ComponentOne

Input for ASP.NET WebForms

ComponentOne, a division of GrapeCity
201 South Highland Avenue, Third Floor
Pittsburgh, PA 15206 USA

Website: http://www.componentone.com
Sales: sales@componentone.com
Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

Trademarks

Warranty

ComponentOne warrants that the media on which the software is delivered is free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective media to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for the defective media by sending it and a check for $2 5 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original media on which the software is delivered is set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. ComponentOne is not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

Copying and Distribution

Table of Contents

Overview

Your complete collection of data-entry and validation controls: choose from masked, date, numeric, and custom editing. Get built-in masks, custom format support, localization, and more with **Input for ASP.NET Web Forms Edition**.	Getting Started To get started, review the following topics: • Key Features • Quick Start

The controls comprising **Input for ASP.NET Web Forms Edition** include:

- C1InputCurrency
  The C1InputCurrency control, derived from C1InputNumeric, is specialized for editing currency values. Using the numeric editor, you can specify input without writing any custom validation logic in your application.
  For details, see Using C1InputCurrency.
- C1InputDate
  The C1InputDate control, derived from **C1InputMask,** is specialized for editing the date and time. The C1InputDate control renders a date editor.
  For details, see Using C1InputDate.
- C1InputMask
  The C1InputMask control is the main Web control used for entering and editing information of any data type in a text form. It supports data formatting, edit mask, data validation and other features. It also supports formatted and masked editing of all data types, including additional functionality. Apart from being the main data editor control, C1InputMask also serves as the base class for specialized controls such as C1InputDate and C1InputNumeric.
  For details, see Using C1InputMask.
- C1InputNumeric
  The C1InputNumeric control, derived from C1InputMask, is specialized for editing numeric values. Using the numeric editor, you can specify input without writing any custom validation logic in your application.
  For details, see Using C1InputNumeric.
- C1InputPercent
  The C1InputPercent control, derived from C1InputNumeric, is specialized for editing percent values. Using the numeric editor, you can specify input without writing any custom validation logic in your application.
  For details, see Using C1InputPercent.

Help with ASP.NET Web Forms Edition

Getting Started

For information on installing **ComponentOne Studio ASP.NET Web Forms Edition**, licensing, technical support, namespaces and creating a project with the control, please visit Getting Started with ASP.NET Web Forms Edition.

Creating a Project

Now that we have a finished base application, we can get started migrating to ASP.NET Web Forms! Oh the anticipation. One important thing to remember starting out is that ASP.NET Controls do not need a **ScriptManager** to function properly.

One of the main differences between the AJAX and ASP.NET versions of the **C1Input** controls is the naming of the controls. The following table lists the control names in both platforms. Keep the control names in mind when you convert your projects to ASP.NET Web Forms.

AJAX	ASP.NET
C1CurrencyInput	C1InputCurrency
C1DateInut	C1InputDate
C1MaskedInput	C1InputMask
C1NumericInput	C1InputNumeric
C1PercentInput	C1InputPercent

We are going to build the same table with the same layout.

1. Create a new Empty ASP.NET Web App in Visual Studio.
2. Select **View | Designer** to switch to Design view.
3. Click **Table | Insert Table** in the Visual Studio menu and make a table with 3 rows and 2 columns.
4. Before we can add the controls to the page, we need to add some assembly references to the project. Select **Project | Add Reference**, and browse to find and select the following assemblies:
   - C1.Web.Wijmo.Controls.4.dll
   - C1.Web.Wijmo.Design.4.dll
5. Once you have those references added, it's time to start adding controls. Place the C1InputMask control in the top right cell, the C1InputDate control in the middle right cell, and the C1InputNumeric control in the bottom right cell.
6. Add the text **Product Number** in the top left cell, **Order Date** in the middle left cell, and **Quantity** in the bottom left cell. The end result should look similar to this:

Setting Properties for ASP.NET Controls

The ASP.NET Controls have a few different options than their predecessors. For instance, select the C1InputMask control and click the smart tag to open the **C1InputMask Tasks** menu. You will find theming options instead of visual styles. The themes are global themes, meaning that if you change the theme of one control on your page, all of the controls will follow suit.

1. Set the **MaskFormat** property to **00-0000**, the **PasswordChar** property to ***** and the **Theme** property to **rocket**.
2. Select the C1InputDate control and click the smart tag to open the **C1InputDate Tasks** menu. Notice that the theme has automatically changed to r**ocket** since the theme was set in the C1InputMask control.
3. Set the culture to **en-US**, which is the culture setting for the United States.
4. Now select the C1InputNumeric control. In the C1InputNumeric **Tasks** menu, set the **minValue** to *0* and the **maxValue** to *10*. Set the **Culture** property to **en-US** for the United States culture setting.

Running the ASP.NET Application

Press **F5** to run the application and check out what we've done.

The theming options of the new ASP.NET Controls go above and beyond what is offered in the AJAX controls and allow the user to develop rich, aesthetically appealing applications.

ASPX Markup Comparison

Here is the aspx markup for both the AJAX and ASP.NET Web Forms projects to show you the slight differences between the two platforms. Note that the ASP.NET markup contains less CSS styling and does not contain a script manager.

AJAX

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="InputMigration._Default" %>

<%@ Register assembly="C1.Web.UI.Controls.4" namespace="C1.Web.UI.Controls.C1Input" tagprefix="cc1" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

    <style type="text/css">
        .style1
        {
            width: 54%;
            height: 115px;
        }

        .style2
        {
            height: 35px;
        }

        .style3
        {
            height: 35px;
            width: 208px;
        }

        .style4
        {
            width: 208px;
        }

        .style5
        {
            height: 36px;
            width: 208px;
        }

        .style6
        {
            height: 36px;
        }

    </style>

</asp:Content>

<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <h2>
        <asp:ScriptManager ID="ScriptManager1" runat="server">
        </asp:ScriptManager>
    </h2>

    <p>
         </p>

    <table class="style1">
        <tr>
            <td class="style3">
                Product Number</td>
            <td class="style2">
                <cc1:C1MaskedInput ID="C1MaskedInput1" runat="server" Width="155px"
                    InvalidInputColor="Red" Mask="00-0000" PasswordChar="*" />
                </td>
        </tr>
        <tr>
            <td class="style5">
                Order Date</td>
            <td class="style6">
                <cc1:C1DateInput ID="C1DateInput1" runat="server" Culture="en-US"
                    style="top: 0px; left: 0px; height: 1.8em" Width="155px" />
                </td>
        </tr>
        <tr>
            <td class="style4">
                Quantity</td>
            <td>
                <cc1:C1NumericInput ID="C1NumericInput1" runat="server" Culture="en-US"
```

ASP.NET

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="InputMigrationWijmo._Default" %>


<%@ Register assembly="C1.Web.Wijmo.Controls.4" namespace="C1.Web.Wijmo.Controls.C1Input" TagPrefix="cc1" %>


<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
    <style type="text/css">

        .style1
        {
            width: 37%;
            height: 96px;
        }

        .style2
        {
            width: 145px;
        }
    </style>
</asp:Content>

<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <h2>
         </h2>


    <table class="style1">
        <tr>
            <td class="style2">
                Product Number</td>
            <td>
                <wijmo:C1InputMask ID="C1InputMask1" runat="server" Mask="00-0000"
                    PasswordChar="*">
                </wijmo:C1InputMask>
            </td>
        </tr>
         <tr>
            <td class="style2">
                Order Date</td>
            <td>
                <wijmo:C1InputDate ID="C1InputDate1" runat="server">
                </wijmo:C1InputDate>
            </td>
        </tr>
        <tr>
            <td class="style2">
                Quantity</td>
            <td>
                <wijmo:C1InputNumeric ID="C1InputNumeric1" runat="server" MaxValue="10"
                    MinValue="0">
                </wijmo:C1InputNumeric>
            </td>
        </tr>
    </table>

</asp:Content>
```

```
                    MaxValue="10" style="top: 0px; left: 0px; height: 1.6em" Width="155px" />
            </td>
        </tr>
    </table>

</asp:Content>
```

Key Features

The following are some of the main features of Input for ASP.NET Web Forms that you may find useful:

- **Rich Client-side Object Model**
  Create a richer user experience on the client with mouse-action and control-focus events. Change the control format, color, mask, minimum and maximum values and more with multiple client-side methods.
- **Over 23 Built-in Masks**
  Select from over 23 built-in masks or customize your own. The C1InputMask control includes 14 built-in standard masks, including time and date formats, day of week chooser, and numeric range. The C1InputDate control includes 9 built-in standard masks, including short and long date and time formats. Both include custom format support.
- **Alerts for Invalid Input**
  Eliminate invalid input such as alphanumeric characters in a numeric input box. You can visually alert your users with red font or display an error message.
- **Rich Formatting Model**
  Format input boxes in almost any way imaginable. A rich formatting model enables developers to customize the appearance of a control's text, border, cell spacing, color scheme, and so on.
- **Design-time Support**
  Use the SmartTag to quickly access the five control-specific designers. Set masks, format, value, and culture properties, and visualize your edits with the WYSIWYG window. See Design-Time Support for more information.
- **Cultural Support**
  Define the cultural setting used by any input control – this applies to string comparison, numeric and date time formats, and special characters. See Selecting the Culture for details.
- **Drop-down and Spin Buttons**
  The specialized Input controls for date/time and numeric editing, C1InputDate, C1InputCurrency, and C1InputNumeric controls support drop-down and spin (up/down) buttons.
- **Date Picker**
  A calendar may be used as a date picker in the C1InputDate control. To enable the trigger button to open the default calendar, simply set the ShowTrigger property to True.
- **Numeric Editing**
  Set value input limits, specify number of decimal places, and indicate use of the thousands separator for the C1InputNumeric, C1InputCurrency, and C1InputPercent controls.
- **Password Protection**
  Using the C1InputMask control, protect input text by displaying password characters. At design time, simply select a character (*, for example) to substitute for the actual input characters. See Using Password Protection for an example.
- **Keyboard Support**
  Quickly edit input values using keyboard support. Move the cursor one position to the left or right or to the beginning or end, increment or decrement the range value, copy and paste, and more!
- **Theming**
  With just a click of the SmartTag, change the input control's look by selecting one of the 6 premium themes (Arctic, Midnight, Aristo, Rocket, Cobalt, and Sterling). Optionally, use ThemeRoller from jQuery UI to create a customized theme! See Input for ASP.NET Web Forms Appearance for additional information.
- **CSS Support**
  （Use a cascading style sheet (CSS) style to define custom skins. You have complete control over the input box's background, colors, borders, styles, padding, cell spacing, and more. CSS support allows you to match the input controls to your organization's standards.

Quick Start

This section will lead you through the creation of a Web form that uses the Input for ASP.NET Web Forms Edition controls. In addition, it will show you how to change the appearance, format, and functionality of the controls. By following the steps outlined in the help, you will be able to create a rich, user-friendly Web form.

Note that for brevity, the Quick Start will show the C1InputMask, C1InputDate, and C1InputCurrency controls. The C1InputNumeric and C1InputPercent controls will not appear in this Quick Start since they share similar properties to the C1InputCurrency control.

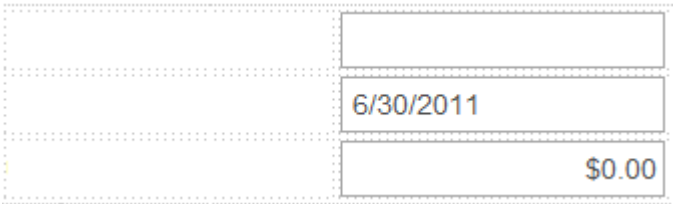
Step 1 of 5: Add Controls to Your Form

To begin, create an ASP.NET project and add the **Input for ASP.NET Web Forms** controls to your Toolbox.

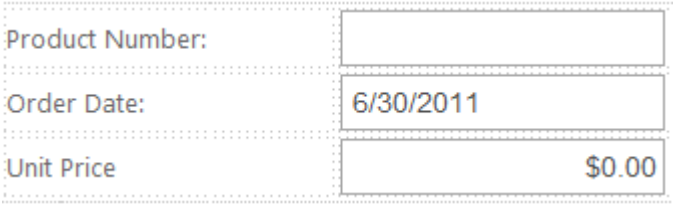
To set up your new Web form, complete the following steps:

1. Click the **Design** tab located below the Document window to switch to Design view, if necessary.
2. On the page, add a table (select **Insert Table** from the **Table menu**) with two columns and three rows. The first column will be used for text and the second column for the **Input for ASP.NET Web Forms** controls. The table appears on the form.
3. From the Toolbox, add the following controls to your page by completing a drag-and-drop operation placing each control in a cell in the table's second column:
   - C1InputMask
   - C1InputDate
   - C1InputCurrency

   The table should look similar to the following image:



4. Add text to the table. For this example, add **Product Number:, Order Date:, and Unit Price:,** respectively. You can resize and format the table to fit your needs.



5. Switch to Source view. You can see the HTML that you created by adding the table and text in Design view.

Step 2 of 5: Change the Appearance

This topic shows how to change the appearance of your Input for ASP.NET Web Forms controls using visual styles. Complete the following steps:

1. Click the Design tab located below the Document window to switch to Design view.
2. Select the first control, C1InputMask, and click the smart tag (). The C1InputMask Tasks menu appears.

3. From the Tasks menu, click the drop-down arrow next to Theme and select rocket. The other C1Input controls will also be updated with the rocket theme. The following image shows the appearance of the updated controls:

4. Switch to Source view. You can see the HTML that you created by changing the scheme in Design view.

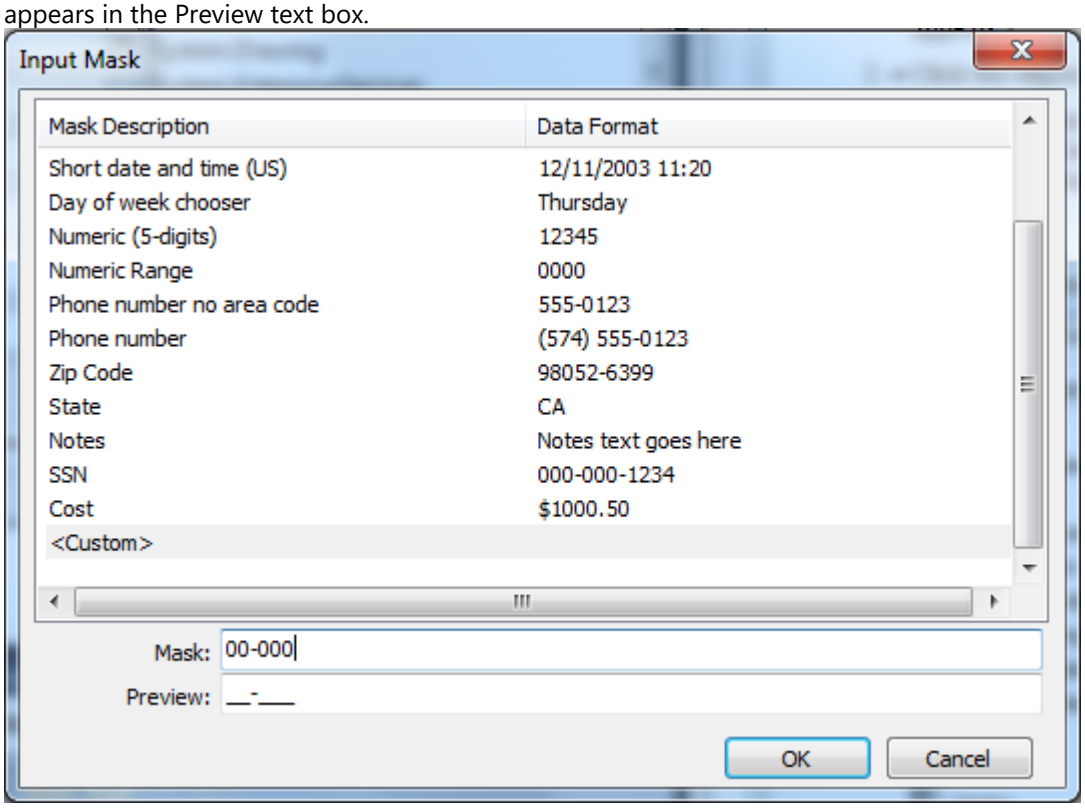
You have successfully changed the appearance of the Input for ASP.NET Web Forms controls. The next topic shows how to format the input boxes.

Step 3 of 5: Formatting the Control

This topic shows how to format the controls using the Tasks menu. To begin, click the Design tab located below the Document window to switch to Design view. Follow the steps below to format each of the Input for ASP.NET Web Forms controls on your Web form.

To format the C1InputMask control:

1. Select the C1InputMask control and click the smart tag (). The C1InputMask Tasks menu appears.
2. Click the ellipsis button next to Mask. The Input Mask dialog box appears.
3. Enter 00-000 in the Mask text box. Note that the Mask Description column automatically switches to when you start typing the mask (if the typed mask is not found in list of masks). The output value from the mask value appears in the Preview text box.



4. Click OK to close the Input Mask dialog box.
5. Select View | Properties Window and notice ui-state-error next to the InvalidClass property. If a user enters invalid input, such as alphanumeric characters, the input color appears red identifying the invalid entry, as specified in the ui-state-error in the CSS.

To format the C1InputDate control:

1. Select the C1InputDate control and click the smart tag (). The C1InputDate Tasks menu appears.
2. Enter a date format in the DateFormat text box. In this example, we'll use D. Standard format characters include the following:

Preset Pattern	Name
d	Short date pattern
D	Long date pattern
t	Short time pattern
T	Long time pattern
F	Full date/time pattern(short time)
g	General date/time pattern (short time)
G	General date/time pattern (long time)
U	Universal sortable date/time pattern

   The Resulting date pattern text box updates automatically.
3. In the C1InputDate Tasks menu, click the drop-down arrow next to Date, and select a date from the drop-down calendar.
4. Locate the ShowSpinner property in the Visual Studio Properties window, click the drop-down arrow and select True.

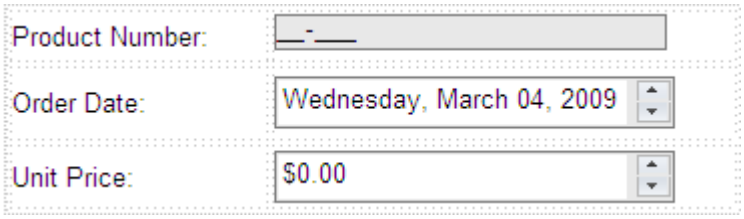
> Note: You may have to resize the controls using the Height and Width properties in the Visual Studio Properties window.

To format the C1InputCurrency control:

1. Select the C1InputCurrency control and locate the ShowSpinner property in the Visual Studio Properties window.
2. Click the drop-down arrow and select True.

You have successfully changed the format of the Input for ASP.NET Web Forms controls. To see the HTML that you created, switch to Source view.

The following image shows the appearance of the updated controls (note that the width of each control has been set to 200px):



The next topic shows how to add buttons to change the culture information for the C1InputDate and C1InputCurrency controls.

Step 4 of 5: Add a Culture Setting

This topic demonstrates how to add code to Button_Click events to set the Culture for the C1InputDate and C1InputCurrency controls. To do this, complete the following steps:

1. Click the Design tab located below the Document window to switch to Design view.
2. From the Toolbox, select the Button control and place it on your Web form (below the table) by performing a drag-and-drop operation. Repeat this step to add a second Button control to your Web form.
3. You should now have two Button controls placed next to each other on your form. Change some basic settings in the Properties window:

Button1 Properties:	Button2 Properties:
(ID) = FrenchBtn	**(ID)** = USEnglishBtn
Text = French Culture	**Text =** U.S. English Culture
Height = 25px	Height = 25px
Width = 130px	**Width** = 140px

4. Double-click the French Culture button to create an event handler for the button's Click event. Enter the following code for the FrenchBtn_Click event:

To write code in Visual Basic

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
```

To write code in C#

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");

}
```

5. The next topic shows how to run the application. It also lists tasks for you to complete to observe the functionality of the Web form.

To write code in Visual Basic

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
```

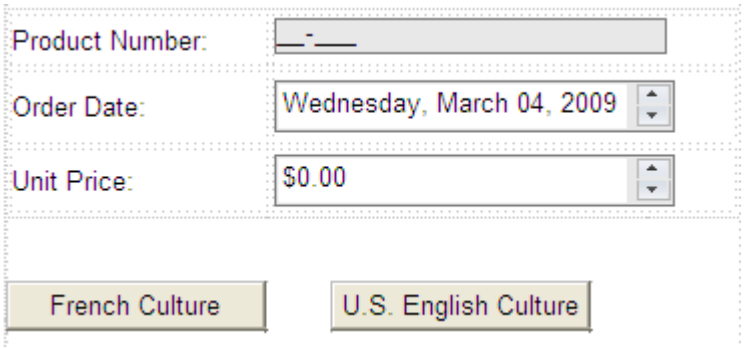
To write code in C#

C#

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

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

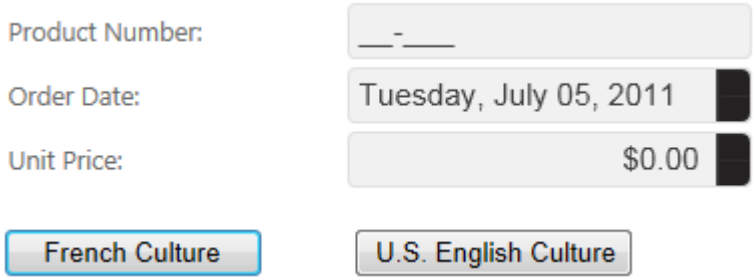
You have successfully added two button controls with culture information to your Web form. The following image shows the appearance of the updated Web form:



The next topic shows how to run the application. It also lists tasks for you to complete to observe the functionality of the Web form.

Step 5 of 5: Run Your WebApplication

Click the Start Debugging button to run your application. The following image shows the Quick Start Web form after completing each main step in the quick start (steps 1 – 4):



To observe the changes, complete the following tasks:

- Enter numeric input in the Product Number input box. Numeric characters are valid. Try entering an alphanumeric value (for example, a) and notice the input box does not allow it.
- To change the Order Date (C1InputDate control) input, complete the following tasks:
  - With your mouse pointer, click the Up/Down spin buttons.
  - Click inside the Order Date input box and press your keyboard UP/DOWN ARROWS.
- To change the Unit Price (C1InputCurrency control) input, complete the following tasks:
  - With your mouse pointer, click the Up/Down spin buttons.
  - Click inside the Unit Price input box and press your keyboard UP/DOWN ARROWS or select the current unit price and type a new unit price.
- To change the culture to French for the C1InputDate and C1InputCurrencycontrols, click the French Culture button.
- To change the culture back to U.S. English for the C1InputDate and C1InputCurrency controls, click the U.S. English Culture button.

Congratulations!

You have successfully created a basic Web form with three different Input for ASP.NET Web Forms controls.

Additionally, you customized the controls and included culture information to increase the performance in your Web form.

Design-Time Support

Input for ASP.NET Web Forms provides visual editing to make it easier to create Web input controls. The following section details each type of support available in **Input for ASP.NET Web Forms**.

Invoking the Tasks Menus

In Visual Studio, each control in **Input for ASP.NET Web Forms** includes a smart tag. A smart tag represents a short-cut **Tasks** menu that provides the most commonly used properties in each control. You can invoke each control's **Tasks** menu by clicking on the smart tag (▶) in the upper-right corner of the control. For more information on how to use the smart tags for each control in **Input for ASP.NET Web Forms**, see Smart Tags.

Invoking the Context Menus

You can easily configure any of the **Input for ASP.NET Web Forms** components at design time by using its associated context menu. For more information on **Input for ASP.NET Web Forms** context menu, see the Context Menu.

Showing the Input for ASP.NET Web Forms Control's Properties

You can access the properties for any of **Input for ASP.NET Web Forms's** controls simply by right-clicking on the control and selecting **Properties** or by selecting the class from the drop-down list box of the **Properties** window.

Smart Tags

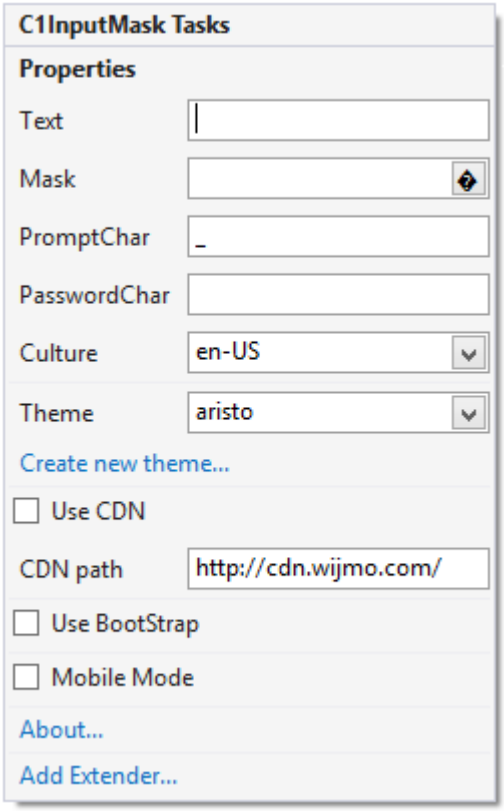
In Visual Studio, each control in **Input for ASP.NET Web Forms** includes a smart tag (). A smart tag represents a short-cut Tasks menu that provides the most commonly used properties in each control.

The following topics introduce each smart tag for the Input for ASP.NET Web Forms controls.

C1InputMask Smart Tag

The C1InputMask control provides quick and easy access to the most common C1InputMask properties through its smart tag.

To access the **C1InputMask Tasks** menu, click the smart tag (▶) in the upper-right corner of the C1InputMask control. This will open the **C1InputMask Tasks** menu.



The **C1InputMask Tasks** menu operates as follows:

Designer

Clicking **Designer** opens the **C1InputMask Designer**. For more information on the designer, see C1InputMask Designer.

Properties

The most common properties of the C1InputMask control. The **C1InputMask Tasks** menu lists the following properties:

- **Text**
  Enter text displayed to the user in the Text box.
- **Mask**
  Click the **ellipsis** button in the MaskFormat box, and the **Input Mask** dialog box appears. You can choose from preformatted masks or enter a custom mask.
- **PromptChar**
  Enter a prompt character, displayed in the absence of user input in the control, in the PromptChar box. The default is an underscore (_).
- **PasswordChar**
  In the PasswordChar box, for a C1InputMask control with a mask, enter a character to be substituted for the actual input characters
- **Culture**
  Click the drop-down arrow in the Culture box to select a culture. Each culture has different conventions for displaying dates, time, numbers, currency, and other information.
- **Theme**
  Click the drop-down arrow in the **Theme** property to select one of the built-in themes to change the appearance of the control.
- **Create new theme...**
  The **Create new theme...** option opens **ThemeRoller for Visual Studio**. This allows you to customize a theme without leaving your development environment. To find more information on using **ThemeRoller** in your

application, see ThemeRoller for Visual Studio.

- **Use CDN**
  Check the **Use CDN** check box to link to the content delivery network to access Wijmo widgets and cascading style sheets.
- **CDN path**
  The CDN path points to the content delivery network where you can access Wijmo widgets and cascading style sheets.
- **Use Bootstrap**
  The **Use Bootstrap** option applies Bootstrap theming to your control. To find more information on using **Bootstrap** theming in your application, see Bootstrap Theming.

See Using C1InputMask for details.

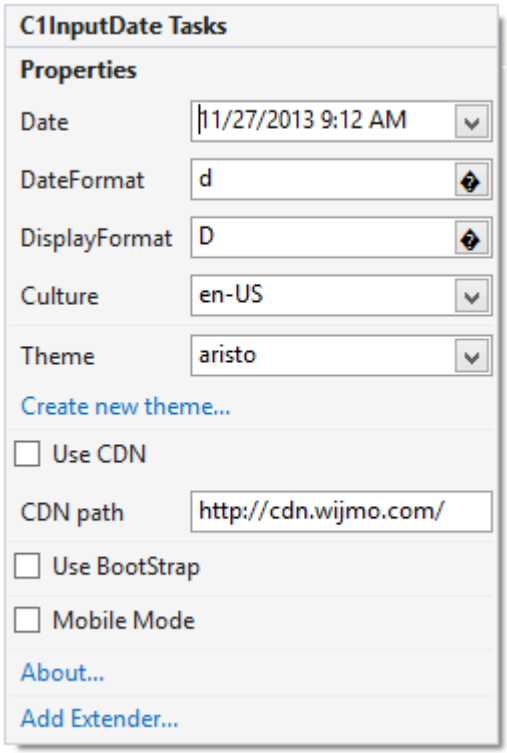
About

Clicking the **About** item displays a dialog box, which is helpful in finding the version number of **ASP.NET Web Forms Edition** and online resources.

C1InputDate Smart Tag

The C1InputDate control provides quick and easy access to the most common C1InputDate properties through its smart tag.

To access the **C1InputDate Tasks** menu, click the smart tag (▶) in the upper-right corner of the C1InputDate control. This will open the **C1InputDate Tasks** menu.



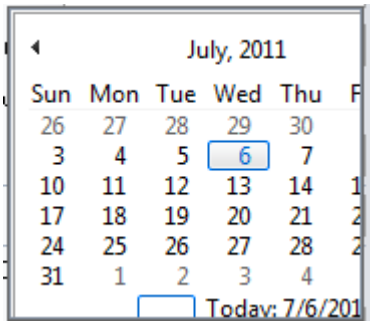
The **C1InputDate Tasks** menu operates as follows:

Designer

Clicking **Designer** opens the **C1InputDate Designer**. For more information on the designer, see C1InputDate Designer.

Properties

The most common properties of the C1InputDate control. The **C1InputDate Tasks** menu lists the following properties:

- **Date**
  Enter a date in the Date box or click the drop-down arrow to select a date from the calendar:
  

- **DateFormat**
  Enter a date format pattern in the DateFormat box. The default value is *d*.
- **Placeholder**
  Checking the Placeholder check box shows null text if the **Date** value is empty and the control loses its focus. Note that the minimal date 01.01.0001 00:00:00 is used as the null date.
- **Culture**
  Click the drop-down arrow in the Culture box to select a culture. Each culture has different conventions for displaying dates, time, numbers, currency, and other information.
- **Theme**
  Click the drop-down arrow in the **Theme** property to select one of the built-in themes to change the appearance of the control.
- **Create new theme**
  The **Create new theme** option opens **ThemeRoller for Visual Studio**. This allows you to customize a theme without leaving your development environment. To find more information on using **ThemeRoller** in your application, see ThemeRoller for Visual Studio.
- **Use CDN**
  Check the **Use CDN** check box to link to the content delivery network to access Wijmo widgets and cascading style sheets.
- **CDN path**
  The CDN path points to the content delivery network where you can access Wijmo widgets and cascading style sheets.
- **Use Bootstrap**
  Selecting the **Use Bootstrap** option applies Bootstrap theming to your control. To find more information on using **Bootstrap** theming in your application, see Bootstrap Theming.

See Using C1InputDate for details.

About

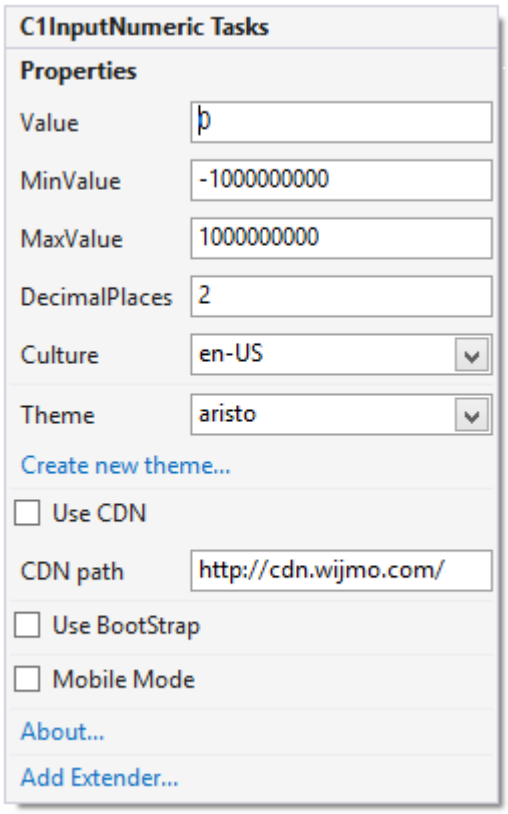
Clicking the **About** item displays a dialog box, which is helpful in finding the version number of **ASP.NET Web Forms Edition** and online resources.

C1InputNumeric Smart Tag

The C1InputNumeric control provides quick and easy access to the most common C1InputNumeric properties through its smart tag.

To access the **C1InputNumeric Tasks** menu, click the smart tag (▶) in the upper-right corner of the C1InputNumeric

control. This will open the **C1InputNumeric Tasks** menu.



The **C1InputNumeric Tasks** menu operates as follows:

Properties

The most common properties of the C1InputNumeric control. The **C1InputNumeric Tasks** menu lists the following properties:

- **Value**
  Enter a numeric value, displayed to the user, in the Value box.
- **MinValue**
  Enter the minimum value that can be entered by the user in the MinValue box.
- **MaxValue**
  Enter the maximum value that can be entered by the user in the MaxValue box.
- **DecimalPlaces**
  In the DecimalPlaces box, enter the number of decimal places to display. The default value is 2.
- **Placeholder**
  Check the Placeholder check box to show null text if the numeric **Value** is empty (equals **MinValue**) and control loses its focus.
- **Culture**
  Click the drop-down arrow in the Culture box to select a culture. Each culture has different conventions for displaying dates, time, numbers, currency, and other information.
- **Theme**
  Click the drop-down arrow in the **Theme** property to select one of the built-in themes to change the appearance of the control.
- **Create new theme...**
  The **Create new theme...** option opens **ThemeRoller for Visual Studio**. This allows you to customize a theme without leaving your development environment. To find more information on using **ThemeRoller** in your application, see ThemeRoller for Visual Studio.

- **Use CDN**
  Check the **Use CDN** check box to link to the content delivery network to access Wijmo widgets and cascading style sheets.
- **CDN path**
  The CDN path points to the content delivery network where you can access Wijmo widgets and cascading style sheets.
- **Use Bootstrap**
  Selecting the **Use Bootstrap** option applies Bootstrap theming to your control. To find more information on using **Bootstrap** theming in your application, see Bootstrap Theming.

See Using C1InputNumeric for details.

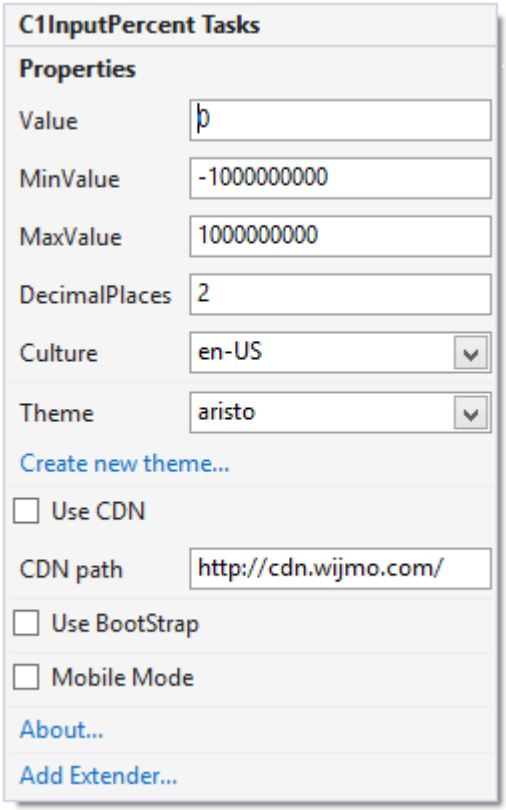
About

Clicking the **About** item displays a dialog box, which is helpful in finding the version number of **ASP.NET Web Forms Edition** and online resources.

C1InputPercent Smart Tag

The C1InputPercent control provides quick and easy access to the **most common** C1InputPercent properties through its smart tag.

To access the **C1InputPercent Tasks** menu, click the smart tag (▶) in the upper-right corner of the C1InputPercent control. This will open the **C1InputPercent Tasks** menu.



The **C1InputPercent Tasks** menu operates as follows:

Properties

The most common properties of the **C1PerecentEdit** control. The **C1InputPercent Tasks** menu lists the following properties:

- **Value**
  Enter a numeric value, displayed to the user, in the Value box.
- **MinValue**
  Enter the minimum value that can be entered by the user in the MinValue box.
- **MaxValue**
  Enter the maximum value that can be entered by the user in the MaxValue box.
- **DecimalPlaces**
  In the DecimalPlaces box, enter the number of decimal places to display. The default value is 2.
- **Placeholder**
  Check the Placeholder check box to show null text if the numeric **Value** is empty (equals **MinValue**) and control loses its focus.
- **Culture**
  Click the drop-down arrow in the Culture box to select a culture. Each culture has different conventions for displaying dates, time, numbers, currency, and other information.
- **Theme**
  Click the drop-down arrow in the **Theme** property to select one of the built-in themes to change the appearance of the control.
- **Create new theme...**
  The **Create new theme...** option opens **ThemeRoller for Visual Studio**. This allows you to customize a theme without leaving your development environment. To find more information on using **ThemeRoller** in your application, see ThemeRoller for Visual Studio.
- **Use CDN**
  Check the **Use CDN** check box to link to the content delivery network to access Wijmo widgets and cascading style sheets.
- **CDN path**
  The CDN path points to the content delivery network where you can access Wijmo widgets and cascading style sheets.
- **Use Bootstrap**
  Selecting the **Use Bootstrap** option applies Bootstrap theming to your control. To find more information on using **Bootstrap** theming in your application, see Bootstrap Theming.

See Using C1InputPercent for details.

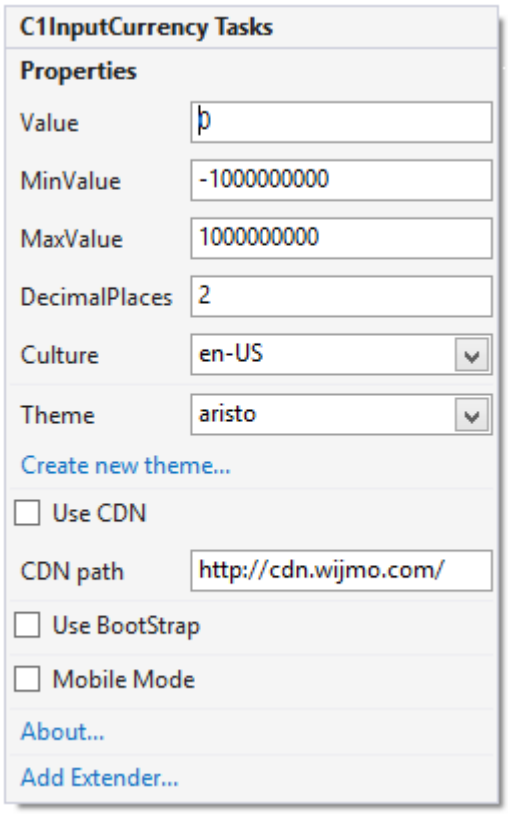
About

Clicking the **About** item displays a dialog box, which is helpful in finding the version number of **ASP.NET Web Forms Edition** and online resources.

C1InputCurrency Smart Tag

The C1InputCurrency control provides quick and easy access to the common methods and properties through its smart tag.

To access the **C1InputCurrency Tasks** menu, click on the smart tag (▶) in the upper-right corner of the **C1InputCurrency** control. This will open the **C1InputCurrency Tasks** menu.



The **C1InputCurrency Tasks** menu operates as follows:

Properties

The most common properties of the C1InputCurrency control. The **C1InputCurrency Tasks** menu lists the following properties:

- **Value**
  Enter a numeric value, displayed to the user, in the Value box.
- **MinValue**
  Enter the minimum value that can be entered by the user in the MinValue box.
- **MaxValue**
  Enter the maximum value that can be entered by the user in the MaxValue box.
- **DecimalPlaces**
  In the DecimalPlaces box, enter the number of decimal places to display. The default value is 2.
- **Placeholder**
  Check the Placeholder check box to show null text if the numeric **Value** is empty (equals **MinValue**) and control loses its focus.
- **Culture**
  Click the drop-down arrow in the Culture box to select a culture. Each culture has different conventions for displaying dates, time, numbers, currency, and other information.
- **Theme**
  Click the drop-down arrow in the **Theme** property to select one of the built-in themes to change the appearance of the control.
- **Create new theme...**
  The **Create new theme...** option opens **ThemeRoller for Visual Studio**. This allows you to customize a theme without leaving your development environment. To find more information on using **ThemeRoller** in your application, see ThemeRoller for Visual Studio.
- **Use CDN**
  Check the **Use CDN** check box to link to the content delivery network to access Wijmo widgets and cascading style sheets.

- **CDN path**
  The CDN path points to the content delivery network where you can access Wijmo widgets and cascading style sheets.
- **Use Bootstrap**
  Selecting the **Use Bootstrap** option applies Bootstrap theming to your control. To find more information on using **Bootstrap** theming in your application, see Bootstrap Theming.

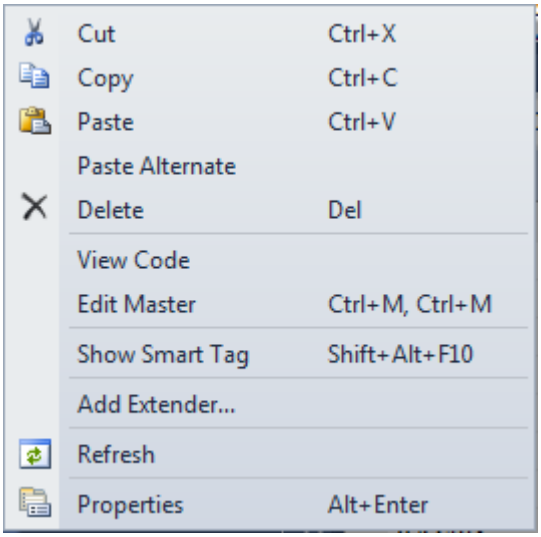
See Using C1InputCurrency for details.

About

Clicking the **About** item displays a dialog box, which is helpful in finding the version number of **ASP.NET Web Forms Edition** and online resources.

Context Menu

Each of the Input for ASP.NET Web Forms controls provide a context menu for additional functionality to use at design time. Right-click on any of the Input for ASP.NET Web Forms controls to open the following context menu:

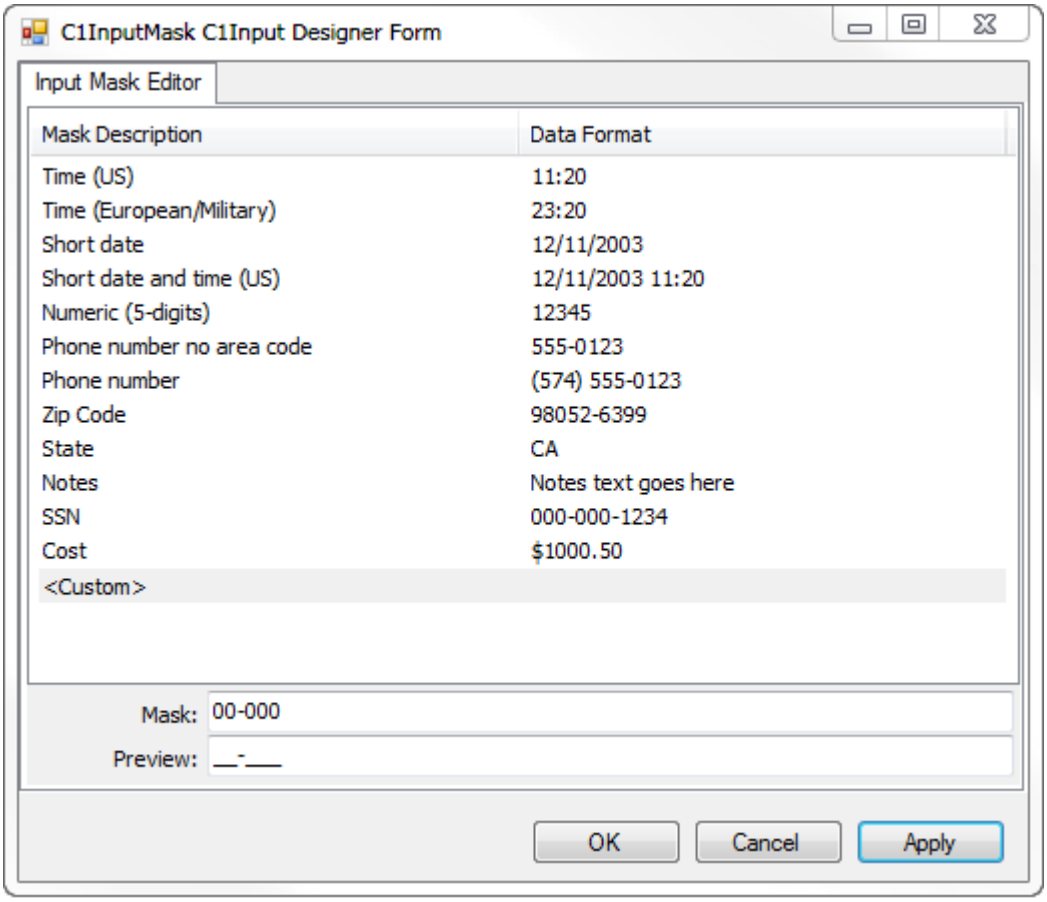


Designers

Input for ASP.NET Web Forms provides designers for C1InputMask and C1InputDate , allowing you to easily specify the mask or date format. These designers are described in the following topics.

C1InputMask Designer

To view the **C1InputMask Designer**, click on the smart tag (▶) in the upper-right corner of the C1InputMask control and select **Designer**. The following designer appears:



Input Mask Editor Tab

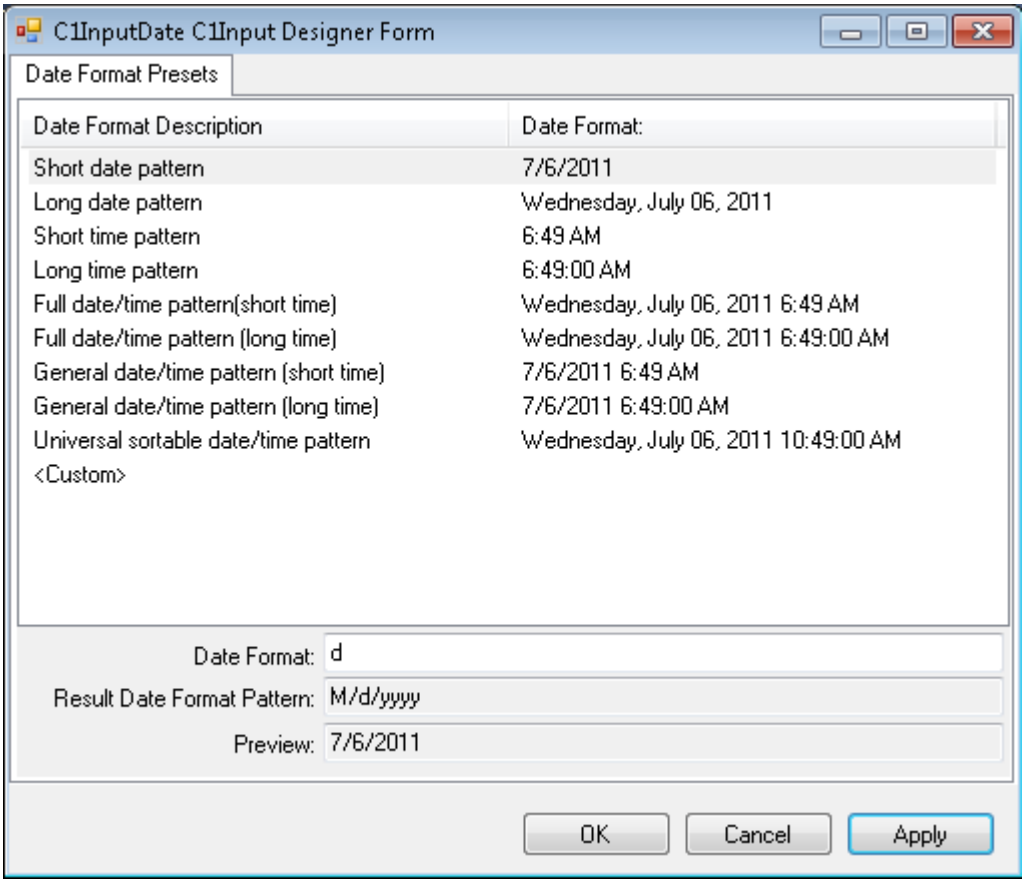
The designer's **Input Mask Editor** tab lists the C1InputMask control's mask options. The **Mask** text box shows the mask string composed of one or more placeholders (for example, 0, 9, #, and so on) and literals (for example, parentheses surrounding the area code of a telephone number). The **Preview** box shows you how the mask will appear in the Web browser.

Note: Not all input masks protect against non-existent values. For example, the preset 9-digit zip code mask will accept an input value of 00000-0000 even though no such zip code exists. Similarly, the preset mask for state abbreviations will allow PD to pass through even though there is no such state. Nonetheless, the preset masks are still useful as a first line defense against blatantly incorrect input.

C1InputDate Designer

To view the **C1InputDate Designer**, click on the smart tag (▶) in the upper-right corner of the C1InputDate control and select **Designer**.

The following designer appears:



Date Format Presets Tab

The designer's **Date Format Presets** tab lists the C1InputDate control's date format options. The **DateFormat** box shows the date format pattern composed of placeholders (dddd) and literals (for example, the divider). The **Preview** box shows you how the mask will appear in the Web browser.

Using C1InputMask

The C1InputMask control is basically an enhanced TextBox control that uses a mask to distinguish between proper and improper user input. It is the main Web control used for entering and editing information of any data type in a text form. C1InputMask serves as a base class for the C1InputDate and C1InputNumeric controls. The following image shows a C1InputMask control with a phone number Mask.

Using the Mask property, you can specify the following input without writing any custom validation logic in your application:

- Mask literals (characters that should appear directly in the C1InputMask control); for example, the hyphen (-) in a phone number.
- The type of input required at a given position in the mask; for example, numeric or alphabetic.
- Custom input characters.

Key Benefits

The key benefits of C1InputMask include the following:

- It's easy to master C1InputMask because its most basic properties and methods are similar in behavior with the System.Windows.Forms.MaskedTextBox control at the input of text. The C1InputMask properties and methods are distinctive with additional functionality.
- Ability to copy and paste to and from Input for ASP.NET Web Forms controls.
- Keyboard support:
  - LEFT/RIGHT ARROWS: move the cursor one position to the left/right.
  - HOME/END: move the cursor to the beginning or end.
  - UP/DOWN ARROWS: for enumerations and numeric ranges, increase or decrease the enumeration/numeric range value.
  - DELETE/BACKSPACE: for enumeration/numeric range, set value of enumeration/numeric range to initial value.
  - [CTRL+C and CTRL+V: support for copy/paste keyboard shortcuts.
- Ability to choose a specific culture for C1InputMask, for example, English, Spanish, German, Russian, and so on.
- Ability to change most properties of C1InputMask "on-the-fly" from client script.
- Client-side events available for you to use to increase the performance of your Web form by eliminating a postback.

Defining C1InputMask

The C1InputMask control uses a mask to distinguish between proper and improper user input. You can define the mask through the visual designers, for example, the C1InputMask Smart Tag or the C1InputMask Designer, or programmatically through the C1InputMask object.

For common C1InputMask tasks, see the C1InputMask Tasks topic.

C1InputMask Mask Types

The following table lists some examples of masks and their behaviors:

Mask	Behavior
00/00/0000	A date (day, numeric month, year) in international date format. The "/" character is a logical date separator, and will appear to the user as the date separator appropriate to the application's current culture. Note that to specify date patterns, you can use the C1InputDate control, which provides a much richer interface for

	entering dates and times.
00->L<LL-0000	A date (day, month abbreviation, and year) in United States format in which the three-letter month abbreviation is displayed with an initial uppercase letter followed by two lowercase letters.
(999) 000-0000	United States phone number, area code optional. If users do not want to enter the optional characters, they can either enter spaces or place the mouse pointer directly at the position in the mask represented by the first 0.
$999,999.00	A currency value in the range of 0 to 999999. The currency, thousandth, and decimal characters will be replaced at run time with their culture-specific equivalents.

MaskFormat is a default property for the C1InputMask control. If you define an edit mask, each character position in the control maps to either a special placeholder or a literal character. Literal characters, or literals, can give visual cues about the type of data being used. For example, the parentheses surrounding the area code of a telephone number and dash are literals: (412) 123-4567. The edit mask prevents you from entering invalid characters into the control and provides other enhancements of the user interface.

C1InputMask Characters

To enable masked input, set the MaskFormat property to a mask string composed of one or more placeholders and literals, the following table lists available placeholders:

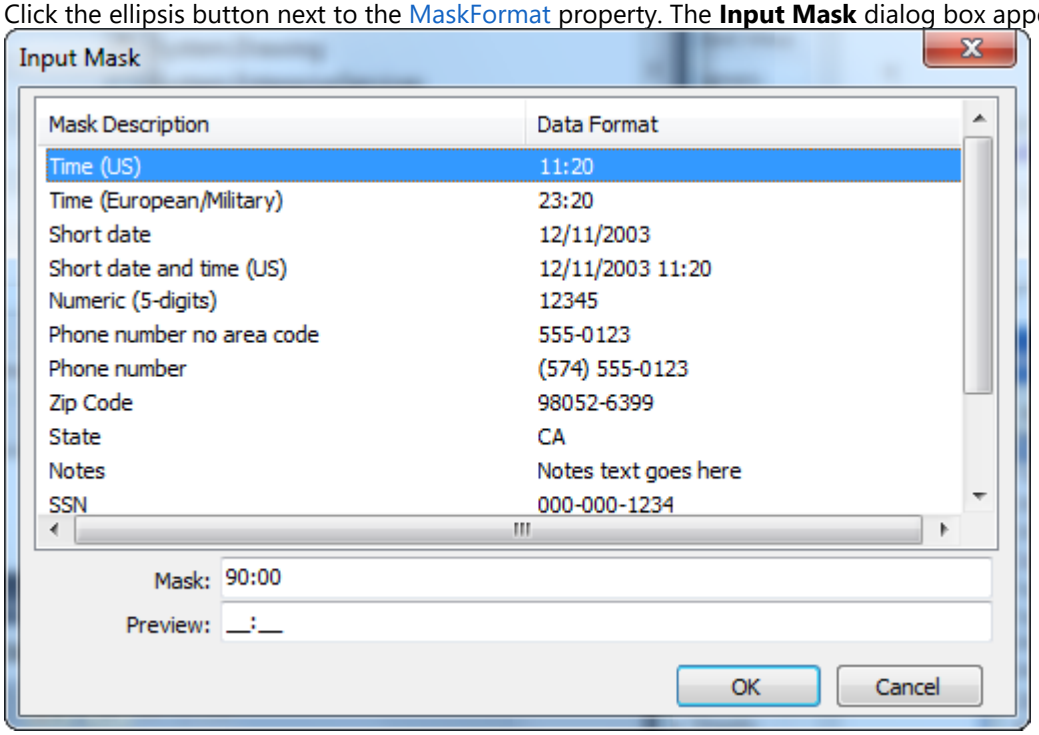
Masking Element	Description
0	Digit, required. This element will accept any single digit between 0 and 9.
9	Digit or space, optional.
#	Digit or space, optional. If this position is blank in the mask, it will be rendered as a space in the Text property. Plus (+) and minus (-) signs are allowed.
L	Letter, required. Restricts input to the ASCII letters a-z and A-Z. This mask element is equivalent to [a-zA-Z] in regular expressions.
?	Letter, optional. Restricts input to the ASCII letters a-z and A-Z. This mask element is equivalent to [a-zA-Z]? in regular expressions.
&	Character, required.
C	Character, optional. Any non-control character.
A	Alphanumeric, optional.
.	Decimal placeholder. The actual display character used will be the decimal placeholder appropriate to the Culture property.
,	Thousands placeholder. The actual display character used will be the thousands placeholder appropriate to the Culture property.
:	Time separator. The actual display character used will be the time placeholder appropriate to the Culture property.
/	Date separator. The actual display character used will be the date placeholder appropriate to the Culture property.

$	Currency symbol. The actual character displayed will be the currency symbol appropriate to the Culture property.
<	Shift down. Converts all characters that follow to lowercase.
>	Shift up. Converts all characters that follow to uppercase.
|	Disable a previous shift up or shift down.
\	Escape. Escapes a mask character, turning it into a literal. "\\" is the escape sequence for a backslash.
All other characters	Literals. All non-mask elements will appear as themselves within the C1InputMask. Literals always occupy a static position in the mask at run time, and cannot be moved or deleted by the user.

If you change a mask when C1InputMask already contains user input filtered by a previous mask, C1InputMask will attempt to migrate that input into the new mask definition.

To set the C1InputMask.Mask property, follow these steps:

1. Select the C1InputMask control, and click its smart tag to open the **C1InputMask Tasks** menu.
2. Click the ellipsis button next to the MaskFormat property. The **Input Mask** dialog box appears.



3. Select a **Data Format** and then define the mask in the **Mask** text box. Notice the **Preview** text box displays a preview of the mask.
4. Click **OK** to close the **Input Mask** dialog box.

Using C1InputDate

The C1InputDate control, derived from C1InputMask, is specialized for editing the date and time. With the date-specific masked editing field, users can enter dates directly in the control, or use the UP/DOWN ARROW keys to increase/decrease the value of the current field. The following image shows a C1InputDate control:

Using the DateFormat property, you can specify the following input without writing any custom validation logic in your application:

- Mask literals (characters that should appear directly in the C1InputDate control); for example, the colon (:) in time or the separator (/) in a date.
- The type of input required at a given position in the mask; for example, numeric or alphabetic.
- Custom input characters.

Key Benefits

The key benefits of C1InputDate include the following:

- C1InputDate control renders a date editor. Use the DateFormat property to set/get the date format character or pattern.
- You can set the C1InputDate control to interact with the C1Calendar control. Use the Calendar property to integrate C1InputDate with C1Calendar.
- Ability to choose a specific culture for C1InputDate, for example, English, Spanish, German, Russian, and so on. The date pattern and other aspects of date string depend on the selected Culture property. プロパティによって決まります。
- Client-side events available for you to use to increase the performance of your Web form by eliminating a postback.

Defining C1InputDate

You can define the date pattern through the visual designers, for example, the C1InputDate Smart Tag or the C1InputDate Designer, or programmatically through the C1InputDate object.

When the user edits the date at run time, note the following:

- Formatted fields represented in string form, such as month or day of the week in Long date pattern, can be typed as numbers on the keyboard and their string representation is updated automatically.
- UP/DOWN ARROWS can be used to increase/decrease the current field.

C1InputDate General Properties

The following table lists general properties of the C1InputDate control:

Property	Description
Date	DateTime value.
DateFormat	Date format pattern or date format character (preset character).
Placeholder	The Placeholder property determines the text that will be displayed for blank status.
Calendar	Determines the Calendar element for a date input.

C1InputDate Format Characters

The C1InputDate format characters are case-sensitive. The following table lists standard format characters:

Preset Pattern	Name
d	Short date pattern
D	Long date pattern
t	Short time pattern
T	Long time pattern
F	Full date/time pattern(short time)
g	General date/time pattern (short time)
G	General date/time pattern (long time)
U	Universal sortable date/time pattern

C1InputDate Format Patterns

The C1InputDate patterns are case-sensitive. The following table lists standard patterns:

Format Pattern	Description
d	The day of the month. Single-digit days will not have a leading zero.
dd	Two-digit day of the month. Single-digit days will have a leading zero.
ddd	The abbreviated name of the day of the week.
dddd	The full name of the day of the week.
M	The numeric month. Single-digit months will not have a leading zero.
MM	The numeric month. Single-digit months will have a leading zero.
MMM	The abbreviated name of the month.
MMMM	The full name of the month.
y	The year without the century. If the year without the century is less than 10, the year is displayed with no leading zero.
yy	The year without the century. If the year without the century is less than 10, the year is displayed with a leading zero.
yyyy	Four-digit year (0000 through 9999).
h	The hour in a 12-hour clock. Single-digit hours will not have a leading zero.
hh	The hour in a 12-hour clock. Single-digit hours will have a leading zero.
H	The hour in a 24-hour clock. Single-digit hours will not have a leading zero.
HH	The hour in a 24-hour clock. Single-digit hours will have a leading zero.
m	The minute. Single-digit minutes will not have a leading zero.
mm	The minute. Single-digit minutes will have a leading zero.
s	The second. Single-digit seconds will not have a leading zero.

ss	The second. Single-digit seconds will have a leading zero.
t	The first character in the AM/PM designator.
tt	The AM/PM designator.

Note: If characters in pattern are enclosed in single quotation marks then these characters are treated as literals. For example, pattern: 'dd:' dd.MM.yyyy for date 03.07.2006 outputs string "dd: 03.07.2006".

Using C1InputNumeric

The C1InputNumeric control, derived from C1InputMask, is specialized for editing numeric values. Using the numeric editor, you can specify input without writing any custom validation logic in your application. The following image shows a C1InputNumeric control:

Key Benefits

The key benefits of C1InputNumeric include the following:

- C1InputNumeric control renders a numeric editor.
- Ability to choose a specific culture for C1InputNumeric, for example, English, Spanish, German, Russian, and so on. Note that C1InputNumeric uses the selected Culture property to render number group separators (thousands separator), decimal separator, and signs.
- Numeric range support with the ability to easily change MinValue and MaxValue properties.
- Client-side events available for you to use to increase the performance of your Web form by eliminating a postback.

Defining C1InputNumeric

The C1InputNumeric control has numeric range support for displaying numerical data. Since the C1InputNumeric control deals strictly with numbers, there is no input mask to specify. To define the C1InputNumeric, you simply supply the minimum and maximum values, the number of decimal places (can be zero), and indicate whether culture-specific thousands separators should be displayed.

You can define the value of the C1InputNumeric control through the C1InputNumeric Smart Tag or programmatically through the C1InputNumeric object.

Note that when the user edits the value at run time, the UP ARROW or DOWN ARROW keys can be used to increase or decrease the current field.

For common C1InputNumeric tasks, see the C1InputNumeric Tasks topic.

C1InputNumeric General Properties

The following table lists general properties of the C1InputNumeric control:

Property	Description
Value	Double, numeric value of the C1InputNumeric control.
MinValue	Minimum value that can be entered.
MaxValue	Maximum value that can be entered.
DecimalPlaces	Indicates the number of decimal places to display (Default: 2).
Placeholder	The Placeholder property determines the text that will be displayed for blank status.

Most of the properties and events of the C1InputNumeric control are the same as the C1InputMask control, except for hidden properties that are not used in numeric controls (such as the following: AllowPromptAsInput, MaskFormat, HidePromptOnLeave, PasswordChar, PromptChar, ResetOnPrompt, ResetOnSpace, SkipLiterals).

Using C1InputPercent

The C1InputPercent control, derived from C1InputNumeric, is specialized for editing percent values. Using the numeric editor, you can specify input without writing any custom validation logic in your application. The following image shows a C1InputPercent control:

Key Benefits

The key benefits of C1InputPercent include the following:

- C1InputPercent control renders a numeric editor. C1InputPercent can be used to input percent values.
- Ability to choose a specific culture for C1InputPercent, for example, English, Spanish, German, Russian, and so on. Note that the number pattern and other aspects of number string (percent symbols and align) depends on the selected Culture property.
- Client-side events available for you to use to increase the performance of your Web form by eliminating a postback.

Note: The C1InputPercent control's properties are the same as the C1InputNumeric control.

Defining C1InputPercent

The C1InputPercent control has numeric range support for displaying numerical data. You can define the value of the C1InputPercent control through the C1InputPercent Smart Tag or programmatically through the C1InputPercent object.

Note that when the user edits the value at run time, the UP ARROW or DOWN ARROW keys can be used to increase or decrease the current field.

Note that the C1InputPercent control's properties are the same as the C1InputNumeric control. For common C1InputPercent tasks, see the C1InputNumeric Tasks topic.

Using C1InputCurrency

The C1InputCurrency control, derived from C1InputNumeric, is specialized for editing currency values. Using the numeric editor, you can specify input without writing any custom validation logic in your application. The following image shows a C1InputCurrency control:

Key Benefits

The key benefits of C1InputCurrency include the following:

- C1InputCurrency control renders a numeric editor. C1InputCurrency can be used to input currency values.
- Ability to choose a specific culture for C1InputCurrency, for example, English, Spanish, German, Russian, and so on. Note that the number pattern and other aspects of number string (currency symbols and align) depends on the selected Culture property.
- Client-side events available for you to use to increase the performance of your Web form by eliminating a postback.

Note: The C1InputCurrency control's properties are the same as the C1InputNumeric control.

Defining C1InputCurrency

The C1InputCurrency control has numeric range support for displaying numerical data. You can define the value of the C1InputCurrency control through the C1InputCurrency Smart Tag or programmatically through the C1InputCurrency object.

Note that when the user edits the value at run time, the UP ARROW or DOWN ARROW keys can be used to increase or decrease the current field.

Note that the C1InputCurrency control's properties are the same as the C1InputNumeric control. For common C1InputCurrency tasks, see the C1InputNumeric Tasks topic.

Appearance

Change the input control's look by selecting one of the six premium themes (*Arctic*, *Midnight*, *Aristo*, *Rocket*, *Cobalt*, and *Sterling*). Or you can use ThemeRoller from jQuery UI to create your own customized theme!

Built-in Themes

Input for ASP.NET Web Forms provides six built-in themes for each **Input for ASP.NET Web Forms** control, allowing you to automatically format the controls. The themes include the following: **arctic**, **aristo**, **cobalt**, **midnight**, **rocket**, and **sterling**.

The examples below show the C1InputMask control; however, the themes for all controls are the same.

arctic

The following image displays the **arctic** theme:

aristo

The following image displays the **aristo** theme. This is the default format for all of the **Input for ASP.NET Web Forms** controls:

cobalt

The following image displays the **cobalt** theme:

midnight

The following image displays the **midnight** theme:

rocket

The following image displays the **rocket** theme:

sterling

The following image displays the **sterling** theme:

CSS Selectors

You can style any **C1Input** elements using CSS styles to make their appearance truly unique. To make the customization process easier, ComponentOne includes CSS selectors for each of its six built-in themes. For more information on themes, see Built-in Themes.

You can apply general CSS properties such as background, text, font, border, outline, margin, padding, list, and table

to applicable CSS selectors.

The following topic details the common individual CSS selectors and grouped CSS selectors. You can combine the individual CSS selectors as a group to make the CSS selector more specific and strong.

CSS Selectors	Description
.wijmo-input-trigger	Applies the style to the trigger button.
.wijmo-wijinput-spinner	Applies the style to the spinner button.
.wijmo-wijinput-spinup	Applies the style to the up spin button.
.wijmo-wijinput-spindown	Applies the style to the down spin button.
.wijmo-wijinput-input	Applies the style to the outmost container for all the input types.
.wijmo-wijinput-mask	Applies the style to the outmost container for the mask input (C1InputMask).
.wijmo-wijinput-numeric	Applies the style to the outmost container for the number input types (C1InputNumeric, C1InputCurrencyand C1InputPercent).
.wijmo-wijinput-date	Applies the style to the outmost container for the date input types (C1InputDate).
.wijmo-wijinput-wrapper	Applies the style to the direct wrapper for the input element.
.wijmo-wijinput ui-state-focus	Applies the style to the outmost container for all the input types when it is in focused state.

Working with the Client-Side

The **Input for ASP.NET Web Forms** controls have a very rich client-side object model since most of their members are identical to the members in the server-side control.

When a **C1Input** control is rendered, an instance of the client-side control will be created automatically. This means that you can enjoy the convenience of accessing properties and methods of the **C1Input** controls without having to postback to the server.

Using client-side code, you can implement many features in your Web page without the need to send information to the Web server, which takes time. Thus, using the client-side object model can increase the efficiency of your Web site.

Client-Side Events

Input for ASP.NET Web Forms includes several client-side events that allow you to manipulate the **C1Input** controls when an action such as an invalid character is entered.

You can use the sever-side properties, listed in the Client Side Event table, to specify the name of the JavaScript function that will respond to a particular client-side event. For example, to assign a JavaScript function called **invalidInput** to respond when an invalid character is entered, you would set the OnClientInvalidInput property to **invalidInput**.

The following table lists the events that you can use in your client scripts. These properties are defined on the server side, but the actual events or the name you declare for each JavaScript function are defined on the client side.

Event Server-Side Property Name	Event Name	Description
OnClientInitialized	initialized	Occurs after the widget is initialized.
OnClientInitializing	initializing	Occurs before the widget is initialized.
OnClientInvalidInput	invalidInput	Occurs when an invalid character is entered.
OnClientTextChanged	textChanged	Occurs when the text of the input is changed.
OnClientTriggerMouseDown	triggerMouseDown	Occurs when the mouse is pressed down on the trigger button.
OnClientTriggerMouseUp	triggerMouseUp	Occurs when the mouse is released on the trigger button.
OnClientDateChanged (InputDate only)	dateChanged	Occurs after the date value changed.
OnClientValueBoundsExceeded (InputNumber only)	valueBoundsExceeded	Occurs when the value of the input exceeds the valid range.
OnClientValueChanged (InputNumber only)	valueChanged	Occurs after the value changed.

For task-based help on client-side events, see the Client-Side Event Tasks topic. Descriptions and syntax examples for the **C1Input** client-side events can also be found at http://wijmo.com/wiki/index.php/InputMask.

Samples

Please be advised that this ComponentOne software tool is accompanied by various sample projects and/or demos which may make use of other development tools included with the ComponentOne Studios.

Samples can be accessed from the **ComponentOne Sample Explorer**.

The following pages within the ControlExplorer sample installed with **ASP.NET Web Forms Edition** detail the **Input for ASP.NET Web Forms** controls' functionality:

C# Sample

Sample	Description
C1InputDate:	
DatePicker	This sample demonstrates that when the **showTrigger** property is set to true, clicking the trigger button opens the default calendar to be used as a date picker.
DropDown	This sample shows how you can easily create a drop-down box for date input.
Overview	This sample shows the C1InputDate control, an input control that is specialized for editing Date/Time values.
C1InputMask:	
DropDown	This sample shows how predefined input values can be specified using the **ComboItems** property.
FirstName	This sample illustrates how you can use the **MaskFormat** property to allow only text input.
Overview	This samples shows the C1InputMask, an input web control that allows users to type text based on the mask.
Password	This sample demonstrates how to display password characters, such as * and #, by setting the **PasswordChar** property.
C1InputNumber:	
Currency	The C1InputCurrency control is an input control that is specialized for editing currency values.
DropDown	This sample demonstrates how you can create a drop-down list with currency items by setting the **ComboItems** properties.
Increment	This sample demonstrates how to use the **Increment** property to create a custom increment.
Overview	The C1InputNumeric control is an input control that is specialized for editing numeric values.
Percent	The C1InputNumeric control is an input control that is specialized for editing numeric values.

Task-Based Help

The task-based help assumes that you are familiar with programming in Visual Studio. By following the steps outlined in the Help, you will be able to create projects demonstrating a variety of Input for ASP.NET Web Forms features and get a good sense of what Input for ASP.NET Web Forms can do.

Each task-based help topic also assumes that you have created a new ASP.NET project.

General C1Input Control Tasks

This section shows how to perform tasks that are not specific to any one C1Input control. The following topics assume that you have added a C1Input control to your Web form.

Creating ToolTip Text

To show ToolTip text for any of the **Input for ASP.NET Web Forms** controls, enter text for the **ToolTip** property. For example, you can create a ToolTip text for the C1InputDate control that reads, "Enter the required date".

Using the Designer

To display a ToolTip when the user hovers over the C1InputDate control, complete the following steps:

1. Open the **C1InputDate Tasks** menu and select **Designer**. The **C1InputDate Designer** appears.
2. Select the **Properties** tab and locate the **ToolTip** property.
3. Enter the following text in the **ToolTip** text box: "Enter the required date".

Note that you can preview the control's ToolTip in the Designer's preview window.

Using HTML Markup

To display a ToolTip when the user hovers over the C1InputDate control, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputDate ID="C1InputDate1" runat="server"
      Date="2006-08-07"
      ToolTip="Enter the required date">
</cc1:C1InputDate>
```

Using Code

To display a ToolTip when the user hovers over the C1InputDate control, double-click the Web page to create an event handler for the **Load** event. Enter the following code for the **Page_Load** event:

To write code in Visual Basic

Visual Basic

```
' Set a Date
Me.C1InputDate1.Date = "2006-08-07"
' Add a ToolTip
Me.C1InputDate1.ToolTip = "Enter the required date"
```

To write code in C#

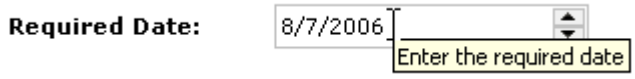
C#

```
// Set a Date
```

```
this.C1InputDate1.Date = "2006-08-07";
// Add a ToolTip
this.C1InputDate1.ToolTip = "Enter the required date";
```

This topic illustrates the following:

Run the project and notice that when you hover over the C1InputDate control, the following ToolTip is displayed:



Changing the Theme

You can format the **Input for ASP.NET Web Forms** controls with one of six built-in themes. We will use C1InputMask in the following examples.

Changing the Theme Using the Smart Tag

You can change the style of the **Input for ASP.NET Web Forms** controls at design time using the control's **Tasks** menu.

1. Click the C1InputMask smart tag to open the **C1InputMask Tasks** menu.
2. Click drop-down arrow next to **Theme**.
3. Select one of the built-in themes listed. The theme is applied to the C1InputMask control.

Changing the Theme in Code

To change the theme of an **Input for ASP.NET Web Forms** control programmatically, use the following code. In this example, **midnight** is used, but you can replace it with any of the built-in themes.

To write code in Visual Basic

Visual Basic

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

To write code in C#

C#

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

Adding a Custom Theme

Input for ASP.NET Web Forms provides six built-in themes, but if you prefer to use a different theme, you can customize a theme using ThemeRoller for Visual Studio, choose an existing theme using a CDN URL or create your own custom theme with the jQuery ThemeRoller Web application. We will use C1InputDate in the following examples.

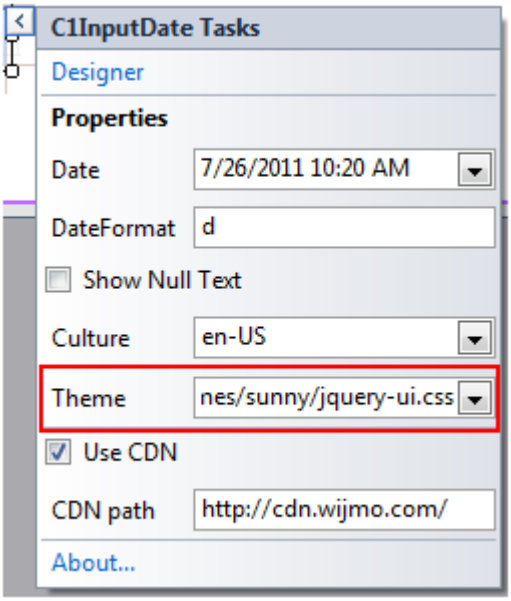
Using ThemeRoller for Visual Studio

The new **ThemeRoller for Visual Studio** makes designing beautiful themes for **ASP.NET Web Forms Edition** controls easy. To find more information on creating and editing a **ThemeRoller for Visual Studio** theme, please see ThemeRoller for Visual Studio.

Using a CDN URL

1. Click the C1InputDate smart tag to open the **Tasks** menu.
2. Select **Use CDN**.

3. In the **Theme** property, enter a CDN URL to specify the theme; CDN URLs can be found at http://blog.jqueryui.com/2011/06/jquery-ui-1-8-14/. In this example, we'll use the *sunny* theme: http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.14/themes/sunny/jquery-ui.css.

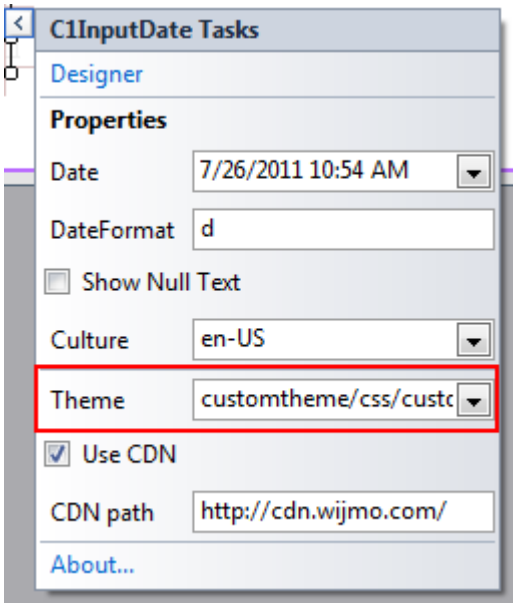


This theme setting is stored in the **<appSettings>** of the **Web.config** file. In the Solution Explorer, double-click the **Web.config** file. Notice the **<appSettings>** tag contains a **WijmoTheme** key and value; this is where the CDN URL you added is specified.

4. Run the project and notice the theme is applied to C1InputDate.

Using jQuery ThemeRoller

1. Go to http://jqueryui.com/themeroller/.
2. On the **Roll Your Own** tab, change the settings to create a custom theme; you can customize fonts, colors, backgrounds, borders, and more. Or click the **Gallery** tab and select an existing theme.
3. Click the **Download** button and then click **Download** again on the **Build Your Download** page.
4. Save and unzip the theme .zip file to a folder within your Visual Studio project folder. In this example, we created a **customtheme** folder.
5. In the Solution Explorer, click **Show All Files** and then right-click the **customtheme** folder and select **Include in Project**.
6. Click the C1InputDate smart tag to open the **Tasks** menu.
7. Select **Use CDN**.
8. In the **Theme** property, enter the path to your custom theme .css, for example, **customtheme/css/custom-theme/jquery-ui-1.8.14.custom.css**.



This theme setting is stored in the **<appSettings>** of the **Web.config** file. In the Solution Explorer, double-click the **Web.config** file. Notice the **<appSettings>** tag contains a **WijmoTheme** key and value; this is where the custom theme you added is specified.

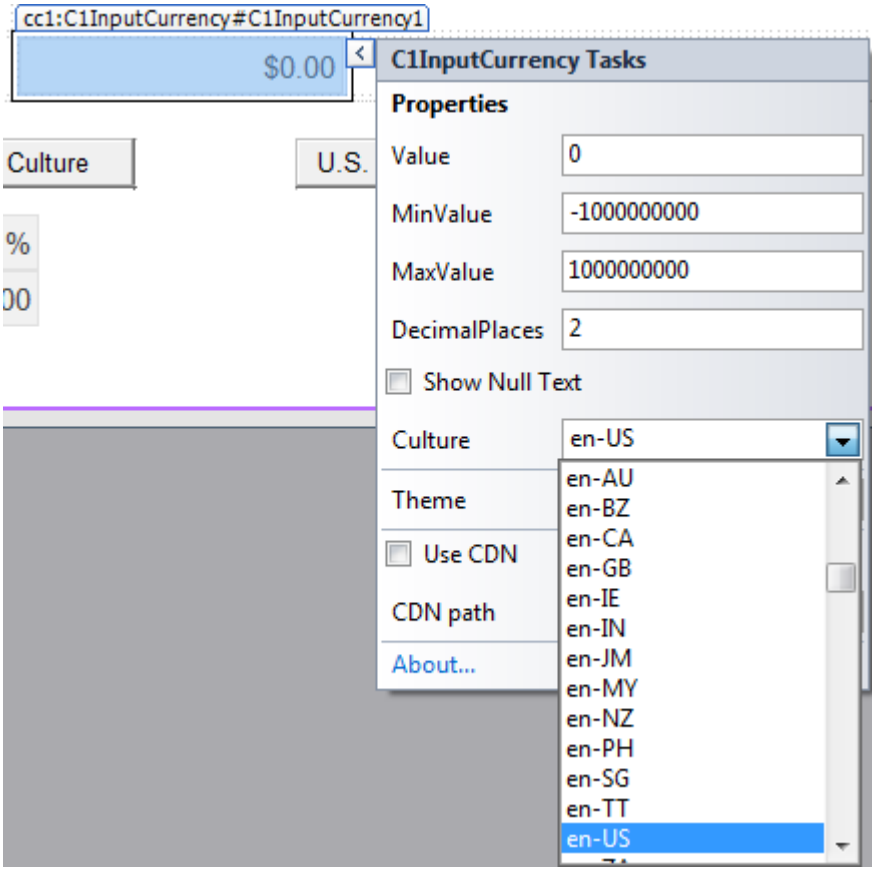
9. Run the project and notice the theme is applied to C1InputDate.

Selecting the Culture

Note that the following topic shows how to use the Culture property for the C1InputCurrency control; however, the Culture property is available for all **Input for ASP.NET Web Forms** controls.

Using the Designer

You can choose a specific culture for any of the **Input for ASP.NET Web Forms** controls. To set the Culture property for the control, simply open its **Tasks** menu and select a culture from its drop-down list.



Using HTML Markup

To set the Culture value, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputCurrency ID="C1InputCurrency1" runat="server"
      Culture="English (United Kingdom)">
</cc1:C1InputCurrency>
```

Using Code

To set the Culture for the C1InputCurrency control, double-click the Web page to create an event handler for the **Load** event. Enter the following code for the **Page_Load** event:

To write code in Visual Basic

Visual Basic

```
Me.C1InputCurrency1.Culture = New System.Globalization.CultureInfo("en-GB")
```

To write code in C#

C#

```
this.C1InputCurrency1.Culture = new System.Globalization.CultureInfo("en-GB");
```

This topic illustrates the following:

The following C1InputCurrency control shows the British Pound:

£0.00

C1InputMask Tasks

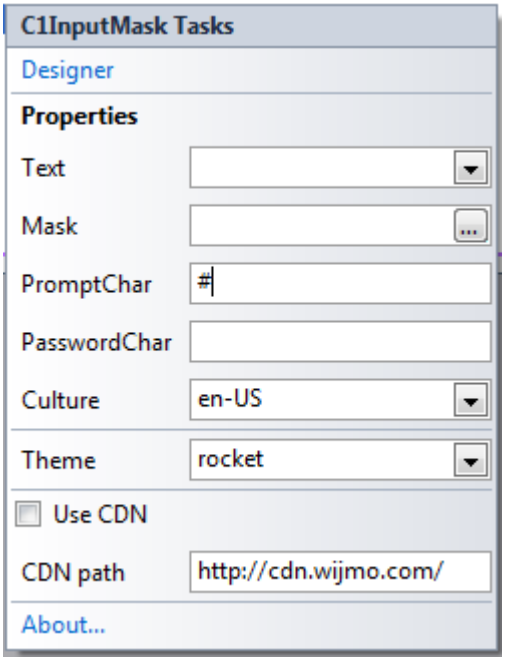
This section shows how to perform specific tasks using the C1InputMask control. The following topics assume that you have added a C1InputMask control to your Web form.

Changing the Prompt Character

At run time, the C1InputMask control displays the mask as a series of prompt characters (for example, # or _). The prompt characters represent each editable mask position. To change the prompt character, use the PromptChar property. This example uses the C1InputMask control with the **Phone number** mask: (999) 000-0000.

To change the prompt character using the Tasks menu:

To change the phone number PromptChar property, open the **C1InputMask Tasks** menu and enter the number sign (#) in the **PromptChar** text box.



To change the prompt character using .html markup:

To change the prompt character to the number sign (#) for the C1InputMask control, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputMask ID="C1InputMask1" runat="server"
    Mask="(999) 000-0000"
    Text="412"
    PromptChar="#">
</cc1:C1InputMask>
```

This topic illustrates the following:

Run the project and notice that the number sign (#) is displayed as the prompt character in the Web browser, as shown here:

Note that the 412 area code appears instead of the number signs since the Text property was specified for the control.

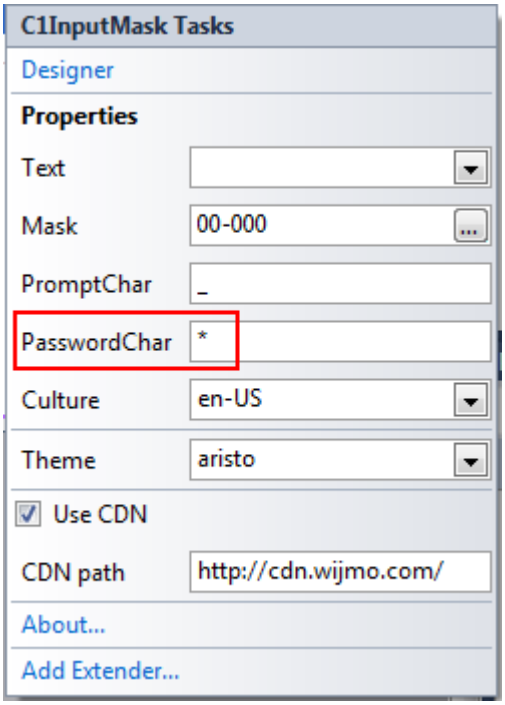
For details on hiding the prompt characters when the input box loses focus, see the Hiding the Prompt Character on Leave topic.

Using Password Protection

You can protect input text by displaying password characters for the C1InputMask control so that the actual characters entered are not visible.

To set password characters using the Tasks menu:

Open the **C1InputMask Tasks** menu and enter an asterisk (*) in the **PasswordChar** text box.



To change the password character using .html markup:

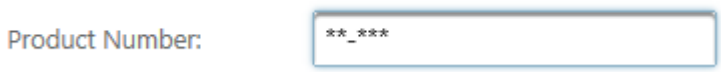
To change the password character to an asterisk (*) for the C1InputMask control, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputMask ID="C1InputMask1" runat="server" Mask="00-000" Width="200px"
                    PasswordChar="*">
                </cc1:C1InputMask>
```

This topic illustrates the following:

Run the project and enter characters in the C1InputMask control. Notice that an asterisk (*) is displayed for each character as it is entered, as shown here:



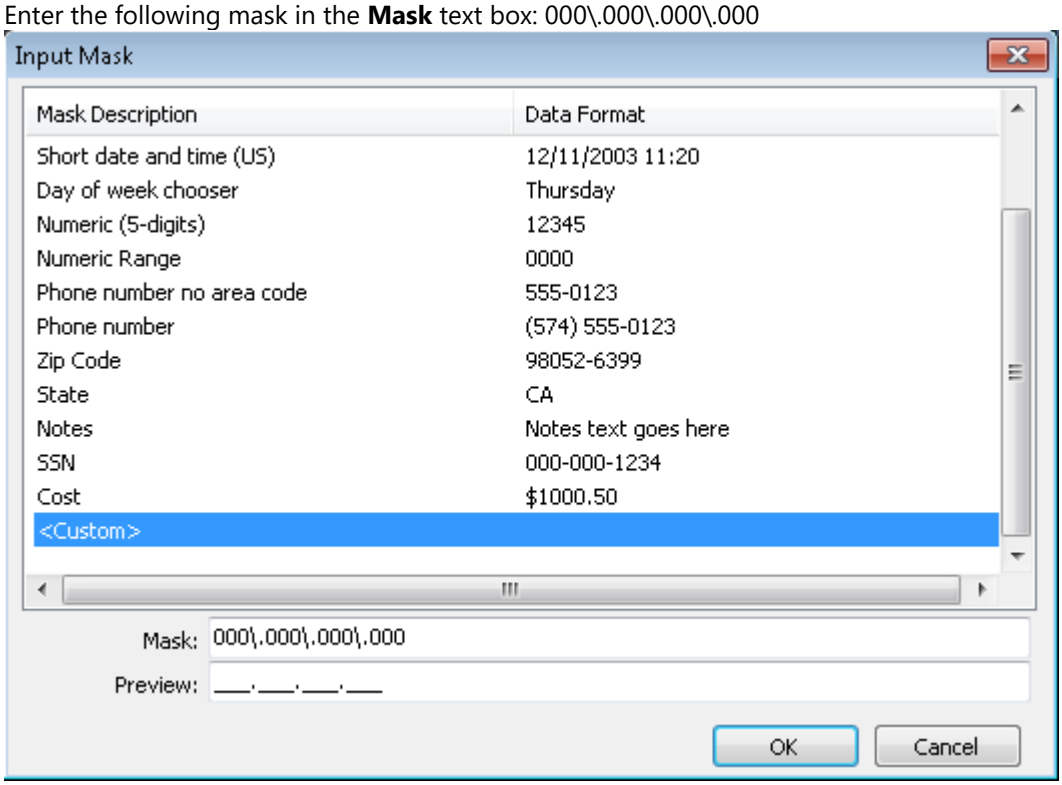
Creating an IP Address Mask

The following example demonstrates how to use numeric ranges to represent a masked text box for editing an IP address. This example uses the C1InputMask control with the custom mask: 000\.000\.000\.000.

To create an IP address mask using the Tasks menu:

To display the **IP address** value with specific text, complete the following tasks:

1. Open the **C1InputMask Tasks** menu and click the MaskFormat property's **ellipsis** button to open the **Input Mask** dialog box.
2. Enter the following mask in the **Mask** text box: 000\.000\.000\.000



   Note that the Designer automatically switches to <Custom> when you start typing the mask (if the typed mask is not found in list of masks).
3. Click **OK**.
4. With the **Tasks** menu still open, enter **192168001001** in the Text text box.

To create an IP address mask using .html markup:

To create the masked value for an IP address, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputMask ID="C1InputMask1" runat="server" Mask="000\.000\.000\.000"
        Text="192168001001 ">
    </cc1:C1InputMask>
```

Note: One character "<" or ">" forces the next characters to shift down or shift up instructions. Character "." without "\" acts as a decimal placeholder and actual display characters used will be the decimal placeholder appropriate to the value of the Culture property.

This topic illustrates the following:

Run the project and notice that the IP address mask with the 192168001001 text is displayed in the Web browser, as shown here:

Creating a Phone Number Mask

The following example demonstrates using enumeration parts in the MaskFormat property. This example uses the C1InputMask control with the **Phone Number** mask: (999) 000-0000.

To create a phone number mask using the Tasks menu:

To display the **Phone Number** value with a 412 area code, complete the following tasks:

1. Open the **C1InputMask Tasks** menu and click the **Mask** property's **ellipsis** button to open the **Input Mask** dialog box.
2. Choose **Phone number** for the mask value and click **OK**.
3. With the **Tasks** menu still open, type **412** in the Text text box.

To create a phone number mask using .html markup:

To display the **Phone Number** value with a 412 area code, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputMask ID="C1InputMask1" runat="server"
    Mask="(999) 000-0000"
    Text="412"
</cc1:C1InputMask>
```

Note: Character "9" acts as a masking element: digit or space, optional. Character "0" acts as a masking element: digit, required. This masking element will accept any single digit between 0 and 9.

This topic illustrates the following:

Run the project and notice that the **Phone Number** mask with the 412 text is displayed in the Web browser, as shown here:

Displaying the Date Mask without Prompt Characters

To create an input box for the date that does not contain prompt characters (for example, " / / "), use the C1InputMask control and set the PromptChar property to space, " ".

To create a short date mask without prompt characters using the Tasks menu:

To create a **Short Date** input box that does not contain prompt characters, complete the following tasks:

1. Open the **C1InputMask Tasks** menu and click the **Mask** property's **ellipsis** button to open the **Input Mask** dialog box.
2. Choose **Short Date** for the mask value and click **OK**.
3. With the **Tasks** menu still open, type a space character (" ") in the PromptChar text box. Note that you must delete the default underscore (_).

To create a short date mask without prompt characters using .html markup:

To create a **Short Date** input box that does not contain prompt characters, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputMask ID="C1InputMask1" runat="server" Mask="00/00/0000" PromptChar=" ">
</cc1:C1InputMask>
```

To create a short date mask without prompt characters using code:

To create a short date mask without prompt characters for the C1InputMask control, double-click the Web page to create an event handler for the **Load** event. Enter the following code for the **Page_Load** event:

To write code in Visual Basic

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

To write code in C#

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

This topic illustrates the following:

Run the project and notice that the **Short Date** mask is displayed without prompt characters, as shown here:

Hiding the Prompt Character on Leave

You can set the HidePromptOnLeave property to **True** to hide the prompt characters when the control loses input focus.

To hide the prompt character on leave using .html markup:

In the markup of the .aspx page insert:

To write code in Source View

```
<cc1:C1InputMask ID="C1InputMask1" runat="server"
Mask="(999) 000-0000"
PromptChar="#"
HidePromptOnLeave="True">
</cc1:C1InputMask>
```

To hide the prompt character on leave using code:

To hide the prompt character on leave for the C1InputMask control

1. Double-click the Web page to create an event handler for the **Load** event.
2. Enter the following code for the **Page_Load** event:

To write code in Visual Basic

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

To write code in C#

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

This topic illustrates the following:

Run the project. Notice that the prompt characters for the phone number mask are hidden:

When you click inside the input box and it gets focus, the prompt characters (for this example, #) appear:

When you click outside of the input box and it loses focus, the prompt characters are hidden again. For details on changing the prompt characters, see the Changing the Prompt Character topic.

C1InputDateTasks

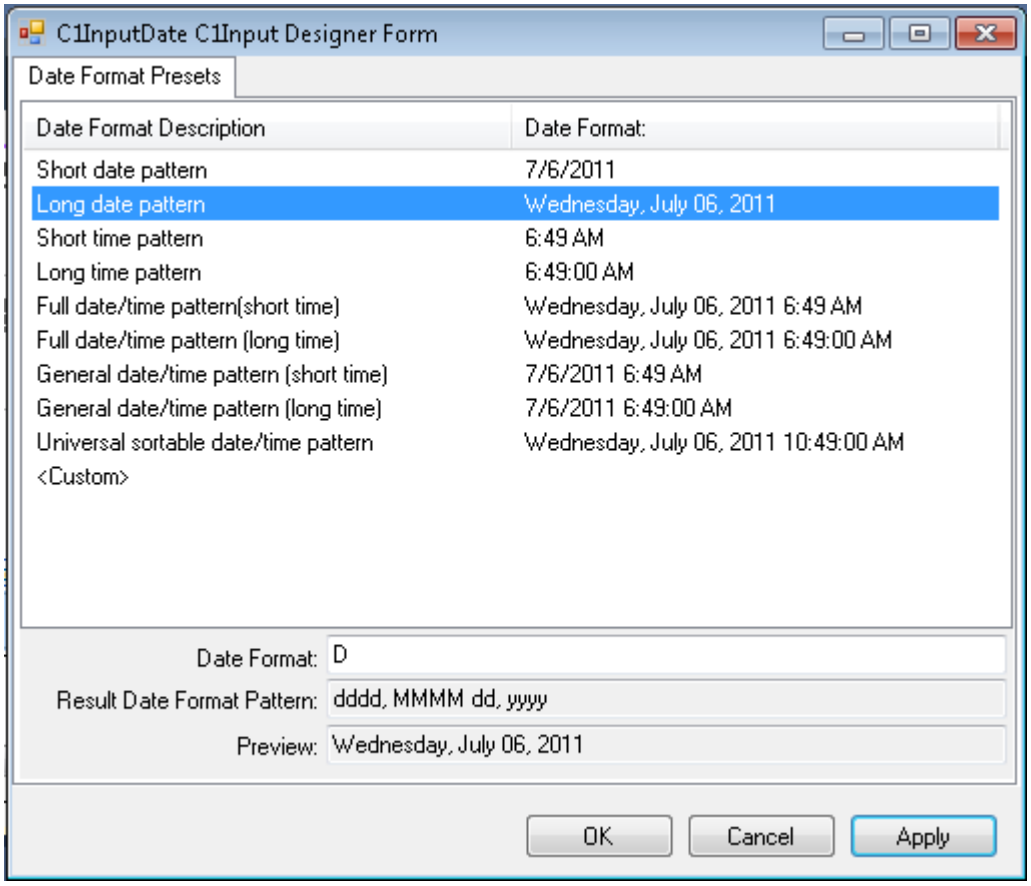
This section shows how to perform specific tasks using the C1InputDate control. The following topics assume that you have added a C1InputDate control to your Web form.

Setting the Date Format Pattern and Date

The following example demonstrates how to set the date format pattern for the C1InputDate control.

To set the date format pattern using the Tasks menu:

1. Open the **C1InputDate Tasks** menu and click **Designer**. The **C1InputDate Designer** appears.
2. Choose a preformatted date pattern. For this example, select **Long date pattern**.



Notice that the designer shows the preview below.

3. Click **OK**.
4. With the Tasks menu still opened, click the **Date** drop-down arrow. The calendar appears.
5. Select the date selected for today's date.

To set the date format pattern using .html markup:

To display the **Long date pattern** format for the **Date Format** value, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputDate ID="C1InputDate1" runat="server"
    Date="2006-12-19"
    DateFormat="D">
</cc1:C1InputDate>
```

To set the date format pattern using code:

To set the date format pattern for the C1InputDate control, double-click the Web page to create an event handler for the **Load** event. Enter the following code for the **Page_Load** event:

To write code in Visual Basic

Visual Basic

```
' Format the control as long date pattern
Me.C1InputDate1.DateFormat = "D"
' Set the date
Me.C1InputDate1.Date = "2006-12-19"
```

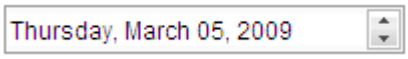
To write code in C#

C#

```
// Format the control as long date pattern
this.C1InputDate1.DateFormat = "D";
// Set the date
this.C1InputDate1.Date = DateTime.Parse("2006-12-19");
```

This topic illustrates the following:

Run the project and notice the date format pattern has been updated.

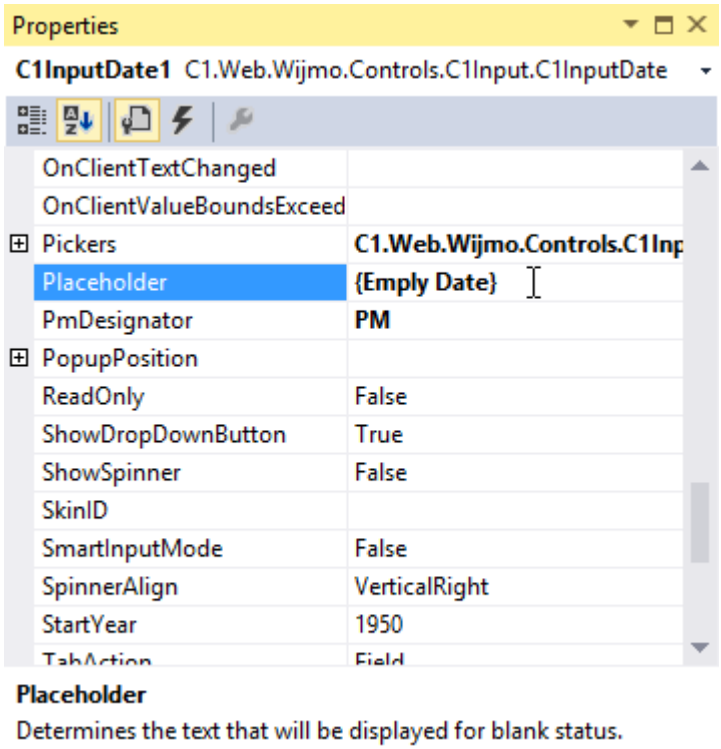


Displaying an Empty Date Value

When you run your project that includes a C1InputDate control, by default the current date automatically appears inside the control, regardless of how you have customized the controls (see Setting the Date Format Pattern and Date). In case, you want the C1InputDate control to display empty date value, instead of displaying the current date value, follow the steps:

Using the Designer

1. From the toolbox, drag and drop the **C1InputDate** control on your web form. You will notice that C1InputDate control will display the current date.
2. Select C1InputDate control from the Design view, and then go to the properties window.
3. From the properties window, set the Placeholder property to **{Empty Date}** or any desired empty data value.



4. Run the project. You will notice that C1InputDate control displays **{Empty Date}**

Note: The empty value which you have assigned to the control will appear when you first run the project. Once you select the control and toggle through the dates, the only way to have the empty value to reappear is to select another control on your page; you cannot "delete" the date within the control, you can only make the empty value text visible.

C1InputNumeric Tasks

This section shows how to perform specific tasks using the C1InputNumeric control.

Note that the C1InputCurrency and C1InputPercent control's properties are the same as the C1InputNumeric control; therefore, the following tasks apply to the C1InputCurrency and C1InputPercent controls as well.

The following topics assume that you have added a C1InputNumeric control to your Web form.

Indicating the Number of Decimal Places

The following example shows how you can easily indicate the number of decimal places to display for the C1InputNumeric control.

To set the decimal places value using the Tasks menu:

1. Open the **C1InputNumeric Tasks** menu.
2. Set the **Value** of the control to **2.345**.
3. Enter **3** for the DecimalPlaces value.

Note that even though you entered 2.345 for the **Value**, if you do not change the DecimalPlaces value to 3, only 2 decimal places (the default) will be displayed. That is, 2.34.

To set the decimal places value using .html markup:

To set the **Value** to **2.345** and the DecimalPlaces value to **3**, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputNumeric ID="C1InputNumeric1" runat="server"
    DecimalPlaces="3"
    Value="2.345">
</cc1:C1InputNumeric>
```

To set the decimal places value using code:

To set the decimal places value for the C1InputNumeric control, double-click the Web page to create an event handler for the **Load** event. Enter the following code for the **Page_Load** event:

To write code in Visual Basic

Visual Basic

```
' Set the numeric value
Me.C1InputNumeric1.Value = 2.345
' Set the number of decimal places
Me.C1InputNumeric1.DecimalPlaces = 3
```

To write code in C#

C#

```
// Set the numeric value
```

```
this.C1InputNumeric1.Value = 2.345;
// Set the number of decimal places
this.C1InputNumeric1.DecimalPlaces = 3;
```

Setting the Min/Max Value

The following example demonstrates the C1InputNumeric control's numeric range support with the ability to easily change MinValue and MaxValue properties.

To set the numeric values using the Tasks menu:

1. Open the **C1InputNumeric Tasks** menu.
2. Enter **1** for the **MinValue**.
3. Enter **1000** for the **MaxValue**.
4. With the Tasks menu still open, enter **1** in the **Value** text box.

To set the numeric values using .html markup:

To set the MinValue to **1**, the MaxValue to **1000**, and the **Value** to **1**, use the following markup in the .aspx page:

To write code in Source View

```
<cc1:C1InputNumeric ID="C1InputNumeric1" runat="server"
    MaxValue="1000"
    MinValue="1"
    Value="1">
</cc1:C1InputNumeric>
```

To set the numeric value using code:

To set the numeric value for the C1InputNumeric control, double-click the Web page to create an event handler for the **Load** event. Enter the following code for the **Page_Load** event:

To write code in Visual Basic

Visual Basic

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

To write code in C#

C#

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

Run the project and observe the following:

- With the input control displaying 1.00, click the Down spin button with your mouse pointer and notice that the control will not display a value lower than 1.00.
- With the input control displaying 1000.00, click the Up spin button with your mouse pointer and notice that the control will not display a value higher than 1000.00.

Client-Side Event Tasks

This section shows how to perform various client-side event tasks using the **Input for ASP.NET Web Forms Edition** controls. These topics require you to add Wijmo CDN references to your application.

Showing a Tooltip when Invalid Input is Entered

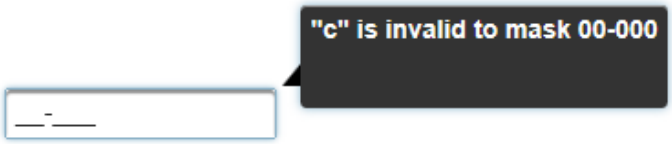
This topic demonstrates how to show a tooltip, we'll use a wijtooltip in this example, when an invalid character is entered into the C1InputMask control.

1. Right-click the C1InputMask control on your form and choose **Properties** to open the Visual Studio Properties window.
2. Next to the MaskFormat property, enter 00-000.
3. Next to the OnClientInvalidInput property, enter **invalidInput**.
4. Select the **Source** tab to open the source view.
5. In the .aspx source, enter the following script markup:

To write code in Source View

```
<script type="text/javascript">
    function invalidInput(e, data) {
        $(data.widget.element).wijtooltip({
            title: '\"' + data.char + '\" is invalid to mask ' + data.widget.options.mask,
            triggers: 'custom',
            showing: function () {
                window.setTimeout(function () {
                    $(data.widget.element).wijtooltip('hide');
                }, 3000);
            }
        });
        $(data.widget.element).wijtooltip('show');
    }
</script>
```

When an invalid character is entered in the C1InputMask control, a tooltip appears, like in the following image:



Client-Side Reference

As part of the amazing ComponentOne Web stack, the Wijmo jQuery UI widgets are optimized for client-side Web development and utilize the power of jQuery for superior performance and ease of use.

The ComponentOne Wijmo website at http://wijmo.com/widgets/ provides everything you need to know about Wijmo widgets, including demos and samples, documentation, theming examples, support and more.

The client-side documentation provides an overview, sample markup, options, events, and methods for each widget. To get started with client-side Web development for **Input for ASP.NET Web Forms**, click one of the external links to view our Wijmo wiki documentation. Note that the **Overview** topic for each of the widgets applies mainly to the widget, not to the server-side ASP.NET Web Forms control.

- InputMask
  - wijinputmask documentation
  - wijinputmask API
- InputNumber
  - wijinputnumber documentation
  - wijinputnumber API
- InputDate
  - wijinputdate documentation
  - wijinputdate API

Using the Wijmo CDN

You can easily load the client-side Wijmo widgets into your web page using a Content Delivery Network (CDN). CDN makes it quick and easy to use external libraries, and deploy them to your users. A CDN is a network of computers around the world that host content. Ideally, if you're in the United States and you access a webpage using a CDN, you'll get your content from a server based in the US. If you're in India or China, and you access the SAME webpage, the content will come from a server a little closer to your location.

When web browsers load content, they commonly will check to see if they already have a copy of the file cached. By using a CDN, you can benefit from this. If a user had previously visited a site using the same CDN, they will already have a cached version of the files on their machine. Your page will load quicker since it doesn't need to re-download your support content.

Wijmo has had CDN support from the very beginning. You can find the CDN page at http://wijmo.com/downloads/cdn/. The markup required for loading Wijmo into your page looks similar to this:

To write code in Source View

```
<!--jQuery References-->
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.7.1.min.js" type="text/javascript"></script>
<script src="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.17/jquery-ui.min.js" type="text/javascript"></script>
<!--Theme-->
<link href="http://cdn.wijmo.com/themes/rocket/jquery-wijmo.css" rel="stylesheet" type="text/css" title="rocket-jqueryui" />
<!--Wijmo Widgets CSS-->
<link href="http://cdn.wijmo.com/jquery.wijmo-complete.all.2.0.0.min.css" rel="stylesheet" type="text/css" />
<!--Wijmo Widgets JavaScript-->
<script src="http://cdn.wijmo.com/jquery.wijmo-open.all.2.0.0.min.js" type="text/javascript"></script>
<script src="http://cdn.wijmo.com/jquery.wijmo-complete.all.2.0.0.min.js" type="text/javascript"></script>
```

In this markup, you'll notice that some of the .js files are labeled as *.min.js. These files have been minified - in other words, all unnecessary characters have been removed - to make the pages load faster. You will also notice that there are no references to individual .js files. The JavaScript for all widgets, CSS, and jQuery references have been combined into one file, respectively, such as wijmo-complete.2.0.0.min.js. If you want to link to individual .js files, see the **Dependencies** topic for each widget.

With the **ASP.NET Web Forms Edition** controls, you can click the **Use CDN** checkbox in the control's **Tasks** menu and specify the **CDN path** if you want to access the client-side widgets.