OK]をクリックします。
- デザインビューで、Visual Studio のツールボックスにある **C1ListView** コントロールを見つけ、ダブルクリックしてページに追加します。マークアップは次のようになります。


デザインビュー

```
<cc1:C1ListView></cc1:C1ListView>
```

この手順では、新しい Visual Studio プロジェクトを作成し、参照アセンブリをプロジェクトに追加し、**C1ListView** コントロールをアプリケーションに追加しました。次の手順では、さまざまなタイプの**C1ListViewItem** を **C1ListView** コントロールに追加します。

手順 2: C1ListViewItem の追加

この手順では、さまざまなタイプの**C1ListViewItem** を既存の **C1ListView** コントロールに追加します。

- `<cc1:C1ListView>` 開始タグをクリックして、コントロールのスマートタグ  を表示します。
- スマートタグをクリックして[C1ListView のタスク]メニューを開き、[モバイルモード]チェックボックスをオンにします。
- カーソルを `<cc1:C1ListView>` `</cc1:C1ListView>` タグの間に挿入します。
- このタグの間に次のマークアップを追加します。

スマートタグ

```
<Items> </Items>
```

- `<Items>` タグの間に、次のマークアップを追加します。

スマートタグ

```
<cc1:C1ListViewItem Text="項目 1" DataIcon="alert"></cc1:C1ListViewItem>
```

これで、汎用の **C1ListViewItem** がアプリケーションに追加されます。

- 次に、**C1ListViewNestedItem** をアプリケーションに追加します。これで、アプリケーションにサブメニューが作成されます。

スマートタグ

```
<cc1:C1ListViewNestedItem Text="他のオプション" DataIcon="arrow-1">
  <Items>
```

Listview for ASP.NET Web Forms

```
<cc1:C1ListViewButtonItem Text="ここ！"></cc1:C1ListViewButtonItem>
<cc1:C1ListViewDividerItem></cc1:C1ListViewDividerItem>
<cc1:C1ListViewItem Text="これは別の項目です"></cc1:C1ListViewItem>
<Items>
</cc1:C1ListViewNestedItem>
```

7. C1ListViewButtonItem をアプリケーションに追加するために、次のマークアップを追加します。

スマートタグ

```
<cc1:C1ListViewButtonItem Text="クリック！"></cc1:C1ListViewButtonItem>
```

8. C1ListViewDividerItem は、リストに分割線を追加します。次のマークアップでは、**C1ListView** コントロール内で簡単に識別できるように、Text プロパティを Divider に設定しています。

スマートタグ

```
<cc1:C1ListViewDividerItem Text="仕切り"></cc1:C1ListViewDividerItem>
```

9. 最後に、C1ListViewLinkItem と C1ListViewInputItem の2つの **C1ListViewItem** を追加します。C1ListViewLinkItem を使用して、外部 Web サイトへのリンクを **C1ListView** コントロールに埋め込むことができます。C1ListViewInputItem は "textarea" に設定します。これで、データに合わせてサイズが自動的に変更される編集可能なテキストボックスになります。

スマートタグ

```
<cc1:C1ListViewLinkItem Text="リンクへ移動"
NavigateUrl="http://www.componentone.com/"></cc1:C1ListViewLinkItem>

<cc1:C1ListViewInputItem Type="textarea" LabelText="ここでテキストを入力してください:">
</cc1:C1ListViewInputItem>
```

10. この手順では、さまざまなタイプの **C1ListViewItem** を **C1ListView** コントロールに追加しました。次の手順では、アプリケーションを実行して、**C1ListView** コントロールの実行時の動作を確認します。

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、**C1ListView** コントロールの実行時の動作を確認します。

1. [F5]キーを押してアプリケーションを実行します。アプリケーションは次の図のようになります。



2. いくつかの **C1ListViewItem** を試してみます。C1ListViewInputItem にテキストを入力すると、textarea タイプの入力領域のサイズがテキストに合わせて自動的に変更されます。
 - a. C1ListViewLinkItem をクリックまたはタップすると、リンク項目にアタッチされた Web サイトが開きます。
 - b. C1ListViewNestedItem をクリックまたはタップすると、作成したサブメニューが開きます。

🟢ここまでの成果

このクイックスタートでは、Visual Studio プロジェクトを作成し、**C1ListView** コントロールをアプリケーションに追加しました。また、いくつかの **C1ListViewItem** をコントロールに追加し、アプリケーションを実行しました。

設計時サポート

Listview for ASP.NET Web Forms は、オブジェクトモデルを簡単に利用できるように、カスタマイズされたコンテキストメニュー、スマートタグ、および設計時サポートを備えています。

以下のセクションでは、Listview for ASP.NET Web Forms 設計時環境(特に C1ListView スマートタグとC1ListView のコンテキストメニューからアクセスできる[C1ListView のタスク]メニュー)を使用して、C1ListView コントロールを設定する方法について説明します。

C1ListView のスマートタグとタスクメニュー

C1ListView コントロールには、次の図のようなスマートタグが含まれています。



スマートタグは、各コンポーネントまたはコマンド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。

[C1ListView のタスク]メニューにアクセスするには、C1ListView コントロールの右上隅にあるスマートタグの矢印をクリックします。[C1ListView のタスク]メニューが表示されます。

[C1ListView のタスク]メニューの機能は次のとおりです。

- **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、データソース設定ウィザードを開きます。

- **ListView の編集**

[**ListView の編集**]リンクをクリックすると、C1ListView デザイナフォームが開きます。デザイナフォームを使用して、C1ListViewItem を追加したり、コントロールのプロパティを編集することができます。

- **ListView データバインディングの編集**

[**ListView データバインディングの編集**]をクリックすると、**Bindings コレクションエディタ**が開きます。このエディタを使用して、バインディングを削除したり、バインディングのプロパティを編集することができます。

- **[テーマ]**

[**C1ListView のタスク**]メニューには[テーマ]ドロップダウンがありますが、これらはアプリケーションに適用されるテーマではありません。テーマはスウォッチを通して適用されます。詳細については、「C1ListView の外観」を参照してください。

- **新しいテーマの作成**

[**新しいテーマの作成**]オプションをクリックすると、**ThemeRoller for Visual Studio** が開きます。したがって、開発環境内でテーマをカスタマイズすることができます。アプリケーションで **ThemeRoller for Visual Studio** を使用方法については、「[ThemeRoller for Visual Studio](#)」を参照してください。

- **CDN の使用**

[**CDN の使用**]チェックボックスをオンにして、コントロールエクステンダがコンテンツ配信ネットワークからクライアントリソースをロードする必要があることを指定します。このボックスは、デフォルトではオフになっています。

- **CDN パス**

コンテンツ配信ネットワークのパスを指定します。パスを変更するには、ここに URL を入力します。

- **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。

- **モバイルモード**

このオプションは、ASP.NET Web Forms コントロールがモバイルモードであるかどうかを示します。**C1AppView** または **C1ListView** コントロールを使用するには、このオプションのチェックボックスをオンにする必要があります。

- **ThemeSwatch**

[**ThemeSwatch**]オプションは、コントロール全体に適用される ThemeSwatch を示します。

- **バージョン情報**

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

- **エクステンダの追加**

Listview for ASP.NET Web Forms

[**エクステンダの追加**]項目をクリックすると、[**エクステンダウィザード**]が開き、**C1ListView** コントロールにエクステンダを追加できます。

C1ListView のコンテキストメニュー

Visual Studio がすべての .NET コントロールに対して提供するコンテキストメニューに、**C1ListView** コントロールのコマンドが追加されます。**C1ListView** コンテキストメニューにアクセスするには、**C1ListView** コントロール内の任意の場所で右クリックします。**C1ListView** コンテキストメニューが表示されます。

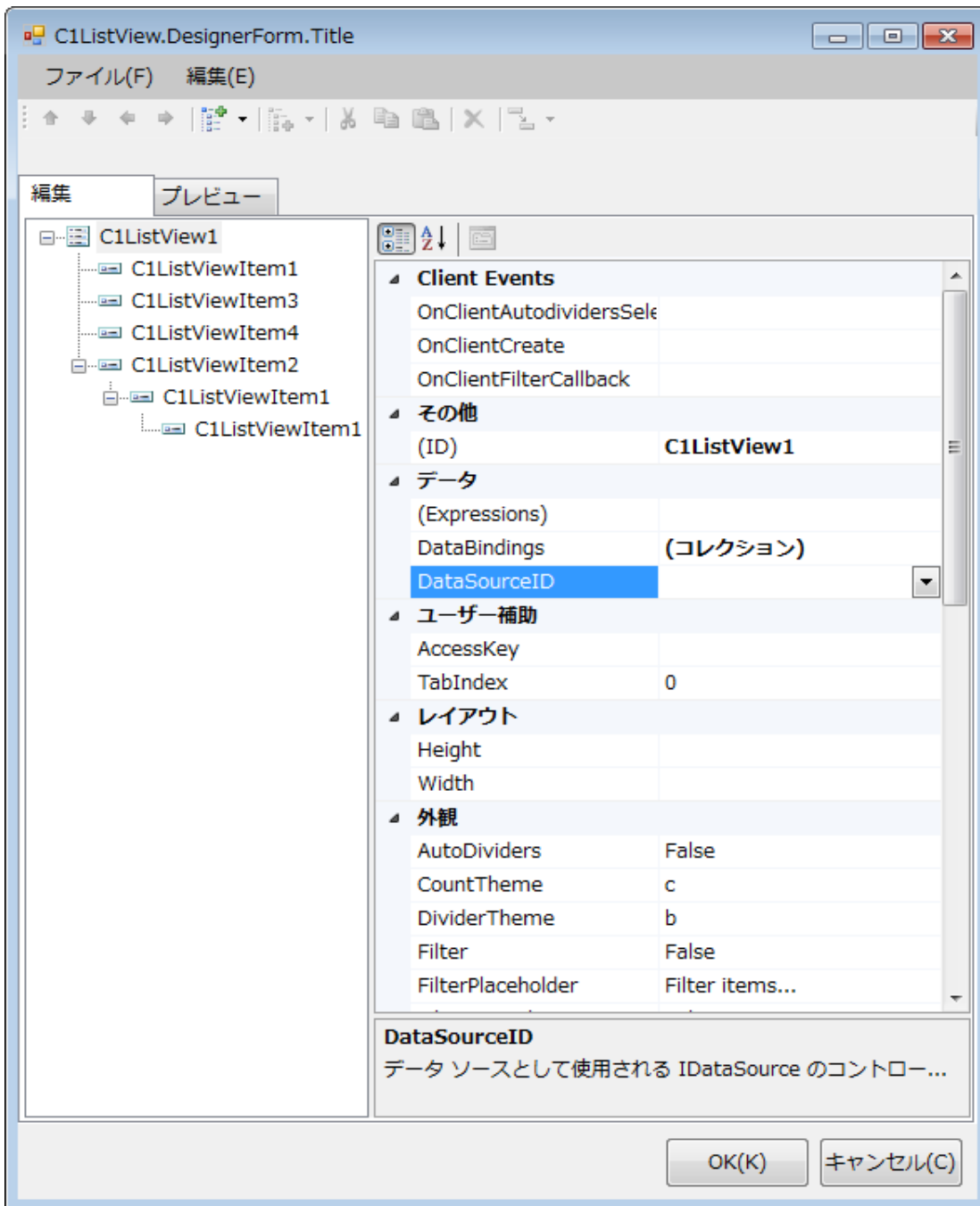
✂	切り取り(T)	Ctrl+X
📄	コピー(Y)	Ctrl+C
📄	貼り付け(P)	Ctrl+V
	代替の貼り付け(E)	
✖	削除(D)	Del
	コードの表示(C)	
🌐	ブラウザーで表示 (Internet Explorer)(B)	Ctrl+Shift+W
	スマート タグの表示(G)	Shift+Alt+F10
	Listviewの編集	
	Listviewデータバインディングの編集	
🔄	最新の情報に更新(F)	
🔧	プロパティ(R)	Alt+Enter

C1ListView コンテキストメニューには、**C1ListView** によって次のカスタムコマンドが追加されます。

- **スマートタグの表示**
[**スマートタグの表示**]オプションを選択すると、**C1ListView** のタスクメニューが開きます。
- **Listview の編集**
[**Listview の編集**]オプションを選択すると **C1ListView** デザイナフォームが開きます。デザイナーフォームから、**C1ListView** コントロールに子項目を追加したり、コントロールのプロパティを編集したり、アプリケーションをプレビューすることができます。
- **Listview データバインディングの編集**
[**Listview データバインディングの編集**]オプションをクリックすると、**Bindings コレクションエディタ**が開きます。このエディタを使用して、バインディングを追加または削除したり、バインディングのプロパティを編集することができます。
- **エクステンダの追加**
[**エクステンダの追加**]オプションを選択すると、[**エクステンダウィザード**]が開き、**C1ListView** コントロールにエクステンダを追加できます。

C1ListView デザイナフォーム

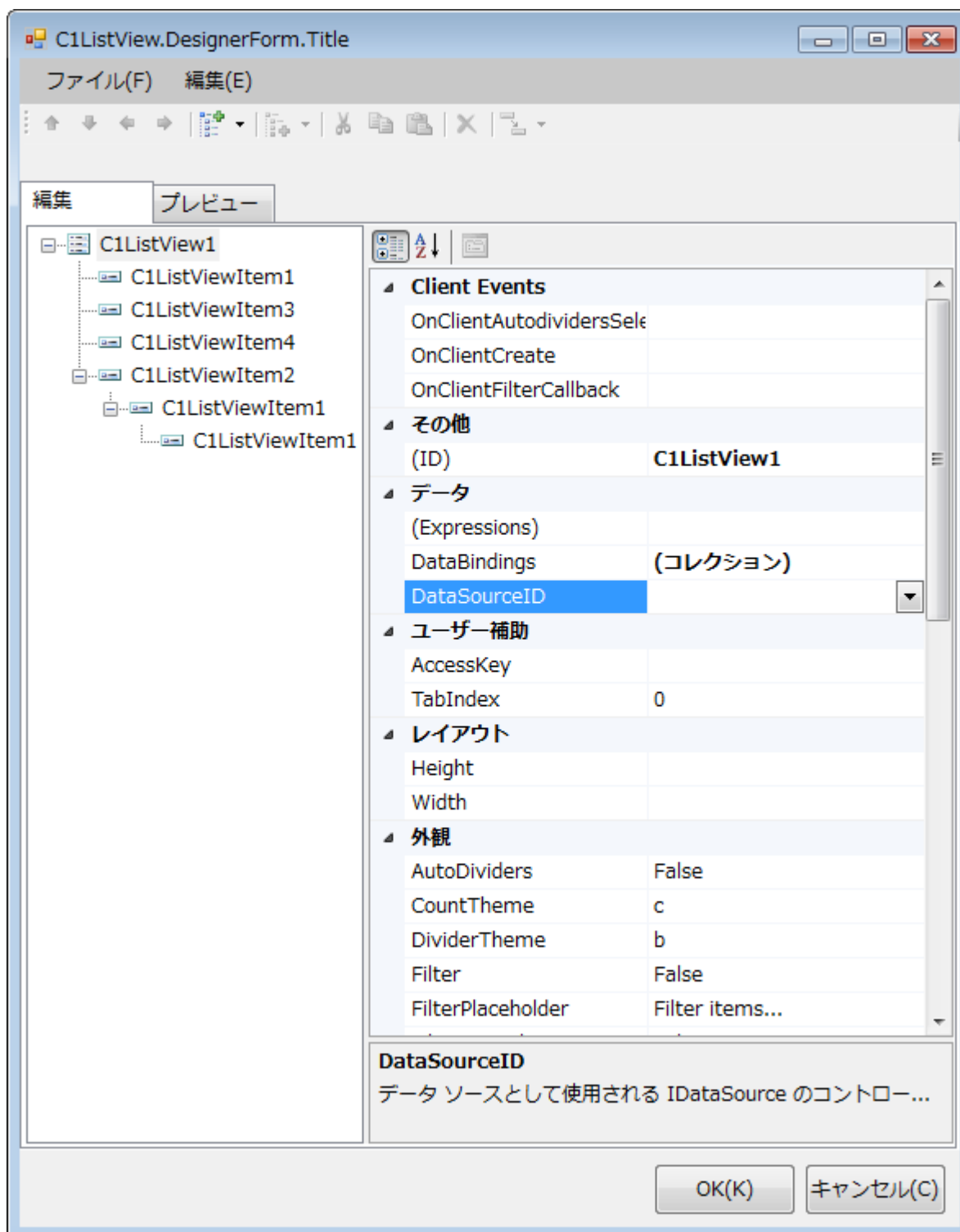
C1ListView デザイナフォームを使用して、**C1ListViewItem** を**C1ListView** コントロールに追加したり、**C1ListView** や**C1ListViewItem** のプロパティを編集したり、アプリケーションをプレビューすることができます。



C1ListView デザイナフォームの操作

C1ListView 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あります。

Listview for ASP.NET Web Forms

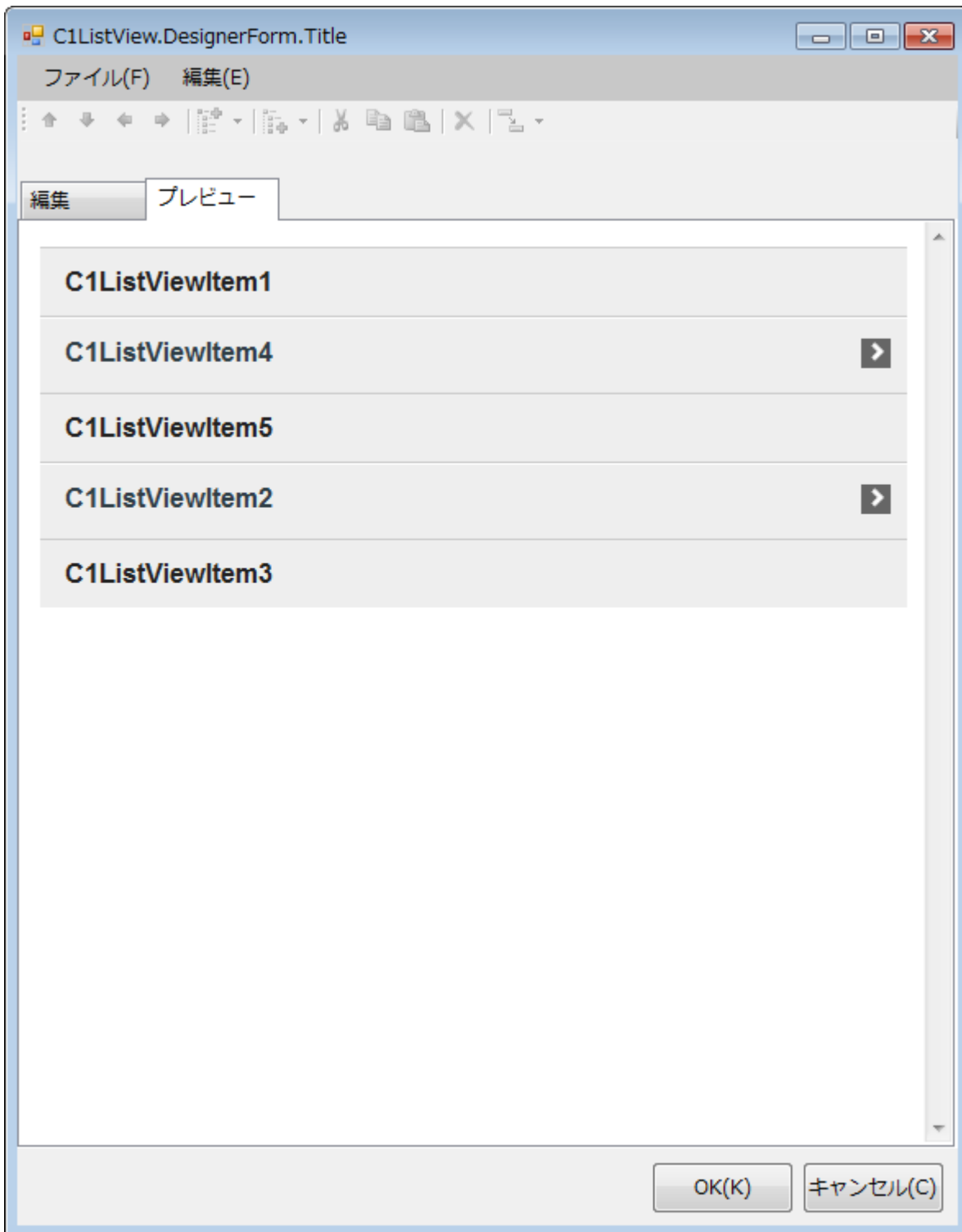


[編集]タブ

[編集]タブをクリックして、編集する**C1ListView** コントロールまたは**C1ListViewItem** を選択します。プロパティペインに、コントロールまたは項目のプロパティが表示されます。

[プレビュー]タブ

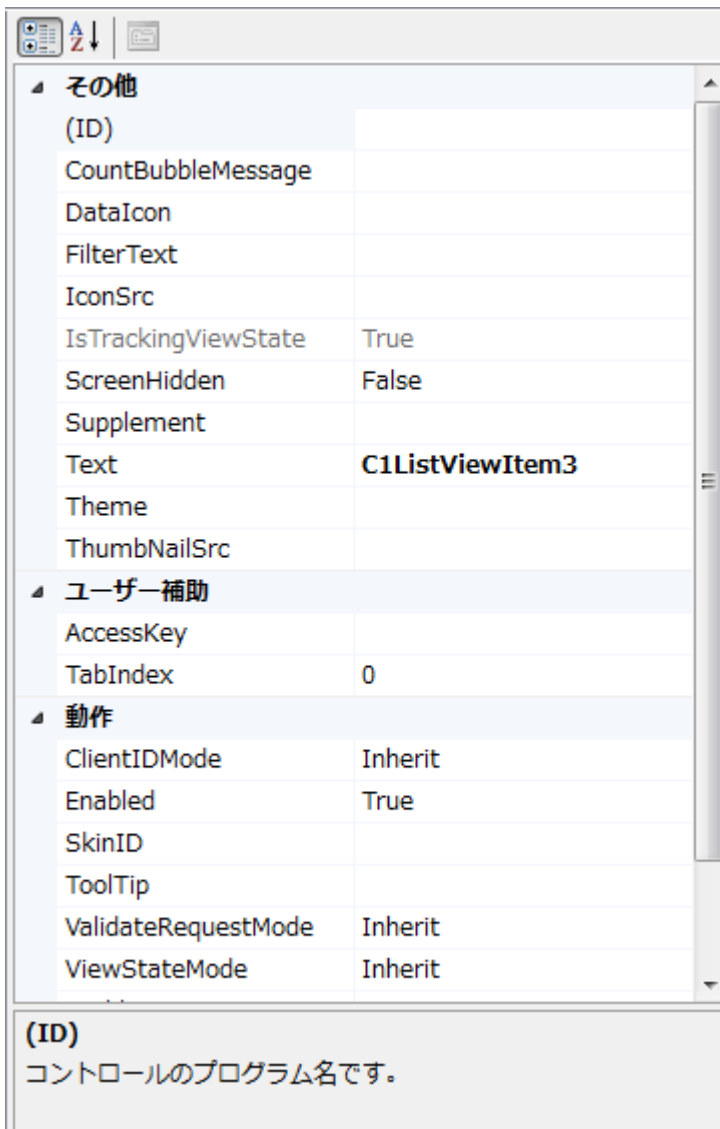
アプリケーションをプレビューするには、[プレビュー]タブを選択します。このタブは、デザイナフォームを終了しないでアプリケーションに加えた変更を確認する場合に便利です。



プロパティペイン

Visual Studio の[プロパティ]ウィンドウと同様に、プロパティペインには、**C1ListView** コントロールと各**C1ListViewItem** の編集可能なプロパティがすべて含まれています。

Listview for ASP.NET Web Forms



コマンドボタン

次の表は、2つのコマンドボタンの説明です。

ボタン	説明
OK	[OK]をクリックすると、 C1ListView コントロールに新しい設定が適用されます。
キャンセル	[キャンセル]をクリックすると、C1ListView デザイナフォームが閉じられます。新しい設定はキャンセルされ、C1ListView コントロールにはデフォルトの設定が適用されます。

C1ListView デザイナフォームメニュー

C1ListView デザイナフォームメニューには、次の表に示されているメニュー項目とサブメニュー項目が含まれています。

メニュー項目	サブメニュー項目	項目のアイコン	説明
ファイル	XML からロード		C1ListView コントロールの書式設定を .xml ファイルからロードします。

	XMLとして保存		C1ListView の書式設定を .xml ファイルに保存します。
	終了		C1ListView デザイナフォームを終了します。
編集	項目の挿入		新しい C1ListViewItem を C1ListView コントロール内の指定された場所に挿入します。このメニュー項目を選択すると、項目タイプを選択するサブメニューが開きます。
	子の追加		新しい C1ListViewItem が C1ListView コントロールまたは C1ListViewItem に追加されます。このメニュー項目を選択すると、項目タイプを選択するサブメニューが開きます。
	切り取り		移動する C1ListViewItem を C1ListView コントロールから切り取ります。
	コピー		選択された C1ListViewItem をコピーします。
	貼り付け		コピーまたは切り取られた C1ListViewItem を C1ListView コントロール内の指定された場所に貼り付けます。
	削除		選択された C1ListViewItem を完全に削除します。
	名前の変更		選択した C1ListView または C1ListViewItem の名前を変更することができます。




C1ListView デザイナフォームタスクバー

C1ListView デザイナフォームタスクバーは、次の図のように表示されます。



C1ListView デザイナフォームタスクバーには、次の表に示されている項目が含まれています。

ボタン	名前	説明
	項目を上に移動	選択された C1ListViewItem を1レベル上に移動します。
	項目を下に移動	選択された C1ListViewItem を1レベル下に移動します。
	項目を左に移動	選択された C1ListViewItem を1レベル左に移動します。選択した項目が C1ListViewItem の子項目である場合、その項目は単独の C1ListViewItem になります。
	項目を右に移動	選択された C1ListViewItem を1レベル右に移動します。選択した項目が単独の C1ListViewItem である場合、その項目は前の項目の子項目になります。
	子項目の追加	選択された C1ListView コントロールまたは C1ListViewItem に子項目を追加します。このボタンのドロップダウン矢印をクリックすると、項目タイプを選択するサブメニューが開きます。
	項目の挿入	現在選択されている項目の下に C1ListViewItem を挿入します。このボタンのドロップダウン矢印をクリックすると、項目タイプを選択するサブメニューが開きます。
	切り取り	選択された C1ListViewItem を削除します。これは、項目リスト内の任意の場所に貼り付けることができます。

	コピー	選択された C1ListViewItem のコピーを作成します。これは、項目リスト内の任意の場所に貼り付けることができます。
	貼り付け	指定された位置に、切り取りまたはコピーされた C1ListViewItem を挿入します。
	削除	選択された C1ListViewItem を完全に削除します。

C1ListView デザイナフォームの使用方法

以下のトピックでは、**C1ListView デザイナフォーム**を使用していくつかのタスクを実行する方法について説明します。

C1ListViewItem の削除

C1ListViewItem を削除するには、以下のいずれかの方法を使用できます。

- **ショートカットメニューを使用して C1ListViewItem を削除する**
コントロールから削除する **C1ListViewItem** を右クリックし、ショートカットメニューから[**削除**]を選択します。
- **[削除]ボタンを押して C1ListViewItem を削除する**
コントロールから削除する **C1ListViewItem** を選択し、[Delete]ボタンを押します。
- **[編集]メニューを使用して C1ListViewItem を削除する**
コントロールから削除する **C1ListViewItem** を選択し、[編集]メニューの[**削除**]を選択します。

C1ListViewItem の名前の変更

次のいずれかの方法で **C1ListViewItem** の名前を変更することができます。

- **[F2]キーを押す**
 1. 名前を変更する **C1ListViewItem** を選択します。
 2. [F2]を押し、**C1ListViewItem** の新しい名前を入力します。
- **ショートカットメニューから[名前の変更]を選択する**
 1. 名前を変更する **C1ListViewItem** を右クリックします。
 2. ショートカットメニューから[名前の変更]を選択し、**C1ListViewItem** の新しい名前を入力します。
- **[編集]メニューから[名前の変更]を選択する**
 1. 名前を変更する **C1ListViewItem** を選択します。
 2. [編集]→[名前の変更]を選択し、**C1ListViewItem** の新しい名前を入力します。
- **プロパティペインで C1ListViewItem の名前を変更する**
 1. 名前を変更する **C1ListViewItem** を選択します。プロパティペインにプロパティが表示されます。
 2. **Text** プロパティを見つけます。[Text]プロパティ領域に **C1ListViewItem** の新しい名前を入力します。

子項目の追加

次のいずれかの方法で、子 **C1ListViewItem** を **C1ListView** コントロールに追加することができます。

- **ショートカットメニューを使用して子項目を追加する**
C1ListView を右クリックし、メニューの[子の追加]→[**Listview 通常項目**]を選択します。これで、子項目が **C1ListView** コントロールに追加されます。
- **[子項目の追加]ボタンを押して子項目を追加する**
子項目を追加する **C1ListView** コントロールを選択します。デザイナフォームタスクバーにある[子項目の追加]ボタンをクリックすると、**C1ListViewItem** が追加されます。
- **[編集]メニューを使用して子項目を追加する**

C1ListViewItem の追加先の **C1ListView** を選択します。[編集]メニューをクリックし、[子項目の追加]→[ListView 通常項目]を選択します。

 **メモ:** 前述のいずれかの方法を使用して、**C1ListViewNestedItem** に子項目を追加することもできます。

C1ListViewItem の挿入

次のいずれかの方法で、**C1ListViewItem** を **C1ListView** アプリケーションに挿入することができます。

- **ショートカットメニューを使用して C1ListViewItem を挿入する**
 1. [編集]タブで、いずれかの **C1ListViewItem** を右クリックします。
 2. ショートカットメニューから[項目の挿入]→[ListView 通常項目]を選択します。
 3. **C1ListViewItem** がアプリケーションに追加されます。
- **[項目の挿入]ボタンを使用して C1ListViewItem を挿入する**

[編集]タブで、いずれかの **C1ListViewItem** を選択します。[項目の挿入]ボタンを押すと、現在選択されている項目の上に、もう1つの **C1ListViewItem** が挿入されます。
- **[編集]メニューを使用して C1ListViewItem を挿入する**

[編集]タブで、いずれかの **C1ListViewItem** を選択します。[編集]メニューの[項目の挿入]→[ListView 通常項目]を選択します。これで、コントロールで現在選択されている **C1ListViewItem** の上に **C1ListViewItem** が追加されます。

C1ListView の要素

このセクションでは、**C1ListView** コントロールを構成する要素について画像を使用してわかりやすく説明します。

C1ListView の項目

さまざまなタイプの **C1ListViewItem** を使用して、**ASP.NET Web Forms** プロジェクトをカスタマイズできます。以下のトピックでは、各タイプの **C1ListViewItem** について画像を使用してわかりやすく説明します。また、各 **C1ListViewItem** タイプの作成に使用されるマークアップの例も示します。次の図は、各タイプの **C1ListViewItem** の名前を示しています。



C1ListViewItem

C1ListViewItem は、表示する基本リスト項目を作成します。汎用の **C1ListViewItem** は、次の図のように表示されます。

基本のListViewItem

基本 **C1ListViewItem** の作成に使用されるマークアップは次のようになります。

ソースビュー

```
<cc1:C1ListViewItem Text="基本のListViewItem"></cc1:C1ListViewItem>
```

ほとんどの **C1ListViewItem** で番号付きリストを作成できます。それには、**CountBubbleMessage** プロパティを設定します。

基本のListViewItem

いち!

ソースビュー

```
<cc1:C1ListViewItem ID="ListViewItem" CountBubbleMessage="いち！" Text="基本のListViewItem"></cc1:C1ListViewItem>C1ListViewItem
```

C1ListViewButtonItem

C1ListViewButtonItem は、次の図のような単純なボタンコントロールを作成します。

このボタンをクリックしてください!

C1ListViewButtonItem を作成するためのマークアップは次のようになります。

ソースビュー

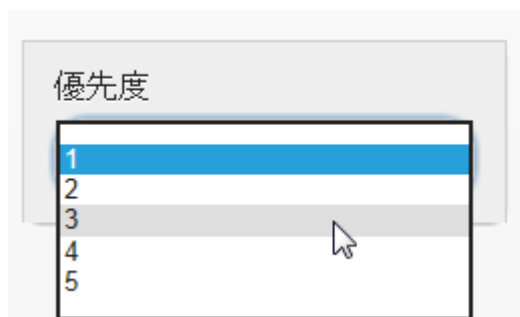
```
<cc1:C1ListViewButtonItem ID="ButtonItem1" Text="このボタンをクリックしてください">
</cc1:C1ListViewButtonItem>
```

C1ListViewControlGroupItem

C1ListViewControlGroupItem は、さまざまなタイプの選択ツールとして表示されます。選択ツールのタイプを指定するには、**ControlGroupType** プロパティを使用します。C1ListViewControlGroupItem には、**normal select**、**radioList**、**checkboxList** の3つのタイプがあります。デフォルトの選択ツールは **normal select** C1ListViewControlGroupItem です。

normal select

これは、C1ListViewControlGroupItem のデフォルト設定です。C1ListViewControlGroupItem が "normal select" に設定されている場合は、次の図に示すように、ドロップダウンリストから項目を選択できます。



normal select 項目の作成に使用されるマークアップは次のようになります。

ソースビュー

```
<cc1:C1ListViewControlGroupItem LabelText="優先度">
  <innerlistcontrols rows="5">
```

Listview for ASP.NET Web Forms

```
<asp:ListItem>1 </asp:ListItem>
<asp:ListItem>2 </asp:ListItem>
<asp:ListItem>3 </asp:ListItem>
<asp:ListItem>4 </asp:ListItem>
<asp:ListItem>5 </asp:ListItem>
<innerlistcontrols>
</cc1:C1ListViewControlGroupItem>
```

選択項目は、<innerlistcontrols>マークアップを使用して作成されます。C1ListView デザインフォームで作業している場合は、マークアップを使用するか、[編集]タブで正しい C1ListViewControlGroupItem を選択して、関連するプロパティを設定することができます。

また、次の図に示すように、オーバーレイして開くドロップダウンメニューに、ポップが移動するように選択項目を表示することもできます。



このタイプの選択項目を作成するには、**NativeMenu** プロパティを False に設定します。

ソースビュー

```
<cc1:C1ListViewControlGroupItem LabelText="優先度" NativeMenu="False">
  <innerlistcontrols rows="5">
    <asp:ListItem>1</asp:ListItem>
    <asp:ListItem>2</asp:ListItem>
    <asp:ListItem>3</asp:ListItem>
    <asp:ListItem>4</asp:ListItem>
    <asp:ListItem>5</asp:ListItem>
  </innerlistcontrols>
</cc1:C1ListViewControlGroupItem>
```

radioList

“radiolist” に設定されている場合、C1ListViewControlGroupItem はラジオボタン項目として表示されます。

このラジオリストの作成に使用されるマークアップは次のようになります。

ソースビュー

```
<cc1:C1ListViewControlGroupItem ControlGroupType="radiolist" LabelText="優先度">
  <innerlistcontrols rows="1" width="400px">
    <asp:ListItem>1</asp:ListItem>
    <asp:ListItem>2</asp:ListItem>
  </innerlistcontrols>
</cc1:C1ListViewControlGroupItem>
```

```
<asp:ListItem Selected="True">3</asp:ListItem>
<asp:ListItem>4</asp:ListItem>
<asp:ListItem>5</asp:ListItem>
</innerlistcontrols>
</cc1:C1ListViewControlGroupItem>
```

checkboxList

“checkboxlist” に設定された C1ListViewControlGroupItem は、チェックボックス型の複数選択項目を作成します。チェックボックス型の C1ListViewControlGroupItem の作成に使用されるマークアップは次のようになります。

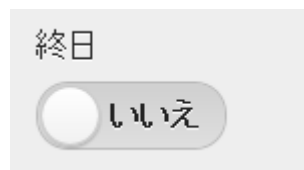
ソースビュー

```
<cc1:C1ListViewControlGroupItem ControlGroupType="checkboxlist" LabelText="優先度">
  <innerlistcontrols rows="1" width="400px">
    <asp:ListItem>1</asp:ListItem>
    <asp:ListItem>2</asp:ListItem>
    <asp:ListItem Selected="True">3</asp:ListItem>
    <asp:ListItem>4</asp:ListItem>
    <asp:ListItem>5</asp:ListItem>
  </innerlistcontrols>
</cc1:C1ListViewControlGroupItem>
```

C1ListViewFlipSwitchItem

C1ListViewFlipSwitchItem を使用して、ユーザーはブール値を選択できます。C1ListViewFlipSwitchItem を使用すると、ラジオボタンやチェックボックス項目から成る UI より魅力的な対話式の UI を作成することができます。また、タッチ環境では、フリップスイッチの方がラジオボタンやチェックボックスより簡単に使用できます。デフォルトでは、FlipSwitch は「はい」または「いいえ」に設定されています。

C1ListViewFlipSwitchItem は、次の図のようになります。



C1ListViewFlipSwitchItem の作成に使用されるマークアップは次のようになります。

ソースビュー

```
<cc1:C1ListViewFlipSwitchItem ID="AllDayInput" LabelText="終日" ONMessage="はい"
ONValue="true" OFFMessage="いいえ" OFFValue="false"></cc1:C1ListViewFlipSwitchItem>
/innovasys:widgetproperty>
```

フリップスイッチでは、ON と OFF に対応する Message と Value をそれぞれ設定します。これらは独自のブール値です。ペアにして使用するオプションに合わせて、これらの値をカスタマイズできます。

C1ListViewInputItem

C1ListViewInputItem には **label**、**text**、**textarea**、**search**、**slider** の5つのタイプがあります。label タイプ以外の C1ListViewInputItem では、データ入力を実装することができます。使用する C1ListViewInputItem のタイプを設定するに

Listview for ASP.NET Web Forms

は、**C1ListViewInputItem.Type** プロパティを使用します。このプロパティは、マークアップまたは [C1ListView デザイナフォーム](#) [C1ListView デザイナフォーム](#) で設定することができます。

label

label タイプの C1ListViewInputItem を使用すると、項目を固定ラベルとして表示できます。次の図は、このタイプの C1ListViewInputItem を示しています。



次のマークアップでは **Type** が設定されています。

```
ソースビュー
<cc1:C1ListViewInputItem ID="SubjectInput" LabelText="件名" Type="Wijmo 会議">
</cc1:C1ListViewInputItem>
```

text

C1ListViewInputItem.Type プロパティを "text" に設定すると、次の図に示すようなテキスト入力ツールが提供されます。



Type プロパティを設定するには、次のようなマークアップを使用します。

```
ソースビュー
<cc1:C1ListViewInputItem ID="SubjectInput" LabelText="件名" Type="Wijmo 会議">
</cc1:C1ListViewInputItem>
```

textarea

C1ListViewInputItem.Type プロパティを "textarea" に設定すると、サイズ変更可能なテキスト入力ツールが提供されます。この入力ツールのサイズは、テキストの入力時に自動的に変更されます。また、サイズ変更ハンドルを使用して、ユーザーが入力ツールのサイズを変更することもできます。



Type プロパティを設定するには、次のようなマークアップを使用します。

```
ソースビュー
<cc1:C1ListViewInputItem ID="SubjectInput" LabelText="件名" Type="Wijmo 会議">
</cc1:C1ListViewInputItem>
```


search

C1ListViewInputItem.Type を "search" に設定すると、次の図のような編集可能な検索バーが提供されます。



Type プロパティを設定するには、次のようなマークアップを使用します。

ソースビュー

```
<cc1:C1ListViewInputItem ID="SubjectInput" LabelText="件名" Type="Wijmo 会議">
</cc1:C1ListViewInputItem>
```

slider

C1ListViewInputItem.Type を "slider" に設定すると、スライダ UI が提供されます。ユーザーは、このスライダ UI を使用して、一定範囲内の値を対話式に選択することができます。

Type プロパティを設定するには、次のようなマークアップを使用します。

ソースビュー

```
<cc1:C1ListViewInputItem ID="SubjectInput" Type="slider" Max="40">
</cc1:C1ListViewInputItem>
```

C1ListViewLinkItem

C1ListViewLinkItem を使用して、外部 Web サイトへのリンクをナビゲーションリストに埋め込むことができます。C1ListViewLinkItem は次の図のように表示されます。



"split" プロパティ (SplitUrl, SplitTheme, SplitTitle) を設定して、1つの C1ListViewLinkItem 内に2つのリンクを作成することもできます。次の図は、SplitUrl プロパティがある Web サイトに設定された C1ListViewLinkItem を示しています。



このリスト項目の作成に使用されるマークアップは次のようになります。Web サイトの URL を指定するには、**NavigateUrl** プロパティを使用します。

ソースビュー

```
<cc1:C1ListViewLinkItem ID="Link" NavigateUrl="http://www.componentone.com/" Text="リンクへ移動!"></cc1:C1ListViewLinkItem>
```

C1ListViewNestedItem

C1ListViewNestedItem を使用すると、メインリスト内に他の **C1ListViewItem** をネストすることができます。C1ListViewNestedItem は次の図のように表示されます。

Listview for ASP.NET Web Forms

C1ListViewNestedItem の作成に使用されるマークアップは次のようになります。

他のオプション



ソースビュー

```
<cc1:C1ListViewNestedItem Text="他のオプション">
  <Items>
    <cc1:C1ListViewButtonItem Text="ここ！"></cc1:C1ListViewButtonItem>
    <cc1:C1ListViewDividerItem></cc1:C1ListViewDividerItem>
    <cc1:C1ListViewItem Text="これは別の項目です"></cc1:C1ListViewItem>
  </Items>
</cc1:C1ListViewNestedItem>
```

C1ListViewDividerItem

C1ListViewDividerItem は、2つの **C1ListViewItem** の間に静的な分割線を挿入します。デフォルトでは、この項目はテーマスウォッチ "b" を使用して表示されます。**C1ListView DividerTheme** プロパティを使用して、アプリケーション全体でこの項目にテーマを設定することも、C1ListViewDividerItem の **Theme** プロパティを使用して個別に設定することもできます。C1ListViewDividerItem は、次の図のようになります。

デフォルトの DividerItem:



Text プロパティが設定された DividerItem:

仕切り

上の図のような C1ListViewDividerItem は、次のマークアップを使用して作成できます。

ソースビュー

```
<cc1:C1ListViewDividerItem Theme="c" Text="仕切り"></cc1:C1ListViewDividerItem>
```

C1ListView でサポートされているリストタイプ

C1ListView コントロールを使用して、さまざまなタイプのリストを作成できます。読み取り専用リスト、リンクリスト、番号付きリスト、ネストされたリストなどを作成できます。

読み取り専用リスト







読み取り専用リストは、基本の **C1ListViewItem** で構成されます。これは、静的なラベルから成る、対話式ではないリストです。

いち
に
さん
し
ご
ろく

ナビゲーションリスト

ナビゲーションリスト内の各 **C1ListViewLinkItem** には、ナビゲーション URL が含まれています。ナビゲーション URL を設定するには、**NavigateUrl** プロパティを使用します。**C1ListView** コントロールのナビゲーションリスト項目は、右矢印インジケータ付きのボタンとして自動的にスタイルが設定されます。各項目はクリックすることができます。項目をクリックまたはタップすると、クリックイベントがトリガーされます。クリックイベントにより、ナビゲーション URL の AJAX 要求が発行され、DOM で新しいページが作成されて、ページの遷移が開始されます。

1つのナビゲーションリスト項目で2つのクリック可能なボタンを提供するには、**SplitUrl** プロパティを使用します。**SplitUrl** プロパティを使用すると、**C1ListViewLinkItem** は、それぞれクリック可能なボタンの間に分割線を入れて表示されます。通常の **C1ListViewLinkItem** と **SplitUrl** プロパティが設定された **C1ListViewLinkItem** との違いは見てわかります。次の図では、項目「Two」と「Five」には **SplitUrl** プロパティが設定されていません。

いち	
に	
さん	
し	
ご	
ろく	

番号付きリスト

C1ListView コントロールで番号付きリストを作成するには、2つの方法があります。1つは、リスト全体の **UseNumberedList** プロパティを設定するという方法です。このプロパティを設定すると、次の図のようなリストになります。

1. いち
2. に
3. さん
4. し
5. ご
6. ろく

もう1つは、各 **C1ListViewItem** に **CountBubbleMessage** プロパティを設定するという方法です。

いち	①
に	②
さん	③
し	④
ご	⑤
ろく	⑥

ネストされたリスト

ネストされたリストを作成すると、サブメニューがネストされたリストを作成することができます。C1ListViewNestedItem のビジュアルスタイルは、通常の C1ListViewLinkItem に似ています。このボタン項目の右側に、右向きの矢印が表示されます。次の図では、項目「One」と「Five」がこのスタイルになっています。

いち	➤
に	
さん	
し	
ご	➤
ろく	

ネストされたリスト項目をクリックまたはタップすると、新しいページが表示され、ヘッダーには上位レイヤのタイトルが、コンテンツには子要素のリストが表示されます。

インセットスタイルのリスト

通常、**C1ListView** コントロールは、使用されているデバイスのタイプに応じて自動的にサイズが変更され、画面の端まで拡張されます。**C1ListView Inset** プロパティを true に設定すると、作成したリストがコンテンツ領域内に収められ、リストの周囲に一定のマージンが挿入されます。

いち
に
さん
し
ご
ろく

このリストタイプは、次のトピック「フォームリスト」で説明されているフォームスタイルのリストに使用すると特に便利です。

フォームリスト

C1ListViewItem を使用して、複雑なフォームを簡単に作成することができます。各項目のコントロール (C1ListViewInputItem、C1ListViewFlipSwitchItem など) には、読み取り専用ラベルと対話式パーツの両方を作成します。

次に、**C1ListViewItem** で作成したフォームの例を示します。

Listview for ASP.NET Web Forms

件名	<input type="text" value="新規のイベント"/>
位置	<input type="text"/>
開始	<input type="text" value="1/13/2014 午前10:00"/>
終了	<input type="text" value="1/13/2014 午後 05:00"/>
説明	<input type="text"/>
終日	<input type="radio"/> いいえ

C1ListView の外観

C1ListView は、簡単にカスタマイズできるように設計されています。**C1ListView** コントロールの外観の変更の際に、その可能性は無限大です。さらなるカスタマイズのために、C1ListView では [jQuery Mobile テーマスウォッチ](#) が使用されます。[jQuery Mobile Themeroller](#) を使用して、独自のテーマをデザインすることもできます。

テーマスウォッチ

C1ListView では [jQuery Mobile テーマスウォッチ](#) がデフォルトテーマとして使用されています。デフォルトテーマには5つのスウォッチが用意されています。**C1ListView** コントロール内でこれらのスウォッチを組み合わせることで、コントロールを完全にカスタマイズできます。C1ListView コントロール全体に1つのスウォッチを適用することも、コントロール内の**C1ListViewItem** ごとに異なるスウォッチを適用することもできます。デフォルトでは、それぞれの **C1ListViewItem** タイプが次のように表示されます。



以下のトピックでは、各スウォッチを**C1ListView** コントロールに適用した例を示します。以下のトピックでスウォッチが適用されたコントロールには、各種の**C1ListViewItem** が含まれています。このため、それぞれの項目の外観をスウォッチが適用された状態で確認することができます。

スウォッチ a



スウォッチ b



スウォッチ c

項目1
他のオプション 
項目2
<div style="border: 1px solid #ccc; border-radius: 15px; padding: 10px; text-align: center; width: fit-content; margin: 0 auto;"> クリック! </div>
項目3
リンクへ移動 
仕切り
テキストを入力してください: <input style="width: 100%; height: 30px; border: 1px solid #ccc;" type="text"/>  

スウォッチ d

項目1
他のオプション 
項目2
<input type="button" value="クリック!"/>
項目3
リンクへ移動 
仕切り
テキストを入力してください: <input type="text"/>

スウォッチ e

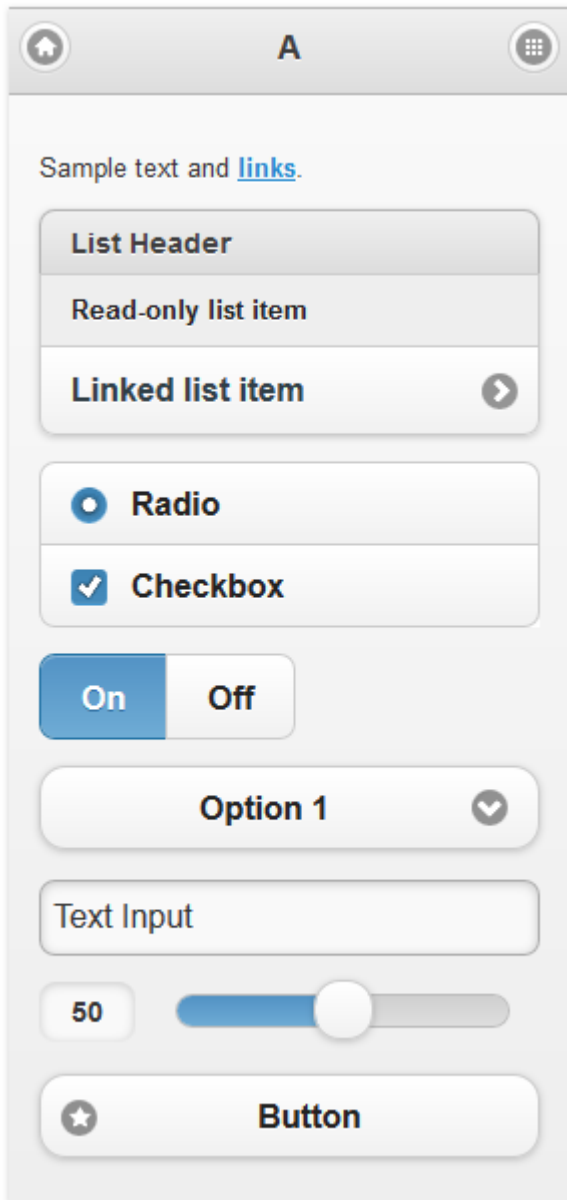
項目1
他のオプション 
項目2
<input type="button" value="クリック!"/>
項目3
リンクへ移動 
仕切り
テキストを入力してください: <input type="text"/>

jQuery Mobile Themeroller の概要

jQuery Mobile Themeroller を使用して、直観的な方法で簡単に独自のテーマを作成できます。カスタムテーマを **C1ListView** アプリケーションに適用することも簡単です。アプリケーションにカスタムテーマを適用する方法については、「カスタムテーマの使用」トピックを参照してください。

新しい Mobile ThemeRoller を使用して、さまざまなパレットから色をドラッグ & ドロップしたり、カスタム色を作成することができます。1つのテーマに対して、最大でラベル A から Z までの 26 のスウォッチを作成できます。

新しいテーマを作成する場合は、まず1つのスウォッチから始めます。



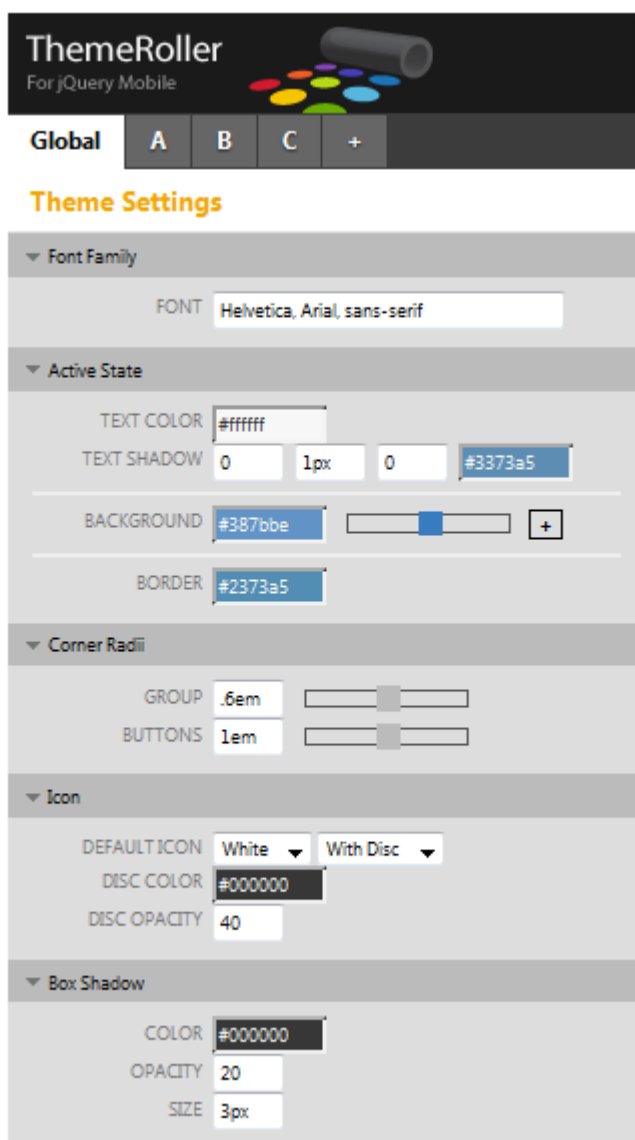
ページの上にあるカラーピッカーから、ドラッグ & ドロップ操作で色を選択できます。



[明度]および[彩度]コントロールを使用して、色を完全にカスタマイズできます。Adobe Kuler スウォッチから色を選択することもできます。

Listview for ASP.NET Web Forms



さらにアプリケーションをカスタマイズするために、いくつかの全体設定を変更できます。これらの設定は、ページの左側の[グローバル]タブにあります。



ページの左側にはスウォッチごとのタブもあり、ここでヘッダー/フッターバー、コンテンツ領域、およびボタンをカスタマイズできます。

アイコン

C1ListView コントロールで使用されるアイコンは、[jQuery Mobile Themeroller の概要](#)システムに含まれています。ファイルサイズを小さくするために、この標準アイコンのコアセットは、1つの白色のアイコンスプライトとして格納されています。モバイルアプリケーションで最もよく使用されるアイコンが用意されています。C1ListViewLinkItem 要素と C1ListViewNestedItem 要素にアイコンを指定することができます。

アイコン名	アイコン参照	アイコン画像
左矢印	arrow-l	
右矢印	arrow-r	

上矢印	arrow-u	
下矢印	arrow-d	
削除	delete	
プラス	plus	
マイナス	minus	
チェック	check	
ギア	gear	
更新	refresh	
進む	forward	
戻る	back	
グリッド	grid	
星	star	
警告	alert	
情報	info	
ホーム	home	
検索	search	

タスク別ヘルプ

タスク別ヘルプセクションは、ユーザーの皆様が Visual Studio ASP.NET 環境でのプログラミングに精通しており、**Listview for ASP.NET Web Forms** コントロールの一般的な使用方法を理解していることを前提としています。

各トピックでは、**C1ListView** 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提供します。各トピックに示される手順に従って作業を進めるだけで、さまざまな **C1ListView** の機能を使用したプロジェクトを作成できます。

また、タスク別ヘルプトピックは、新しい ASP.NET プロジェクトが既に作成されていることを前提としています。

ナビゲーションリストの作成

C1ListView コントロールを使用して、リンクナビゲーションリストを簡単に作成できます。ナビゲーションリストを作成するには、次の手順に従います。

1. **C1ListView** コントロールをアプリケーションに追加します。それには、Visual Studio のツールボックスでコントロールを見つけてダブルクリックするか、次のマークアップをアプリケーションに追加します。

ソースビュー

```
<cc1:C1ListView ID="C1ListView1" runat="server"></cc1:C1ListView>
```

2. `<cc1:C1ListView>` `</cc1:C1ListView>` タグの間にカーソルを置き、次のマークアップを挿入します。

ソースビュー

```
<Items> </Items>
```

3. `<Items>` タグの間に次のマークアップを追加して、3つの **C1ListViewLinkItem** を追加します。**NavigateUrl** プロパティを指定すると、外部 Web サイトへのリンクが埋め込まれます。

ソースビュー

```
<cc1:C1ListViewLinkItem Text="ComponentOne"
NavigateUrl="http://www.componentone.com/"></cc1:C1ListViewLinkItem>

<cc1:C1ListViewLinkItem Text="Google" NavigateUrl="http://www.google.com/"
</cc1:C1ListViewLinkItem>

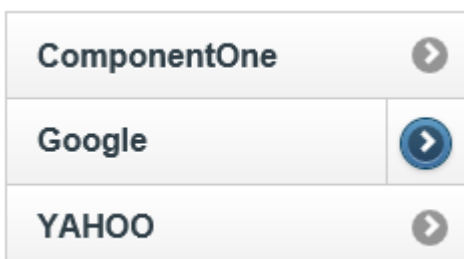
<cc1:C1ListViewLinkItem Text="YAHOO" NavigateUrl="http://www.yahoo.com/"
Title="yahoo"></cc1:C1ListViewLinkItem>
```

4. **SplitUrl** プロパティを使用すると、1つの **C1ListViewLinkItem** に2つのリンクを含めることができます。このプロパティを設定するには、2つ目の **C1ListViewLinkItem** を次のように編集します。

ソースビュー

```
<cc1:C1ListViewLinkItem Text="Google" NavigateUrl="http://www.google.com/"
SplitUrl="http://news.google.com/" SplitTitle="google news">
</cc1:C1ListViewLinkItem>
```

5. アプリケーションを実行すると、次の図のようになります。



フィルタ機能と検索機能の使用

C1ListView コントロールのフィルタ機能と検索機能は関連しています。**Filter** プロパティが "true" に設定されている場合は、**C1ListView** コントロールに自動的に検索ボックスが付加されます。

フィルタ機能の設定

C1ListView コントロールを設定して、検索とフィルタを簡単に使用することができます。次の手順に従います。

1. フィルタを有効にするデータのリストを表示するための **C1ListView** を作成します。次のようなマークアップを使用します。

ソースビュー

```
<cc1:C1ListView ID="C1ListView1" runat="server">
  <Items>
    <cc1:C1ListViewLinkItem Text="愛知工業大学"></cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="足利工業大学"></cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="大阪工業大学"></cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="愛知県立芸術大学"></cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="大阪芸術大学"></cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="愛知術大学"></cc1:C1ListViewLinkItem>
  </Items>
</cc1:C1ListView>
```

2. 次のマークアップのように、**Filter** プロパティを "true" に設定します。

ソースビュー

```
<cc1:C1ListView ID="C1ListView1" runat="server" Filter="true">
```

3. **Filter** プロパティを "true" に設定して **C1ListView** リストに付加する検索ボックスに、プレースホルダテキストを設定することもできます。

ソースビュー

```
<cc1:C1ListView ID="C1ListView1" runat="server" Filter="true"
  FilterPlaceholder="大学を検索してください...">
```

4. [F5]キーを押すか、デバッグを開始して、アプリケーションを実行します。次の図のように表示されます。



検索バーへの入力を開始すると、結果がフィルタされます。



FilterText プロパティの使用

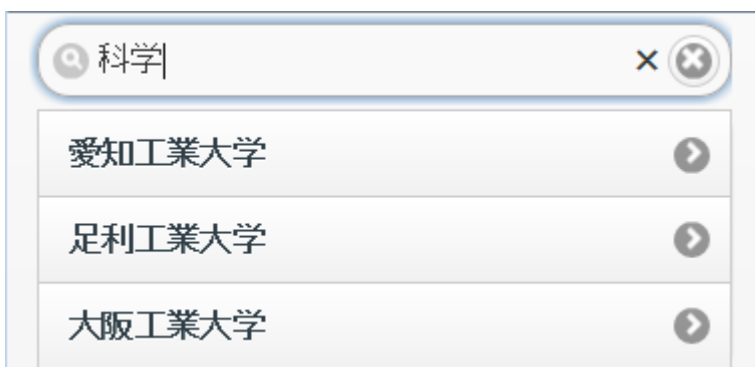
FilterText プロパティを使用すると、**C1ListViewItem** 以外のコンテンツを対象に検索を実行することができます。トピック「[フィルタ機能の設定](#)」で作成したアプリケーションを使用して、次の手順を実行します。

1. 次のようなマークアップを使用して、**C1ListView** コントロールを編集します。一部の**C1ListViewItem** に **FilterText** プロパティを設定していることに注意してください。

ソースビュー

```
<cc1:C1ListView ID="C1ListView1" runat="server" Filter="true" Inset="true">
    <Items>
        <cc1:C1ListViewLinkItem Text="愛知工業大学" FilterText="科学">
    </cc1:C1ListViewLinkItem>
        <cc1:C1ListViewLinkItem Text="足利工業大学" FilterText="科学">
    </cc1:C1ListViewLinkItem>
        <cc1:C1ListViewLinkItem Text="大阪工業大学" FilterText="科学">
    </cc1:C1ListViewLinkItem>
        <cc1:C1ListViewLinkItem Text="愛知県立芸術大学" FilterText="芸術">
    </cc1:C1ListViewLinkItem>
        <cc1:C1ListViewLinkItem Text="大阪芸術大学" FilterText="芸術">
    </cc1:C1ListViewLinkItem>
        <cc1:C1ListViewLinkItem Text="愛知術大学" FilterText="術">
    </cc1:C1ListViewLinkItem>
    </Items>
</cc1:C1ListView>
```

2. [F5]キーを押すか、デバッグを開始して、アプリケーションを実行します。検索バーに「Adventure」と入力します。結果は次の図のようになります。



フィルタ機能のカスタマイズ

フィルタ機能のカスタマイズして、ユーザーに表示する結果をより詳細に制御することができます。**OnClientFilterCallback** 関数を使用して、フィルタをカスタマイズすることができます。

トピック「フィルタ機能の設定」で作成したアプリケーションを使用して、次の手順を実行します。

1. ページ内のデータをフィルタするための関数を作成します。次のスクリプトをページ内の `<head>` `</head>` タグの間に挿入します。

ソースビュー

```
<script type="text/javascript">
    function filterFunction(text, searchValue) {
        // この関数は、文字列の先頭からフィルタとの一致を調べます。
        return !(text.slice(0, searchValue.length).toLowerCase() ==
searchValue.toLowerCase());
    }
</script>
```

2. **OnClientFilterCallback** イベントを "filterFunction" に設定します。

ソースビュー

```
<c1:C1ListView ID="C1ListView1" runat="server" Filter="true"
OnClientFilterCallback="filterFunction">
```

3. [F5]キーを押すか、デバッグを開始して、アプリケーションを実行します。検索バーに入力すると、入力した文字列に先頭から一致するリスト項目のみがフィルタから返されます。



XML テンプレートの作成とロード

C1ListView デザイナフォームを使用すると、XML テンプレートを簡単に作成してロードすることができます。次のトピックでは、XML テンプレートを作成してロードする方法について説明します。

XML テンプレートの作成

XML テンプレートを作成するには、次の手順に従います。

1. **C1ListView** コントロールを作成します。以前に作成したコントロールを使用することも、任意の書式設定で新しいコントロールを作成することもできます。次に、**C1ListView** コントロールを作成するためのマークアップを示します。

ソースビュー

```
<c1:C1ListView ID="C1ListView1" runat="server">
<Items>
    <c1:C1ListViewLinkItem Text="愛知工業大学"></c1:C1ListViewLinkItem>
```

Listview for ASP.NET Web Forms

```
<cc1:C1ListViewLinkItem Text="足利工業大学"></cc1:C1ListViewLinkItem>
<cc1:C1ListViewLinkItem Text="大阪工業大学"></cc1:C1ListViewLinkItem>
<cc1:C1ListViewLinkItem Text="愛知県立芸術大学"></cc1:C1ListViewLinkItem>
<cc1:C1ListViewLinkItem Text="大阪芸術大学"></cc1:C1ListViewLinkItem>
<cc1:C1ListViewLinkItem Text="愛知術大学"></cc1:C1ListViewLinkItem>
</Items>
</cc1:C1ListView>
```

2. デザインビューに切り替えて、デザイナフォームを開きます。それには、コントロールのスマートタグをクリックし、[C1ListView のタスク]メニューの[ListView の編集]を選択します。
3. デザイナフォームで、[ファイル]→[XML として保存]を選択します。
4. [名前を付けて保存]ウィンドウが開いたら、XML ファイルを保存する場所を参照します。XML ファイルの名前を入力し、[保存]をクリックします。

このトピックでは、C1ListView コントロールを作成し、XML ファイルとして保存しました。

XML テンプレートのロード

XML テンプレートをロードするには、次の手順に従います。

1. 新しい ASP.NET アプリケーションで、空の C1ListView コントロールを追加します。
2. C1ListView デザイナフォームを開きます。それには、コントロールのスマートタグをクリックし、[C1ListView のタスク]メニューの[ListView の編集]を選択します。
3. デザイナフォームで、[ファイル]→[XML からロード]を選択します。
4. XML ファイルを見つけて選択し、[開く]をクリックします。データが C1ListView コントロールにロードされます。

テーマ

jQuery Mobile テーマを使用すると、モバイルアプリケーションのテーマも簡単に作成できます。デフォルトのテーマスウォッチを組み合わせると独自のテーマを作成したり、jQuery の新しい Mobile ThemeRoller を使用してデザインしたカスタムテーマを適用することができます。

デフォルトのスウォッチの適用

デフォルトのテーマスウォッチを組み合わせると、独自のアプリケーションを作成できます。C1ListView アプリケーションにデフォルトのスウォッチを適用するには、次の手順に従います。

1. C1ListView コントロールのマークアップは次のようになります。

ソースビュー

```
<cc1:C1ListView ID="C1ListView1" runat="server" Filter="true" Inset="true">
  <Items>
    <cc1:C1ListViewLinkItem Text="日本の言葉" FilterText="日本">
  </cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="日本の文化" FilterText="日本">
  </cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="インドの言葉" FilterText="インド">
  </cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="インドの文化" FilterText="インド">
  </cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="インドの料理" FilterText="インド">
  </cc1:C1ListViewLinkItem>
    <cc1:C1ListViewLinkItem Text="日本の料理" FilterText="日本">
  </cc1:C1ListViewLinkItem>
  </Items>
</cc1:C1ListView>
```

```
</cc1:C1ListViewLinkItem>
    </Items>
</cc1:C1ListView>
```

2. **デザイナーフォーム**を開きます。それには、コントロールのスマートタグをクリックし、[**C1ListView のタスク**]メニューの [**ListView の編集**]を選択します。
3. **C1ListView** コントロールを選択し、プロパティペインで Appearance プロパティを見つけます。
4. **FilterTheme** プロパティを "a" に、**ThemeSwatch** プロパティを "e" に設定し、[OK]を押します。アプリケーションの <cc1:C1ListView> 開始タグは次のサンプルのようになります。

ソースビュー

```
<cc1:C1ListView ID="C1ListView1" runat="server" Filter="true" Inset="true"
FilterTheme="a" ThemeSwatch="e">
```

5. [F5]キーを押すか、デバッグを開始して、アプリケーションを実行します。次の図のようになります。



カスタムテーマの使用

jQuery Mobile Themerollerを使用すると、独自のテーマを直観的な方法で簡単にデザインできます。それには、カラーピッカーからスウォッチの目的のセクションまで色をドラッグするだけです。新しいテーマをダウンロードして保存したら、次の手順に従ってアプリケーションにテーマを適用します。

1. ソリューションエクスプローラで、[**すべてのファイルを表示**]を選択して、アプリケーションの非表示ファイルを表示します。
2. **Content** フォルダを見つけ、右クリックしてリストを表示します。[**プロジェクトに追加**]を選択します。
3. **Content** フォルダを再度右クリックし、リストから[**追加**]→[**既存の項目**]を選択します。
4. カスタムテーマを含むファイルを参照して選択し、縮小テーマファイルを選択します。[**追加**]をクリックして、ファイルをアプリケーションに追加します。
5. アプリケーションの <head> <head> タグに次の参照を追加します。テーマ名は、独自のテーマに付けた名前を参照するように変更してください。

ソースビュー

```
<link rel="stylesheet" href="~/Content/mycustomtheme1.min.css" type="text/css" />
```

6. アプリケーションで使用するスウォッチを選択し、**FilterTheme** プロパティと **ThemeSwatch** プロパティを設定します。マークアップは次のようになります。

ソースビュー

Listview for ASP.NET Web Forms

```
<cc1:C1ListView ID="C1ListView1" runat="server" Filter="true" Inset="true"  
FilterTheme="g" ThemeSwatch="g">
```

7. [F5]キーを押すか、デバッグを開始して、アプリケーションを実行します。カスタムテーマがアプリケーションに適用されたことを確認します。