

# AutoComplete for ASP.NET Web Forms

2018.04.12 更新

グレープシティ株式会社

## 目次

<a href="#">製品の概要</a>	3
<a href="#">ComponentOne for ASP.NET Web Forms のヘルプ</a>	3
<a href="#">主な特長</a>	4
<a href="#">クイックスタート</a>	5
<a href="#">手順 1: ページへのコントロールの追加</a>	5
<a href="#">手順 2: 検索ボックスのカスタマイズ</a>	5-6
<a href="#">手順 3: アプリケーションの実行</a>	6
<a href="#">デザイン時サポート</a>	7
<a href="#">C1AutoComplete のスマートタグ</a>	7-8
<a href="#">C1AutoComplete のコンテキストメニュー</a>	8-9
<a href="#">C1AutoComplete デザイナフォーム</a>	9
<a href="#">AutoComplete デザイナフォームの機能</a>	9-12
<a href="#">デザイナの使用方法</a>	12
<a href="#">C1AutoCompleteDataItem の削除</a>	12
<a href="#">デザイナでの AutoCompleteDataItem の名前変更</a>	12-13
<a href="#">子項目の追加</a>	13
<a href="#">C1AutoComplete の要素</a>	14
<a href="#">C1AutoComplete の外観</a>	15
<a href="#">テーマ</a>	15-16
<a href="#">CSS セレクタ</a>	16
<a href="#">クライアント側での作業</a>	17
<a href="#">クライアント側イベント</a>	17
<a href="#">タスク別ヘルプ</a>	18
<a href="#">コードによる C1AutoComplete コントロールの作成</a>	18
<a href="#">C1AutoComplete の幅と高さのカスタマイズ</a>	18-19
<a href="#">C1AutoComplete CSS セレクタでの作業</a>	19-20
<a href="#">AutoPostBack プロパティの使用</a>	20-22
<a href="#">DataBinding の使用</a>	22
<a href="#">LoadOnDemand プロパティの使用</a>	22-23
<a href="#">C1AutoComplete でのテーマの使用</a>	23
<a href="#">スマートタグを使用した組み込みテーマの適用</a>	23-24



## 製品の概要

この AJAX 対応 AutoComplete コントロールを使用してオートコンプリート検索ボックスを作成します。この検索ボックスは、DataBinding、AutoPostBack での値の検索、および Ajax での値のロードをサポートします。**C1AutoComplete** コントロールは、クライアントまたはサーバー側で作成できる、ユーザーによる検索用語の入力を完結させる特別な種類の検索ボックスです。

## 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

**C1AutoComplete** は、LoadOnDemand、DataBinding、AutoPostBack、AutoComplete の位置決め、MaxCount プロパティ、MinLength プロパティ、および組み込みテーマなど、いくつかの独特な機能を備えています。

- **LoadOnDemand**  
**AutoComplete for ASP.NET Web Forms** は Ajax に対応し、情報を動的にロードできます。子項目は、検索ボックスに入力するときにロードされます。
- **DataBinding**  
**C1AutoComplete** では、ラベル、値、およびカテゴリフィールドを希望するデータソースにバインドできます。
- **AutoPostBack**  
**AutoPostBack** プロパティにより、ユーザーがコントロールでインタラクティブ操作を行うたびにコントロールがサーバーにポストバックするかどうかを指定できます。
- **位置決め**  
AutoComplete では、関連付けられた入力要素に対するコントロールの位置を識別できます。
- **MaxCount**  
**C1AutoComplete** では、コールバックの結果の最大数を制御できます。このプロパティを使用するには、データソースを指定し、**LoadOnDemand** プロパティを "true" に設定する必要があります。
- **MinLength**  
**MinLength** では、Autocomplete をアクティブ化する前にユーザーが入力する必要がある最小文字数を制御できます。
- **テーマ**  
スマートタグを単にクリックして、5種類のプレミアムテーマ (Midnight、Aristo、Rocket、Cobalt、および Sterling) のいずれかを選択して AutoComplete の外観を変更します。オプションとして、jQuery UI から ThemeRoller を使用してカスタマイズしたテーマを作成します。この機能についての詳細は、[テーマ](#)を参照してください。
- **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。CSS のサポートにより、検索ボックスを組織の基準に合致させることができます。

## クイックスタート

このクイックスタートでは、**AutoComplete for ASP.NET Web Forms** を始める方法について説明します。クイックスタートでは、Visual Studio でプロジェクトを作成して、ページに **C1AutoComplete** コントロールを追加し、コントロールの外観と動作を変更し、DataItem の一覧を追加し、**C1AutoComplete** の実行時の動作を確認します。

## 手順 1: ページへのコントロールの追加

この手順では、新しいプロジェクトを作成して、**AutoComplete for ASP.NET Web Forms** 検索ボックスをプロジェクトに追加します。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. Visual Studio の[ファイル]メニューから、[新規]→[プロジェクト]を選択します。[新規プロジェクト]ダイアログボックスが表示されます。
2. [新規プロジェクト]ダイアログボックスで、左ペインにある言語を展開し、[Web]を選択します。右ペインで、[空の ASP.NET Web アプリケーション]を選択し、アプリケーションの[名前]を入力して<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。
3. ソリューションエクスプローラで、プロジェクトを右クリックして[参照の追加]を選択します。
4. [参照の追加]ダイアログボックスで、**C1.Web.Wijmo.Controls** および **C1.Web.Wijmo.Controls.Design** アセンブリを検出して選択し、<OK>をクリックします。参照が追加されます。
5. デザインビューで、Visual Studio ツールボックスに移動し、**C1AutoComplete** アイコンをダブルクリックして **C1AutoComplete** コントロールをページに追加します。

新規プロジェクトを作成して、**C1AutoComplete** 検索ボックスをプロジェクトに追加する操作が完了しました次の手順では、検索ボックスの外観と動作をカスタマイズします。

## 手順 2: 検索ボックスのカスタマイズ

この手順では、前の手順で作成した検索ボックスの外観と動作を変更します。次の手順を実行して、**AutoComplete for ASP.NET Web Forms** 検索ボックスをカスタマイズします。

1. **C1AutoComplete** スマートタグを選択して、[**C1AutoComplete タスク**]メニューを開き、以下のように設定します。
  - [幅]テキストボックスに **300px** を入力する
  - [高さ]テキストボックスに **50px** を入力する
  - [テーマ]ドロップダウンボックスで **midnight** を選択して、AutoComplete コントロールの外観を変更します。
2. [**AutoCompleteItem の追加**]をクリックして、**C1AutoComplete デザインフォーム**を開きます。Use the <子項目の追加>ボタンを使用して、**C1AutoComplete** コントロールに 13 個の子項目を追加します。
3. リスト内の最初の子項目の **C1AutoCompleteItem1** をクリックし、**label** および **value** プロパティを探します。両方のプロパティに **c++** を入力します。残りの各 **C1AutoCompleteDataItem** の[ラベル]および[値]テキストボックスに下表に示すテキストを入力します。

C1AutoCompleteDataItem	ラベル	値
C1AutoCompleteItem2	java	java
C1AutoCompleteItem3	php	php
C1AutoCompleteItem4	coldfusion	coldfusion
C1AutoCompleteItem5	javascript	javascript
C1AutoCompleteItem6	asp	asp
C1AutoCompleteItem7	ruby	ruby
C1AutoCompleteItem8	python	python
C1AutoCompleteItem9	c	c

C1AutoCompleteItem10	scala	scala
C1AutoCompleteItem11	groovy	groovy
C1AutoCompleteItem12	haskell	haskell
C1AutoCompleteItem13	perl	perl

上記の値の入力が完了したら<OK>をクリックします。これらの **C1AutoCompleteItem** はユーザーの検索を完了させる項目です。

4. **C1AutoComplete** コントロールをクリックし、プロパティウィンドウでプロパティを表示します。**MinCount** プロパティを指定し、その値を 1 から 2 に変更します。

**AutoComplete for ASP.NET Web Forms** クイックスタートの手順2が完了しました。次の手順では、プロジェクトを実行します。

## 手順 3: アプリケーションの実行

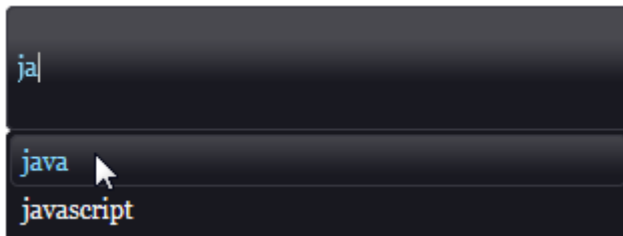
これまでは、検索ボックスの外観と動作をカスタマイズしました。残された作業は、アプリケーションを実行して、**C1AutoComplete** 検索ボックスのランタイム機能のいくつかを確認することです。

以下の手順を実行します。

1. [F5]を押して、アプリケーションを実行します。**C1AutoComplete** 検索ボックスが次の図のように表示されます。



2. 検索ボックスに検索トピックを入力し始めます。検索ボックスに「ja」と入力します。次のように表示されます。



おめでとうございます。**C1AutoComplete** が作成され、カスタマイズされ、**AutoComplete for ASP.NET Web Forms** クイックスタートが無事完了しました！

## デザイン時サポート

**AutoComplete for ASP.NET Web Forms** は、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、およびデザイン時サポートを備え、オブジェクトモデルでの作業を単純化します。

以下のセクションでは、**AutoComplete for ASP.NET Web Forms** のデザイン時環境、特に **C1AutoComplete スマートタグ** および **C1AutoComplete** コンテキストメニューからアクセス可能な **[C1AutoComplete タスク]** メニューを使用して、**C1AutoComplete** コントロールを設定する方法について説明します。

## C1AutoComplete のスマートタグ

**C1AutoComplete** コントロールはスマートタグを備えています。スマートタグは、各コンポーネント/コマンドで最もよく使われるプロパティを提供するショートカットのタスクメニューを表します。

**[C1AutoComplete タスク]** メニューにアクセスするには、**C1AutoComplete** コントロールの右上端にあるスマートタグ矢印をクリックします。**[C1AutoComplete タスク]** メニューが開きます。

C1AutoComplete タスク	
データソースの選択:	(なし) ▼
幅	<input type="text"/>
高さ	<input type="text"/>
<a href="#">オートコンプリート項目の追加</a>	
テーマ	aristo ▼
<a href="#">新しいテーマの作成...</a>	
<input type="checkbox"/> CDN の使用	
CDN パス	<input type="text" value="http://cdn.wijmo.com/"/>
<input type="checkbox"/> Bootstrap の使用	
<input type="checkbox"/> Mobile Mode	
<a href="#">バージョン情報...</a>	
<a href="#">テンプレートの編集</a>	

**[C1AutoComplete タスク]** メニューは、以下のように動作します。

- **データソースの選択**  
[データソースの選択] ドロップダウンボックスを使用して、**C1AutoComplete** コントロールのデータソースを設定できます。
- **幅**  
[幅] テキストボックスを使用して、コントロールの幅を設定できます。
- **高さ**  
[高さ] テキストボックスを使用して、コントロールの高さを設定できます。
- **オートコンプリート項目の追加**  
[オートコンプリート項目の追加] をクリックすると **[AutoComplete デザイナフォーム]** が開き、そこで **<子項目の追加>** をクリックして **C1AutoComplete** コントロールに項目を追加できます。



- **テーマ**  
[テーマ] ドロップダウンボックスをクリックすれば、各種の視覚スキームから選択できます。使用可能な視覚スタイルの詳細については、[テーマ](#)を参照してください。
- **新しいテーマの作成**  
[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。
- **CDN の使用**  
チェックボックスを ON にすると、CDN からクライアントリソースがロードされます。これはデフォルトで OFF です。
- **CDN パス**  
CDN の URL パスを表示します。
- **Bootstrap の使用**  
[Bootstrap の使用] オプションを選択すると、コントロールに Bootstrap テーマを適用することができます。アプリケーションで Bootstrap テーマを使用する方法については、[Bootstrap for ASP.NET Web Forms クイックスタート](#)を参照してください。
- **バージョン情報**  
[バージョン情報] をクリックすると、製品のバージョン情報を確認できるダイアログボックスが表示されます。
- **テンプレートの編集**  
[テンプレートの編集] 項目をクリックすると、**C1AutoComplete タスク**がテンプレート編集モードになります。

## C1AutoComplete のコンテキストメニュー

**C1AutoComplete** コントロールには、Visual Studio がすべての .NET コントロールに提供しているコンテキストメニューで利用できる追加的なコマンドがあります。**C1AutoComplete** コンテキストメニューにアクセスするには、**C1AutoComplete** コントロール上のどこかを右クリックします。**C1AutoComplete** コンテキストメニューが開きます。

	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)	
	コンテンツ ページの追加(N)	Ctrl+M, Ctrl+C
	スマート タグの表示(G)	Shift+Alt+F10
	オートコンプリート項目の追加	
	テンプレートの編集(I)	▶
	テンプレート編集の終了(N)	
	最新の情報に更新(F)	
	プロパティ(R)	Alt+Enter

**C1AutoComplete** のコンテキストメニューには、**C1AutoComplete**で追加される以下のカスタムコマンドが含まれます。

# AutoComplete for ASP.NET Web Forms

- **スマートタグの表示**  
**C1AutoComplete** コントロールのスマートタグを表示します。スマートタグの使用法と使用可能な機能についての詳細は、[C1AutoComplete のスマートタグ](#)を参照してください。
- **オートコンプリート項目の追加**  
[オートコンプリート項目の追加] 項目をクリックすると[AutoComplete デザイナフォーム]が開き、**C1AutoComplete**コントロールに子項目を追加できます。
- **テンプレートの編集**  
[テンプレートの編集]項目をクリックすると[**C1CompleteItem** タスク]のテンプレート編集モードに入り、テンプレートを作成して、**C1AutoComplete** コントロールに適用できます。

## C1AutoComplete デザイナフォーム

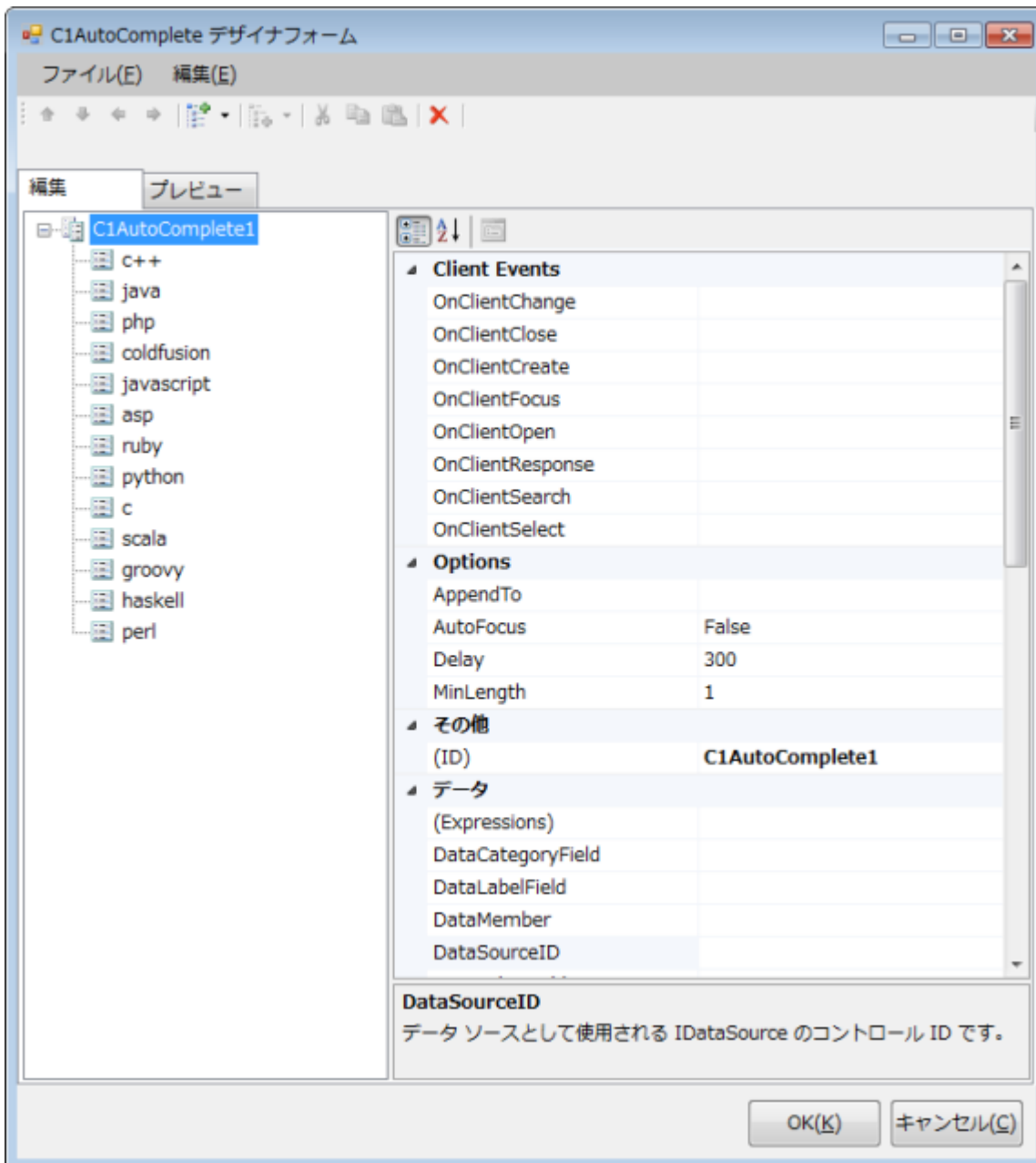
AutoComplete デザイナフォームは、AutoComplete のプロパティや **C1AutoCompleteDataItem** のプロパティを編集するための **C1AutoComplete** のデザイナーです。**AutoComplete デザイナフォーム**は、プログラマーがコントロールを視覚的に修正できる点でプロパティウィンドウに似ています。しかし、このデザイナーでは、**C1AutoCompleteDataItem** の選択、そのプロパティの設定、および**C1AutoComplete**コントロールの外観のプレビューまで、すべてをフォーム内で行うことができます。

このトピックでは、**AutoComplete デザイナフォーム**のデザインインターフェイスに精通することにより、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 **C1AutoComplete**を編集できるようになります。

**AutoComplete デザイナフォーム**を開くには、**C1AutoComplete** のスマートタグをクリックし、[**C1AutoComplete** タスク]メニューから[オートコンプリート項目の追加]リンクを選択します。

## AutoComplete デザイナフォームの機能

AutoComplete 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」タブをクリックして、プロパティを設定する**C1AutoComplete** コントロールまたは**C1AutoCompleteDataItem** を選択します。

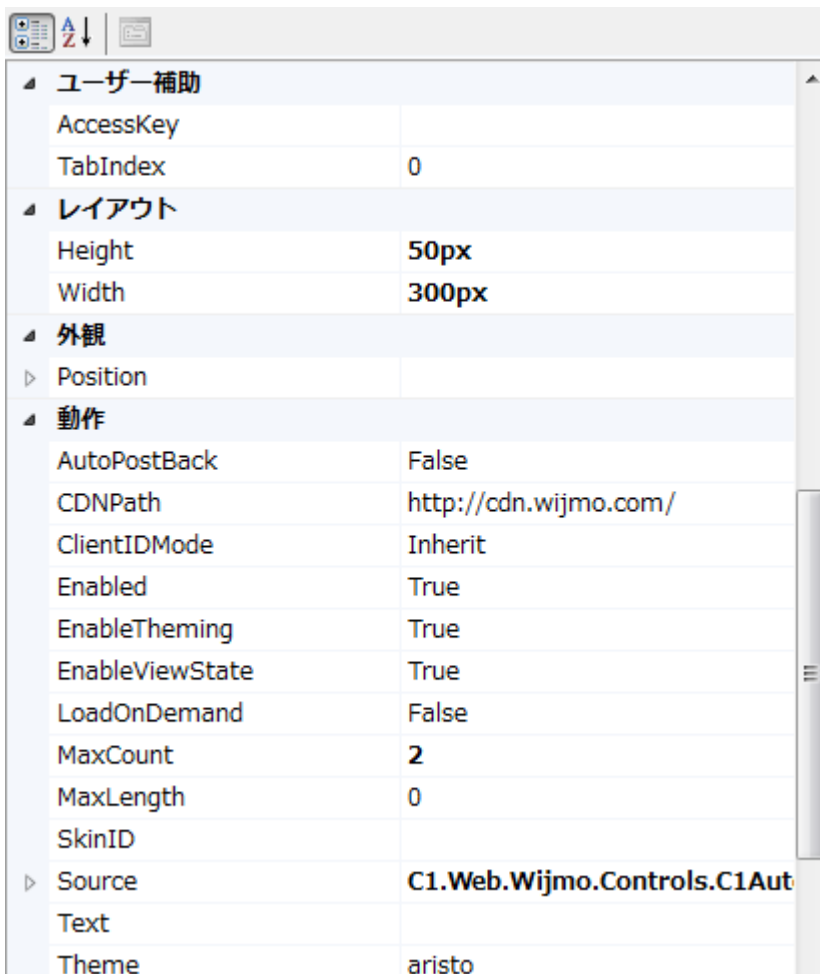
## 「プレビュー」タブ

「プレビュー」タブをクリックして、**C1AutoComplete** コントロールのプレビューを表示します。

## プロパティペイン

AutoComplete デザイナー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。プロパティウィンドウを使用して、**C1AutoComplete** コントロールまたは**C1AutoCompleteDataItem** のプロパティを変更します。

# AutoComplete for ASP.NET Web Forms



▲ ユーザー補助	
AccessKey	
TabIndex	0
▲ レイアウト	
Height	50px
Width	300px
▲ 外観	
▷ Position	
▲ 動作	
AutoPostBack	False
CDNPath	http://cdn.wijmo.com/
ClientIDMode	Inherit
Enabled	True
EnableTheming	True
EnableViewState	True
LoadOnDemand	False
MaxCount	2
MaxLength	0
SkinID	
▷ Source	C1.Web.Wijmo.Controls.C1Aut
Text	
Theme	aristo

## Command Buttons

2つのコマンドボタンには、次に示す機能があります。

ボタン	説明
OK	〈OK〉をクリックすると、新しい設定値が <b>C1AutoComplete</b> 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、 <b>AutoComplete デザイナフォーム</b> が閉じて新しい設定値は破棄され、デフォルト設定値が <b>C1AutoComplete</b> コントロールに適用されます。


## AutoComplete デザイナフォームのメニュー

AutoComplete デザイナフォームのメニューでは、次に示すメニュー項目とサブメニュー項目が利用可能です。

メニュー項目	サブメニュー項目	説明
ファイル	終了	AutoComplete デザイナフォームを閉じます。
編集	子の追加	C1AutoComplete または別の C1AutoCompleteDataItem の子として、新しい C1AutoCompleteDataItem を追加します。
	削除	選択した C1AutoCompleteDataItem を削除します。
	名前の変更	<b>C1AutoCompleteDataItem</b> の名前を変更できます。





## AutoComplete デザインフォームのツールバー

AutoComplete デザインフォームのツールバーには、**C1AutoComplete** コントロールまたは **C1AutoCompleteDataItem** の1つを選択したかどうかに応じて変わるオプションが用意されています。

 **注意:**〈項目を左に移動〉、〈項目を右に移動〉、〈項目を挿入〉、〈切り取り〉、〈コピー〉、および〈貼り付け〉ボタンは利用できません。



下の表で、使用可能なツールバーの各ボタンとそれらが使用可能になる場合について説明します。

ボタン	名前	説明
	項目を上移動	DataItem のリスト内で、選択した <b>C1AutoCompleteDataItem</b> を上に移動します。このボタンは <b>C1AutoCompleteDataItem</b> を選択すると使用可能になります。
	項目を下移動	DataItem のリスト内で、選択した <b>C1AutoCompleteDataItem</b> を下に移動します。このボタンは <b>C1AutoCompleteDataItem</b> を選択すると使用可能になります。
	子項目の追加	<b>C1AutoCompleteDataItem</b> を <b>C1AutoComplete</b> コントロールに追加します。このボタンは <b>C1AutoComplete</b> コントロールを選択すると使用可能になります。
	削除	選択した <b>C1AutoCompleteDataItem</b> または <b>C1AutoComplete</b> コントロールを削除します。このボタンは常に使用可能です。

## デザイナの使用法

以下のトピックでは、**AutoComplete デザインフォーム**を使用したいいくつかの作業について説明します。

### C1AutoCompleteDataItem の削除

デザイナで **C1AutoCompleteDataItem** 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項目の削除  
削除する **C1AutoCompleteDataItem** 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項目の削除  
削除する項目を選択し、〈削除〉をクリックします。
- [編集]メニューによる子項目の削除  
[編集]メニューをクリックし、〈削除〉を選択します。

### デザイナでの AutoCompleteDataItem の名前変更

デザイナで **C1AutoCompleteDataItem** 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。

1. [F2]の押下
  - a. 名前を変更する **C1AutoCompleteDataItem** を選択します。
  - b. [F2]キーを押し、新しい名前を入力します。
2. ショートカットメニューからの[名前の変更]の選択
  - a. 名前を変更する **C1AutoCompleteDataItem** を右クリックします。
  - b. 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。

### 3. [編集]メニューからの[名前の変更]の選択

- a. 名前を変更する **C1AutoCompleteDataItem** を選択します。
- b. [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。

## 子項目の追加

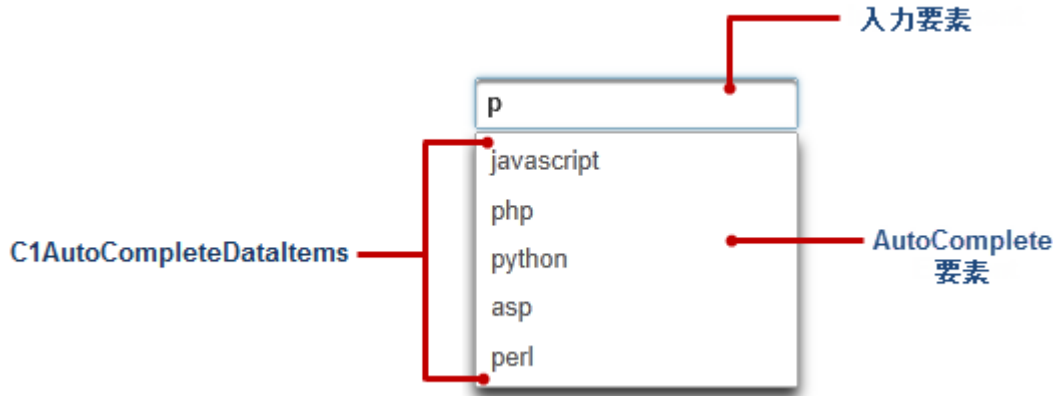
デザイナーで **C1AutoCompleteDataItem** の名前を変更するときは、次の3つの方法のいずれかを使用できます。

- **ショートカットメニューによる子項目の追加**  
デザイナーで **C1AutoCompleteDataItem** 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。
- **〈子の追加〉ボタン押下によるデータ項目の追加**  
**C1AutoComplete** コントロールを右クリックし、[子の追加]→[AutoComplete]を選択します。
- **[編集]メニューによる子項目の追加**  
**C1AutoComplete** コントロールを選択し、[編集]メニューをクリックして、[子の追加]→[AutoComplete]を選択します。

## C1AutoComplete の要素

このセクションでは、**C1AutoComplete** コントロールを構成する各要素の視覚的および記述的な概要を提供します。

**C1AutoComplete** コントロールは、入力要素と**C1AutoCompleteDataItem** を含むAutoComplete 要素で構成されます。



各 C1AutoComplete 要素の説明については、以下を参照してください。

- **入力要素**  
入力要素を使用して、ユーザーは検索する項目を入力したり、AutoComplete 要素から選択した項目を表示したりできます。
- **AutoComplete 要素**  
AutoComplete要素には、**C1AutoCompleteDataItems クラス** が含まれています。これらは、ユーザーがデザイン時に作成した項目か、または別のデータソースから取得した項目のいずれかです。
- **C1AutoCompleteDataItems**  
**C1AutoCompleteDataItem**は、デザイン時に作成するか、または外部のデータソースから呼び出すことができます。

## C1AutoComplete の外観

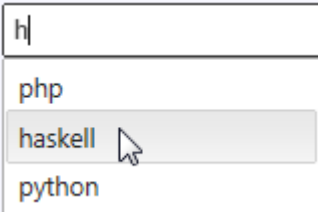
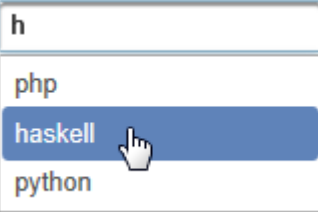
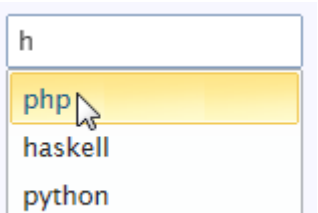
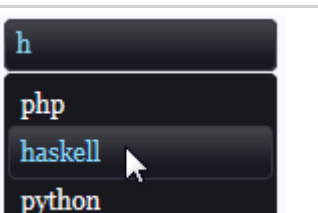
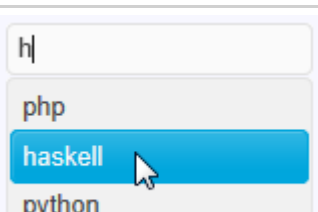
**C1AutoComplete** は、簡単にカスタマイズできるように設計されています。**C1AutoComplete** ダイアログウィンドウの外観を変更する際に、その可能性は限りなく広がります。**C1AutoComplete** には、その入力領域および組み込みテーマに関して、Office 2007 が提供するものを含め、いくつかのスタイルが用意されています。

各ダイアログウィンドウのプロパティのスタイルに加えて、**C1AutoComplete** は CSS スタイルを完全にサポートするので、ユーザーは CSS スタイルを使用して各ダイアログウィンドウをさらにカスタマイズできます。

## テーマ

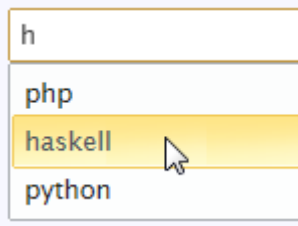
**C1AutoComplete** は、コントロール用に **Arctic**、**Aristo**、**Cobalt**、**Midnight**、**Rocket**、および **Sterling** という6種類の組み込みテーマを備えています。これらは **Theme** プロパティを設定して簡単にコントロールに適用できます。

マウスが上に置かれた状態で表示される6種類の組み込みテーマは次の通りです。

視覚スタイル	プレビュー
<b>Arctic</b>	
<b>Aristo</b>	
<b>Cobalt</b>	
<b>Midnight</b>	
<b>Rocket</b>	



Sterling



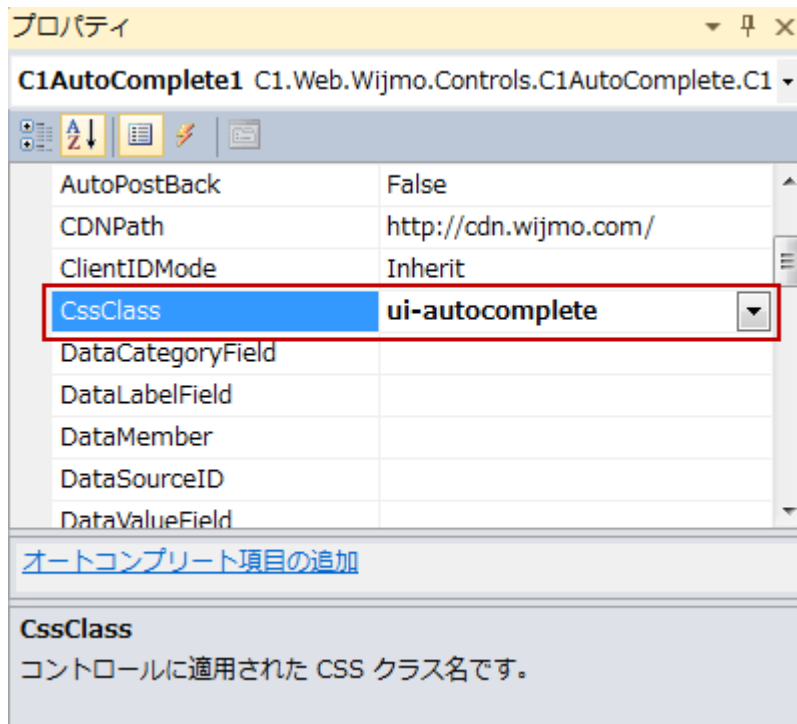
## CSS セレクタ

CSS を使用し、数多くの **C1AutoComplete** 要素のスタイルを設定して、独特の外観を生み出すことができます。カスタマイズを簡素化するために、その6種類の組み込みテーマごとに CSS セレクタが組み込まれています。

枠、背景、テキスト、フォント、マージン、埋め込み、リスト、輪郭、表などの一般的な CSS プロパティを該当する CSS セレクタに適用できます。

一般に使用される個々の CSS セレクタとグループ化された CSS セレクタのリストについては、プロジェクトの **C1AutoComplete** コントロールを選択し、Visual Studio プロパティウィンドウで **CssClass** プロパティの横にあるドロップダウンリストを表示します。

**C1AutoControl** クラス CSS セレクタは、下図に示すように ui-autocomplete から開始します。



個々の CSS セレクタをグループとして組み合わせ、CSS セレクタをより具体的かつ強力なものにすることができます。

## クライアント側での作業

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

**C1AutoComplete** コントロールが表示されると、クライアント側コントロールのインスタンスが自動的に作成されます。これにより、サーバーへポストバックする必要なしに、ユーザーは**C1AutoComplete** のプロパティとメソッドに簡単にアクセスすることができます。

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

## クライアント側イベント

AutoComplete for ASP.NET Web Forms には、複数のクライアント側イベントが含まれています。それらを利用すれば、検索ボックスでの検索などのアクションが行われたときに、**C1AutoComplete** コントロールを操作できます。

クライアント側イベントの表にリストされたサーバー側プロパティを使用して、特定のクライアント側イベントに反応する JavaScript 関数の名前を指定できます。たとえば、「Search」という JavaScript 関数を割り当てて、AutoComplete ウィンドウがサイズ変更されたときに応答させるには、**OnClientSearch** プロパティを search に設定します。

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

イベントのサーバー側プロパティ名	イベント名	説明
OnClientChange	change	コントロールが変更されたときに発生します。
OnClientClose	close	コントロールが閉じられたときに発生します。
OnClientCreate	create	コントロールが作成されたときに発生します。
OnClientFocus	focus	コントロールがフォーカスを取得するときに発生します。
OnClientOpen	open	コントロールが開かれたときに発生します。
OnClientSearch	search	検索が行われたときに発生します。
OnClientSelect	select	コントロールが選択されたときに発生します。

## タスク別ヘルプ

タスク別ヘルプセクションでは、Visual Studio ASP.NET 環境でのプログラミングに精通し、**AutoComplete for ASP.NET Web Forms** コントロールを全般的に理解しているユーザーを対象としています。

各トピックでは、**C1AutoComplete** を使用した特定のタスクのソリューションを示します。各トピックで概説されている手順に従うことによって、さまざまな C1AutoComplete 機能を使用したプロジェクトを作成できます。

タスク別ヘルプの各トピックでは、読者が新しい ASP.NET プロジェクトを作成した経験を持っていることが前提となります。

## コードによる C1AutoComplete コントロールの作成

コードによる **C1AutoComplete** コントロールの作成は簡単な作業です。このトピックでは、**PlaceHolder** コントロールをページに追加し、重要なステートメントを追加し、**C1AutoComplete** コントロールをカスタマイズして、コントロールを **PlaceHolder** に追加します。

以下の手順を実行します。

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

### Visual Basic コードの書き方

```
Visual Basic
Imports C1.Web.Wijmo.Controls.C1AutoComplete
```

### C# コードの書き方

```
C#
using C1.Web.Wijmo.Controls.C1AutoComplete;
```

4. 次のコードを **Page\_Load** イベントに追加してコントロールを作成し、コントロールを **PlaceHolder** に追加します。

### Visual Basic コードの書き方

```
Visual Basic
' 新しい C1AutoComplete を作成
C1AutoComplete C1A = new C1AutoComplete();
' コントロールを PlaceHolder に追加
PlaceHolder1.Controls.Add(C1A)
```

### C# コードの書き方

```
C#
// 新しい C1AutoComplete を作成
C1AutoComplete C1A = new C1AutoComplete();
//C1AutoComplete を PlaceHolder に追加
PlaceHolder1.Controls.Add(C1A);
```

## C1AutoComplete の幅と高さのカスタマイズ

# AutoComplete for ASP.NET Web Forms

このトピックでは、**C1AutoComplete** の幅と高さをカスタマイズする手順について説明します。これは、デザインビュー、ソースビュー、またはコードのいずれかで実行できます。

## デザインビューの場合

以下の手順を実行します。

1. プロジェクトで **C1AutoComplete** コントロールを選択し、スマートタグをクリックして、[**C1AutoComplete タスク**]メニューを開きます。
2. コントロールの **Height** プロパティを 150px に設定します。
3. コントロールの **Width** プロパティを 200px に設定します。
4. [F5]を押してプログラムを実行し、加えた変更を確認します。

## ソースビューの場合

以下の手順を実行します。

1. プロジェクトのソースビューに移動します。
2. **Height**および **Width** プロパティは両方とも<cc1:C1AutoComplete> タグ内にあります。

ソースビュー

```
<cc1:C1Complete ID="C1AutoComplete1" runat="server" Height="200px" Width="200px">
```

3. **Height** および **Width** プロパティを 200px に設定します。
4. [F5]を押して、プログラムを実行します。

## コードの場合

次のコードを **Page\_Load** イベントに追加し、**C1AutoComplete** コントロールの**Height** および **Width** プロパティを設定します。

### Visual Basic コードの書き方

Visual Basic

```
Me.C1AutoComplete1.Height = 150  
Me.AutoComplete1.Width = 200
```

### C# コードの書き方

C#

```
this.C1AutoComplete1.Height = 150;  
this.C1AutoComplete1.Width = 200;
```

## C1AutoComplete CSS セレクタでの作業

**C1AutoComplete** は、CSS スタイリングをサポートします。これにより、ユーザーは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、**C1AutoComplete** コントロールに CSS スタイリングを適用する手順について説明します。

1. デザインビューで、Visual Studio のメニューから、[表示]→[プロパティ]を選択します。プロパティウィンドウの上部にあるドロップダウンリストから、**C1AutoComplete**を選択し、CssClass プロパティを探します。
2. ドロップダウンリストを使用して適切な CSS セレクタ(このトピックでは **ui-autocomplete** セレクタ)を選択します。
3. ソースビューに切り替えて、<asp:Content> タグの最初のセットを探します。
4. <asp:Content> タグの間に <style type="text/css"></style> タグを挿入します。このタグのセットを

使って、コントロールに CSS スタイル設定を追加できます。

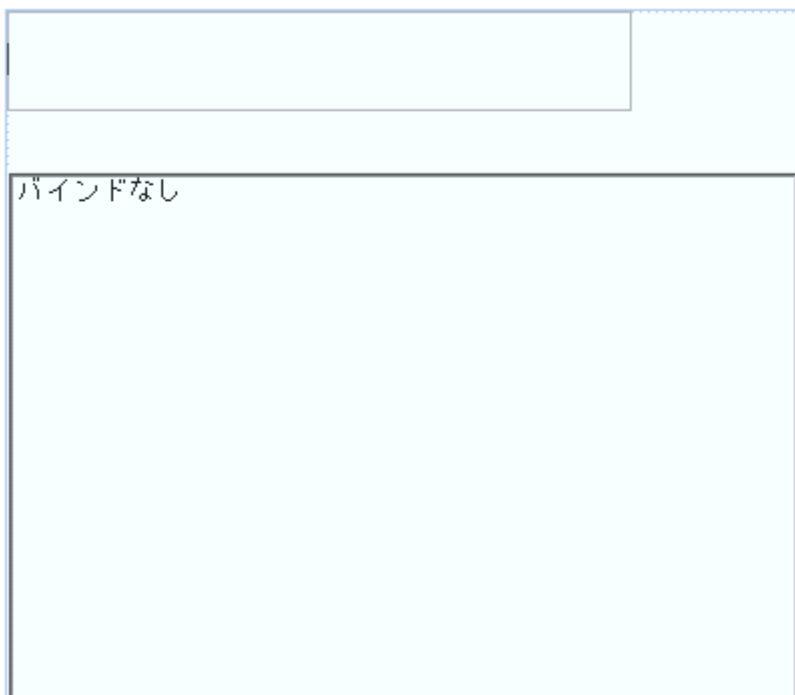
5. `<style>` タグの間に以下のスクリプトを挿入して、コントロールに CSS スタイル設定を追加します。
6. プログラムを実行します。**C1AutoComplete** コントロールが次の図のように表示されます。

## AutoPostBack プロパティの使用

**AutoPostBack** プロパティは、ユーザーがコントロールでインタラクティブ操作を行うたびにコントロールがサーバーにポストバックするかどうかを指定します。このトピックでは、**C1AutoComplete** コントロールの作成、コンテンツの設定、および **AutoPostBack** プロパティの設定を行う手順について説明します。

以下の手順を実行します。

1. **C1AutoComplete** コントロールおよび標準の **ListBox** コントロールをアプリケーションに追加します。デザインビューは、次の図のようになります。



2. ページで **C1AutoComplete** コントロールを選択し、Visual Studio のプロパティウィンドウに移動します。リストで **AutoPostBack** プロパティを探し、それを **True** に設定します。
3. **<イベント>** ボタン  をクリックし、**C1AutoComplete** イベントを表示します。**ItemSelected** イベントを指定し、**C1AutoComplete1\_ItemSelected** を入力します。
4. ソースビューに切り替えて `<cc1:C1AutoComplete>` `</cc1:C1AutoComplete>` タグを探します。タグの間に次のマークアップを挿入し、**C1AutoCompleteDataItem** を作成します。

ソースビュー

```
<Items>
  <cc1:C1AutoCompleteDataItem Label="c++" Value="c++" />
  <cc1:C1AutoCompleteDataItem Label="java" Value="java" />
  <cc1:C1AutoCompleteDataItem Label="php" Value="php" />
  <cc1:C1AutoCompleteDataItem Label="coldfusion" Value="coldfusion" />
  <cc1:C1AutoCompleteDataItem Label="javascript" Value="javascript" />
  <cc1:C1AutoCompleteDataItem Label="asp" Value="asp" />
  <cc1:C1AutoCompleteDataItem Label="ruby" Value="ruby" />
  <cc1:C1AutoCompleteDataItem Label="python" Value="python" />
  <cc1:C1AutoCompleteDataItem Label="c" Value="c" />
</Items>
```

# AutoComplete for ASP.NET Web Forms

```
<cc1:C1AutoCompleteDataItem Label="scala" Value="scala" />
<cc1:C1AutoCompleteDataItem Label="groovy" Value="groovy" />
<cc1:C1AutoCompleteDataItem Label="haskell" Value="haskell" />
<cc1:C1AutoCompleteDataItem Label="perl" Value="perl" />
</Items>
```

5. `<asp:ListBox></asp:ListBox>` タグを探し、次のマークアップを挿入して、**ListBox** コントロールの幅と高さを制御します。

ソースビュー

```
Width="300px" Height="300px"
```

6. ソースビューを右クリックし、リストから[**コードの表示**]を選択します。以前に `ItemSelected` イベントを処理するために Visual Studio で作成された次のようなコードがあることを確認します。

## Visual Basic コードの書き方

Visual Basic

```
Protected Sub C1AutoComplete1_ItemSelected(ByVal sender As Object, ByVal args As
C1.Web.Wijmo.Controls.C1AutoComplete.C1AutoCompleteDataItemEventArgs)
End Sub
```

## C# コードの書き方

C#

```
protected void C1AutoComplete1_ItemSelected(object sender,
C1.Web.Wijmo.Controls.C1AutoComplete.C1AutoCompleteDataItemEventArgs args)
```

7. 次のコードを `ItemSelected` イベントハンドラの下に配置します。このコードは、**C1AutoComplete** コントロールから `ListBox` コントロールへの新しい項目の追加を処理します。

## Visual Basic コードの書き方

Visual Basic

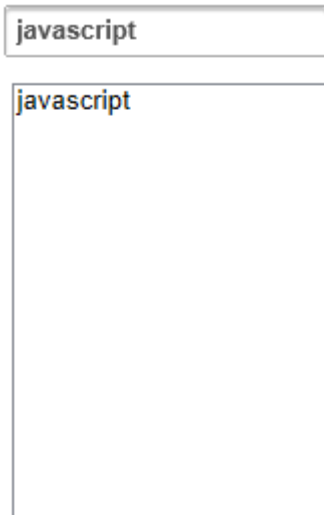
```
ListBox1.Items.Add(New ListItem(C1AutoComplete1.Text))
```

## C# コードの書き方

C#

```
ListBox1.Items.Add(new ListItem(C1AutoComplete1.Text));
```

8. [F5]を押して、アプリケーションを実行します。検索ボックスに「ja」と入力し、表示される `AutoCompleteDataItem` の1つを選択します。項目を選択し、**AutoPostBack** プロパティを確認します。アプリケーションは、次の図のようになります。



## DataBinding の使用

**C1AutoComplete** は、ラベル、値、およびカテゴリフィールドを希望するデータソースにバインドできるデータバインディングをサポートします。

以下の手順を実行します。

1. ソリューションエクスプローラで、**App\_Data** フォルダを右クリックして、リストから[追加]→[既存の項目]を選択します。
2. C:\Northwind データベースの格納場所に移動しNwind.mdbを選択して、[追加]をクリックします。
3. プロジェクトで **C1AutoComplete** コントロールを選択します。
4. スマートタグボタンをクリックし、[**C1AutoComplete タスク**]メニューを開きます。
5. [データソースの選択]ドロップダウンリストの矢印をクリックして、リストから[新しいデータソース]を選択します。これによって、[データソース構成ウィザード]が開きます。
6. リストから **Sql データベース** を選択して、<OK>をクリックします。データ接続の選択ウィンドウで、データ接続として **接続文字列** を選択し、<次へ>をクリックします。
7. **Select ステートメントの構成**ウィンドウで、[テーブルまたはビューから列を指定する]の下にあるドロップダウンリストから[商品]を選択します。
8. <クエリーのテスト>をクリックし、データベースが正しくクエリーされていることを確認して、<完了>をクリックします。
9. [F5]を押して、アプリケーションを実行します。検索ボックスに何文字か入力して、AutoComplete 機能をテストします。

## LoadOnDemand プロパティの使用

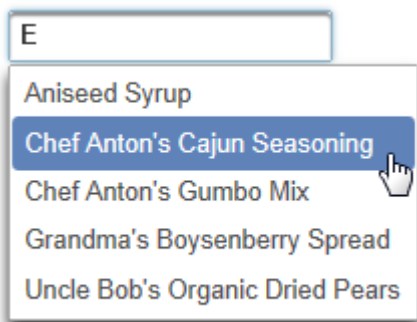
**LoadOnDemand** プロパティを使用して、データを動的に取得できます。子項目は、ユーザーが検索ボックスに入力を始めるとロードされます。

以下の手順を実行します。

1. ソリューションエクスプローラで、**App\_Data** フォルダを右クリックして、リストから[追加]→[既存の項目]を選択します。
2. Northwind データベースの格納場所に移動してNwind.mdbを選択し、<追加>をクリックして、プロジェクトにデータベースを追加します。
3. プロジェクトで **C1AutoComplete** コントロールを選択します。
4. スマートタグボタンをクリックし、[**C1AutoComplete タスク**]メニューを開きます。
5. [データソースの選択]ドロップダウンリストの矢印をクリックして、リストから[新しいデータソース]を選択します。これによって、[データソース構成ウィザード]が開きます。

# AutoComplete for ASP.NET Web Forms

6. リストから Sql データベースを選択して、〈OK〉をクリックします。**データ接続の選択**ウィンドウで、ドロップダウンメニューを使用して、データ接続として **C1Nwind.mdb** を選択し、〈次へ〉をクリックします。
7. **Select ステートメントの構成**ウィンドウで、[**テーブルまたはビューから列を指定する**]の下にあるドロップダウンリストから[**商品**]を選択します。
8. 〈**クエリーのテスト**〉をクリックし、データベースが正しくクエリーされていることを確認して、〈**完了**〉をクリックします。
9. プロジェクトで **C1AutoComplete** コントロールを選択し、プロパティウィンドウに移動します。
10. プロパティウィンドウで**LoadOnDemand** プロパティを探し、それを **True** に設定します。
11. そのままプロパティウィンドウで、**MaxCount** プロパティを探し、それを **5** に設定します。このプロパティは、AutoComplete での検索結果の数を制限し、**LoadOn Demand** プロパティとともに使用する必要があります。
12. [F5]を押してアプリケーションを実行し、検索ボックスに「E」と入力します。結果は、次の図のようになります。




## C1AutoComplete でのテーマの使用

**C1AutoComplete** には、6種類の組み込み CSS テーマが用意されています。CDN を使用するか、または jQuery ThemeRoller を使って独自のテーマを作成することで、さまざまなテーマを選択することもできます。このトピックでは、組み込みテーマの適用、CDN を使用したカスタムテーマの選択、または ThemeRoller を使った独自のテーマの作成を行う手順について説明します。

## スマートタグを使用した組み込みテーマの適用

スマートタグを使用して組み込みテーマのいずれかを適用して、簡単に AutoComplete 検索ボックスをカスタマイズできます。以下の手順を実行します。

1. プロジェクトで **C1AutoComplete** コントロールを選択します。
2. スマートタグボタン  をクリックして、[**C1AutoComplete タスク**]メニューを開きます。
3. [**テーマ**]ドロップダウンボックスを使用して新しいテーマを選択します。




The screenshot shows the 'C1AutoComplete タスク' (C1AutoComplete Task) pane. The 'データソースの選択' (Data Source Selection) is set to 'SqlDataSource1'. The 'テーマ' (Theme) dropdown menu is open, showing a list of themes: arctic, aristo (highlighted), cobalt, midnight, rocket, and sterling. Other options like '幅' (Width), '高さ' (Height), and 'CDN の使用' (Use CDN) are also visible.

4. [F5]または<デバッグ開始>を押して、プログラムを実行します。選択したテーマが AutoComplete コントロールに適用されます。

## カスタムテーマの追加

以下の手順を実行します。

1. **C1AutoComplete**のスマートタグをクリックして、[C1AutoComplete タスク]メニューを開きます。
2. [テーマ]ドロップダウンボックスで、URL を入力してテーマを指定します。URL は<http://blog.jqueryui.com/2012/06/jquery-ui-1-8-21/>で確認できます。この例では、次の le-frog テーマを使用します：<http://jquery-ui.googlecode.com/svn/tags/1.8.21/themes/le-frog/>

The screenshot shows the 'C1AutoComplete タスク' (C1AutoComplete Task) pane. The 'データソースの選択' (Data Source Selection) is set to 'SqlDataSource1'. The 'テーマ' (Theme) dropdown menu is set to 'j/jquery.ui.theme.css'. The 'CDN の使用' (Use CDN) checkbox is checked, and the 'CDN パス' (CDN Path) is set to 'http://cdn.wijmo.com/'. Other options like '幅' (Width), '高さ' (Height), and 'バージョン情報...' (Version Information...) are also visible.

このテーマ設定は **Web.config** ファイルの <appSettings> に格納されています。ソリューションエクスプローラ

# AutoComplete for ASP.NET Web Forms

で、**Web.config** ファイルをダブルクリックします。<appSettings> タグに **WijmoTheme** キーと値が含まれていることに注意してください。これは、追加した CDN URL が指定されている場所です。

3. プロジェクトを実行して、テーマが **C1AutoComplete** コントロールに適用されていることを確認します。

