

Input for ASP.NET Web Forms

2020.08.25 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: フォームへのコントロールの追加 | 5 |
| 手順 2: コントロールの外観の変更 | 5-6 |
| 手順 3: コントロールの書式設定 | 6-7 |
| 手順 4: カルチャ設定の追加 | 7-9 |
| 手順 5: クイックスタート Web アプリケーションの実行 | 9 |
| デザイン時のサポート | 10 |
| C1Input の各スマートタグ | 10 |
| C1InputMask スマートタグ | 10-12 |
| C1InputDate スマートタグ | 12-13 |
| C1InputNumeric スマートタグ | 13-15 |
| C1InputPercent スマートタグ | 15-16 |
| C1InputCurrency スマートタグ | 16-18 |
| C1Input コンテキストメニュー | 18 |
| C1Input の各デザイナー | 18 |
| C1InputMask C1Input デザイナフォーム | 18-19 |
| C1InputDate C1Input デザイナフォーム | 19-20 |
| C1InputMask の使い方 | 21 |
| C1InputMask の定義 | 21-23 |
| C1InputDate の使い方 | 24 |
| C1InputDate の定義 | 24-28 |
| 和暦の設定 | 28-29 |
| C1InputNumeric の使い方 | 30 |
| C1InputNumeric の定義 | 30 |
| C1InputPercent の使い方 | 31 |
| C1InputPercent の定義 | 31 |
| C1InputCurrency の使い方 | 32 |
| C1InputCurrency の定義 | 32 |

| | |
|--|-------|
| C1Input の外観 | 33 |
| テーマ | 33 |
| C1Input CSS セレクタ | 33-34 |
| クライアント側の機能 | 35 |
| クライアント側イベント | 35 |
| タスク別ヘルプ | 36 |
| C1InputMask タスク | 36 |
| プロンプト文字列の変更 | 36-37 |
| IP アドレスマスクの作成 | 37-38 |
| 電話番号マスクの作成 | 38 |
| プロンプト文字列を含まない日付マスクの表示 | 38-39 |
| フォーカス消失時のプロンプト文字列の非表示化 | 39-40 |
| C1InputDate タスク | 40 |
| 日付書式パターンと日付の設定 | 40-42 |
| 空の日付値の表示 | 42-43 |
| C1InputNumeric タスク | 43 |
| 小数点以下の桁数の指定 | 43-44 |
| 最小値／最大値の設定 | 44-45 |
| テーマの変更 | 45-46 |
| カスタムテーマの追加 | 46-47 |
| カルチャの選択 | 47-48 |
| クライアント側イベントタスク | 49 |
| 無効な入力の入力時のツールチップの表示 | 49 |
| トリガーを使用したカスタム UI の表示 | 49-51 |

製品の概要

Input for ASP.NET Web Forms では、データエントリと検証コントロールの完全なコレクションが提供されます。マスク、日付、数値、およびカスタムの編集から選択します。組み込みマスク、カスタム書式サポート、ローカライズ、その他多くの機能を使用できます。

Input for ASP.NET Web Forms を構成する各コントロールは、以下のとおりです。

- **C1InputCurrency**
C1InputCurrency コントロールは、**C1InputNumeric** から派生した通貨値編集専用コントロールです。この数値エディタを使用すると、アプリケーションでカスタム検証論理を記述せずに入力を指定できます。
詳細については、「[C1InputCurrency の使い方](#)」を参照してください。
- **C1InputDate**
C1InputDate コントロールは、**C1InputMask** から派生した日付／時刻編集専用コントロールです。**C1InputDate** コントロールは、日付エディタを表示します。
詳細については、「[C1InputDate の使い方](#)」を参照してください。
- **C1InputMask**
C1InputMask コントロールは、テキストフォーム内ですべてのデータ型の情報の入力と編集に使用するメイン Web コントロールです。このコントロールは、データの書式設定、マスクの編集、データ検証などの機能をサポートします。また、追加機能を含む、すべてのデータ型の書式およびマスクの編集もサポートします。メインデータの編集コントロールであることに加え、**C1InputMask** は、**C1InputDate** や **C1InputNumeric** などの専用コントロールの基本クラスとしても機能します。
詳細については、「[C1InputMask の使い方](#)」を参照してください。
- **C1InputNumeric**
C1InputNumeric コントロールは、**C1InputMask** から派生した数値の編集専用コントロールです。この数値エディタを使用すると、アプリケーションでカスタム検証論理を記述せずに入力を指定できます。
詳細については、「[C1InputNumeric の使い方](#)」を参照してください。
- **C1InputPercent**
C1InputPercent コントロールは、**C1InputNumeric** から派生したパーセント値編集専用コントロールです。数値エディタを使用すると、アプリケーションでカスタム検証論理を記述せずに入力を指定できます。
詳細については、「[C1InputPercent の使い方](#)」を参照してください。

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

Input for ASP.NET Web Forms の主な特長として、次の事項が挙げられます。

- **充実したクライアント側オブジェクトモデル**
マウスアクションやコントロールフォーカスイベントによって、さらに充実したクライアント側のユーザー操作性を作成します。多数のクライアント側メソッドによって、コントロールの書式、色、マスク、最小値／最大値、その他多くを変更します。
- **23 種類以上の組み込みマスク**
23 種類以上の組み込みマスクから選択するか、独自のマスクをカスタマイズします。**C1InputMask** コントロールは、日付時刻書式、曜日チューザ、および数値範囲など、14 種類の組み込み標準マスクを備えています。**C1InputDate** コントロールは、長い／短い日付時刻書式など、9種類の組み込み標準マスクを備えています。どちらもカスタム書式サポートが含まれています。
- **無効な入力に対する警告**
数値入力ボックス内の英数字などの無効な入力を排除します。赤色のフォントを使用してユーザーに視覚的に警告したり、エラーメッセージを表示したりできます。
- **豊富な書式モデル**
可能なほぼすべての書式で入力ボックスを設定します。豊富な書式モデルによって、開発者はコントロールのテキスト、枠線、セルスペース、カラースキームなどの外観をカスタマイズできます。
- **デザイン時のサポート**
SmartTag を使用して、5つのコントロール固有のデザイナーにすばやくアクセスできます。マスク、書式、値、およびカルチャの各プロパティを設定し、WYSIWYG ウィンドウで編集を視覚化します。詳細については、「[デザイン時のサポート](#)」を参照してください。
- **カルチャサポート**
任意の入力コントロールによって使用されるカルチャ設定を定義します。これは文字列比較、数値および日付時刻書式、特殊文字に適用されます。詳細については、「[カルチャの選択](#)」を参照してください。
- **ドロップダウンとスピンボタン**
日付／時刻と数値編集専用の入力コントロールである、**C1InputDate**、**C1InputCurrency**、および**C1InputNumeric** コントロールは、ドロップダウンとスピン(上／下)ボタンをサポートします。
- **日付ピッカー**
カレンダーを **C1InputDate** コントロールの日付ピッカーとして使用できます。トリガーボタンを有効にしてデフォルトカレンダーを開くには、単に **ShowTrigger** プロパティを **True** に設定します。
- **数値編集**
C1InputNumeric、**C1InputCurrency**、および **C1InputPercent** コントロールに対し、値の入力制限を設定して、小数点以下の桁数を指定し、桁区切り文字の使用を指定します。
- **パスワード保護**
C1InputMask コントロールを使用し、パスワード文字を表示させて入力テキストを保護します。デザイン時、文字(* など)を単に選択して、実際の入力文字の代用にします。
- **キーボードサポート**
キーボードサポートを使用して、入力値をすばやく編集します。カーソル位置を1つ左／右や先頭／末尾に移動させたり、範囲値を増減したり、コピー&ペースト、その他多くの操作ができます。
- **テーマ**
スマートタグをクリックするだけで、6種類のプレミアムテーマ(Arctic、Midnight、Aristo、Rocket、Cobalt、および Sterling)のいずれかを選択して外観を変更します。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。詳細については、「[C1Input の外観](#)」を参照してください。
- **CSS のサポート**
(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。

クイックスタート

このセクションでは、**C1Input** コントロールを使用する Web フォームの作成方法について紹介します。また、コントロールの外観、書式、および機能の変更方法についても示します。ヘルプの各手順に従えば、リッチでユーザーフレンドリな Web フォームを作成できます。

簡潔にするために、クイックスタートでは**C1InputMask**、**C1InputDate**、および **C1InputCurrency** コントロールについて説明することに注意してください。**C1InputNumeric** および **C1InputPercent** コントロールは、**C1InputCurrency** コントロールと同様のプロパティを共有しているため、このクイックスタートでは取り扱いません。

手順 1: フォームへのコントロールの追加

まず、ASP.NET 対応 Web サイトを作成し、C1Input コントロールをツールボックスに追加します。まず、ASP.NET プロジェクトを作成し、**C1Input** コントロールをツールボックスに追加します。

新規 Web フォームを設定するには、以下の手順を実行します。

1. ドキュメントウィンドウのすぐ下の「**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(必要な場合)。
2. このページに、2列3行のテーブルを追加します([**テーブル**]メニューから[**テーブルの挿入**]を選択)。1番目の列はテキストに使用し、2番目の列は **C1Input** コントロールに使用します。テーブルがフォームに表示されます。
3. ツールボックスから、ドラッグ & ドロップ操作を実行して、以下の各コントロールをテーブルの2番目の列のセルに配置して、ページに追加します。
 - **C1InputMask**
 - **C1InputDate**
 - **C1InputCurrency**

テーブルは、次のような表示になります。

| | |
|--|-----------|
| | |
| | 6/30/2011 |
| | \$0.00 |

4. テキストをテーブルに追加します。この例では、それぞれ**商品番号**、**注文日**、および**単価**を追加します。必要に応じて、テーブルのサイズを変更したり、書式を設定したりできます。テキストが追加されたテーブルは、次のような表示になります。

| | |
|-----------------|-----------|
| Product Number: | |
| Order Date: | 6/30/2011 |
| Unit Price | \$0.00 |

5. ソースビューに切り替えます。デザインビューでテーブルとテキストを追加して作成した HTML が表示されます。**C1Input** コントロールが Web フォームに正常に追加されました。次のトピックでは、入力ボックスの外観を変更する方法を示します。

手順 2: コントロールの外観の変更

このトピックでは、テーマを使用して **C1Input** コントロールの外観を変更する方法を示します。以下の手順を実行します。

1. ドキュメントウィンドウのすぐ下の「**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。

- 1 番目のコントロール、C1InputMask を選択して、スマートタグ (🔗) をクリックします。[C1InputMask タスク]メニューが表示されます。
- [タスク]メニューで、[テーマ]の横のドロップダウン矢印をクリックし、**rocket** を選択します。その他の **C1Input** コントロールも **rocket** テーマで更新されます。更新されたコントロールの外観は、次のような表示になります。

| | |
|-------|------------|
| 商品番号: | |
| 注文日: | 2012/07/25 |
| 単価: | ¥0.00 |

4. ソースビューに切り替えます。デザインビューでスキームを変更して作成した HTML が表示されます。

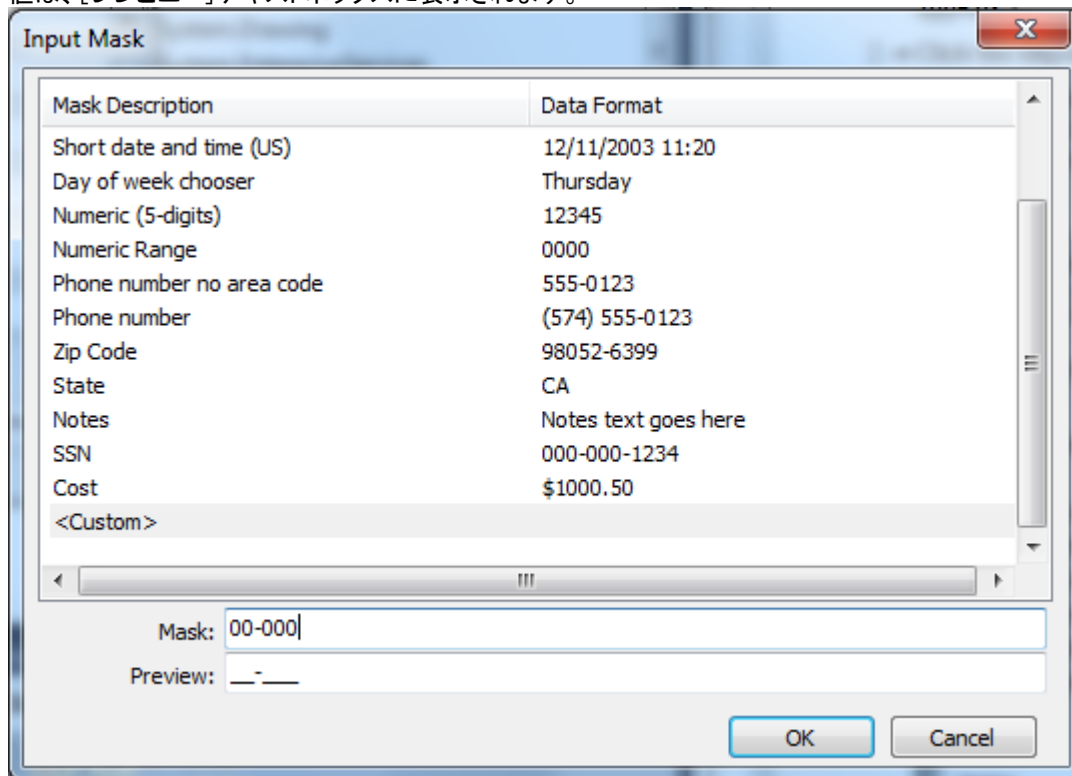
C1Input コントロールの外観が正常に変更されました。次のトピックでは、入力ボックスの書式を設定する方法を示します。

手順 3: コントロールの書式設定

このトピックでは、[タスク]メニューを使用して、コントロールを書式設定する方法を示します。まず、ドキュメントウィンドウのすぐ下の「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。以下の手順に従って、Web フォーム上の **Input for ASP.NET Web Forms** の各コントロールを書式設定します。

C1InputMask コントロールを書式設定するには、以下の手順を実行します。

1. **C1InputMask** を選択して、スマートタグ (🔗) をクリックします。[C1InputMask タスク]メニューが表示されます。
2. [マスク]プロパティの横の<...>ボタンをクリックします。[定型入力]ダイアログボックスが表示されます。
3. [マスク]テキストボックスに 00-000 を入力します。マスクの入力を開始すると、[マスクの説明]列が <ユーザー設定> に自動的に切り替わる(入力したマスクがマスクのリストになかった場合)ことに注意してください。マスク値からの出力値は、[プレビュー]テキストボックスに表示されます。



4. <OK>をクリックして、[定型入力]ダイアログボックスを閉じます。
5. [表示]→[プロパティウィンドウ]を選択し、InvalidClass プロパティの横の ui-state-error に注意します。ユーザーが英数字などの無効な入力を入力した場合、入力の色は、CSS の ui-state-error に指定された、無効なエントリを識別

Input for ASP.NET WebForms

する赤で表示されます。

C1InputDate コントロールを書式設定するには、以下の手順を実行します。

1. **C1InputDate** を選択して、スマートタグ()をクリックします。**[C1InputDate タスク]**メニューが表示されます。
2. 日付書式を**[日付書式]**テキストボックスに入力します。この例では、**D**を使用します。標準書式文字は、次のとおりです。

| プリセットパターン | 名前 |
|-----------|---------------------|
| d | 短い日付パターン |
| D | 長い日付パターン |
| t | 短い時刻パターン |
| T | 長い時刻パターン |
| F | 完全な日付/時刻パターン(短い時刻) |
| g | 一般的な時刻パターン(短い時刻) |
| G | 一般的な日付/時刻パターン(長い時刻) |
| U | 世界時のソート可能な日付/時刻パターン |

[結果の日付パターン]テキストボックスは自動的に更新されます。

3. **[C1InputDate タスク]**メニューで、**[日付]**の横のドロップダウン矢印をクリックして、ドロップダウンカレンダーから日付を選択します。
4. Visual Studio のプロパティウィンドウで**ShowSpinner** プロパティを指定し、ドロップダウン矢印をクリックして **True** を選択します。

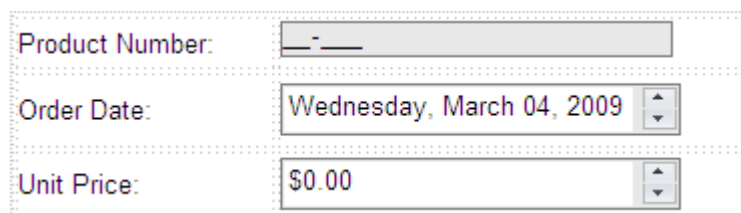
 **注意:** Visual Studio のプロパティウィンドウの **Height** と **Width** プロパティを使用して、コントロールをサイズ変更する必要がある場合があります。

C1InputCurrency コントロールを書式設定するには、以下の手順に従います。

1. **C1InputCurrency** コントロールを選択して、Visual Studio のプロパティウィンドウで **ShowSpinner** プロパティを表示します。
2. ドロップダウン矢印をクリックして、**True** を選択します。

C1Input コントロールの書式が正常に変更されました。作成した HTML を表示するには、ソースビューに切り替えます。

更新されたコントロールの外観は、次のような表示になります(各コントロールの幅が **200px** に設定されていることに注意してください)。



The screenshot shows three controls within a dotted border. The first is a text box labeled 'Product Number' containing '1-____'. The second is a calendar control labeled 'Order Date' showing 'Wednesday, March 04, 2009'. The third is a spinner control labeled 'Unit Price' showing '\$0.00'.

次のトピックでは、ボタンを追加して **C1InputDate** と **C1InputCurrency** コントロールのカルチャ情報を変更する方法を示します。

手順 4:カルチャ設定の追加

このトピックでは、コードを **Button_Click** イベントに追加して、**C1InputDate** と **C1InputCurrency** コントロールの **Culture** を設定する方法を例示します。これを行うには、以下の手順を実行します。

- ドキュメントウィンドウのすぐ下の「**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。
- ツールボックスから、**Button** コントロールを選択して、ドラッグ&ドロップ操作で Web フォーム(テーブルのすぐ下)に配置します。この手順を繰り返して、2番目の **Button** コントロールを Web フォームに追加します。
- これで、2つの **Button** コントロールが横に並んでフォームに配置されました。**プロパティ**ウィンドウで、いくつかの基本設定を変更します。

| Button1 のプロパティ: | Button2 のプロパティ: |
|--------------------------|----------------------------|
| ((ID)) = JapanBtn | (ID) = USEnglishBtn |
| Text = 日本語 | Text = 英語(アメリカ) |
| Height = 25px | Height = 25px |
| Width = 130px | Width = 140px |

- 〈**日本語**〉ボタンをダブルクリックして、ボタンの **Click** イベントのイベントハンドラを作成します。**JapanBtn_Click** イベントに、次のコードを入力します。

Visual Basic コードの書き方

```

Visual Basic

Protected Sub FrenchBtn_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles FrenchBtn.Click
    C1InputDate1.Culture = New System.Globalization.CultureInfo("fr-FR")
    C1InputCurrency1.Culture = New System.Globalization.CultureInfo("fr-FR")
End Sub
    
```

C# コードの書き方

```

C#

protected void FrenchBtn_Click(object sender, System.EventArgs e)
{
    C1InputDate1.Culture = new System.Globalization.CultureInfo("fr-FR");
    C1InputCurrency1.Culture = new System.Globalization.CultureInfo("fr-
FR");
}
    
```

- 〈**英語(アメリカ)**〉ボタンをダブルクリックして、ボタンの **Click** イベントのイベントハンドラを作成します。**USEnglishBtn_Click** イベントに、次のコードを入力します。

Visual Basic コードの書き方

```

Visual Basic

Protected Sub USEnglishBtn_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles USEnglishBtn.Click
    C1InputDate1.Culture = New System.Globalization.CultureInfo("en-US")
    C1InputCurrency1.Culture = New System.Globalization.CultureInfo("en-US")
End Sub
    
```

C# コードの書き方

```

C#

protected void USEnglishBtn_Click(object sender, System.EventArgs e)
{
    
```

Input for ASP.NET WebForms

```
C1InputDate1.Culture = new System.Globalization.CultureInfo("en-US");
    C1InputCurrency1.Culture = new System.Globalization.CultureInfo("en-US");
}
```

カルチャ情報を含む2つのボタンコントロールが Web フォームに正常に追加されました。更新された Web フォームの外観は、次のような表示になります。

| | |
|---|---|
| 商品番号: | <input type="text" value="--"/> |
| 注文日: | <input type="text" value="2012年7月25日"/> |
| 単価: | <input type="text" value="¥0.00"/> |
| <input type="button" value="日本語"/> <input type="button" value="英語 (アメリカ)"/> | |

次のトピックでは、アプリケーションを実行する方法を示します。Web フォームの機能を確認する際に実行するタスクも示します。

手順 5: クイックスタート Web アプリケーションの実行

〈**デバッグ開始**〉ボタンをクリックして、アプリケーションを実行します。クイックスタートのメイン手順(手順 1~4)を実行した後のクイックスタート Web フォームは、次のような表示になります。

| | |
|--|---|
| 商品番号: | <input type="text" value="--"/> |
| 注文日: | <input type="text" value="2012年7月25日"/> |
| 単価: | <input type="text" value="¥0.00"/> |
| <input type="button" value="日本語"/> <input type="button" value="英語(アメリカ)"/> | |

変更を確認するには、次のタスクを実行します。


- [商品番号] 入力ボックスに数値を入力します。数字が有効です。英数字(a など)を入力してみます。入力ボックスがこれを許容しないことに注意してください。
- [注文日] (C1InputDate コントロール) 入力を変更するには、次のタスクを実行します。
 - マウスポインタで、上/下スピンボタンをクリックします。
 - [注文日] 入力ボックス内でクリックして、キーボードの上/下矢印を押します。
- [単価] (C1InputCurrency コントロール) 入力を変更するには、次のタスクを実行します。
 - マウスポインタで、上/下スピンボタンをクリックします。
 - [単価] 入力ボックス内でクリックして、キーボードの上/下矢印を押すか、現在の単価を選択して新しい単価を入力します。
- C1InputDate と C1InputCurrency コントロールのカルチャを日本語に変更するには、〈日本語〉ボタンをクリックします。
- C1InputDate と C1InputCurrency コントロールのカルチャを英語(アメリカ)に戻すには、〈英語(アメリカ)〉ボタンをクリックします。

おめでとうございます！3種類の C1Input コントロールを含む基本的な Web フォームが正常に作成されました。また、コントロールがカスタマイズされ、カルチャ情報が組み込まれて Web フォームの性能も向上しました。

デザイン時のサポート

C1Input は、視覚的な編集を提供しており、Web 入力コントロールの作成がより簡単になります。以下のセクションでは、C1Input で使用可能な各種のサポートについて詳しく説明します。

タスクメニューの起動

Visual Studio では、**C1Input** の各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、各コントロールで最もよく使用されるプロパティを提供するショートカットの**タスクメニュー**を表します。コントロールの右上端のスマートタグ () をクリックして、各コントロールの**タスクメニュー**を呼び出すことができます。**C1Input** の各コントロールのスマートタグの使用方法についての詳細は、「[C1Input の各スマートタグ](#)」を参照してください。


コンテキストメニューの起動

各 **C1Input** コンポーネントは、関連付けられたコンテキストメニューを使用して、デザイン時に簡単に設定できます。**C1Input** のコンテキストメニューについての詳細は、「[C1Input コンテキストメニュー](#)」を参照してください。

C1Input コントロールのプロパティの表示

C1Input の任意のコントロールのプロパティにアクセスするには、コントロールを右クリックして、[**プロパティ**]を選択するか、**プロパティ**ウィンドウのドロップダウンリストボックスからクラスを選択するだけです。

C1Input の各スマートタグ

Visual Studio では、**C1Input** の各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、各コントロールで最もよく使用されるプロパティを提供するショートカットの**タスクメニュー**を表します。

以下の各トピックでは、**C1Input** コントロールの各スマートタグについて紹介します。

C1InputMask スマートタグ

C1InputMask 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最も一般的な **C1InputMask** のプロパティにすばやく簡単にアクセスできます。

[**C1InputMask タスク**]メニューにアクセスするには、**C1InputMask** コントロールの右上端のスマートタグ () をクリックします。[**C1InputMask タスク**]メニューが開きます。

Input for ASP.NET WebForms



[C1InputMask タスク]メニューは、次のように操作します。

デザイナー

[デザイナー]をクリックすると、[C1InputMask C1Input デザイナフォーム]が開きます。デザイナーについての詳細は、「C1InputMask C1Input デザイナフォーム」を参照してください。

プロパティ

C1InputMask コントロールの最もよく使用されるプロパティです。[C1InputMask タスク]メニューには、以下の各プロパティが表示されます。

- **テキスト**
Text ボックスには、ユーザーに表示するテキストを入力します。
- **マスク**
マスクボックスの(<...>)ボタンをクリックすると、[定型入力]ダイアログボックスが表示されます。書式付きマスクから選択するか、カスタムマスクを入力できます。
- **プロンプト文字列**
PromptChar ボックスには、コントロールにユーザー入力がないときに表示するプロンプト文字列を入力します。デフォルトは、アンダースコア(_)です。
- **パスワード文字列**
PasswordChar ボックスには、マスクを指定した C1InputMask コントロールで実際の入力文字の代用となる文字を入力します。
- **カルチャ**
Culture ボックスのドロップダウン矢印をクリックして、カルチャを選択します。各カルチャには、日付、時刻、数字、通貨、およびその他の情報を表示するためのさまざまな表記規則があります。
- **テーマ**
Theme プロパティのドロップダウン矢印をクリックして、組み込みテーマの1つを選択し、コントロールの外観を変更します。
- **新しいテーマの作成**
[新しいテーマの作成]オプションをクリックすると、ThemeRoller for Visual Studio が開きます。したがって、開発環境内でテーマをカスタマイズすることができます。アプリケーションで ThemeRoller for Visual Studio を使用する方

法については、「[ThemeRoller for Visual Studio](#)」を参照してください。

- **CDN の使用**
[**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。

また、「[C1InputMask の使い方](#)」もあわせて参照してください。

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。

C1InputDate スマートタグ

C1InputDate 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最も一般的な **C1InputDate** のプロパティにすばやく簡単にアクセスできます。

[**C1InputDate タスク**]メニューにアクセスするには、**C1InputDate** コントロールの右上端のスマートタグ () をクリックします。[**C1InputDate タスク**]メニューが開きます。



| C1InputDate タスク | |
|-------------------------------------|-----------------------|
| プロパティ | |
| 日付 | 2014/01/08 13:48 |
| 日付書式 | d |
| 表示書式 | |
| カルチャ | ja-JP |
| テーマ | aristo |
| 新しいテーマの作成... | |
| <input checked="" type="checkbox"/> | CDN の使用 |
| CDN パス | http://cdn.wijmo.com/ |
| <input type="checkbox"/> | BootStrap の使用 |
| <input type="checkbox"/> | Mobile Mode |
| バージョン情報... | |

[**C1InputDate タスク**]メニューは、次のように操作します。

デザイナー

[**デザイナー**]をクリックすると、[**C1InputDate C1Input デザイナーフォーム**]が開きます。デザイナーについての詳細は、「[C1InputDate C1Input デザイナーフォーム](#)」を参照してください。

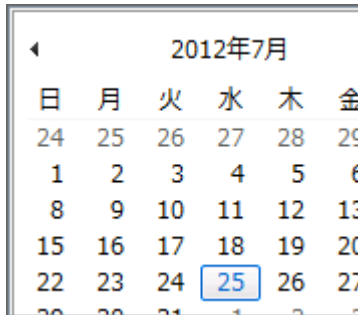
プロパティ

Input for ASP.NET WebForms

C1InputDate コントロールの最もよく使用されるプロパティです。[**C1InputDate タスク**]メニューには、以下の各プロパティが表示されます。

- **日付**

カレンダーから日付を選択するには、[**Date**]ボックスに日付を入力するか、ドロップダウン矢印をクリックします。



- **日付書式**

[**DateFormat**]ボックスには、日付書式パターンを入力します。デフォルト値は d です。

- **カルチャ**

[**Culture**]ボックスのドロップダウン矢印をクリックして、カルチャを選択します。各カルチャには、日付、時刻、数字、通貨、およびその他の情報を表示するためのさまざまな表記規則があります。

- **テーマ**

Theme プロパティのドロップダウン矢印をクリックして、組み込みテーマの1つを選択し、コントロールの外観を変更します。

- **新しいテーマの作成**

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

- **CDN の使用**

[**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。


また、「[C1InputDate の使い方](#)」もあわせて参照してください。

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。

C1InputNumeric スマートタグ

C1InputNumeric コントロールでは、スマートタグから最も一般的な **C1InputNumeric** のプロパティにすばやく簡単にアクセスできます。

[**C1InputNumeric タスク**]メニューにアクセスするには、**C1InputNumeric** コントロールの右上端のスマートタグ () をクリックします。[**C1InputNumeric タスク**]メニューが開きます。

| C1InputNumeric タスク | |
|--|--|
| プロパティ | |
| 値 | <input type="text" value="0"/> |
| 最小有効値 | <input type="text" value="-1000000000"/> |
| 最大有効値 | <input type="text" value="1000000000"/> |
| 小数点以下桁数 | <input type="text" value="2"/> |
| カルチャ | <input type="text" value="ja-JP"/> |
| テーマ | <input type="text" value="aristo"/> |
| 新しいテーマの作成... | |
| <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 | |
| バージョン情報... | |

[C1InputNumeric タスク]メニューは、次のように操作します。

プロパティ

C1InputNumeric コントロールの最もよく使用されるプロパティです。[C1InputNumeric タスク]メニューには、以下の各プロパティが表示されます。

- **値**
[Value]ボックスには、ユーザーに表示する数値を入力します。
- **最小有効値**
[MinValue]ボックスには、ユーザーが入力できる最小値を入力します。
- **最大有効値**
[MaxValue]ボックスには、ユーザーが入力できる最大値を入力します。
- **小数点以下桁数**
[DecimalPlaces]ボックスには、表示する小数点以下の桁数を入力します。デフォルト値は 2 です。
- **カルチャ**
[Culture]ボックスのドロップダウン矢印をクリックして、カルチャを選択します。各カルチャには、日付、時刻、数字、通貨、およびその他の情報を表示するためのさまざまな表記規則があります。
- **テーマ**
Theme プロパティのドロップダウン矢印をクリックして、組み込みテーマの 1 つを選択し、コントロールの外観を変更します。
- **新しいテーマの作成**
[新しいテーマの作成]オプションをクリックすると、ThemeRoller for Visual Studio が開きます。したがって、開発環境内でテーマをカスタマイズすることができます。アプリケーションで ThemeRoller for Visual Studio を使用方法については、「[ThemeRoller for Visual Studio](#)」を参照してください。
- **CDN の使用**
[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**

Input for ASP.NET WebForms

[**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。

また、「[C1InputNumeric の使い方](#)」もあわせて参照してください。

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。

C1InputPercent スマートタグ

C1InputPercent コントロールでは、スマートタグから最も一般的な **C1InputPercent** のプロパティにすばやく簡単にアクセスできます。

[**C1InputPercent タスク**] メニューにアクセスするには、**C1InputPercent** コントロールの右上端のスマートタグ () をクリックします。[**C1InputPercent タスク**] メニューが開きます。

| C1InputPercent タスク | |
|--|--|
| プロパティ | |
| 値 | <input type="text" value="0"/> |
| 最小有効値 | <input type="text" value="-1000000000"/> |
| 最大有効値 | <input type="text" value="1000000000"/> |
| 小数点以下桁数 | <input type="text" value="2"/> |
| カルチャ | <input type="text" value="ja-JP"/> |
| テーマ | <input type="text" value="aristo"/> |
| 新しいテーマの作成... | |
| <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 | |
| バージョン情報... | |

[**C1InputPercent タスク**] メニューは、次のように操作します。

プロパティ

C1PercentEdit コントロールの最もよく使用されるプロパティです。[**C1InputPercent タスク**] メニューには、以下の各プロパティが表示されます。

- **値**
[**Value**] ボックスには、ユーザーに表示する数値を入力します。
- **最小有効値**
[**MinValue**] ボックスには、ユーザーが入力できる最小値を入力します。
- **最大有効値**
[**MaxValue**] ボックスには、ユーザーが入力できる最大値を入力します。
- **小数点以下桁数**

[**DecimalPlaces**]ボックスには、表示する小数点以下の桁数を入力します。デフォルト値は 2 です。

- **カルチャ**

[**Culture**]ボックスのドロップダウン矢印をクリックして、カルチャを選択します。各カルチャには、日付、時刻、数字、通貨、およびその他の情報を表示するためのさまざまな表記規則があります。

- **テーマ**

Theme プロパティのドロップダウン矢印をクリックして、組み込みテーマの1つを選択し、コントロールの外観を変更します。

- **新しいテーマの作成**

[**新しいテーマの作成**]オプションをクリックすると、**ThemeRoller for Visual Studio** が開きます。したがって、開発環境内でテーマをカスタマイズすることができます。アプリケーションで **ThemeRoller for Visual Studio** を使用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**

[**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。

また、「[C1InputNumeric の使い方](#)」もあわせて参照してください。

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。

C1InputCurrency スマートタグ

C1InputCurrency コントロールでは、スマートタグから、最も一般的なメソッドやプロパティにすばやく簡単にアクセスできます。

[**C1InputCurrency タスク**]メニューにアクセスするには、**C1InputCurrency** コントロールの右上端のスマートタグ () をクリックします。[**C1InputCurrency タスク**]メニューが開きます。

Input for ASP.NET WebForms

| C1InputCurrency タスク | |
|--|--|
| プロパティ | |
| 値 | <input type="text" value="0"/> |
| 最小有効値 | <input type="text" value="-1000000000"/> |
| 最大有効値 | <input type="text" value="1000000000"/> |
| 小数点以下桁数 | <input type="text" value="2"/> |
| カルチャ | <input type="text" value="ja-JP"/> |
| テーマ | <input type="text" value="aristo"/> |
| 新しいテーマの作成... | |
| <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 | |
| バージョン情報... | |

[C1InputCurrency タスク]メニューは、次のように操作します。

プロパティ

C1InputCurrency コントロールの最もよく使用されるプロパティです。[C1InputCurrency タスク]メニューには、以下の各プロパティが表示されます。

- **値**
[Value]ボックスには、ユーザーに表示する数値を入力します。
- **最小有効値**
[MinValue]ボックスには、ユーザーが入力できる最小値を入力します。
- **最大有効値**
[MaxValue]ボックスには、ユーザーが入力できる最大値を入力します。
- **小数点以下桁数**
[DecimalPlaces]ボックスには、表示する小数点以下の桁数を入力します。デフォルト値は 2 です。
- **カルチャ**
[Culture]ボックスのドロップダウン矢印をクリックして、カルチャを選択します。各カルチャには、日付、時刻、数字、通貨、およびその他の情報を表示するためのさまざまな表記規則があります。
- **テーマ**
Theme プロパティのドロップダウン矢印をクリックして、組み込みテーマの 1 つを選択し、コントロールの外観を変更します。
- **新しいテーマの作成**
[新しいテーマの作成]オプションをクリックすると、ThemeRoller for Visual Studio が開きます。したがって、開発環境内でテーマをカスタマイズすることができます。アプリケーションで ThemeRoller for Visual Studio を使用方法については、「[ThemeRoller for Visual Studio](#)」を参照してください。
- **CDN の使用**
[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。








また、「[C1InputNumeric の使い方](#)」もあわせて参照してください。

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。

C1Input コンテキストメニュー


各 **C1Input** コントロールは、デザイン時に使用する追加機能のコンテキストメニューを備えています。**C1Input** コントロールのどれかを右クリックすると、次のような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) | |
|  | ブラウザで表示(B) | |
| | スマート タグの表示(G) | Shift+Alt+F10 |
|  | 最新の情報に更新(F) | |
|  | プロパティ(R) | Alt+Enter |

C1Input の各デザイナー

C1Input は、マスクや日付書式を簡単に指定できる、**C1InputMask** と **C1InputDate** の各デザイナーを備えています。以下の各トピックでは、これらのデザイナーについて説明します。

C1InputMask C1Input デザイナーフォーム

[**C1InputMask C1Input デザイナーフォーム**] を表示するには、**C1InputMask** コントロールの右上端のスマートタグ () をクリックして、[**デザイナー**] を選択します。次のようなデザイナーが表示されます。

Input for ASP.NET WebForms



「定型入力エディタ」タブ

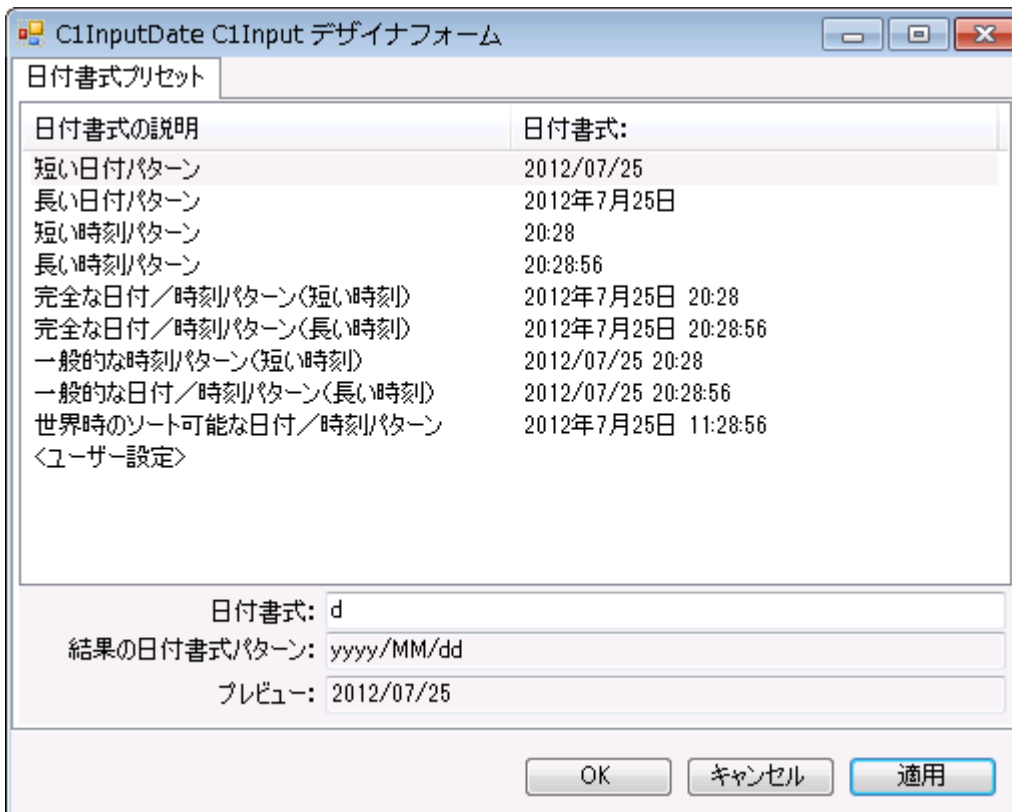
デザイナーの「**定型入力エディタ**」タブには、**C1InputMask** コントロールのマスクオプションが表示されます。[**マスク**]テキストボックスは、1つ以上のプレースホルダ(0、9、# など)とリテラル(電話番号の局番を囲むカッコなど)で構成されるマスク文字列を表示します。[**プレビュー**]ボックスは、Web ブラウザに表示されるマスクの外観を表示します。

注意: すべての入力マスクが存在しない値から保護されるわけではありません。たとえば、プリセットされた9桁郵便番号マスクは、00000~00000 のような郵便番号が存在しなくても、この入力値を許容します。同様に、状態省略時のプリセットマスクにより、PD はそのような状態が存在しない場合でも通過させることができます。それでも、プリセットマスクは、明らかな不正入力に対する防御の最前線として役立ちます。

C1InputDate C1Input デザイナフォーム

[**C1InputDate C1Input デザイナフォーム**]を表示するには、**C1InputDate** コントロールの右上端のスマートタグ(🔗)をクリックして、[**デザイナー**]を選択します。

次のようなデザイナーが表示されます。



「日付書式プリセット」タブ

デザイナの「日付書式プリセット」タブには、**C1InputDate** コントロールの日付書式オプションが表示されます。[日付書式] ボックスは、プレースホルダ(ddd)とリテラル文字(区切り文字など)で構成される日付書式パターンを表示します。[プレビュー] ボックスは、Web ブラウザに表示されるマスクの外観を表示します。

C1InputMask の使い方

C1InputMask コントロールは基本的に、ユーザー入力の有効／無効を識別するマスクを使用する高度な **TextBox** コントロールです。これは、テキスト形式で任意のデータ型の情報を入力／編集するために使用するメイン Web コントロールです。**C1InputMask** は、**C1InputDate** と **C1InputNumeric** コントロールの基本クラスとして機能します。次の図は、電話番号 **MaskFormat** を持つ **C1InputMask** コントロールを示しています。



MaskFormat プロパティを使用すると、アプリケーション内にカスタム検証論理を記述せずに、以下の入力を指定できます。

- マスクリテラル(**C1InputMask** コントロールに直接表示される文字)。例: 電話番号のハイフン(-)
- マスク内の特定の位置に要求される入力タイプ。例: 数字または英字
- カスタム入力文字

主な利点

C1InputMask の主な利点は、次のとおりです。

- **C1InputMask** の大部分の基本プロパティとメソッドは、テキスト入力時の動作が **System.Windows.Forms.MaskedTextBox** コントロールと同じであるため、習得が簡単。**C1InputMask** の各プロパティとメソッドは、追加機能によって区別されます。
- **C1Input** コントロール間でコピー＆ペーストが可能
- キーボードサポート:
 - 左／右矢印: カーソルを左／右位置に1つ移動します。
 - [Home]／[End]: カーソルを先頭または末尾に移動します。
 - 上／下矢印: 列挙／数値範囲の場合、列挙／数値範囲の値を増減します。
 - [Del]／[Back Space]: 列挙／数値範囲の場合、列挙／数値範囲の値を初期値に設定します。
 - [Ctrl]+[C]／[Ctrl]+[V]: コピー＆ペーストキーボードショートカットのサポート
- **C1InputMask** に特定のカルチャを選択可能。例: 英語、スペイン語、ドイツ語、ロシア語など。
- クライアントスクリプトからオンザフライで **C1InputMask** の大部分のプロパティを変更可能
- クライアント側イベントが使用可能なため、ポストバックが不要となり Web フォームのパフォーマンスが向上

C1InputMask の定義

C1InputMask コントロールは、有効なユーザー入力と無効なユーザー入力を区別するマスクを使用します。マスクを定義するには、**C1InputMask** スマートタグまたは **C1InputMask** デザイナなどの視覚的なデザイナを使用するか、プログラムで **C1InputMask** オブジェクトを使用します。

よく使用される **C1InputMask** タスクについては、「[C1InputMask タスク](#)」トピックを参照してください。

C1InputMask のマスクタイプ

下の表に、マスクのいくつかの例とその動作を示します。

| マスク | 動作 |
|---------------|--|
| 00/00/0000 | 国際的な日付書式の日付(日、数字月、年)。「/」文字は、論理日付区切り記号で、アプリケーションの現在のカルチャに適切な日付区切り記号としてユーザーに表示されます。日付パターンを指定する場合、日付や時刻を入力するための充実したインターフェースを提供する C1InputDate コントロールを使用することに注意してください。 |
| 00->L<LL-0000 | 米国書式の日付(日、省略月名、および年)。大文字の語頭の後に小文字が続く3文字の省略月名を表示します。 |
| (999) 000- | 米国電話番号、局番(オプション)。ユーザーがオプション文字を入力しないときは、スペースを入力するか、 |

| | |
|--------------|---|
| 0000 | 最初の0で示される位置にマウスポインタを直接入れることができます。 |
| \$999,999.00 | 0~999999 範囲の通貨値。通貨、桁区切り、および小数点の各文字は実行時に、カルチャ固有の等価に置換されます。 |

MaskFormat は、**C1InputMask** コントロールのデフォルトプロパティです。編集マスクを定義する場合、コントロール内の各文字位置は特殊プレースホルダまたはリテラル文字のいずれかにマップします。リテラル文字(リテラル)は、使用されているデータのタイプについて視覚的なキューを提供します。たとえば、電話番号の局番を囲むカッコとダッシュは、リテラルです。例: (412) 123-4567。マスクの編集機能により、無効な文字がコントロールに入力されないようにして、ユーザーインターフェースをさらに強化することができます。

C1InputMask 文字

マスク入力を有効にするには、**MaskFormat** プロパティを、1つまたは複数のプレースホルダとリテラルで構成されるマスク文字列に設定します。下の表に、使用可能なプレースホルダを示します。

| マスク要素 | 説明 |
|-------|---|
| 0 | 数字(必須)。この要素は 0~9 の1桁の数字を受け付けます。 |
| 9 | 数字またはスペース(オプション) |
| # | 数字またはスペース(オプション)。マスク内でこの位置が空白のとき、 Text プロパティではスペースとして表示されます。プラス(+)とマイナス(-)符号が許容されます。 |
| L | 文字(必須)。入力を ASCII 文字 a-z と A-Z に制限します。このマスク要素は、正規表現の [a-zA-Z] と等価です。 |
| ? | 文字(オプション)。入力を ASCII 文字 a-z と A-Z に制限します。このマスク要素は、正規表現の [a-zA-Z]? と等価です。 |
| & | 文字(必須) |
| C | 文字(オプション)。任意の非制御文字。 |
| A | 英数字(オプション) |
| . | 小数点プレースホルダ。使用される実際の表示文字は、 Culture プロパティに適切な小数点プレースホルダとなります。 |
| , | 3桁プレースホルダ。使用される実際の表示文字は、 Culture プロパティに適切な3桁プレースホルダとなります。 |
| : | 時刻区切り文字。使用される実際の表示文字は、 Culture プロパティに適切な時刻プレースホルダとなります。 |
| / | 日付区切り文字。使用される実際の表示文字は、 Culture プロパティに適切な日付プレースホルダとなります。 |
| \$ | 通貨記号。使用される実際の表示文字は、 Culture プロパティに適切な通貨記号となります。 |
| < | シフトダウン。後続のすべての文字を小文字に変換します。 |
| > | シフトアップ。後続のすべての文字を大文字に変換します。 |
| | 直前のシフトアップまたはシフトダウンを無効にします。 |
| H | すべての半角文字 |
| K | 半角カタカナ |
| 9 | 全角の数字 |
| K | 全角カタカナ |

Input for ASP.NET WebForms

| | |
|------------|--|
| J | ひらがな |
| Z | すべての全角文字 |
| \ | エスケープ。マスク文字をエスケープして、リテラルに変えます。「\\」は、バックスラッシュのエスケープシーケンスです。 |
| その他のすべての文字 | リテラル。すべての非マスク要素は、 C1InputMask 内にそのまま表示されます。リテラルは常に、実行時にマスク内の静的位置を占め、ユーザーによって移動または削除できません。 |

C1InputMask が前のマスクでフィルタされたユーザー入力をすでに含んでいる場合に、マスクを変更すると、C1InputMask はその入力を新しいマスク定義に移植しようとします。

MaskFormat プロパティを設定するには、以下の手順に従います。

1. **C1InputMask** コントロールを選択し、そのスマートタグをクリックして、[**C1InputMask タスク**]メニューを開きます。
2. [**マスク**]プロパティの横の〈...〉ボタンをクリックします。[**定型入力**]ダイアログボックスが表示されます。

| マスクの説明 | データ書式 |
|--------------|------------------|
| 時刻(米国) | 11:20 |
| 時刻(欧州/軍隊) | 23:20 |
| 短い日付 | 12/11/2003 |
| 短い日付と時刻(米国) | 12/11/2003 11:20 |
| 数値(5桁) | 12345 |
| 電話番号(市外局番なし) | 555-0123 |
| 電話番号 | (574) 555-0123 |
| 郵便番号 | 98052-6399 |
| 州 | CA |
| メモ | メモテキストをここに挿入 |
| SSN | 000-000-1234 |
| 費用 | \$1000.50 |
| <ユーザー設定> | |

マスク: 90:00
プレビュー: _:_

OK キャンセル

3. **データ書式**を選択して、[マスク]テキストボックスにマスクを定義します。[プレビュー]テキストボックスにマスクのプレビューが表示されることに注意してください。
4. 〈OK〉をクリックして、[**定型入力**]ダイアログボックスを閉じます。

C1InputDate の使い方

C1InputDate コントロールは、**C1InputMask** から派生した日付／時刻編集専用コントロールです。日付固有のマスクの編集フィールドを使用して、ユーザーは、日付をコントロールに直接入力するか、上／下矢印キーを使用して現在のフィールドの値を増減できます。次の図は、**C1InputDate** コントロールを示しています。

2012/07/25

DateFormat プロパティを使用すると、アプリケーション内にカスタム検証論理を記述せずに、以下の入力を指定できます。

- マスクリテラル(**C1InputDate** コントロールに直接表示される文字)。例: 時間のコロン(:)、日付の区切り文字(/)。
- マスク内の特定の位置に要求される入力タイプ。例: 数字または英字
- カスタム入力文字

主な利点

C1InputDate の主な利点は、次のとおりです。

- **C1InputDate** コントロールは、日付エディタを表示します。**DateFormat** プロパティを使用して、日付書式文字またはパターンを設定または取得します。
- **C1InputDate** コントロールは、**C1Calendar** コントロールをインタラクティブ操作するように設定できます。**Calendar** プロパティを使用して **C1InputDate** を **C1Calendar** と統合します。
- **C1InputDate** に特定のカルチャを選択可能。例: 英語、スペイン語、ドイツ語、ロシア語など。日付パターン、その他の日付文字列の側面は、選択された **Culture** プロパティによって決まります。
- クライアント側イベントが使用可能なため、ポストバックが不要となり Web フォームのパフォーマンスが向上

C1InputDate の定義

マスクを定義するには、**C1InputDate** スマートタグまたは **C1InputDate** **C1Input** デザイナフォームなどの視覚的なデザイナを使用するか、プログラムで **C1InputDate** オブジェクトを使用します。

実行時にユーザーが日付を編集する際は、以下の点に注意します。

- 文字列形式で表された書式付きフィールド(長い日付パターンの月名または曜日名など)は、キーボードで数値として入力でき、それらの文字列表現は自動的に更新されます。
- 上／下矢印を使用して、現在のフィールドを増減できます。

C1InputDate の一般プロパティ

下の表に、**C1InputDate** コントロールの一般プロパティを示します。

| プロパティ | 説明 |
|-------------------------|--|
| Date | DateTime 値 |
| DateFormat | 日付書式パターンまたは日付書式文字(プリセット文字) |
| DateFormatResultPattern | カルチャによって決まる結果の日付書式パターン(取得のみ) |
| Placeholder | 空の状態に表示されるテキストを示します。 |
| WebCalendar | C1InputDate コントロールとインタラクティブ操作する C1WebCalendar コントロールを取得または設定します。 |
| DisplayFormat | コントロールがフォーカスを失ったときに表示する日付書式パターン |

C1InputDate 書式文字

Input for ASP.NET WebForms

C1InputDate 書式文字は、大文字と小文字を区別します。下の表に、標準書式文字を示します。


| プリセットパターン | 名前 |
|-----------|---------------------|
| d | 短い日付パターン |
| D | 長い日付パターン |
| t | 短い時刻パターン |
| T | 長い時刻パターン |
| F | 完全な日付／時刻パターン(短い時刻) |
| g | 一般的な時刻パターン(短い時刻) |
| G | 一般的な日付／時刻パターン(長い時刻) |
| U | 世界時のソート可能な日付／時刻パターン |

C1InputDate 書式パターン

C1InputDate パターンは、大文字と小文字を区別します。下の表に、標準パターンを示します。

| 書式パターン | 説明 |
|--------|--|
| d | 月の日。1桁の日は、前にゼロが付きません。 |
| dd | 月の2桁の日。1桁の日は、前にゼロが付きます。 |
| ddd | 曜日の省略名 |
| dddd | 曜日の完全名 |
| M | 数字月。1桁の月は、前にゼロが付きません。 |
| MM | 数字月。1桁の月は、前にゼロが付きます。 |
| MMM | 月の省略名 |
| MMMM | 月の完全名 |
| y | 下2桁の年(西暦)。下2桁の年(西暦)が10未満の場合、年は前にゼロを付けずに表示されます。 |
| yy | 下2桁の年(西暦)。下2桁の年(西暦)が10未満の場合、年は前にゼロを付けて表示されます。 |
| yyyy | 4桁の年(0000~9999) |
| h | 12時間表示の時。1桁の時は、前にゼロが付きません。 |
| hh | 12時間表示の時。1桁の時は、前にゼロが付きます。 |
| H | 24時間表示の時。1桁の時は、前にゼロが付きません。 |
| HH | 24時間表示の時。1桁の時は、前にゼロが付きます。 |
| m | 分。1桁の分は、前にゼロが付きません。 |
| mm | 分。1桁の分は、前にゼロが付きます。 |
| s | 秒。1桁の秒は、前にゼロが付きません。 |
| ss | 秒。1桁の秒は、前にゼロが付きます。 |
| t | AM/PM 指定子の最初の文字 |

| | |
|---------------------------|--|
| tt | AM/PM 指定子 |
| e | 年号を1桁以上の数値で表示します(1-??)。 |
| ee | 年号を2桁以上の数値で表示します(01-??)。 |
| eee(さらに任意の個数の文字「e」を付加可能) | 「ee」と同様に機能します。 |
| g | 年号をアルファベットで表示します(M、T、S、H)。 |
| gg | 年号の最初の文字(DBCS)を表示します(明、大、昭、平) |
| ggg | 完全な年号(DBCS)を表示します(明治、大正、昭和、平成)。 |
| gggg(さらに任意の個数の文字「g」を付加可能) | 書式文字列に他の「g」または「gg」または「ggg」パターンがない場合は、「ggg」と同様に機能します。 |

 **注意:** パターン内の文字がシングルクォーテーションマーク(')で囲まれている場合、これらの文字はリテラルとして処理されます。たとえば、日付 03.07.2006 のパターン 'dd:' dd.MM.yyyy は、文字列「dd: 03.07.2006」を出力します。

C1InputDate 表示書式パターン

| キーワード | 説明 |
|---------------------------|--|
| e | 年号を1桁以上の数値で表示します(1-??)。 |
| ee | 年号を2桁以上の数値で表示します(01-??)。 |
| eee(さらに任意の個数の文字「e」を付加可能) | 「ee」と同様に機能します。 |
| E | 和暦年を2桁で表示し開始年を元と表示(先頭に0を付けない)(例:平成元年) |
| EE | 和暦年を2桁で表示し開始年を元と表示(先頭に0を付ける)(例:平成元年) |
| d | 先頭に0を付けない数字で日を表示します(1 - 31)。 |
| dd | 先頭に0を付けた数字で日を表示します(01 - 31)。 |
| ddd | 現在のスレッドに関連付けられた DateTimeFormatInfo.AbbreviatedDayNames() プロパティまたは指定された書式プロバイダによって定義された日を表示します。 |
| dddd | 現在のスレッドに関連付けられた DateTimeFormatInfo.DayNames() プロパティまたは指定された書式プロバイダによって定義された日を表示します。 |
| dddd(さらに任意の個数の文字「d」を付加可能) | 「dddd」と同様に機能します。 |
| g | 年号をアルファベットで表示します(M、T、S、H)。 |
| gg | 年号の最初の文字(DBCS)を表示します(明、大、昭、平)。 |
| ggg | 完全な年号(DBCS)を表示します(明治、大正、昭和、平成)。 |
| gggg(さらに任意の個数の文字「g」を付加可能) | 「ggg」と同様に機能します。 |
| h | 先頭に0を付けない数字で12時間制の時間を表示します(0 - 11)。 |
| hh | 先頭に0を付けた数字で12時間制の時間を表示します(00 - 11)。 |
| hhh(さらに任意の個数の文字「h」を付加可能) | 「hh」と同様に機能します。 |

Input for ASP.NET WebForms

| | |
|----------------------------|---|
| H | 先頭に0を付けない数字で 24 時間制の時間を表示します (0 - 23)。 |
| HH | 先頭に0を付けた数字で 24 時間制の時間を表示します (00 - 23)。 |
| HHH(さらに任意の個数の文字「H」を付加可能) | 「HH」と同様に機能します。 |
| m | 先頭に0を付けない数字で分を表示します (0 - 59)。 |
| mm | 先頭に0を付けない数字で分を表示します (00 - 59)。 |
| mmm(さらに任意の個数の文字「m」を付加可能) | 「mm」と同様に機能します。 |
| M | 先頭に0を付けない数字で月を表示します (1 - 12)。 |
| MM | 先頭に0を付けた数字で月を表示します (01 - 12)。 |
| MMM | 現在のスレッドに関連付けられた <code>DateTimeFormatInfo.AbbreviatedMonthNames()</code> プロパティまたは指定された書式プロバイダによって定義された月を表示します。 |
| MMMM | 現在のスレッドに関連付けられた <code>DateTimeFormatInfo.MonthNames()</code> プロパティまたは指定された書式プロバイダによって定義された月を表示します。 |
| MMMMM(さらに任意の個数の文字「M」を付加可能) | 「MMMM」と同様に機能します。 |
| s | 先頭に0を付けない数字で秒を表示します (0 - 59)。 |
| ss | 先頭に0を付けた数字で秒を表示します (00 - 59)。 |
| sss(さらに任意の個数の文字「s」を付加可能) | 「ss」と同様に機能します。 |
| t | A.M./P.M. 指定子の最初の文字を表示します。 |
| tt | A.M./P.M. 指定子を表示します。 |
| ttt(さらに任意の個数の文字「t」を付加可能) | 「tt」と同様に機能します。 |
| y | 先頭に0を付けない数字で年を表示します (0 - 99)。 |
| yy | 先頭に0を付けた数字で年を表示します (00 - 99)。 |
| yyy | 先頭に0を付けない数字で年を表示します (1 - 9999)。 |
| yyyy | 4桁の数字で年を表示します (0001 - 9999)。 |
| yyyyy(さらに任意の個数の文字「y」を付加可能) | 「yyyy」と同様に機能します。 |
| A | AD 表現「A.D.」を表示します。 |
| d | ShortDatePattern |
| D | LongDatePattern |
| f/F | 完全な日時(長い日付と短い時刻) |
| g | 一般(短い日付と短い時刻) |
| G | 一般(短い日付と長い時刻) |
| m/M | MonthDayPattern |
| r/R | RFC1123Pattern |

| | |
|-----|----------------------------------|
| s | SortableDateTimePattern |
| t | shortTimePattern |
| T | LongTimePattern |
| u | UniversalSortableDateTimePattern |
| U | グリニッジ標準時による完全な日時(長い日付と長い時刻) |
| y/Y | YearMonthPattern |

和暦の設定

「C1InputDate コントロール」で使用する和暦設定の定義について解説します。

和暦設定

「C1InputDate コントロール」で利用する和暦設定をカスタマイズすることができます。ページ上に和暦定義が存在する場合、wijinputdate ウィジェットではページ上設定を参照します。ページ上に和暦定義が存在しない場合はデフォルトの設定を有効にします。

和暦の定義方法

以下に和暦定義の設定例を示します。wijinputdate ウィジェットのデフォルトは下記設定と同じです。

和暦定義の設定例

```
<script type="text/javascript">
  window.eras = [{ name: "明治", abbreviation: "明", symbol: "Meiji", startDate: "1868-09-08", shortcuts: "1,M" },
    { name: "大正", abbreviation: "大", symbol: "Taisho", startDate: "1912-07-30", shortcuts: "2,T" },
    { name: "昭和", abbreviation: "昭", symbol: "Showa", startDate: "1926-12-25", shortcuts: "3,S" },
    { name: "平成", abbreviation: "平", symbol: "Heisei", startDate: "1989-01-08", shortcuts: "4,H" },
    { name: "令和", abbreviation: "令", symbol: "Reiwa", startDate: "2019-05-01", shortcuts: "5,R" }
  ];
</script>
```

和暦定義の詳細

以下に和暦定義の各設定の詳細を解説します。

| | |
|--------------|---|
| name | 年号の正式名称 キーワード"ggg"で使します。 |
| abbreviation | 年号の略称 キーワード"gg"で使します。 |
| symbol | 年号のアルファベット略称 キーワード"g"で使します。 |
| startDate | 年号の開始日。ハイフン(-)区切りで年月日を指定します。 |
| shortcuts | 和暦を入力する際のショートカットキー (数字、アルファベットを設定できます。複数キーワードを設定する場合はカンマ区切りで設定します) |

Input for ASP.NET WebForms

使用例

和暦設定ファイルを参照してwijinput ウィジェットを使用したサンプルです。和暦設定は独立したjsファイルに定義し、ページ上でjsファイルを参照することで和暦設定をアプリケーション内で一元管理することをお勧めします。

ソースビュー

```
<script type="text/javascript">
    window.eras = [{ name: "明治", abbreviation: "明", symbol: "Meiji", startDate:
"1868-09-08", shortcuts: "1,M" },
    { name: "大正", abbreviation: "大", symbol: "Taisho", startDate: "1912-07-30",
shortcuts: "2,T" },
    { name: "昭和", abbreviation: "昭", symbol: "Showa", startDate: "1926-12-25",
shortcuts: "3,S" },
    { name: "平成", abbreviation: "平", symbol: "Heisei", startDate: "1989-01-08",
shortcuts: "4,H" },
    { name: "令和", abbreviation: "令", symbol: "Reiwa", startDate: "2019-05-01",
shortcuts: "5,R" }
    ];
</script>
<c1:C1InputDate ID="C1InputDate1" runat="server" Date="04/11/2014 09:53:00"
DisplayFormat="ggg ee/MM/dd">
</c1:C1InputDate>
```

C1InputNumeric の使い方

C1InputNumeric コントロールは、**C1InputMask** から派生した数値の編集専用コントロールです。この数値エディタを使用すると、アプリケーションでカスタム検証論理を記述せずに入力を指定できます。次の図は、**C1InputNumeric** コントロールを示しています。

主な利点

C1InputNumeric の主な利点は、次のとおりです。

- **C1InputNumeric** コントロールは、数値エディタを表示します。
- **C1InputNumeric** に特定のカルチャを選択可能。例: 英語、スペイン語、ドイツ語、ロシア語など。**C1InputNumeric** は、選択された **Culture** プロパティを使用して、数値グループ区切り文字(桁区切り文字)、小数点文字、および符号を表示することに注意してください。
- **MinValue** と **MaxValue** プロパティを簡単に変更可能な数値範囲をサポート
- クライアント側イベントが使用可能なため、ポストバックが不要となり Web フォームのパフォーマンスが向上

C1InputNumeric の定義

C1InputNumeric コントロールは、数値データを表示する数値範囲サポートを備えています。**C1InputNumeric** コントロールは数値を厳密に処理するため、指定する入力マスクはありません。**C1InputNumeric** を定義するには、最小値と最大値、小数点以下の桁数(ゼロ可能)を入力して、カルチャ固有の桁区切り文字を表示するかどうかを指定するだけです。

C1InputNumeric コントロールの値を定義するには、**C1InputNumeric** スマートタグを使用するか、プログラムで **C1InputNumeric** オブジェクトを使用します。

ユーザーが実行時に値を編集するとき、上矢印または下矢印キーを使用して現在のフィールドを増減できることに注意してください。

よく使用される **C1InputNumeric** タスクについては、「[C1InputNumeric タスク](#)」トピックを参照してください。

C1InputNumeric の一般プロパティ

下の表に、**C1InputNumeric** コントロールの一般プロパティを示します。

| プロパティ | 説明 |
|---------------|---|
| Value | Double 型。 C1InputNumeric コントロールの数値 |
| Text | String 型。カルチャ情報(グループ区切り文字など)に準じて表示可能なテキスト |
| MinValue | 入力可能な最小値 |
| MaxValue | 入力可能な最大値 |
| DecimalPlaces | 表示する小数点以下の桁数(デフォルト: 2)を示します。 |
| ShowGroup | 桁グループ区切り文字を3桁ごとに挿入するかどうかを示します(桁グループの桁数は選択されたカルチャによって異なります)。 |
| Placeholder | 空の状態に表示されるテキストを示します。 |

C1InputNumeric コントロールのプロパティとイベントの大部分は、**C1InputMask** コントロールと同じです。ただし、数値コントロールでは使用されない隠しプロパティ (**AllowPromptAsInput**、**MaskFormat**、**HidePromptOnLeave**、**PasswordChar**、**PromptChar**、**ResetOnPrompt**、**ResetOnSpace**、**SkipLiterals** など)は除きます。


C1InputPercent の使い方

C1InputPercent コントロールは、**C1InputNumeric** から派生したパーセント値編集専用コントロールです。この数値エディタを使用すると、アプリケーションでカスタム検証論理を記述せずに入力を指定できます。次の図は、**C1InputPercent** コントロールを示しています。

主な利点

C1InputPercent の主な利点は、次のとおりです。

- **C1InputPercent** コントロールは、数値エディタを表示します。**C1InputPercent** を使用して、パーセント値を入力できます。
- **C1InputPercent** に特定のカルチャを選択可能。例：英語、スペイン語、ドイツ語、ロシア語など。数値パターン、および数値文字列のその他の側面（パーセント記号と配置）は、選択された **Culture** プロパティによって異なることに注意してください。
- クライアント側イベントが使用可能なため、ポストバックが不要となり Web フォームのパフォーマンスが向上。

 **注意:** **C1InputPercent** コントロールのプロパティは、**C1InputNumeric** コントロールと同じです。

C1InputPercent の定義

C1InputPercent コントロールは、数値データを表示する数値範囲サポートを備えています。**C1InputPercent** コントロールの値を定義するには、**C1InputPercent** [スマートタグ](#)を使用するか、プログラムで **C1InputPercent** オブジェクトを使用します。ユーザーが実行時に値を編集するとき、上矢印または下矢印キーを使用して現在のフィールドを増減できることに注意してください。

C1InputPercent コントロールのプロパティは、**C1InputNumeric** コントロールと同じです。よく使用される **C1InputPercent** タスクについては、「[C1InputNumeric](#) タスク」トピックを参照してください。


C1InputCurrency の使い方

C1InputCurrency コントロールは、**C1InputNumeric** から派生した通貨値編集専用コントロールです。この数値エディタを使用すると、アプリケーションでカスタム検証論理を記述せずに入力を指定できます。次の図は、**C1InputCurrency** コントロールを示しています。

主な利点

C1InputCurrency の主な利点は、次のとおりです。

- **C1InputCurrency** コントロールは、数値エディタを表示します。**C1InputCurrency** を使用して、通貨値を入力できます。
- **C1InputCurrency** に特定のカルチャを選択可能。例：英語、スペイン語、ドイツ語、ロシア語など。数値パターン、および数値文字列のその他の側面（パーセント記号と配置）は、選択された **Culture** プロパティによって異なることに注意してください。
- クライアント側イベントが使用可能なため、ポストバックが不要となり Web フォームのパフォーマンスが向上。

 **注意:** **C1InputCurrency** コントロールのプロパティは、**C1InputNumeric** コントロールと同じです。

C1InputCurrency の定義

C1InputCurrency コントロールは、数値データを表示する数値範囲サポートを備えています。**C1InputCurrency** コントロールの値を定義するには、**C1InputCurrency** スマートタグを使用するか、プログラムで **C1InputCurrency** オブジェクトを使用します。

ユーザーが実行時に値を編集するとき、上矢印または下矢印キーを使用して現在のフィールドを増減できることに注意してください。

C1InputCurrency コントロールのプロパティは、**C1InputNumeric** コントロールと同じです。よく使用される **C1InputCurrency** タスクについては、「[C1InputNumeric タスク](#)」トピックを参照してください。

C1Input の外観

6つのプレミアムテーマ (*Arctic*、*Midnight*、*Aristo*、*Rocket*、*Cobalt*、および *Sterling*) の1つを選択して、入力コントロールの外観を変更します。あるいは、jQuery UI から ThemeRoller を使用して、カスタマイズした独自のテーマを作成できます。

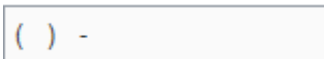
テーマ

C1Input は、**C1Input** コントロールごとに6種類の組み込みスタイルを備えているため、コントロールを自動的に書式設定できます。組み込みテーマには、**arctic**、**aristo**、**cobalt**、**midnight**、**rocket**、および **sterling** が含まれます。

以下の例では、**C1InputMask** コントロールを示していますが、すべてのコントロールのテーマは同じです。

arctic

以下の画面は **arctic** テーマを表示しています。



aristo

以下の画面は **aristo** テーマを表示しています。これは、すべての **C1Input** コントロールのデフォルト書式です。



cobalt

以下の画面は **cobalt** テーマを表示しています。



midnight

以下の画面は **midnight** テーマを表示しています。



rocket

以下の画面は **rocket** テーマを表示しています。



sterling

以下の画面は **sterling** テーマを表示しています。



C1Input CSS セレクタ

CSS スタイルを使用すれば、任意の **C1Input** 要素をスタイル設定して外観を独特なものにできます。カスタマイズ処理を簡素化するために、ComponentOne には、6種類の組み込みテーマごとに CSS セレクタが組み込まれています。テーマについて

の詳細は、「[テーマ](#)」を参照してください。

背景、テキスト、フォント、枠線、輪郭、マージン、埋め込み、リスト、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。

以下のトピックでは、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 セレクタについて詳細に説明します。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。

| CSS セレクタ | 説明 |
|--------------------------------|--|
| .wijmo-input-trigger | トリガーボタンにスタイルを適用します。 |
| .wijmo-wijinput-spinner | スピナーボタンにスタイルを適用します。 |
| .wijmo-wijinput-spinup | 上スピンボタンにスタイルを適用します。 |
| .wijmo-wijinput-spindown | 下スピンボタンにスタイルを適用します。 |
| .wijmo-wijinput-input | すべての入力タイプの最も外側のコンテナにスタイルを適用します。 |
| .wijmo-wijinput-mask | すべてのマスク入力 (C1InputMask) の最も外側のコンテナにスタイルを適用します。 |
| .wijmo-wijinput-numeric | 数値入力タイプ (C1InputNumeric 、 C1InputCurrency 、および C1InputPercent) の最も外側のコンテナにスタイルを適用します。 |
| .wijmo-wijinput-date | すべての日付入力タイプ (C1InputDate) の最も外側のコンテナにスタイルを適用します。 |
| .wijmo-wijinput-wrapper | 入力要素の直接のラッパーにスタイルを適用します。 |
| .wijmo-wijinput ui-state-focus | フォーカス状態にあるすべての入力タイプの最も外側のコンテナにスタイルを適用します。 |

クライアント側の機能

C1Input コントロールは、サーバー側コントロールのメンバとほぼ同等の非常に充実したクライアント側オブジェクトモデルを備えています。

C1Input コントロールが表示されると、クライアント側コントロールのインスタンスが自動的に生成されます。これは、サーバーにポストバックしなくても、**C1Input** コントロールのプロパティやメソッドにアクセスできるということです。

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

クライアント側イベント

C1Input には、無効な文字の入力などのアクションが発生したときに **C1Input** コントロールを操作可能にする複数のクライアント側イベントが含まれています。

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、**invalidInput** という JavaScript 関数を無効な文字が入力されたときに反応するように割り当てるには、`OnClientInvalidInput` プロパティを **invalidInput** 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 |
|---|----------------------------------|----------------------------|
| <code>OnClientInitialized</code> | <code>initialized</code> | コントロールを初期化した後に発生します。 |
| <code>OnClientInitializing</code> | <code>initializing</code> | コントロールを初期化する前に発生します。 |
| <code>OnClientInvalidInput</code> | <code>invalidInput</code> | 無効な文字が入力されたときに発生します。 |
| <code>OnClientTextChanged</code> | <code>textChanged</code> | 入力のテキストが変更されたときに発生します。 |
| <code>OnClientTriggerMouseDown</code> | <code>triggerMouseDown</code> | マウスがトリガーボタン上で押されたときに発生します。 |
| <code>OnClientTriggerMouseUp</code> | <code>triggerMouseUp</code> | マウスがトリガーボタン上で離されたときに発生します。 |
| <code>OnClientDateChanged (InputDate only)</code> | <code>dateChanged</code> | 日付値が変更された後に発生します。 |
| <code>OnClientValueBoundsExceeded (InputNumber only)</code> | <code>valueBoundsExceeded</code> | 入力の値が有効範囲を超えたときに発生します。 |
| <code>OnClientValueChanged (InputNumber only)</code> | <code>valueChanged</code> | 値が変更された後に発生します。 |

C1Input クライアント側イベントの説明と構文の例については、「クライアント側リファレンス」でも確認できます

タスク別ヘルプ

タスク別ヘルプは、Visual Studio のプログラミングに精通しているユーザーを対象としています。ヘルプに記述された手順に従うことによって、C1Input のさまざまな機能をデモンストレーションするプロジェクトを作成して、C1Input の用途を理解することができます。

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。

C1InputMask タスク

このセクションでは、**C1InputMask** コントロールを使用して、個別のタスクを実行する方法について説明します。以下の各トピックでは、**C1InputMask** コントロールを Web フォームに追加してあるものと想定しています。

プロンプト文字列の変更

実行時に、**C1InputMask** コントロールはマスクを一連のプロンプト文字列 (# または _ など) として表示します。プロンプト文字列は、各編集可能なマスク位置を表します。プロンプト文字列を変更するには、**PromptChar** プロパティを使用します。この例では、(999) 000-0000 の電話番号マスクを持つ C1InputMask コントロールを使用します。

タスクメニューを使用してプロンプト文字列を変更するには、以下の手順を実行します。

電話番号の **PromptChar** プロパティを変更するには、[**C1InputMask タスク**]メニューを開き、[**プロンプト文字列**]テキストボックスに数値記号 (#) を入力します。

| C1InputMask タスク | |
|----------------------------------|-----------------------|
| デザイン | |
| プロパティ | |
| テキスト | <input type="text"/> |
| マスク | <input type="text"/> |
| プロンプト文字列 | # |
| パスワード文字列 | <input type="text"/> |
| カルチャ | ja-JP |
| テーマ | aristo |
| <input type="checkbox"/> CDN の使用 | |
| CDN パス | http://cdn.wijmo.com/ |
| バージョン情報... | |

.html マークアップを使用してプロンプト文字列を変更するには、以下の手順を実行します。

プロンプト文字列を **C1InputMask** の数値記号 (#) に変更するには、.aspx ページで次のマークアップを使用します。

```

ソースビュー
<cc1:C1InputMask runat="server"
    MaskFormat="(999) 000-0000"
  >
</cc1:C1InputMask>

```

Input for ASP.NET WebForms

```
Text="412"  
PromptChar="#">  
</cc1:C1InputMask>
```

このトピックの作業結果

プロジェクトを実行します。次のように、Web ブラウザに数値記号(#)がプロンプト文字列として表示されることに注意してください。

(412) #### #####

コントロールに**Text** プロパティが指定されたため、412 の局番が数値記号の代わりに表示されていることに注意してください。入力ボックスがフォーカスを失くしたときにプロンプト文字列を非表示にする方法についての詳細は、「フォーカス消失時のプロンプト文字列の非表示化」トピックを参照してください。

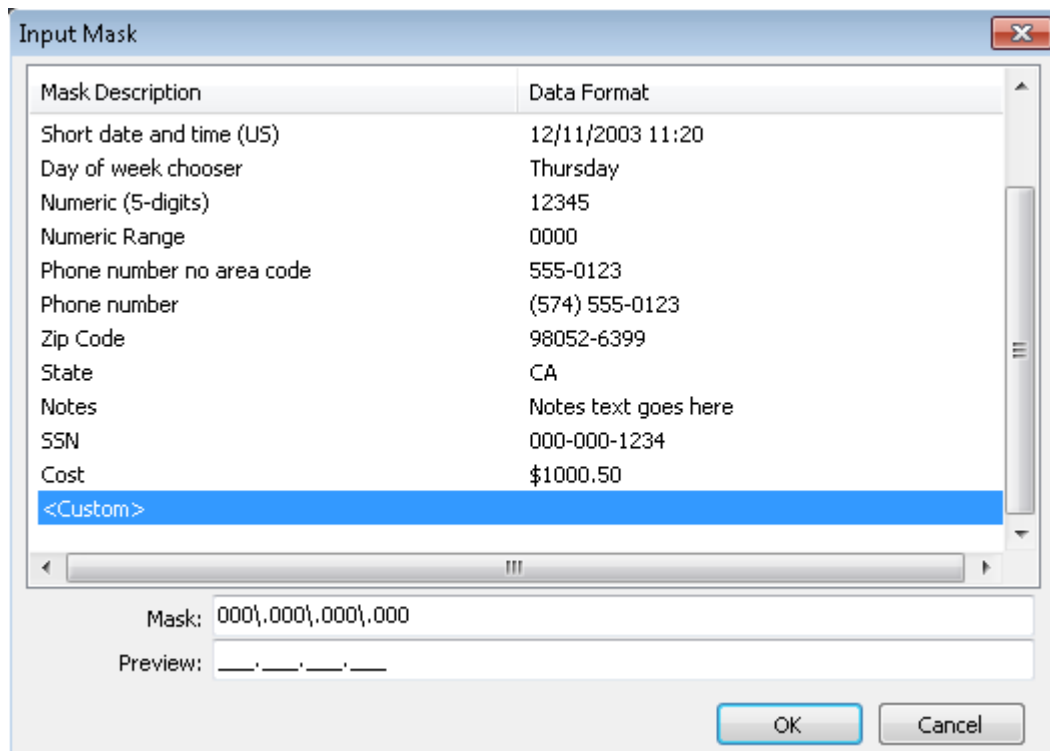
IP アドレスマスクの作成

以下の例では、数値範囲を使用して、IP アドレスを編集するためのマスク付きテキストボックスを表す方法について示します。この例では、「000\,000\,000\,000」のカスタムマスクを持つ**C1InputMask** コントロールを使用します。

タスクメニューを使用して IP アドレスマスクを作成するには、以下の手順を実行します。

特定のテキストを指定した IP アドレス値を表示するには、以下のタスクを実行します。

1. [**C1InputMask** タスク]メニューを開き、マスクボックスの<...>ボタンをクリックして、[**定型入力**]ダイアログボックスを開きます。
2. [**マスク**]テキストボックスに「000\,000\,000\,000」と、マスクを入力します。



マスクの入力を開始すると、デザイナーが <ユーザー設定> に自動的に切り替わる(入力したマスクがマスクリストになかった場合)ことに注意してください。


3. <OK>をクリックします。
4. [**タスク**]メニューがまだ開いた状態で、[**Text**]テキストボックスに、**192168001001** を入力します。

ソースビュー

.html マークアップを使用して IP アドレスマスクを作成するには、以下の手順を実行します。

IP アドレスのマスク値を作成するには、.aspx ページで次のマークアップを使用します。

```
<cc1:C1InputMask runat="server" MaskFormat="000\.000\.000\.000"
    Text="192168001001">
</cc1:C1InputMask>
```

 **注意:**「<」または「>」の文字により、後続の文字は命令をシフトダウンまたはシフトアップします。「\」を付けない「.」文字は、小数点プレースホルダとして機能し、使用される実際の表示文字は、**Culture** プロパティの値に適切な小数点プレースホルダとなります。

このトピックの作業結果

プロジェクトを実行します。次のように、192168001001 のテキストを持つ IP アドレスマスクが Web ブラウザに表示されることに注意してください。

192.168.001.001

電話番号マスクの作成

以下の例では、**MaskFormat** プロパティの列挙体部分を使用する方法を示します。この例では、(999) 000-0000 の電話番号マスクを持つ**C1InputMask** コントロールを使用します。

タスクメニューを使用して電話番号マスクを作成するには、以下の手順を実行します。

412 局番を持つ電話番号値を表示するには、以下のタスクを実行します。


1. [**C1InputMask** タスク]メニューを開き、**マスク** プロパティの<...>ボタンをクリックして、[**定型入力**]ダイアログボックスを開きます。
2. マスク値に**電話番号**を選択して、<OK>をクリックします。
3. [**タスク**]メニューがまだ開いた状態で、[**テキスト**]テキストボックスに 412 を入力します。

ソースビュー

.html マークアップを使用して電話番号マスクを作成するには、以下の手順を実行します。

412 局番を持つ電話番号値を表示するには、.aspx ページで次のマークアップを使用します。

```
<cc1:C1InputMask runat="server"
    MaskFormat="(999) 000-0000"
    Text="412">
</cc1:C1InputMask>
```

 **注意:**文字「9」は、数字またはスペース(オプション)のマスク要素として機能します。文字「0」は、数字(必須)のマスク要素として機能します。このマスク要素は、0~9の1桁の数字を受け付けます。

このトピックの作業結果

プロジェクトを実行します。次のように、412 のテキストを持つ**電話番号**マスクが Web ブラウザに表示されることに注意してください。

(412) ___ - ___

プロンプト文字列を含まない日付マスクの表示

プロンプト文字列(「//」など)を含まない日付の入力ボックスを作成するには、**C1InputMask** コントロールを使用し、**PromptChar** プロパティをスペース「 」に設定します。

タスクメニューを使用してプロンプト文字列を含まない短い日付マスクを作成するには、以下の手順を実行します。

プロンプト文字列を含まない[**短い日付**]入力ボックスを作成するには、以下のタスクを実行します。

1. [**C1InputMask タスク**]メニューを開き、**MaskFormat** プロパティの<...>ボタンをクリックして、[**定型入力**]ダイアログボックスを開きます。
2. マスク値に**短い日付**を選択して、<OK>をクリックします。
3. [**タスク**]メニューがまだ開いた状態で、[**PromptChar**]テキストボックスに、スペース文字(「 」)を入力します。デフォルトのアンダスコア(_)は削除する必要があることに注意してください。

.html マークアップを使用してプロンプト文字列を含まない短い日付マスクを作成するには、以下の手順を実行します。

プロンプト文字列を含まない[**短い日付**]入力ボックスを作成するには、.aspx ページで次のマークアップを使用します。

ソースビュー

```
<c1:C1InputMask runat="server" MaskFormat="00/00/0000" PromptChar=" ">  
</c1:C1InputMask>
```

コードを使用してプロンプト文字列を含まない短い日付マスクを作成するには、以下の手順を実行します。

C1InputMask コントロールにプロンプト文字列を含まない短い日付マスクを作成するには、Web ページをダブルクリックして、**Load** イベントのイベントハンドラを作成します。**Page_Load** イベントに、次のコードを入力します。

Visual Basic コードの書き方

Visual Basic

```
With C1InputMask1  
    .MaskFormat = "00/00/0000"  
    .PromptChar = " "  
End With
```

C# コードの書き方

C#

```
this.C1InputMask1.MaskFormat = "00/00/0000";  
this.C1InputMask1.PromptChar = char.Parse(" ");
```

✔ このトピックの作業結果

プロジェクトを実行します。次のように、**短い日付**マスクがプロンプト文字列を含まずに表示されることに注意してください。

//

フォーカス消失時のプロンプト文字列の非表示化

コントロールが入力フォーカスを消失するときプロンプト文字列を非表示にするには、**HidePromptOnLeave** プロパティを

True に設定します。

.html マークアップを使用してフォーカス消失時にプロンプト文字列を非表示にするには、以下の手順を実行します。

.aspx ページのマークアップで、以下を挿入します。

ソースビュー

```
<cc1:C1InputMask runat="server"
    MaskFormat="(999) 000-0000"
    PromptChar="#"
    HidePromptOnLeave="True">
</cc1:C1InputMask>
```

コードを使用してフォーカス消失時にプロンプト文字列を非表示にするには、以下の手順を実行します。

フォーカス消失時に **C1InputMask** コントロールのプロンプト文字列を非表示にするには、以下の手順を実行します。

1. Web ページをダブルクリックして、**Load** イベントのイベントハンドラを作成します。
2. **Page_Load** イベントに、次のコードを入力します。

Visual Basic コードの書き方

Visual Basic

```
With C1InputMask1
    .MaskFormat = "(999) 000-0000"
    .PromptChar = "#"
    .HidePromptOnLeave = True
End With
```

C# コードの書き方

C#

```
this.C1InputMask1.MaskFormat = "(999) 000-0000";
this.C1InputMask1.PromptChar = char.Parse("#");
this.C1InputMask1.HidePromptOnLeave = true;
```

🟢 このトピックの作業結果

プロジェクトを実行します。電話番号マスクのプロンプト文字列が非表示になっていることに注意してください。

入力ボックス内でクリックして、フォーカスを取得すると、プロンプト文字列 (# など) が表示されます。

入力ボックスの外でクリックして、フォーカスを失くすと、プロンプト文字列が再び非表示になります。プロンプト文字列の変更についての詳細は、「[プロンプト文字列の変更](#)」トピックを参照してください。

C1InputDate タスク

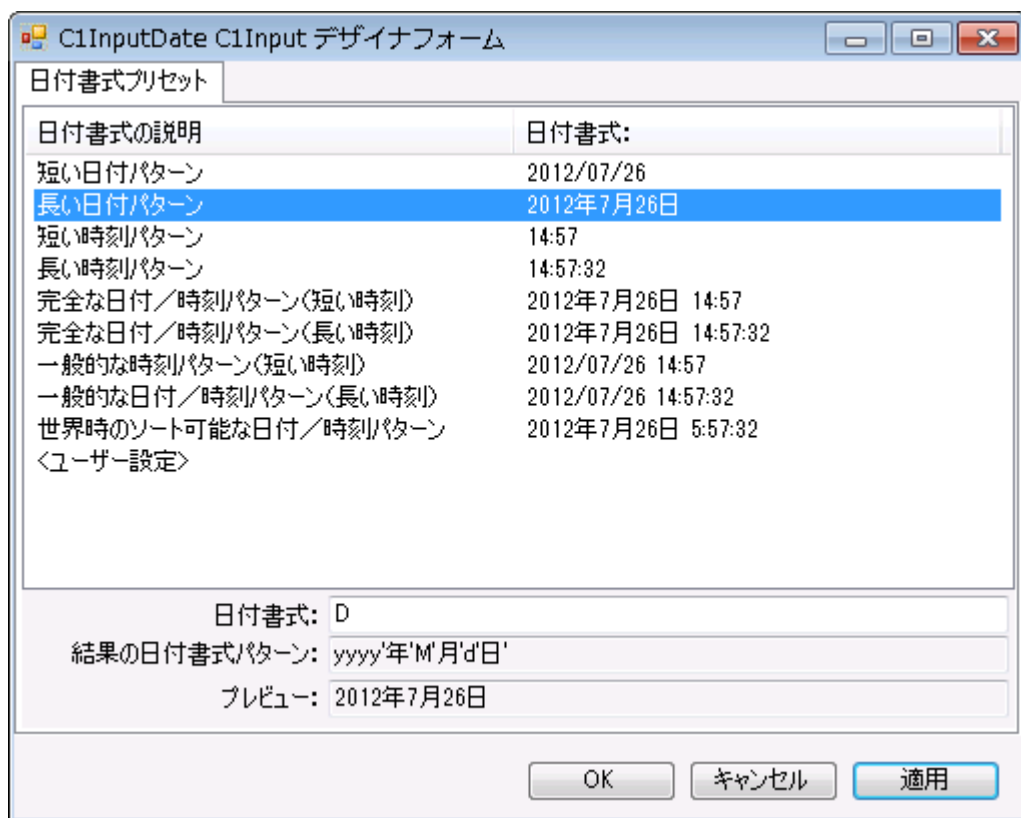
このセクションでは、**C1InputDate** コントロールを使用して、個別のタスクを実行する方法について説明します。以下の各トピックでは、**C1InputDate** コントロールを Web フォームに追加してあるものと想定しています。

日付書式パターンと日付の設定

以下の例では、C1InputDate コントロールの日付書式パターンの設定方法について例示します。

タスクメニューを使用して日付書式パターンを設定するには、以下の手順を実行します。

1. [C1InputDate タスク]メニューを開き、[デザイナー]を選択します。C1InputDate C1Input デザイナフォームが表示されます。
2. 書式付き日付パターンを選択します。この例では、長い日付パターンを選択します。



デザイナーに次のようなプレビューが表示されることに注意してください。

3. <OK>をクリックします。
4. タスクメニューがまだ開いている状態で、[日付]のドロップダウン矢印をクリックします。カレンダーが表示されます。
5. 今日の日付に選択する日付を選択します。

.html マークアップを使用して日付書式パターンを設定するには、以下の手順を実行します。

日付書式値に長い日付パターン書式を表示するには、.aspx ページに次のマークアップを使用します。

ソースビュー

```
<cc1:C1InputDate runat="server"
Date="2012-07-26"
DateFormat="D">
</cc1:C1InputDate>
```

コードを使用して日付書式パターンを設定するには、以下の手順を実行します。

C1InputDate コントロールに日付書式パターンを設定するには、Web ページをダブルクリックして、Load イベントのイベントハンドラを作成します。Page_Load イベントに、次のコードを入力します。

Visual Basic コードの書き方

Visual Basic

```
' コントロールを長い日付パターンとして書式設定します。
Me.C1InputDate1.DateFormat = "D"
' 日付を設定します。
Me.C1InputDate1.Date = "2012-07-26"
```

C# コードの書き方

C#

```
// コントロールを長い日付パターンとして書式設定します。
this.C1InputDate1.DateFormat = "D";
// 日付を設定します。
this.C1InputDate1.Date = DateTime.Parse("2012-07-26");
```

✔ このトピックの作業結果

プロジェクトを実行して、日付書式パターンが更新されたことに注意してください。

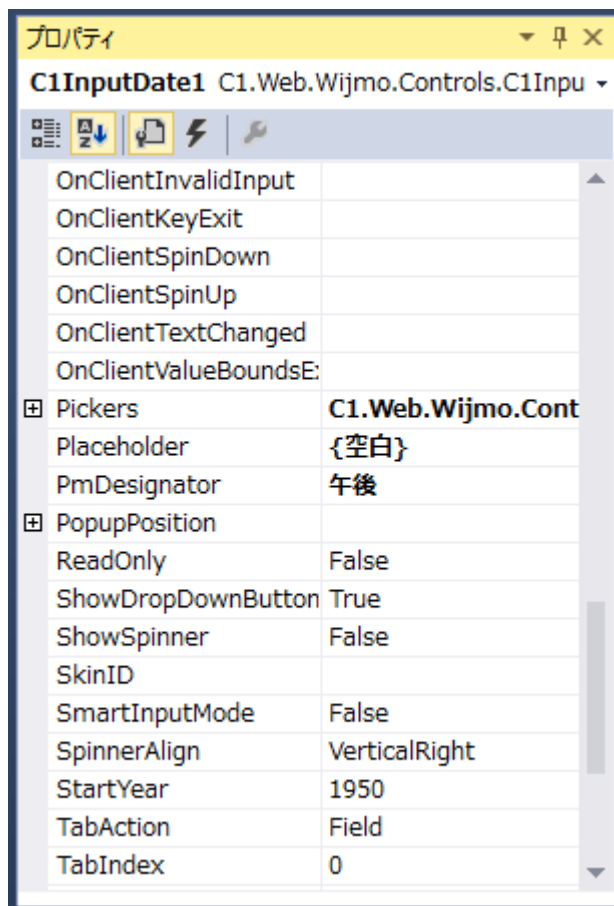
2012年7月26日

空の日付値の表示

C1InputDate コントロールを含むプロジェクトを実行すると、コントロールをどのようにカスタマイズしたかに関係なく、コントロール内にデフォルト日付で現在の日付が自動的に表示されます(「日付書式パターンと日付の設定」を参照)。日付を空の状態にして、このデフォルトの日付を表示しない場合、以下の手順を実行します。

デザイナの使用

1. ツールボックスから**C1InputDate** コントロールをWebフォーム上にドラッグドロップします。**C1InputDate** コントロールでは、現在の日付が表示されます。
2. デザインビューで**C1InputDate** 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。
3. [プロパティ] ウィンドウに、**Placeholder** プロパティをご希望の値や{空白}値に設定します。



4. プロジェクトを実行して、**C1InputDate** コントロールの {空白} に注意してください。



注意: コントロールに割り当てた空値は、プロジェクトを初めて実行するときに表示されます。いったんコントロールを選択して、日付を切り替えた場合、空値を再表示する唯一の方法は、ページ上の別のコントロールを選択することです。コントロール内の日付は「削除」できません。空値のテキストしか表示できます。

C1InputNumeric タスク

このセクションでは、**C1InputNumeric** コントロールを使用して、個別のタスクを実行する方法について説明します。

C1InputCurrency と **C1InputPercent** コントロールのプロパティが **C1InputNumeric** コントロールと同じであることに注意してください。そのため、以下の各タスクは、**C1InputCurrency** と **C1InputPercent** コントロールにも適用されます。

以下の各トピックでは、**C1InputNumeric** コントロールが Web フォームに追加されていることを前提にしています。

小数点以下の桁数の指定

以下の例では、**C1InputNumeric** コントロールに表示する小数点以下の桁数を簡単に指定する方法について説明します。

タスクメニューを使用して小数点以下の桁数値を設定するには、以下の手順を実行します。

1. [**C1InputNumeric** タスク]メニューを開きます。
2. コントロールの値を **2.345** に設定します。
3. 小数点以下桁数値に **3** を入力します。
値に 2.345 を入力しても、小数点以下桁数値を3に変更しないと、小数点以下2桁(デフォルト)のみが表示されることに注意してください。すなわち、2.34 となります。

.html マークアップを使用して小数点以下の桁数を設定するには、以下の手順を実行します。

値を **2.345** に、**DecimalPlaces** 値を **3** に設定するには、.aspx ページで、次のマークアップを使用します。

ソースビュー

```
<cc1:C1InputNumeric runat="server"
DecimalPlaces="3"
Value="2.345">
</cc1:C1InputNumeric>
```

コードを使用して小数点以下の桁数を設定するには、以下の手順を実行します。

C1InputNumeric コントロールの小数点以下の桁数値を設定するには、Web ページをダブルクリックして、**Load** イベントのイベントハンドラを作成します。**Page_Load** イベントに、次のコードを入力します。

Visual Basic コードの書き方

Visual Basic

```
' 数値を設定します。
Me.C1InputNumeric1.Value = 2.345
' 小数点以下の桁数を設定します。
Me.C1InputNumeric1.DecimalPlaces = 3
```

C# コードの書き方

C#

```
// 数値を設定します。
this.C1InputNumeric1.Value = 2.345;
// 小数点以下の桁数を設定します。
this.C1InputNumeric1.DecimalPlaces = 3;
```

最小値／最大値の設定

以下の例では、**MinValue** と **MaxValue** プロパティを簡単に変更可能な、**C1InputNumeric** コントロールの数値範囲サポートについて示します。

タスクメニューを使用して数値を設定するには、以下の手順を実行します。

1. [**C1InputNumeric** タスク]メニューを開きます。
2. **最小有効値** に **1** を入力します。
3. **最大有効値** に **1000** を入力します。
4. タスクメニューがまだ開いた状態で、[値]テキストボックスに、**1** を入力します。

.html マークアップを使用して数値を設定するには、以下の手順を実行します。

MinValue を **1** に、**MaxValue** 値を **1000** に、**Value** を **1** に設定するには、.aspx ページで、次のマークアップを使用します。

ソースビュー

```
<cc1:C1InputNumeric runat="server"
MaxValue="1000"
MinValue="1"
Value="1">
```

Input for ASP.NET WebForms

```
</cc1:C1InputNumeric>
```

コードを使用して数値を設定するには、以下の手順を実行します。

C1InputNumeric コントロールの数値を設定するには、Web ページをダブルクリックして、**Load** イベントのイベントハンドラを作成します。**Page_Load** イベントに、次のコードを入力します。

Visual Basic コードの書き方

Visual Basic

```
With C1InputNumeric1
    .MaxValue = 1000
    .MinValue = 1
    .Value = 1
```

C# コードの書き方

C#

```
this.C1InputNumeric1.MaxValue = 1000;
this.C1InputNumeric1.MinValue = 1;
this.C1InputNumeric1.Value = 1;
```

プログラムを実行して、次のことを確認します。

- 入力コントロールが 1.00 を表示している状態で、マウスポインタで下スピンボタンをクリックします。コントロールが 1.00 より小さい値を表示しないことに注意してください。
- 入力コントロールが 1000.00 を表示している状態で、マウスポインタで上スピンボタンをクリックします。コントロールが 1000.00 より大きい値を表示しないことに注意してください。

テーマの変更

6つの組み込みテーマの1つを使用して、**C1Input** コントロールを書式設定できます。以下の例では、**C1InputMask** コントロールを使用します。

スマートタグを使用したテーマの変更

コントロールの[タスク]メニューを使用して、デザイン時に **C1Input** コントロールのスタイルを変更できます。

1. **C1InputMask** スマートタグをクリックして、[**C1InputMask** タスク]メニューを開きます。
2. [テーマ]の横のドロップダウン矢印をクリックします。
3. リストされた組み込みテーマの1つを選択します。テーマが**C1InputMask** コントロールに適用されます。

コードでのテーマの変更

プログラムで **C1Input** コントロールのスタイルを変更するには、次のコードを使用します。この例では、**midnight** が使用されますが、任意の組み込みテーマに置き換えることができます。

Visual Basic コードの書き方

Visual Basic

```
C1InputMask1.Theme = "midnight"
```

C# コードの書き方

C#

```
C1InputMask1.Theme = "midnight";
```

カスタムテーマの追加

C1Input は、6つの組み込みテーマを提供していますが、異なるテーマを使用したい場合、CDN URL を使用して既存のテーマを選択するか、jQuery ThemeRoller Web アプリケーションを使用して独自のテーマを作成できます。以下の例では、**C1InputDate** コントロールを使用します。

CDN URL の使用

1. **C1InputDate** スマートタグをクリックして、[タスク]メニューを開きます。
2. [CDN の使用]を選択します。
3. テーマ プロパティで、CDN URL を入力して、テーマを指定します。CDN URL は <http://blog.jqueryui.com/2011/06/jquery-ui-1-8-14/> にあります。この例では、sunny テーマ (<http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.14/themes/sunny/jquery-ui.css>)を使用します。

このテーマ設定は **Web.config** ファイルの `<appSettings>` に格納されます。ソリューションエクスプローラで、**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。追加した CDN URL がここに指定されています。

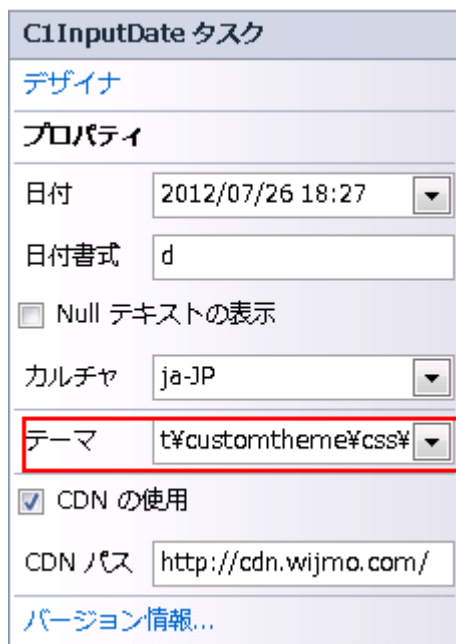
4. プロジェクトを実行して、テーマが **C1InputDate** に適用されることに注意してください。

jQuery ThemeRoller の使用

1. <http://jqueryui.com/themeroller/> に進みます。
2. 「Roll Your Own」タブで、設定値を変更してカスタムテーマを作成します。フォント、色、背景などをカスタマイズできます。あるいは、「Gallery」タブをクリックして、既存のテーマを選択します。
3. 「Download」ボタンをクリックし、次に **Build Your Download** ページで「Download」をもう一度クリックします。
4. テーマの .zip ファイルを Visual Studio プロジェクトフォルダ内のフォルダに保存して解凍します。この例では、**customtheme** フォルダを作成しました。
5. 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]をクリックし、**customtheme** フォルダを右クリックして、[プ

Input for ASP.NET WebForms

- プロジェクトに含める]を選択します。
6. **C1InputDate** スマートタグをクリックして、[**タスク**]メニューを開きます。
 7. [**CDN の使用**]を選択します。
 8. テーマプロパティで、カスタムテーマの .css へのパス (**customtheme/css/custom-theme/jquery-ui-1.8.14.custom.css** など)を入力します。



C1InputDate タスク

デザイナー

プロパティ

日付 2012/07/26 18:27

日付書式 d

Null テキストの表示

カルチャ ja-JP

テーマ t%customtheme%css%

CDN の使用

CDN パス http://cdn.wijmo.com/

バージョン情報...

このテーマ設定は **Web.config** ファイルの <appSettings> に格納されます。ソリューションエクスプローラで、**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。追加したカスタムテーマはここに指定されています。

9. プロジェクトを実行して、テーマが **C1InputDate** に適用されることに注意してください。

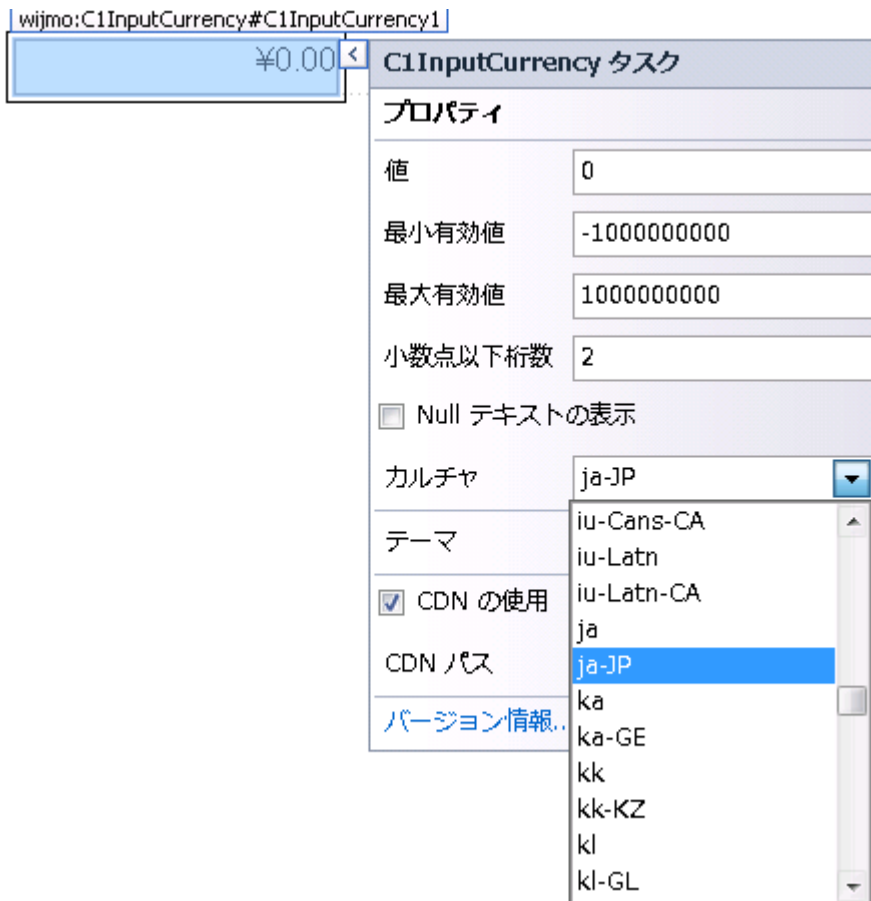
2012/07/26

カルチャの選択

以下のトピックでは、**C1InputCurrency** コントロールの **Culture** プロパティの使用方法について示します。ただし、**Culture** プロパティは、すべての **C1Input** コントロールで使用可能であることに注意してください。

デザイナーの使用

任意の **C1Input** コントロールに特定のカルチャを選択できます。コントロールの **Culture** プロパティを設定するには、その [**タスク**]メニューを単に開き、ドロップダウンリストからカルチャを選択します。



HTML マークアップの使用

Culture 値を設定するには、.aspx ページに次のマークアップを使用します。

ソースビュー

```
<cc1:C1InputCurrency runat="server"
    Culture="ja-JP">
</cc1:C1InputCurrency>
```

コードの使用

C1InputCurrency コントロールの **Culture** を設定するには、Web ページをダブルクリックして、**Load** イベントのイベントハンドラを作成します。**Page_Load** イベントに、次のコードを入力します。

Visual Basic コードの書き方

Visual Basic

```
Me.C1InputCurrency1.Culture = New System.Globalization.CultureInfo("ja-JP")
```

C# コードの書き方

C#

```
this.C1InputCurrency1.Culture = new System.Globalization.CultureInfo("ja-JP");
```

✔ このトピックの作業結果

以下の **C1InputCurrency** コントロールでは、円を示します。

クライアント側イベントタスク

このセクションでは、**C1Input** コントロールを使用して、さまざまなクライアント側イベントタスクを実行する方法について説明します。

無効な入力の入力時のツールチップの表示

このトピックでは、ツールチップを表示する方法について示します。この例では、無効な文字が **C1InputMask** コントロールに入力されたときに `c1tooltip` を使用します。

1. フォーム上の **C1InputMask** コントロールを右クリックし、[プロパティ]を選択して、Visual Studio のプロパティウィンドウを開きます。
2. **MaskFormat** プロパティの横に、**00-000** を入力します。
3. **OnClientInvalidInput** プロパティの横に、**invalidInput** を入力します。
4. 「ソース」タブを選択して、ソースビューを開きます。
5. .aspx ソースで、次のスクリプトマークアップを入力します。

ソースビュー

```
<script type="text/javascript">
    function invalidInput(e, data) {
        $(data.widget.element).wijtooltip({
            title: '\' + data.char + '\' は '\' + data.widget.options.mask + '\' のマ
            スクに対して無効な入力です。',
            triggers: 'custom',
            showing: function () {
                window.setTimeout(function () {
                    $(data.widget.element).wijtooltip('hide');
                }, 3000);
            }
        });
        $(data.widget.element).wijtooltip('show');
    }
</script>
```

無効な文字が **C1InputMask** コントロールに入力されると、ツールチップが表示され、次の図のようになります。

トリガーを使用したカスタム UI の表示

このトピックでは、トリガーボタンがクリックされたときにカスタムインターフェイスを表示する方法について例示します。この例では、ユーザーが **C1InputNumeric** コントロール内の矢印をクリックすると、ドロップダウンスライダが表示され、スライダのつまみがドラッグされると、入力値が変化します。

1. フォーム上の **C1InputNumeric** コントロールを右クリックし、[プロパティ]を選択して、Visual Studio のプロパティウィンドウを開きます。
2. **ShowDropDownButton** プロパティを「True」に設定し、**DecimalPlaces** プロパティを「0」に設定します。
3. **OnClientDropDownButtonMouseDown** プロパティの横に、**triggerClicked** を入力します。
4. 「ソース」タブを選択して、ソースビューを開きます。
5. .aspx ソースで、最後の `</asp:Content>` 直前に次のスクリプトマークアップを入力します。

ソースビュー

```
<script type="text/javascript">
```

```

$(function () {
    $(".dropdown-container").clpopup({
        autoHide: true,
        showEffect: 'drop',
        hideEffect: 'fade'
    });
});
function triggerClicked(e) {
    var $input = $('#<%=C1InputNumeric1.ClientID%>');
    var val = $input.clinputnumeric('option', 'value');
    var $volumeSlider = $('.valueslider');
    $volumeSlider.slider({
        min: 0,
        max: 5,
        value: val,
        step: 1,
        orientation: 'vertical',
        range: 'min',
        slide: function (e, ui) {
            $input.clinputnumeric('option', 'value', ui.value);
        }
    });
    $(".dropdown-cntainer").clpopup('show', {
        of: $('.wijmo-wijinput'),
        at: 'right bottom',
        my: 'right top',
        offset: "4 2"
    });
}

```

6. <asp:Content ContentPlaceHolderID="HeadContent"> タグ内に .dropdown-container と .valueslider に対して以下のマークアップを追加します。

ソースビュー

```

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
<style type="text/css">
    .dropdown-container
    {
        height: 127px;
        margin: 0;
        padding: 0;
        width: 25px;
    }
    .valueslider
    {
        left: 4px;
        height: 103px;
    }
</style>
</asp:Content>

```

Input for ASP.NET WebForms

- その後、`<asp:Content ContentPlaceHolderID="MainContent">` 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

```
<div class="dropdown-container">  
<div class="valueslider ui-state-default ui-corner-top"></div>  
</div>
```

ユーザーがドロップダウン矢印をクリックしたとき、スライダがポップアップします。スライダが動くと、数量が変化し、次のような表示になります。

数量：

