
ComponentOne

Dialog for ASP.NET Web Forms

Copyright © 1987-2015 GrapeCity, Inc. All rights reserved.

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

The ComponentOne product name is a trademark and ComponentOne is a registered trademark of GrapeCity, Inc. All other trademarks used herein are the properties of their respective owners.

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 \$25 (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

While you are welcome to make backup copies of the software for your own use and protection, you are not permitted to make copies for the use of anyone else. We put a lot of time and effort into creating this product, and we appreciate your support in seeing that it is used by licensed users only.

Table of Contents

Overview	7
Help with ASP.NET Web Forms Edition	7
Key Features.....	7
Quick Start	8
Step 1 of 3: Add C1Dialog to the Page	9
Step 2 of 3: Customize the Dialog Window	9
Step 3 of 3: Run your Application	10
Design-Time Support	11
Smart Tag	11
Context Menu	12
Modal and Modeless Dialog Windows	13
Modal Dialog Windows.....	14
Modeless Dialog Windows.....	15
Elements	15
Content Element	15
Caption Bar Element	16
Appearance.....	18
Themes.....	18
Content Templates	20
C1Dialog CSS Selectors.....	20
Working with the Client-Side	21
Client-Side Events	21
Task-Based Help	22
Creating a C1Dialog Control in Code.....	23
Creating a C1Dialog Control on the Client Side	24
Setting Content in the Content Area	28
Setting Content at Design Time	28
Showing External Content in the Content Area	29
Setting Custom HTML Content	30
Using Partial Rendering.....	31
Customizing the C1Dialog Control	40
Customizing the Foreground and Background Colors	40

Customizing the Font Style	42
Customizing the Caption Bar	42
Customizing the Width and Height of C1Dialog	44
CSS Selectors	45
Applying Themes.....	46
Applying a Built-in Theme Using the Smart Tag	46
Adding a Custom Theme	47
Using jQuery ThemeRoller	47
Using the Modal Dialog Option.....	48
Creating a Modal Dialog Box.....	49
Creating an Alert Dialog Box	50
Creating a Confirm Dialog Box	51
Animating C1Dialog.....	52
Client-Side Reference.....	53
Using the Wijmo CDN	54
C1.Web.Wijmo.Controls.4 Assembly	55
Namespaces	55
C1.Web.Wijmo.Controls Namespace.....	55
Overview	55
Classes.....	56
C1BaseStateManager.....	56
Overview	57
Members.....	57
C1TargetControlBase	58
Overview	58
Members.....	59
Methods.....	64
GetProperty<V> Method	66
GetScriptDescriptors Method	67
GetScriptReferences Method	67
RegisterDesignTimeStyleSheet Method	68
SetProperty<V> Method.....	69
Properties.....	69
CDNDependencyPaths Property	73

CDNPath Property	74
DisplayVisible Property	74
EnableCombinedJavaScripts Property	75
Enabled Property	76
Theme Property	77
ThemeSwatch Property	78
UseCDN Property	79
WijmoControlMode Property	80
WijmoCssAdapter Property	80
C1TargetControlHelper	81
Overview	82
Members	83
C1TargetControlHelper Constructor	83
Methods	84
ResolveEmbeddedResourceUrl Method	84
C1TargetControlHelper<T>	85
Overview	85
Members	86
C1TargetControlHelper<T> Constructor	88
Methods	88
EnsureEnabledState Method	89
GetScriptDescriptors Method	90
GetScriptDescriptors() Method	90
GetScriptDescriptors(String) Method	91
RegisterDesignTimeStyleSheet Method	91
RegisterDesignTimeStyleSheets Method	92
RegisterIncludes Method	93
RegisterOnSubmitStatement Method	93
RegisterRunTimeStylesheets Method	94
RegisterScriptDescriptors Method	94
RegisterScriptDescriptorsWithoutScriptManager Method	95
RegisterScriptDescriptorsWithScriptManager Method	95
RenderJsonDataField Method	95
ShowAbout Method	96

Properties.....	96
Control Property	97
JsonSerializableHelper Property	97
Page Property.....	98
ScriptManager Property.....	98
WijChartNavigator	98
Overview	99
Members.....	100
WijChartNavigator Constructor	100
C1.Web.Wijmo.Controls.C1Dialog Namespace	101
Overview	101
Classes.....	101
C1Dialog	101
Overview	102
Members.....	104
C1Dialog Constructor	112
Methods.....	113
LoadLayout Method.....	115
LoadLayout(String) Method	115
LoadLayout(Stream) Method.....	116
LoadLayout(String,LayoutType) Method	117
LoadLayout(Stream,LayoutType) Method	117
SaveLayout Method	118
SaveLayout(String) Method	119
SaveLayout(Stream) Method	119
Properties.....	120
AppendTo Property.....	127
AutoExpand Property.....	127
Buttons Property.....	128
CaptionButtons Property	129
CloseOnEscape Property.....	130
CloseText Property.....	131
CollapsingAnimation Property	132
Content Property	133

ContentUrl Property	134
Draggable Property	134
Enabled Property	135
ExpandingAnimation Property	136
Height Property.....	137
Hide Property	137
MaintainStatesOnPostback Property.....	138
MaxHeight Property.....	138
MaxWidth Property	139
MinHeight Property	140
MinWidth Property	140
Modal Property	141
OnClientBeforeClose Property.....	142
OnClientClose Property.....	143
OnClientCreate Property.....	144
OnClientDrag Property.....	144
OnClientDragStart Property	145
OnClientDragStop Property	146
OnClientFocus Property	147
OnClientOpen Property	148
OnClientResize Property	148
OnClientResizeStart Property	149
OnClientResizeStop Property.....	150
Position Property	151
Resizable Property	152
Show Property.....	153
ShowOnLoad Property	153
Stack Property.....	154
Title Property	155
Width Property	156
ZIndex Property.....	156
DialogButton	157
Overview	157
Members.....	158

DialogButton Constructor	159
DialogButton Constructor(String,String).....	159
DialogButton Constructor().....	160
Properties.....	160
OnClick Click Property.....	161
Text Property	161
DialogCaptionButton.....	162
Overview	163
Members.....	163
DialogCaptionButton Constructor	164
DialogCaptionButton Constructor()	165
DialogCaptionButton Constructor(String,String)	165
Properties.....	166
IconClassOff Property	166
IconClassOn Property.....	167
OnClick Click Property.....	168
Visible Property.....	168
DialogCaptionButtons	169
Overview	170
Members.....	171
DialogCaptionButtons Constructor	171
Properties.....	172
Close Property.....	173
Maximize Property.....	173
Minimize Property	174
Pin Property	175
Refresh Property	176
Toggle Property.....	177

Overview

Create robust, interactive, customizable dialog windows with **Dialog for ASP.NET Web Forms**. This special type of dialog window can be created on the client-side or server-side to display information and receive input from the user.

Getting Started

Get started with the following topics:

- [Key Features](#)
- [Quick Start](#)
- [Control Elements](#)

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](#).

See Also

[Key Features](#)

[Quick Start](#)

Key Features

The [C1Dialog](#) control is a special type of dialog window that can be created on the client-side or server-side to display information and receive input from the user. You can use [C1Dialog](#) to create modal or modeless dialog windows by calling either the `Open()` methods.

Dialog windows can be used to:

- Prompt users for additional information when using items that require user input.
- Retrieve input for an application.
- Display information or options relative to the application while the user works in another window.

Feature Overview

[C1Dialog](#) provides several unique features including, a rich object model, Modal and Modeless Dialog windows, Partial Page Rendering, style properties, built-in themes, template support,

HTML content, external content, window positioning, minimizing options, movable dialog windows, and resizable dialog windows.

- **Modal and Modeless Dialog Windows**
Dialog for ASP.NET Web Forms provides two different types of dialog windows: Modal and Modeless dialog windows. A modal dialog window is a child window that must be closed before the user can continue working on the current application. A modeless dialog window enables users to interact with other windows while the dialog window is present. For more information on these types of dialog windows see, [Modal and Modeless Dialog Windows](#).
- **Partial-Page Rendering**
Improve application performance and provide more direct feedback when users perform actions with Partial Page Rendering (PPR) technology.
- **Docking**
The dialog box can be docked to zones when the window is minimized.
- **Styles**
Dialog for ASP.NET Web Forms provides unique property styles for its caption bar and content elements.
- **Template Support**
Templates can be added to the content area of the dialog window. Dynamic Templates can be used in the content area of the dialog window for achieving rich presentation of the dialog window. For more information on templates, see [Content Templates](#).
- **Animation**
Customize interaction with the dialog using any of the built-in animation effects. Change how the dialog window is shown and hidden, and how the dialog window expands and collapses. By default, the dialog window does not use any animation effects.
- **Theming**
With just a click of the SmartTag, change the dialog box's look by selecting one of the 5 premium themes (Midnight, Aristo, Rocket, Cobalt, and Sterling). Optionally, use ThemeRoller from jQuery UI to create a customized theme! For more information on this feature, see [Themes](#).
- **CSS Support**
Use a cascading style sheet (CSS) style to define custom skins. CSS support allows you to match the window to your organization's standards.

Quick Start

This quick start describes how to get started with **Dialog for ASP.NET Web Forms**. In the quick start you'll create a project in Visual Studio, add a [modeless C1Dialog](#) window to the page, change the appearance and behavior of the dialog window, and observe the dialog window's run-time behavior.

See Also

[Step 1 of 3: Add C1Dialog to the Page](#)

[Step 2 of 3: Customize the Dialog Window](#)

[Step 3 of 3: Run your Application](#)

Step 1 of 3: Add C1Dialog to the Page

In this step you'll create a new project and add a **Dialog for ASP.NET Web Forms** dialog window to your project. To begin the Quick Start, complete the following steps:

1. From the Visual Studio **File** menu select **New | Project**. The **New Project** dialog box will appear.
2. In the **New Project** dialog box expand a language in the left-hand pane and select **Web**. In the right pane, choose **ASP.NET Empty Web Application**, enter a **Name** for your application, and select **OK**. A new application will be created.
3. In the Solution Explorer, right-click the project and choose **Add Reference**.
4. In the Add Reference dialog box, locate and select the **C1.Web.Wijmo.Controls** and **C1.Web.Wijmo.Controls.Design** assemblies and click **OK**. The references will be added.
5. Right-click the project in the Solution Explorer and from the context menu choose **Add | New Item**.
6. In the **Add New Item** dialog box choose **Web Form** from the list of templates, name the item "Default.aspx", and click **Add**. The new page should open.
7. While in Design view navigate to the Visual Studio Toolbox and double-click the **Input (Button)** and **C1Dialog** icons to add the **Button** and **C1Dialog** controls to your page.
8. Select **Button1**, navigate to the Properties window, and set the **Value** property of the control to "Click Me!".
9. In Source View add the **onclick** event handler to **Button1** so that control's markup looks like the following:

[To write code in Source View](#)

```
<input id="Button1" type="button" value="Click Me!"  
  
onclick="$ ('#<%=c1dialog1.ClientID%>').c1dialog('open')"/>  
Now clicking on the Button will open the C1Dialog dialog window.
```

You've completed creating a new project and adding a **C1Dialog** dialog window to your project. In the next step you'll customize the dialog window's appearance and behavior.

Step 2 of 3: Customize the Dialog Window

In this step you'll change the appearance and behavior of the dialog window you created in the last step. Complete the following steps to customize your **Dialog for ASP.NET Web Forms** dialog window:

1. Select the **C1Dialog** smart tag to open the **C1Dialog Tasks** menu and set the following:
 - Uncheck the **ShowOnLoad** check box so that the control is not initially visible when the application is run.
 - Select **rocket** in the **Theme** drop-down box to change the appearance of the dialog window.

- Click once on the **C1Dialog** control to select it and set the following properties in the Properties window:
 - Set the **Title** property to "C1Dialog says..." to set the caption bar title.
 - Set the **Height** to **200** to increase the size of the dialog window.
- Click once in the [Content Area](#) of the dialog window and type "Hello World!" to add text content to your dialog window. Note that the markup now appears similar to the following:

To write code in Source View

```
<cc1:C1Dialog ID="C1Dialog1" runat="server" Height="200px"
Title="C1Dialog says...">
  <CaptionButtons>
    <Pin IconClassOn="ui-icon-pin-w" IconClassOff="ui-icon-
pin-s"></Pin>
    <Refresh IconClassOn="ui-icon-refresh"></Refresh>
    <Minimize IconClassOn="ui-icon-minus"></Minimize>
    <Maximize IconClassOn="ui-icon-extlink"></Maximize>
    <Close IconClassOn="ui-icon-close"></Close>
  </CaptionButtons>
  <Content>
    Hello World!
  </Content>
</cc1:C1Dialog>
```

You've completed step two of the **Dialog for ASP.NET Web Forms** quick start guide. In the next step you'll run the project.

Step 3 of 3: Run your Application

Now that you've customized your dialog window's appearance and behavior, the only thing left to do is to run your application and observe some of the run-time functionality of the [C1Dialog](#) dialog window.

Complete the following steps:

- Click the **Click Me** button.
The dialog window will appear in the Web page.



2. Move the dialog window on the Web page by clicking on the dialog window's header and performing a drag-and-drop operation.
3. Resize the dialog window by clicking on the dialog window's bottom right corner and performing a drag-and-drop operation.
4. Close the dialog window by clicking the **Close** button in the upper right-hand corner of the dialog box.

Congratulations, you've created and customized a [C1Dialog](#) dialog window control and completed the **Dialog for ASP.NET Web Forms** Quick Start guide!

Design-Time Support

Dialog for ASP.NET Web Forms includes customized context menus, smart tags, and design-time support that simplify working with the object model.

The following sections describe how to use the **Dialog for ASP.NET Web Forms** design-time environment, particularly the **C1Dialog Tasks** menu accessible from the [C1Dialog](#) smart tag and [context menu](#), to configure the [C1Dialog](#) control.

See Also

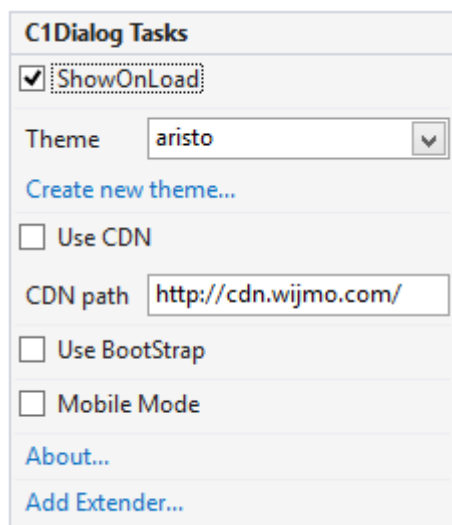
[Smart Tag](#)

[Context Menu](#)

Smart Tag

The [C1Dialog](#) control includes a smart tag. A smart tag represents a short-cut tasks menu that provides the most commonly used properties in each component/command.

To access the **C1Dialog Tasks** menu, click on the Smart Tag arrow in the upper right corner of the [C1Dialog](#) control. This will open the **C1Dialog Tasks** menu.

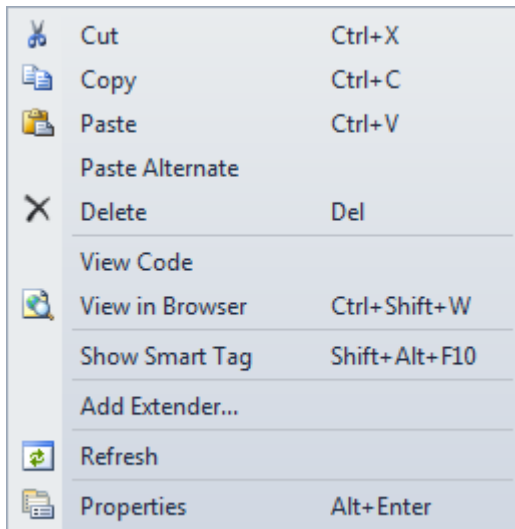


The **C1Dialog Tasks** menu operates as follows:

- **ShowOnLoad**
When the **ShowOnLoad** check box is checked, the [C1Dialog.ShowOnLoad](#) property is set to **True** and the dialog window will appear on the page on load. By default the **ShowOnLoad** check box is checked and the [C1Dialog.ShowOnLoad](#) property is set to **True**.
- **Theme**
Clicking the **Theme** drop-down box allows you to select from various visual schemes. For more information about available visual styles, see [Themes](#).
- **Create new theme...**
The **Create new theme...** option opens the **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**
Selecting the **Use CDN** check box will indicate that the widget extender must load client resources from a content delivery network. By default this box is not checked.
- **CDN Path**
Indicates the path for the content delivery network. Enter a URL here to change the path.
- **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](#).
- **About**
Clicking on the **About** item displays a dialog box, which is helpful in finding the version number of **ASP.NET Web Forms Edition** and online resources.
- **Add Extender**
Clicking the **Add Extender** item opens the **Extender Wizard**, allowing you to add an extender to the **C1Expander** control.

Context Menu

The [C1Dialog](#) control has additional commands with the context menu that Visual Studio provides for all .NET controls. To access the [C1Dialog](#) context menu, right-click anywhere on the [C1Dialog](#) control. This will open the [C1Dialog](#) context menu.



The [C1Dialog](#) context menu includes the following custom commands added by [C1Dialog](#):

- **Show Smart Tag**
Shows the smart tag for the [C1Dialog](#) control. For more information on how to use the smart tag and available features, see [C1Dialog Smart Tag](#).
- **Add Extender**
Clicking the **Add Extender** item opens the **Extender Wizard**, allowing you to add an extender to the **C1Expander** control.

Modal and Modeless Dialog Windows

Dialog boxes are commonly used in applications to retrieve input from the user. In some applications a dialog box is used to prompt the user for input and once the application retrieves the input the dialog box is automatically closed or destroyed.

On the other hand, some applications use dialog boxes to display information while the user works in other windows. For example, when you check spelling in Microsoft Word a dialog box remains open so you can go through and edit your text in the document while the spell checker looks for the next misspelled word. To support the different ways applications use dialog boxes, **C1Dialog** supports two different types of dialog windows: modal and [modeless](#) dialog windows.

See Also

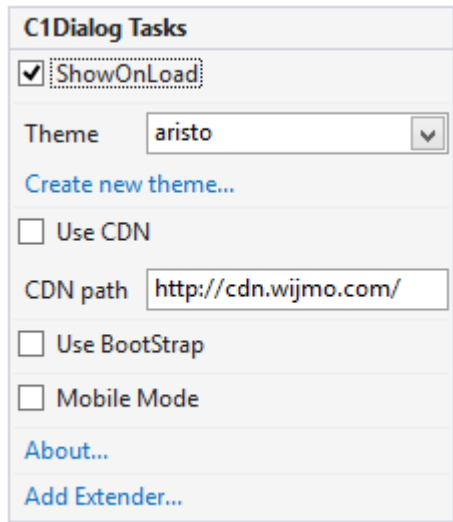
[Modal Dialog Windows](#)

[Modeless Dialog Windows](#)

Modal Dialog Windows

The [C1Dialog](#) control includes a smart tag. A smart tag represents a short-cut tasks menu that provides the most commonly used properties in each component/command.

To access the **C1Dialog Tasks** menu, click on the Smart Tag arrow in the upper right corner of the [C1Dialog](#) control. This will open the **C1Dialog Tasks** menu.



The **C1Dialog Tasks** menu operates as follows:

- ShowOnLoad**
 When the **ShowOnLoad** check box is checked, the [C1Dialog.ShowOnLoad](#) property is set to **True** and the dialog window will appear on the page on load. By default the **ShowOnLoad** check box is checked and the [C1Dialog.ShowOnLoad](#) property is set to **True**.
- Theme**
 Clicking the **Theme** drop-down box allows you to select from various visual schemes. For more information about available visual styles, see [Themes](#).
- Create new theme...**
 The **Create new theme...** option opens the **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**
 Selecting the **Use CDN** check box will indicate that the widget extender must load client resources from a content delivery network. By default this box is not checked.
- CDN Path**
 Indicates the path for the content delivery network. Enter a URL here to change the path.
- 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](#).

- **About**
Clicking on the **About** item displays the **About** dialog box, which is helpful in finding the version number of **Dialog for ASP.NET Web Forms** and online resources.
- **Add Extender**
Clicking the **Add Extender** item opens the **Extender Wizard**, allowing you to add an extender to the **C1Expander** control.

Modeless Dialog Windows

A modeless dialog window enables users to interact with other windows while the dialog window is present. Use this type of dialog window when the requested information is not necessary to continue. Modeless dialog windows do not keep the input focus so you can work on two applications at once. Show a modeless dialog window by setting the [C1Dialog.Modal](#) property to **False** (default) and in JavaScript using the **Open()** method.

A modeless dialog window is commonly used in menus and help systems where the user can use the dialog window and the application window concurrently. For example, a toolbar is a modeless dialog window because it can be detached from the application and the user can select items in the toolbar to apply features to the detached or separated application. You can use the [AppendTo](#) property to specify to which element on the page your **C1Dialog** control is attached.

Elements

This section provides a visual and descriptive overview of the elements that comprise the [C1Dialog](#) control. The topics are categorized into the three distinct elements, the content element, the caption bar element, and the status bar element that represent different aspects of the dialog window control.

See Also

[Content Element](#)

[Caption Bar Element](#)

Content Element

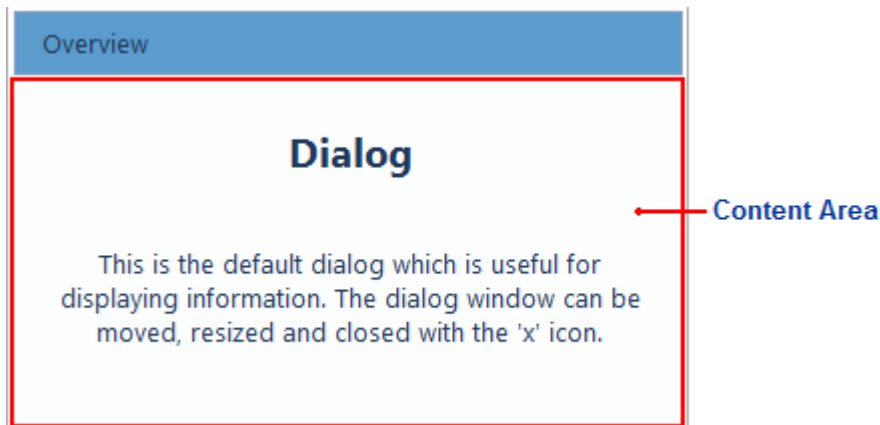
The main part of the [C1Dialog](#) control is the content area. In the content area you can add rich text through custom HTML content, URL links through its [C1Dialog.ContentUrl](#) property, and add arbitrary controls through its content template. Elements in the content area of the control can be added and moved on the control through a simple drag-and-drop operation.

[C1Dialog](#) includes the following properties to make it simple to add and customize any type of item such as text, images, arbitrary controls, and links to the content area:

- [C1Dialog.Content](#)

- [C1Dialog.ContentUrl](#)

The following image labels the content area in the [C1Dialog](#) control:



You can use a style sheet to apply unique styles to the content in the [C1Dialog](#) control.

You can enter text in the content area of the dialog window at design time. When you enter text into the content area, [C1Dialog](#) adds a `<ContentTemplate>` tag inside the `<cc1:C1Dialog>` tag like the following:

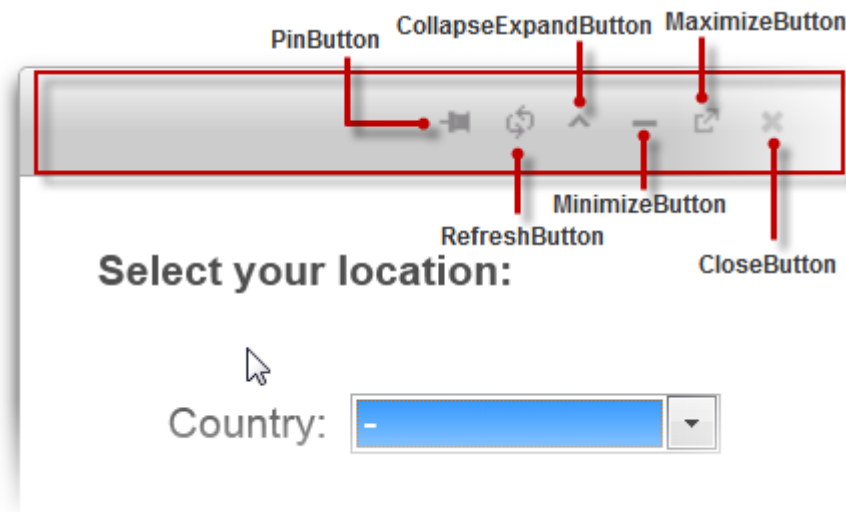
To write code in Source View

```
<cc1:C1Dialog ID="C1Dialog1" runat="server" Height="200px"
Title="C1Dialog says...">
  <CaptionButtons>
    <Pin IconClassOn="ui-icon-pin-w" IconClassOff="ui-icon-pin-
s"></Pin>
    <Refresh IconClassOn="ui-icon-refresh"></Refresh>
    <Minimize IconClassOn="ui-icon-minus"></Minimize>
    <Maximize IconClassOn="ui-icon-extlink"></Maximize>
    <Close IconClassOn="ui-icon-close"></Close>
  </CaptionButtons>
  <Content>
    This is where the content information is placed.
  </Content>
  ...
</cc1:C1Dialog>
```

Caption Bar Element

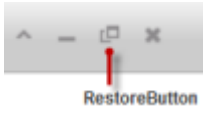
The caption bar appears at the top of the [C1Dialog](#) dialog window, and appears similar to a traditional dialog box with a few differences. From left to right the caption bar includes an icon, title text, and several buttons. It has optional pin, reload, collapse/expand, minimize, maximize, and close buttons that appear on the right side of the caption bar when you enable the [DialogCaptionButtons.Pin](#), [DialogCaptionButtons.Refresh](#), [DialogCaptionButtons.Minimize](#), [DialogCaptionButtons.Maximize](#), and [DialogCaptionButtons.Close](#) buttons. You can specify text for the title on the caption bar by setting the string for the [C1Dialog.Title](#) property

The following diagram labels the properties that appear on the Caption Bar:



The following table describes each of the elements on the caption bar:

Element	Description
Title	Gets or sets text which displayed on the caption bar of the C1Dialog dialog window. By default Title is blank.
Pin	Button that pins and unpins the C1Dialog dialog window. By default Pin is not Visible.
Refresh	Button that reloads the C1Dialog dialog window content set by the ContentUrl property. By default is not Visible.
Minimize	Button that minimizes the C1Dialog dialog window. By default Minimize is Visible.
Maximize	Button that resizes the C1Dialog dialog window to fit all available space in the browser window. By default Maximize is Visible.
Close	Button that closes the C1Dialog dialog window. By default Close is Visible.

 <p>Restore</p>	<p>Button that is shown instead of the Minimize button or the Maximize button of C1Dialog dialog window when the window is minimized or maximized. Clicking this button restores the window to the previous size.</p>
--	---

Appearance

[C1Dialog](#) is designed to make customization easy for you. The possibilities are endless when it comes to changing your [C1Dialog](#) dialog window's appearance. [C1Dialog](#) provides several styles for its caption bar, content area, and status bar elements and built-in themes, including those for Office 2007.

In addition to the property styles for each dialog window element, [C1Dialog](#) provides full support for CSS styles so you can further customize each dialog window element through CSS styles.

See Also

[Themes](#)

[Content Templates](#)




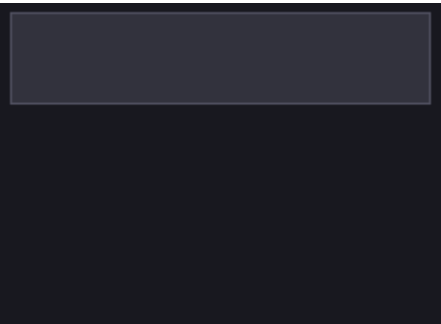
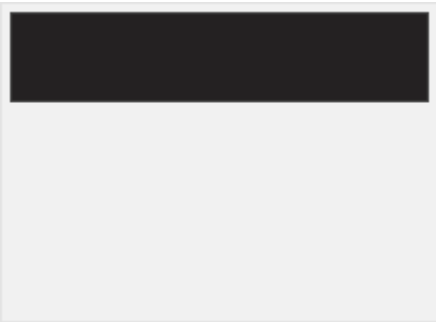
[C1Dialog CSS Selectors](#)

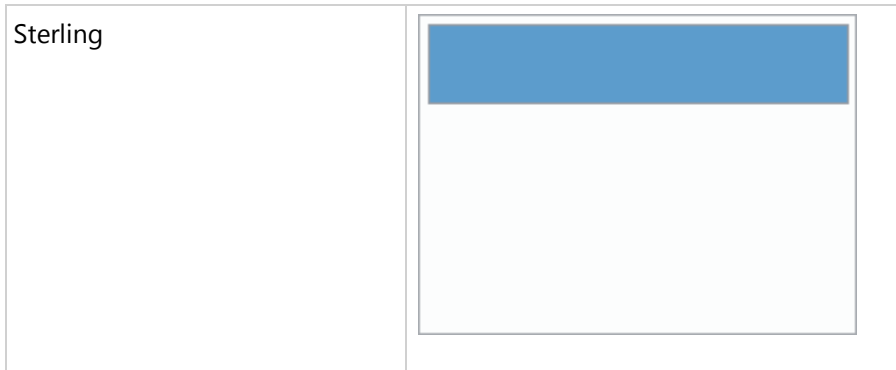
Themes

[C1Dialog](#) includes themes allowing you to easily change the control's appearance. The control includes several built-in themes allowing you to customize the control's appearance to your application. You can easily change themes from the **C1Dialog Tasks** menu, from the Properties window, and in code.

The following themes are included in **Dialog for ASP.NET Web Forms**:

Visual Styles	Preview
---------------	---------

Arctic	
Aristo	
Cobalt	
Midnight	
Rocket	



Content Templates

The contents of the content area of the dialog window can be controlled by using templates. [C1Dialog](#) has a special property, [C1Dialog.Content](#), that is used to apply templates to the content area and status bar area of the [C1Dialog](#) control.

The dialog window content templates are useful for customizing your dialog window so its appearance integrates into your application and for adding content to the content area of the dialog window.

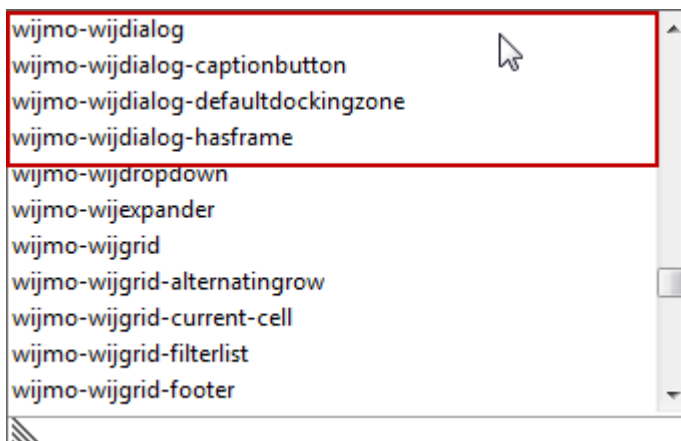
C1Dialog CSS Selectors

You can style many [C1Dialog](#) elements using CSS to make their appearance unique. To make this customization easier, ComponentOne includes CSS selectors with each of its six built-in themes.

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

For a list of common individual CSS selectors and grouped CSS selectors, select the [C1Dialog](#) control in your project and view the drop-down list next to the **CssClass** property in the Visual Studio Properties window.

[C1Dialog](#) **CSS** selectors begin with `wijmo-wijdialog`:



You can combine the individual CSS selectors as a group to make the CSS selector more specific and strong.

Working with the Client-Side

Dialog 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 [C1Dialog](#) control is rendered, an instance of the client-side control will be created automatically. This means that you can enjoy the convenience of accessing the properties and methods of the [C1Dialog](#) 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. Using the client-side object model increases the efficiency of your Web site.

See Also

[Client-Side Events](#)

Client-Side Events

Dialog for ASP.NET Web Forms includes several client-side events that allow you to manipulate the **C1Dialog** controls when an action such as resizing the dialog window occurs.

You can use the server-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 "Resize" to respond when the Dialog window is resized, you would set the [OnClientResize](#) property to "resize".

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 of the name you declare for each JavaScript function are defined on the client side.

Event Server-Side Property Name	Event Name	Description
OnClientBeforeClose	beforeClose	Occurs before the widget closes.
OnClientClose	close	Occurs when the widget is closed.
OnClientDrag	drag	Occurs when the widget is

		dragged.
OnClientDragStart	dragStart	Occurs when dragging the widget starts.
OnClientDragStop	dragStop	Occurs when dragging the widget stops.
OnClientFocus	focus	Occurs when the widget gains focus.
OnClientOpen	open	Occurs when the widget is opened.
OnClientResize	resize	Occurs when the widget is resized.
OnClientResizeStart	resizeStart	Occurs at the start of resizing.
OnClientResizeStop	resizeStop	Occurs when resizing stops.

Task-Based Help

The task-based help section assumes that you are familiar with programming in the Visual Studio ASP.NET environment and have a general understanding of the **Rating for ASP.NET Web Forms** control.

Each topic provides a solution for specific tasks using [C1Dialog](#). By following the steps outlined in each topic, you will be able to create projects using a variety of [C1Dialog](#) features.

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

See Also

[Creating a C1Dialog Control in Code](#)

[Creating a C1Dialog Control on the Client Side](#)

[Setting Content in the Content Area](#)[Customizing the C1Dialog Control](#)[CSS Selectors](#)[Applying Themes](#)[Using the Modal Dialog Option](#)[Animating C1Dialog](#)

Creating a C1Dialog Control in Code

Creating a [C1Dialog](#) control in code is an easy process. In this topic, you will add a **PlaceHolder** control to the page, add an import statement, customize the [C1Dialog](#) control, and add the control to the **PlaceHolder**.

Complete the following steps:

1. In Design View, navigate to the Visual Studio Toolbox and add a **PlaceHolder** control to the page.
2. Double-click the page to add a **Page_Load** event and switch to Code view.
3. Add the following statement to the top of the Code Editor to import the appropriate namespace.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Imports C1.Web.Wijmo.Controls.C1Dialog</pre>	

[To write code in C#](#)

C#	Copy Code
<pre>using C1.Web.Wijmo.Controls.C1Dialog;</pre>	

4. Add the following code to the **Page_Load** event to create the control, set **C1Dialog**'s height and width, and add the control to the **PlaceHolder**.

[To write code in Visual Basic](#)

Visual Basic	Copy Code

```
'Create a new C1Dialog
Dim C1D As New C1Dialog()

'Set the control's size
C1D.Height = 200
C1D.Width = 200

'Add the control to the Placeholder
Placeholder1.Controls.Add(C1D)
```

To write code in C#

C#	Copy Code
<pre>//Create a new C1Dialog C1Dialog C1D = new C1Dialog(); //Set the control's size C1D.Height = 200; C1D.Width = 200; //Add the C1Dialog to the Placeholder control Placeholder1.Controls.Add(C1D);</pre>	

Creating a C1Dialog Control on the Client Side

The [C1Dialog](#) control can be created and displayed on the client side. In this help, you will place a standard Button control and a standard Placeholder control on your page, use code to create the [C1Dialog](#) control, and display it using the button_click event.

Complete these steps:

1. Add a standard **Placeholder** control and a standard **Button** control to an ASP.NET project.
2. Switch to Source View and add the following script in the first `<asp:Content>` tag:

To write code in Source View

```
<!--jQuery References-->

<script src="http://code.jquery.com/jquery-1.8.2.min.js"
    type="text/javascript"></script>

<script src="http://code.jquery.com/ui/1.9.1/jquery-ui.min.js"
    type="text/javascript"></script>

<asp:Placeholder runat="server">

<script type="text/javascript">
    $(document).ready(function () {

        $("#<%=btnShowDialog.ClientID %>").click(function ()

            $("#<%=C1Dialog1.ClientID
            %>").cldialog("open");

            return false;

        });
    });
</script>

</asp:Placeholder>
```

3. Still in the Source View, locate the markup for the **Button** control and the **Placeholder** control and edit it to resemble the following markup:

To write code in Source View

```
<asp:Button ID="btnShowDialog" runat="server" Text="Show Dialog
Created At Run Time" />

<asp:Placeholder ID="C1Dialog1"
runat="server"></asp:Placeholder>
```

4. Right-click the Source View page and select **View Code** from the menu.
5. In Code View, add the following statement at the top of your page:

To write code in Visual Basic

Visual Basic	Copy Code
Imports C1.Web.Wijmo.Controls.C1Dialog	

To write code in C#

C#	Copy Code
using C1.Web.Wijmo.Controls.C1Dialog;	

6. Add the following code to create the button_click event and to create the [C1Dialog](#) control:

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Protected Sub btnShow_Click(sender As Object, e As EventArgs) Handles btnShow.Click Dim dlg As C1Dialog = New C1Dialog() dlg.Title = "RunTime Dialog" dlg.Modal = True dlg.Content = New MyTemplateClass() C1Dialog1.Controls.Add(dlg) End Sub</pre>	

To write code in C#

C#	Copy Code
<pre>protected void btnShow_Click(object sender, EventArgs e) { C1Dialog dlg = new C1Dialog(); dlg.Title = "RunTime Dialog"; dlg.Modal = true; dlg.Content = new MyTemplateClass(); C1Dialog1.Controls.Add(dlg); }</pre>	

7. Begin a new class with the following code. This will be the template for your [C1Dialog](#) control:

[To write code in Visual Basic](#)

Visual Basic	Copy Code
<pre>Public Class MyTemplateClass Implements ITemplate Public Sub InstantiateIn(container As Control) Implements ITemplate.InstantiateIn Dim label As New Label() label.ID = "lblmyLabel" label.Text = "You are seeing a C1Dialog..!!" Dim btnOK As New Button() btnOK.Text = "OK" container.Controls.Add(label) container.Controls.Add(btnOK) End Sub End Class</pre>	

[To write code in C#](#)

C#	Copy Code
<pre>public class MyTemplateClass : ITemplate { public void InstantiateIn(Control container) { Label label = new Label(); label.ID = "lblmyLabel"; label.Text = "You are seeing a C1Dialog..!!"; Button btnOK = new Button(); btnOK.Text = "OK"; container.Controls.Add(label); container.Controls.Add(btnOK); } }</pre>	

```
}
```

8. When you run your application it will resemble the following:

Show Dialog Created At Run Time

On clicking the button, the [C1Dialog](#) should appear as in the following image:



Setting Content in the Content Area

The following procedures show how to add different types of content to [C1Dialog](#) such as: content in the content template, external website content, custom HTML content, and partial rendering content.

See Also

[Setting Content at Design Time](#)

[Setting Custom HTML Content](#)

[Using Partial Rendering](#)

Setting Content at Design Time

Showing External Content in the Content Area You can easily set the content in the Content Area at design time either in the Design View or in the Source View.

Complete these steps to display content in the Content Area:

In Design View

Complete the following steps:

1. Select the Dialog control in your project.
2. Place your cursor in the Dialog Content Area.
3. Type or paste the content you wish to have displayed in the Content Area.
4. Press F5 to run the project.

In Source View

Complete the following steps:

1. Go to the Source View for your project.
2. After the closing `<CaptionButtons/>` tag, insert the following tag set:

```
<Content> </Content>
```

3. Type or insert your text between the `<Content>` tags.
4. Press F5 to run your project to see your text in the Dialog window.

Showing External Content in the Content Area

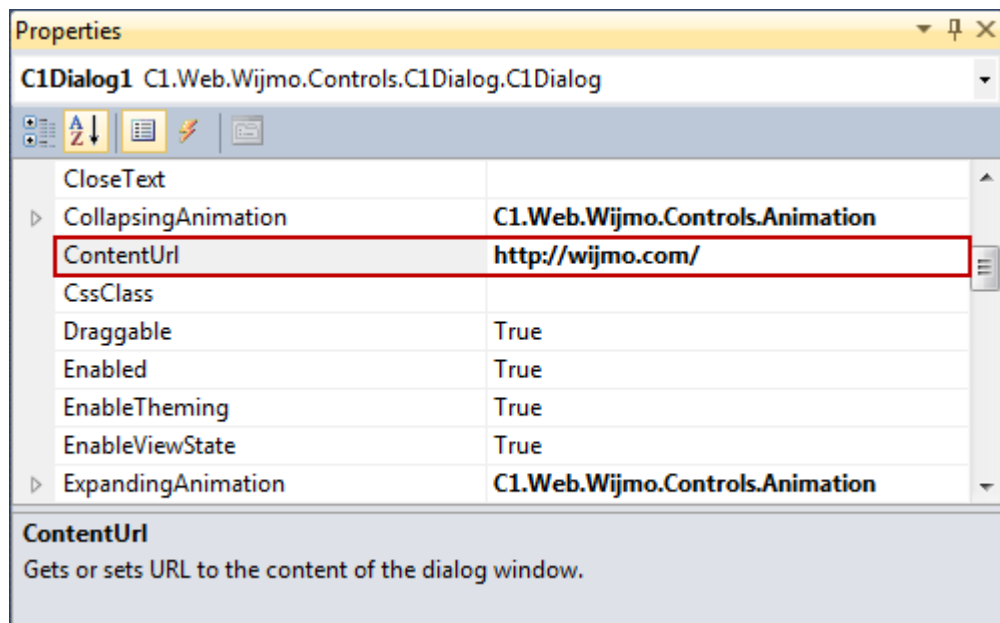
In [C1Dialog](#), you can show external URL content in the Dialog Content Area. You can have the content from another Web page within your project or the content from a URL outside your project display in the dialog content area.

Complete these steps to display URL content in the Content Area:

In Design View

Complete the following steps:

1. Select **View | Properties** from the Visual Studio menu. And choose [C1Dialog](#) from the drop-down list at the top of the Properties window.
2. Scroll down to the **ContentURL** property.
3. Set the **ContentURL** property to the URL you wish to have display in your project.



4. Press F5 to run the program. Your chosen Web page should display in the dialog window.

In Source View

Complete these steps to set a ContentURL in Source View:

1. Add the following markup to the opening `<cc1:C1Dialog>` tag:

[To write code in Source View](#)

```
ContentUrl="http://wijmo.com/wiki/index.php/Main_Page"
```

The markup in the `<cc1:C1Dialog>` tags should resemble the following:

[To write code in Source View](#)

```
<cc1:C1Dialog ID="C1Dialog1" runat="server"
Width="760px" Height="460px" Stack="True" CloseText="True"
MaintainVisibilityOnPostBack="False" CloseOnEscape="False"
ContentURL="http://wijmo.com">
```

2. Press F5 or start debugging to run your program. Your Web page will display in the Dialog window.

In Code

Add the following code to the **Page_Load** event to set the **ContentUrl** property to display external content in the **C1Dialog** Control.

[To write code in Visual Basic](#)

```
Me.C1Dialog1.ContentUrl = "http://www.wikipedia.com/"
```

[To write code in C#](#)

```
this.C1Dialog1.ContentUrl = "http://wikipedia.com";
```

Setting Custom HTML Content

You can set custom HTML content to display in the Content Area. This is done in the Source View of your project. For this Task-Based Help, we will insert HTML Markup that will allow users to enter text in a text box and have it displayed in a formatted, pop-up dialog box.

Complete the following steps:

1. Within the `<cc1:C1Dialog>` tags, insert the following tag set.

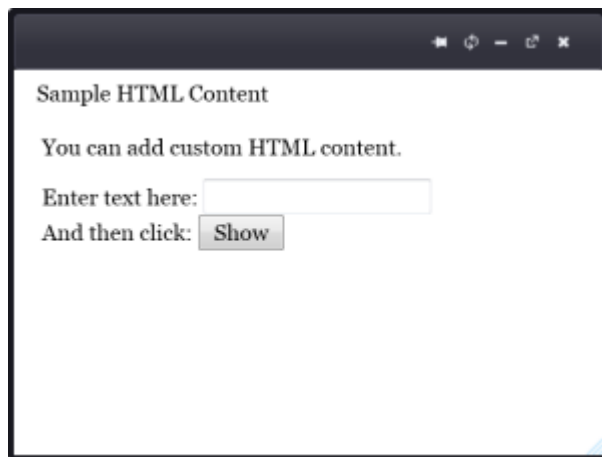
```
<Context></Context>
```


2. Insert the following markup between the <Content> tags.

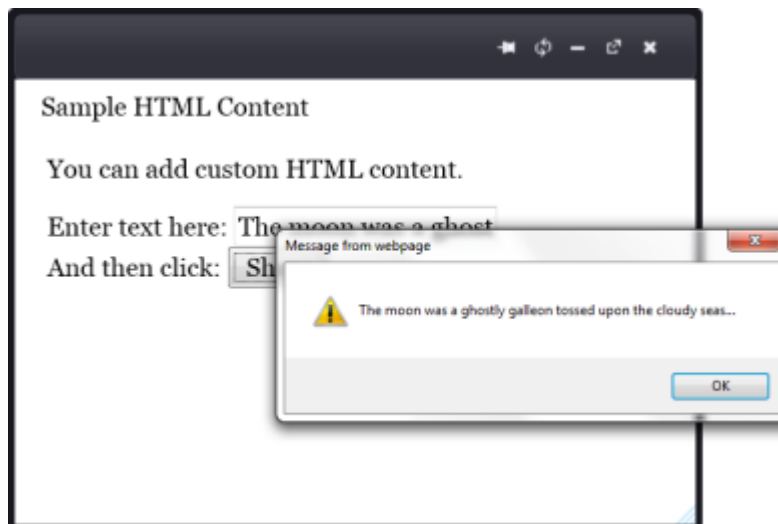
To write code in Source View

```
Sample HTML Content</h2><p>&nbsp;You can add custom HTML  
content.  
<br /></p>&nbsp;Enter text here: <input type=text id="input1" />  
<br>&nbsp;And then click: <input type="button" value="Show"  
onclick="alert(document.getElementById('input1').value)"
```

3. Press F5 to run your project. Your [C1Dialog](#) should look like the following image:

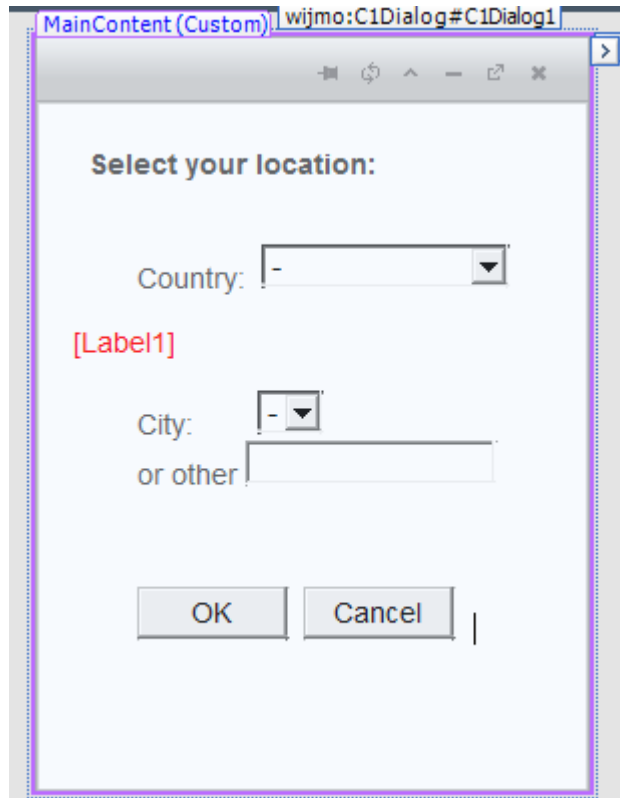


When the user enters text and clicks the "Show" button, the following dialog box appears:



Using Partial Rendering

[C1Dialog](#) supports Partial Page Rendering (PPR) technology, which enables partial page updates without requiring custom JavaScript. This topic will take you through creating a project that uses PPR.



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

To write code in Visual Basic

Visual Basic	Copy Code
<pre>[String].Format("javascript:openWindow({0});", C1Window1.ClientID) If Me.IsPostBack Then Dim dl As DropDownList = DirectCast(C1Window1.FindControl("DropDownList1"), DropDownList) Dim dlc As DropDownList = DirectCast(C1Window1.FindControl("DropDownList2"), DropDownList) Dim tb As TextBox = DirectCast(C1Window1.FindControl("TextBox1"), TextBox) If dl.Text <> "-" AndAlso (dlc.Text <> "-" OrElse tb.Text <> "") Then Dim text As String = "You selected "</pre>	

```
        If dlc.Text <> "-" Then
            text += dlc.Text
        Else

            text += tb.Text

        End If

        text += ", " + dl.Text
        Label2.Text = text

    End If

    Else

        Label2.Text = ""

    End If
```

To write code in C#

C#	Copy Code
<pre>protected void Page_Load(object sender, EventArgs e) String.Format("javascript:openWindow({0});", C1Dialog1.ClientID); if (this.IsPostBack) { DropDownList dl = (DropDownList)C1Dialog1.FindControl("DropDownList1"); DropDownList dlc = (DropDownList)C1Dialog1.FindControl("DropDownList2"); TextBox tb = (TextBox)C1Dialog1.FindControl("TextBox1"); if (dl.Text != "-" && (dlc.Text != "-" tb.Text != "")) { string text = "You selected "; if (dlc.Text != "-")</pre>	

```

        {
            text += dlc.Text;
        }

        else
        {
            text += tb.Text;
        }

        text += ", " + dl.Text;
        Label1.Text = text;
    }
}

else
{
    Label1.Text = "";
}
}

```

6. Add the following **SelectIndexChanged** event to your code:

To write code in Visual Basic

Visual Basic	Copy Code
<pre> Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object, ByVal e As EventArgs) Dim l As Label = DirectCast(C1Cialog1.FindControl("Label1"), Label) Dim dl As DropDownList = DirectCast(C1Dialog1.FindControl("DropDownList1"), DropDownList) Dim dlc As DropDownList = DirectCast(C1Dialog1.FindControl("DropDownList2"), DropDownList) dlc.Items.Clear() dlc.Items.Add(New ListItem("-")) If dl.Text <> "-" Then </pre>	

```
l.Text = "Select a city in " + dl.Text
```

```
Select Case dl.Text
```

```
Case "UK"
```

```
    dlc.Items.Add(New ListItem("London"))
    dlc.Items.Add(New ListItem("Birmingham"))
    dlc.Items.Add(New ListItem("Leeds"))
    dlc.Items.Add(New ListItem("Glasgow"))
    dlc.Items.Add(New ListItem("Glasgow"))
    dlc.Items.Add(New ListItem("Sheffield"))
    dlc.Items.Add(New ListItem("Bradford"))
    dlc.Items.Add(New ListItem("Edinburgh"))
    dlc.Items.Add(New ListItem("Liverpool"))
```

```
Exit Select
```

```
Case "USA"
```

```
    dlc.Items.Add(New ListItem("New York, New York"))
    dlc.Items.Add(New ListItem("Los Angeles, California"))
    dlc.Items.Add(New ListItem("Chicago, Illinois"))
    dlc.Items.Add(New ListItem("Houston, Texas"))
    dlc.Items.Add(New ListItem("Philadelphia, Pennsylvania"))
    dlc.Items.Add(New ListItem("Phoenix, Arizona"))
    dlc.Items.Add(New ListItem("San Diego, California"))
    dlc.Items.Add(New ListItem("Dallas, Texas"))
    dlc.Items.Add(New ListItem("Detroit, Michigan"))
```

```
Exit Select
```

```
Case "Russia"
```

```
    dlc.Items.Add(New ListItem("Moscow"))
    dlc.Items.Add(New ListItem("Chelyabinsk"))
    dlc.Items.Add(New ListItem("Ekaterinburg"))
    dlc.Items.Add(New ListItem("Irkutsk"))
    dlc.Items.Add(New ListItem("St. Petersburg"))
    dlc.Items.Add(New ListItem("Volgograd"))
    dlc.Items.Add(New ListItem("Petrozavodsk"))
    dlc.Items.Add(New ListItem("Nizhni Novgorod"))
```

```

        dlc.Items.Add(New ListItem("Novosibirsk"))

        Exit Select

    Case "Canada"

        dlc.Items.Add(New ListItem("Toronto, Ontario"))
        dlc.Items.Add(New ListItem("Montreal, Quebec"))
        dlc.Items.Add(New ListItem("Vancouver, British Columbia"))
        dlc.Items.Add(New ListItem("Calgary, Alberta"))
        dlc.Items.Add(New ListItem("Edmonton, Alberta"))
        dlc.Items.Add(New ListItem("Ottawa - Gatineau,
Ontario/Quebec"))

        Exit Select

    Case Else

        Exit Select

    End Select

End If

End Sub

```

[To write code in C#](#)

C#	Copy Code
<pre> protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e) { Label l = (Label)C1Dialog1.FindControl("Label1"); DropDownList dl = (DropDownList)C1Dialog1.FindControl("DropDownList1"); DropDownList dlc = (DropDownList)C1Dialog1.FindControl("DropDownList2"); dlc.Items.Clear(); dlc.Items.Add(new ListItem("-")); </pre>	

```
if (dl.Text != "-")
{
    l.Text = "Select a city in " + dl.Text;

    switch (dl.Text) {

        case "UK":

            dlc.Items.Add(new ListItem("London"));
            dlc.Items.Add(new ListItem("Birmingham"));
            dlc.Items.Add(new ListItem("Leeds"));
            dlc.Items.Add(new ListItem("Glasgow"));
            dlc.Items.Add(new ListItem("Glasgow"));
            dlc.Items.Add(new ListItem("Sheffield"));
            dlc.Items.Add(new ListItem("Bradford"));
            dlc.Items.Add(new ListItem("Edinburgh"));
            dlc.Items.Add(new ListItem("Liverpool"));
            break;

        case "USA":

            dlc.Items.Add(new ListItem("New York, New York"));
            dlc.Items.Add(new ListItem("Los Angeles,
California"));

            dlc.Items.Add(new ListItem("Chicago, Illinois"));
            dlc.Items.Add(new ListItem("Houston, Texas"));
            dlc.Items.Add(new ListItem("Philadelphia,
Pennsylvania"));

            dlc.Items.Add(new ListItem("Phoenix, Arizona"));
            dlc.Items.Add(new ListItem("San Diego, California"));
            dlc.Items.Add(new ListItem("Dallas, Texas"));
            dlc.Items.Add(new ListItem("Detroit, Michigan"));
            break;

        case "Russia":

            dlc.Items.Add(new ListItem("Moscow"));
            dlc.Items.Add(new ListItem("Chelyabinsk"));
            dlc.Items.Add(new ListItem("Ekaterinburg"));
            dlc.Items.Add(new ListItem("Irkutsk"));
            dlc.Items.Add(new ListItem("St. Petersburg"));
            dlc.Items.Add(new ListItem("Volgograd"));
    }
}
```



```
        dlc.Items.Add(new ListItem("Petrozavodsk"));
        dlc.Items.Add(new ListItem("Nizhni Novgorod"));
        dlc.Items.Add(new ListItem("Novosibirsk"));
        break;

    case "Canada":

        dlc.Items.Add(new ListItem("Toronto, Ontario"));
        dlc.Items.Add(new ListItem("Montreal, Quebec"));
        dlc.Items.Add(new ListItem("Vancouver, British
Columbia"));

        dlc.Items.Add(new ListItem("Calgary, Alberta"));
        dlc.Items.Add(new ListItem("Edmonton, Alberta"));
        dlc.Items.Add(new ListItem("Ottawa - Gatineau,
Ontario/Quebec"));

        break;

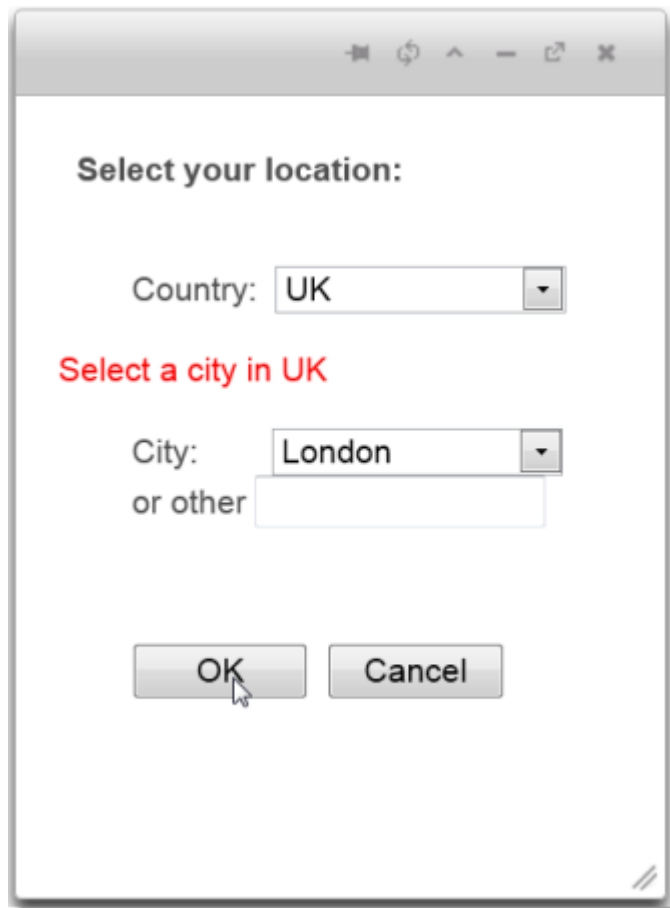
    default:
        break;

    }
}
```

This topic illustrates the following:

Run the application and complete the following steps:

1. Select a country from the first drop-down list.
2. Notice that the next drop-down list of selectable cities is refreshed from the server.
3. Select a city from the drop-down list and click OK.



The name of the city you select is reflected on the Web page.

Customizing the C1Dialog Control

Customizing the Foreground and Background Colors

See Also

[Customizing the Foreground and Background Colors](#)

[Customizing the Font Style](#)

[Customizing the Caption Bar](#)



[Customizing the Width and Height of C1Dialog](#)

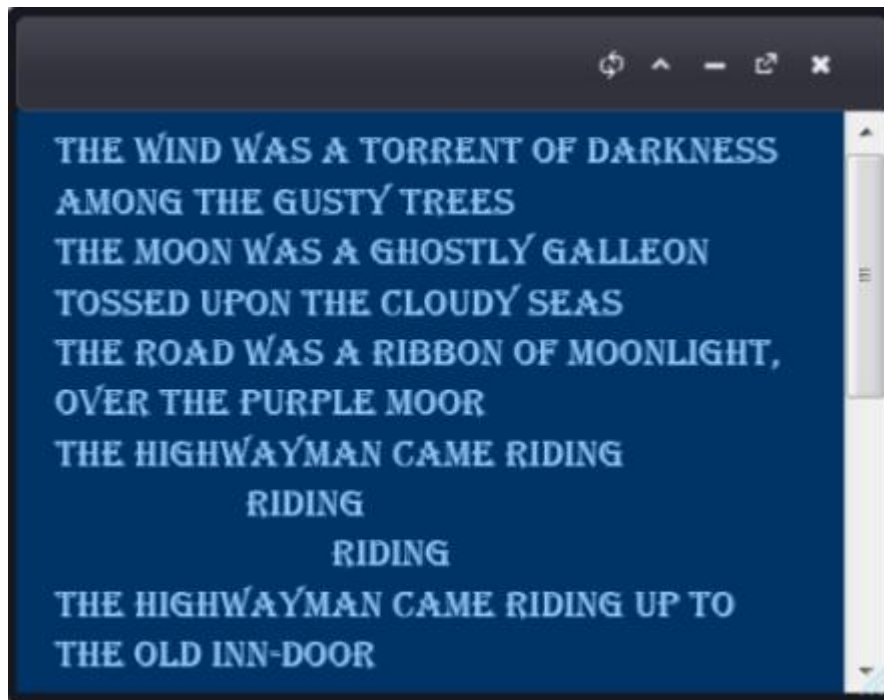
Customizing the Foreground and Background Colors

This topic illustrates setting the foreground and background colors of the Content Area in both the Design View and in the Source View.

In Design View

Complete the following steps:

1. Add the [C1Dialog](#) control to an ASP.NET project.
2. Select the [C1Dialog](#) control.
3. Locate the **Foreground Color**  and **Background**  **Color** buttons in the **Visual Studio Task toolbar**.
4. Click the **Foreground Color** button to choose your text color.
5. Click the **Background Color** button to choose your background color.
6. Enter text in the [C1Dialog](#) control.
7. Press F5 or start debugging to run your project. Your [C1Dialog](#) window should resemble the following image:



In Source View

Complete the following steps:

1. Insert the following .html markup between the `<cc1:C1Dialog>` tags:

[To write code in Source View](#)

```
style="color: #99CCFF; background-color: #003366"
```

```
<Content>  
Enter your text here  
</Content>
```

2. Run your program to see the formatted [C1Dialog](#) and text.

Customizing the Font Style

This topic will walk you through customizing the font style of your Dialog window in Design View and in Source View.

In Design View

Complete the following steps:

1. Select your [C1Dialog](#) control.
2. Go to the Font drop-down list in the Visual Studio toolbar.
3. Choose a font from the list. For this topic, choose **Blackadder ITC**.
4. Type or paste text into your [C1Dialog](#) control Content Area.
5. Press F5 or start debugging to run your program. The text in your Dialog window will appear in the font you chose.

In Source View

Complete the following steps:

1. Go to the Source View of your project.
2. Find the `<cc1:C1Dialog>` tags. Add the following markup and `<Content>` tags to the `<cc1:C1Dialog>` tag:

[To write code in Source View](#)

```
Style="font-family:Blackadder ITC"
```

```
<Content> </Content>
```

3. Type or paste your content between the `<Content>` tags.
4. Press F5 or start debugging to run your program. Your text will appear in the [Dialog](#) window in the font you chose.

Customizing the Caption Bar

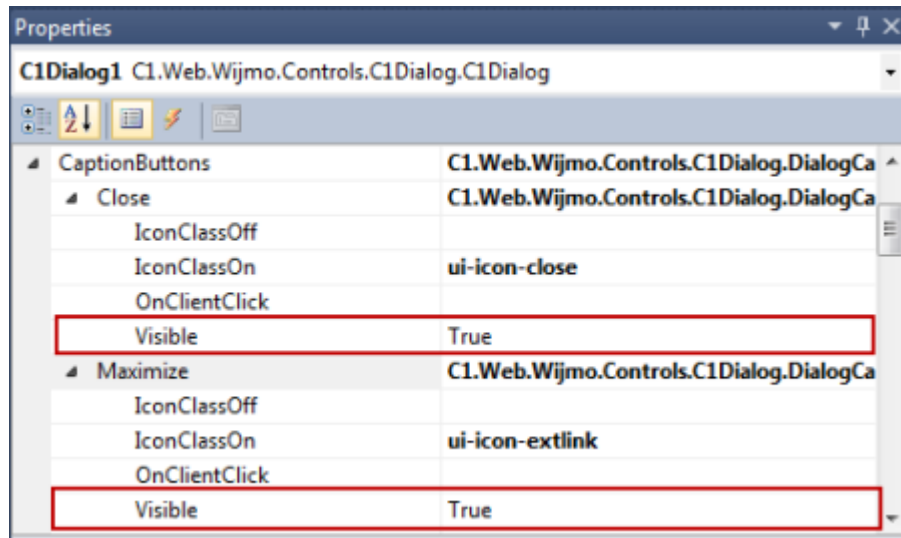
This topic will take you through customizing the buttons that appear on the Caption Bar both in the Design View and in the Source View.

In Design View

Complete the following steps:

1. Navigate to the [C1Dialog](#) Properties window to find the [CaptionButtons](#) property.
2. Use the arrow to expand the property node.
3. Expand each of the [CaptionButtons](#) nodes to see their properties.

- Find the **Visible** property for each **CaptionButton** and set it to **True** or **False**. Your **Property** window should resemble the following image:



- Press F5 or start debugging to run your program.

Note that the Caption Bar Button items you set to **False** are no longer visible.

In Source View

Complete the following steps:

- Insert the following markup between the `<cc1:C1Dialog>` tags:

To write code in Source View

```
<CaptionButtons>

    <Pin IconClassOn="ui-icon-pin-w" IconClassOff="ui-icon-pin-s"></Pin>
    <Refresh IconClassOn="ui-icon-refresh"
Visible="False"></Refresh>
    <Minimize IconClassOn="ui-icon-minus"></Minimize>
    <Maximize IconClassOn="ui-icon-extlink"></Maximize>
    <Close IconClassOn="ui-icon-close" Visible="False"></Close>

</CaptionButtons>
```

- Press F5 or start debugging to run the program. Note that the Caption Bar buttons with a **Visible** property set to **False** do not appear.

Customizing the Width and Height of C1Dialog

This topic will take you through customizing the width and height of [C1Dialog](#). This can be done in the Design View, in the Source View, in Code, or at run time.

In Design View

Complete the following steps:

1. Navigate to the [C1Dialog](#) Properties window and find the [Height](#) property in the list.
2. Set the height of the control in pixels, for this topic, set it to 400px.
3. Find the [Width](#) property in the list. Set the width of the control in pixels, for this topic, set it to 400px.
4. Press F5 to run the program and see the changes you made.

In Source View

Complete the following steps:

1. Go to your project's Source View.
2. The [Height](#) and [Width](#) properties are both within the `<cc1:C1Dialog>` tags:

To write code in Source View

```
<cc1:C1Dialog ID="C1Dialog1" runat="server"
CloseOnEscape="False"
    Height="400px" Width="400px">
```

3. Set the [Height](#) and [Width](#) properties to 400px.
4. Press F5 to run your program.

In Code

Add the following code to the **Page_Load** event to set the [Height](#) and [Width](#) properties of the [C1Dialog](#) control.

To write code in Visual Basic

Visual Basic	Copy Code
<pre>Me.C1Dialog1.Height = 150 Me.C1Dialog1.Width = 150</pre>	

To write code in C#

C#	Copy Code
<pre>this.C1Dialog1.Height = 150; this.C1Dialog1.Width = 150;</pre>	

At Run Time

Complete the following steps:

You can set the width and height of your dialog window at run time, as well.

1. Press F5 to run your program.
2. Hover over a corner or edge of your C1Dialog|tag=C1Dialog_Class control.
3. Use the arrows to resize your C1Dialog|tag=C1Dialog_Class control.

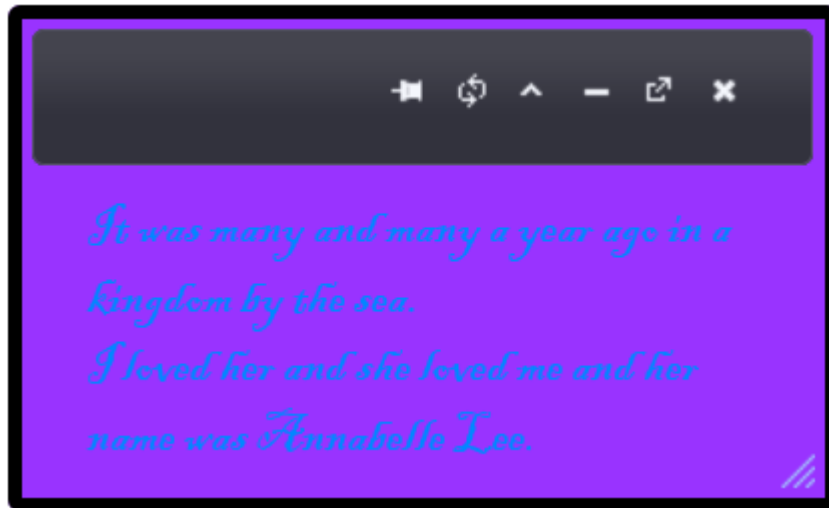
CSS Selectors

[C1Dialog](#) supports CSS styling, enabling you to completely customize the control's appearance. This topic will walk you through applying CSS styling to the [C1Dialog](#) control.

1. In Design View, select **View | Properties** from the Visual Studio menu. Select [C1Dialog](#) from the drop-down list at the top of the Properties window and locate the **CssClass** property.
2. Use the drop-down list to choose the appropriate CSS selector, for this topic, choose the **wijmo-wijdialog** selector.
3. Switch to Source View and locate the first set of <asp:Content> tags.
4. Insert <style type="text/css"></style> tags between the <asp:Content> tags. This set of tags will allow you to add CSS styling to your control.
5. Insert the following script between the <style> tags to add CSS styling to your control.

```
.wijmo-wijdialog
{
    color: #0088FF;
    background: #9933FF;
    border-color: #000000;
    border-width: thick;
    font-family: Blackadder ITC;
    font-size: large;
}
```

6. Run your program. The [C1Dialog](#) control should appear as in the following image.



Using Themes with C1Dialog

[C1Dialog](#) provides six built-in CSS themes. You can also choose a different theme by using the CDN or by rolling your own theme with jQuery ThemeRoller. This topic will walk you through applying a built-in theme, choosing a custom theme using the CDN, or rolling your own theme using ThemeRoller.

Applying Themes

The following topics will walk you through the many ways to theme the [C1Dialog](#) control.

See Also

[Applying a Built-in Theme Using the Smart Tag](#)


[Adding a Custom Theme](#)

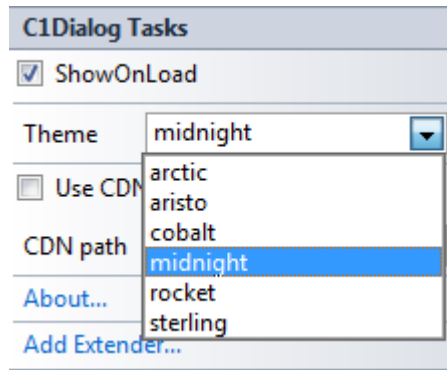
[Using jQuery ThemeRoller](#)

Applying a Built-in Theme Using the Smart Tag

Applying one of the six built-in themes using the smart tag is an easy way to customize your Dialog window.

Complete the following steps:

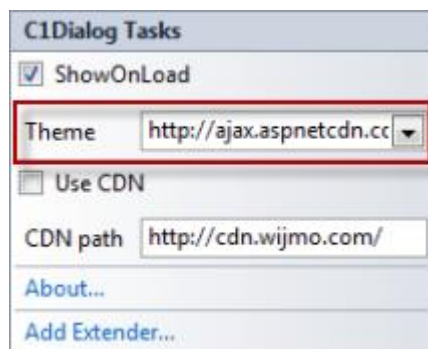
1. Select the [C1Dialog](#) control in your project.
2. Click the smart tag button  to open the [C1Dialog Tasks](#) menu.
3. Use the **Theme** drop-down box to choose a new theme.



4. Press F5 or start debugging to run the program. Your chosen theme will be applied to the Dialog window.

Adding a Custom Theme

1. Click the **C1Dialog** smart tag  to open the **C1Dialog Tasks** menu.
2. In the Theme drop-down box, enter a URL to specify the theme; URLs can be found at <http://blog.jqueryui.com/2011/06/jquery-ui-1-8-14/>. In this example, we'll use the *swanky-purse* theme: <http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.14/themes/swanky-purse/jquery-ui.css>




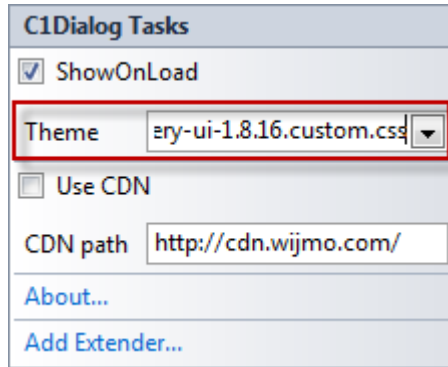
This theme setting is stored in the `<appSettings>` of the **Web.config** file. In the Solution Explorer, double-click the **Web.config** file. Notice the `<appSettings>` tag contains a **WijmoTheme** key and value; this is where the CDN URL you added is specified.

3. Run the project and notice that the theme has been applied to **C1Dialog**.

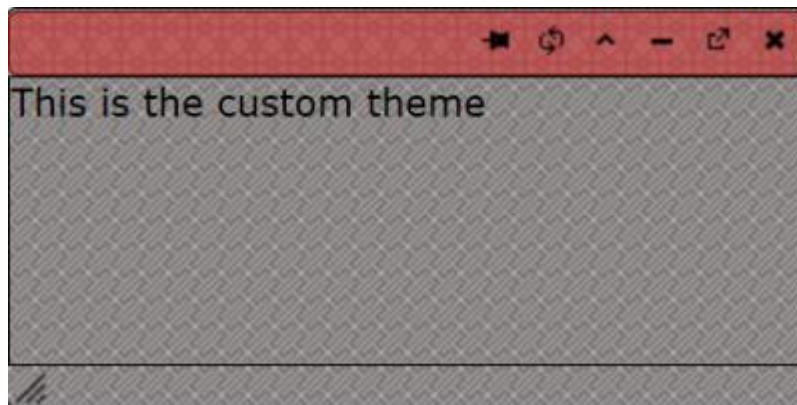
Using jQuery ThemeRoller

1. Go to <http://jqueryui.com/themeroller/>.
2. Go to the **Roll Your Own** tab. Change the settings to create a custom theme; you can customize fonts, backgrounds, colors, and more. Or you can click the **Gallery** tab to 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**. Right-click **customtheme** and click **Include In Project**.
6. Click the [C1Dialog](#) smart tag  to open the **Tasks** menu.
7. Enter the path to your custom theme .css in the **Theme** property, for example, `\customtheme\css\custom-theme\jquery-ui-1.8.16.custom.css`.



8. Run your project and note that the new theme is applied to [C1Dialog](#).



Using the Modal Dialog Option

You can make [C1Dialog](#) a modal dialog box. This child window will take control of the current application until it is closed. The Modal option allows you to use [C1Dialog](#) to impart important information or to require user interaction. In this section, you will learn how to create a Modal dialog box, an Alert dialog box, and a Confirm dialog box.

See Also

[Creating a Modal Dialog Box](#)

[Creating a Confirm Dialog Box](#)

Creating a Modal Dialog Box

This topic will walk you through setting [C1Dialog](#) as a modal dialog box in Design View, in Source View, and in Code.

In Design View

Complete the following steps:

1. In the [C1Dialog](#) Properties window, scroll down to the [Modal](#) property.
2. Use the drop-down menu to set the property to **True**.
3. Press F5 or start debugging to run your program. Note that until you close the Modal window, you have no control over the rest of the page.

In Source View

Complete the following steps:

1. Add the following markup within the `<cc1:C1Dialog>` tags:

[To write code in Source View](#)

```
Modal="True"
```

2. Run your program and note the effect the Modal window has on the rest of the Web page.

In Code

Add the following code to the Page_Load event to set [C1Dialog](#)'s [Modal](#) property.

[To write code in Visual Basic](#)

Visual Basic	Copy Code
Me.C1Dialog1.Modal = True	

[To write code in C#](#)

C#	Copy Code
this.C1Dialog1.Modal = true;	

Creating an Alert Dialog Box

You can use [C1Dialog](#) to alert users to information they need to know, such as when an upload completes. For this topic, we will create a Modal Alert window that tells users that their upload has completed.

Complete the following steps:

1. Go to the Source View of your project.
2. Locate the first pair of `<asp:Content>` tags. Insert the following script between them:

[To write code in Source View](#)

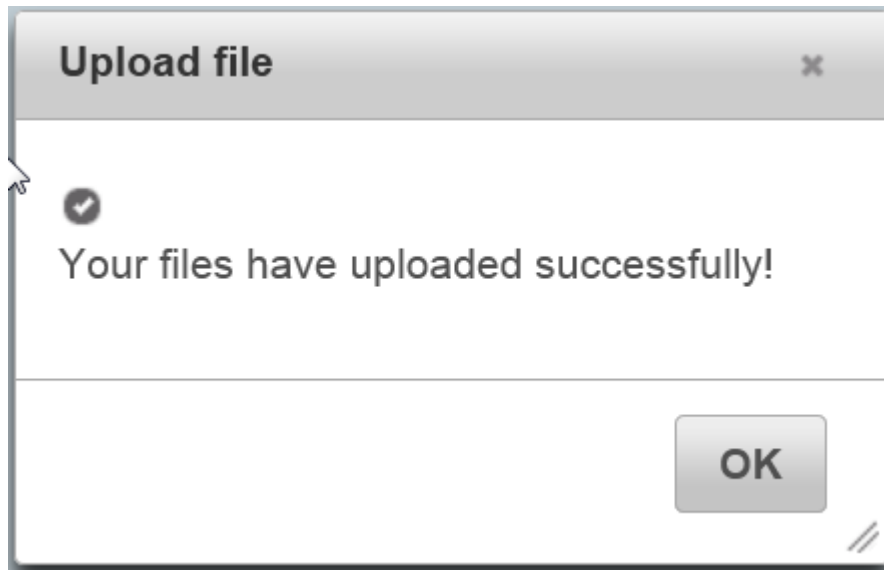
```
<script type="text/javascript">
    function btnClick() {
        $(this).c1dialog("close");
    }
</script>
```

3. Locate the `<cc1:C1Dialog>` tags and insert the following markup after the `runat="server"` property:

[To write code in Source View](#)

```
Width="450px" Height="285px" Modal="true"
    Stack="True" CloseText="Close" Title="Upload file">
    <Content>
        <p>
            <span class="ui-icon ui-icon-circle-check"></span>Your files have uploaded successfully!
        </p>
    </Content>
    <ExpandingAnimation Duration="400">
    </ExpandingAnimation>
    <Buttons>
        <cc1:DialogButton OnClientClick="btnClick" Text="OK"
/>
    </Buttons>
    <CaptionButtons>
        <Pin Visible="False" />
        <Refresh Visible="False" />
        <Toggle Visible="False" />
        <Minimize Visible="False" />
        <Maximize Visible="False" />
    </CaptionButtons>
    <CollapsingAnimation Duration="300" />
```

4. Run your program. The Alert window should resemble the following image:



Note that all the Caption Bar buttons except for the Close button have been disabled.

Creating a Confirm Dialog Box

[C1Dialog](#) can also be used to confirm user actions, such as emptying the recycle bin. You will create a Confirm Dialog window in this topic.

Complete the following steps:

1. Go to the Source View of your project.
2. Locate the first pair of `<asp:Content>` tags and insert the following script after the `runat="server"` property:

To write code in Source View

```
<script type="text/javascript">
    function delClick() {
        $(this).c1dialog("close");
    }
    function cancelClick() {
        $(this).c1dialog("close");
    }
</script>
</asp:Content>
```

3. Locate the `<cc1:C1Dialog>` tag. Insert the following markup after the `runat="server"` property:

To write code in Source View

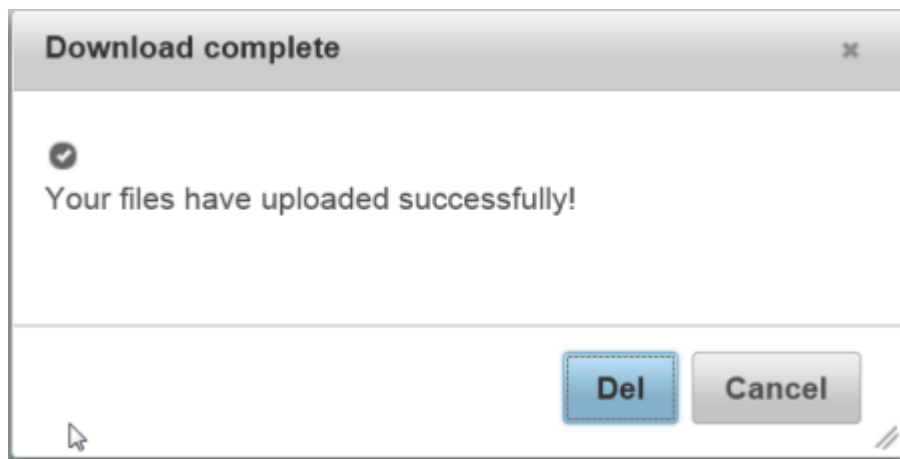
```
<cc1:C1Dialog ID="C1Dialog1" runat="server" Width="400"
Height="200" Title="Download complete"
Modal="True" CloseText="Close">
    <Content>
        <p>
```

```

        <span class="ui-icon ui-icon-circle-
check"></span>Your files have uploaded successfully!
    </p>
</Content>
<ExpandingAnimation Duration="400" />
<CollapsingAnimation Duration="300" />
<Buttons>
    <ccl:DialogButton onclick="delClick"
text="Del" />
    <ccl:DialogButton onclick="cancelClick"
text="Cancel" />
</Buttons>
<CaptionButtons>
    <Pin Visible="false" />
    <Refresh Visible="False" />
    <Toggle Visible="False" />
    <Minimize Visible="False" />
    <Maximize Visible="False" />
</CaptionButtons>

```

4. Run your program. The Confirm Dialog window should resemble the following image:



Animating C1Dialog

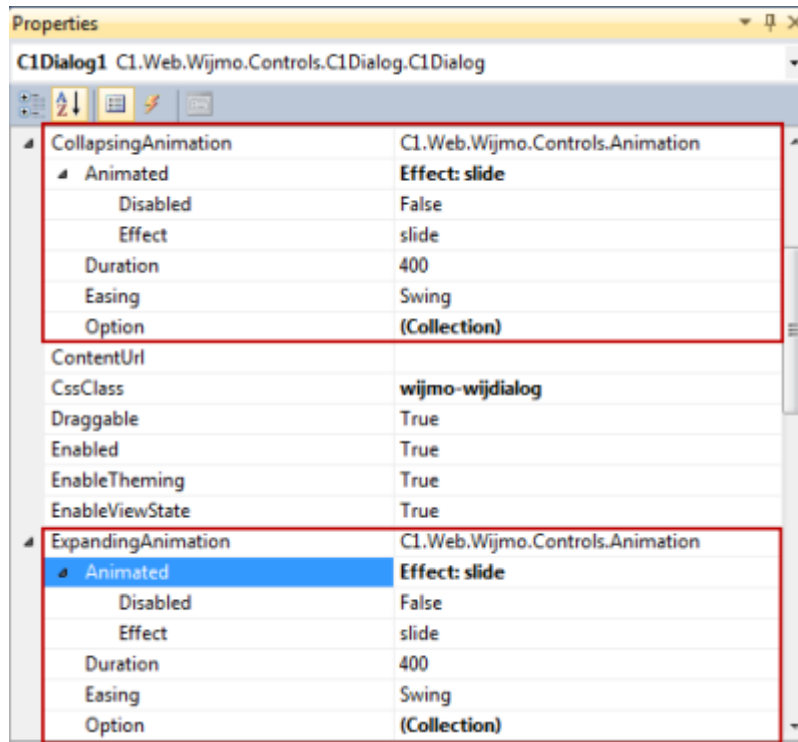
You can add animation effects to four [C1Dialog](#) properties in either Design View or Source View. There are four properties that you can alter to animate [C1Dialog:Closing Animation](#), [Expanding Animation](#), [Hide](#), and [Show](#). This topic will walk you through applying animation effects in Design View and in Source View.

In Design View

Complete the following steps:

1. Select **View | Properties** from the Visual Studio menu and select [C1Dialog](#) from the drop-down list at the top of the window.
2. Locate the **ClosingAnimation** property in the list.
3. Use the arrows to expand the property node.
4. Go to **ClosingAnimation:Animated:Effect** and enter **bounce**.
5. Find the **ExpandingAnimation** property and use the arrows to expand the property node.

- Go to **ExpandingAnimation:Animation:Effect** and enter **shake**. Your **Properties** window should resemble the following image:



- Press F5 or start debugging to run the program.

In Source View

- Add the following markup within your `<cc1:C1Dialog>` tags to set your Show and Hide animation effects.
- Add the following markup before the `<Content>` tag to set your Expanding and Collapsing animation effects:
- Run your program to see the animation effects you have applied to your project.

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 Web site 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 **Dialog 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.

- [wijdialog documentation](#)
- [wijdialog API](#)

See Also

[Using the Wijmo CDN](#)

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.



C1.Web.Wijmo.Controls.4 Assembly






Namespaces

C1.Web.Wijmo.Controls Namespace

Overview

[Inheritance Hierarchy](#)

Classes

	Class	Description
	C1BaseStateManager	Base class to support view state management for a server control.
	C1TargetControlBase	Class to be used as base for the wijmo extenders target controls. Class to be used as base for the wijmo target controls.
	C1TargetControlHelper	
	C1TargetControlHelper<T>	
	WijChartNavigator	

See Also

Reference

[C1.Web.Wijmo.Controls.4 Assembly](#)

Classes

C1BaseStateManager

Base class to support view state management for a server control.

Object Model

C1BaseStateManager

Syntax

Visual Basic (Declaration)	
Public MustInherit Class C1BaseStateManager	
C#	
public abstract class C1BaseStateManager	

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1BaseStateManager

See Also

Reference

[C1BaseStateManager Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Overview

Base class to support view state management for a server control.

Object Model

C1BaseStateManager

Syntax

Visual Basic (Declaration)

```
Public MustInherit Class C1BaseStateManager
```

C#

```
public abstract class C1BaseStateManager
```

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1BaseStateManager

See Also

Reference

[C1BaseStateManager Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Members

The following tables list the members exposed by [C1BaseStateManager](#).

See Also

Reference

[C1BaseStateManager Class](#)[C1.Web.Wijmo.Controls Namespace](#)

C1TargetControlBase

Class to be used as base for the wijmo extenders target controls. Class to be used as base for the wijmo target controls.

Object Model

C1TargetControlBase

Syntax

Visual Basic (Declaration)

```
<WidgetDependenciesAttribute(System.Object[])>  
Public MustInherit Class C1TargetControlBase  
    Inherits System.Web.UI.WebControls.WebControl
```

C#

```
[WidgetDependencies(System.Object[])]  
public abstract class C1TargetControlBase :  
    System.Web.UI.WebControls.WebControl
```

Inheritance Hierarchy

```
System.Object  
    System.Web.UI.Control  
        System.Web.UI.WebControls.WebControl  
            C1.Web.Wijmo.Controls.C1TargetControlBase  
                C1.Web.Wijmo.Controls.C1Dialog.C1Dialog
```

See Also

Reference

[C1TargetControlBase Members](#)[C1.Web.Wijmo.Controls Namespace](#)

Overview

Class to be used as base for the wijmo extenders target controls. Class to be used as base for the wijmo target controls.

Object Model

C1TargetControlBase

Syntax

Visual Basic (Declaration)	
<pre><WidgetDependenciesAttribute(System.Object[])> Public MustInherit Class C1TargetControlBase Inherits System.Web.UI.WebControls.WebControl</pre>	
C#	
<pre>[WidgetDependencies(System.Object[])] public abstract class C1TargetControlBase : System.Web.UI.WebControls.WebControl</pre>	

Inheritance Hierarchy

System.Object
 System.Web.UI.Control
 System.Web.UI.WebControls.WebControl
 C1.Web.Wijmo.Controls.C1TargetControlBase
 [C1.Web.Wijmo.Controls.C1Dialog.C1Dialog](#)

See Also

Reference



[C1TargetControlBase Members](#)















[C1.Web.Wijmo.Controls Namespace](#)
















Members










The following tables list the members exposed by [C1TargetControlBase](#).

Public Properties

	Name	Description
	AccessKey	(Inherited from System.Web.UI.WebControls.WebControl)
	AppRelativeTemplateSourceDirectory	(Inherited from System.Web.UI.Control)


	Attributes	(Inherited from System.Web.UI.WebControls.WebControl)
	CDNDependencyPaths	A string array value specifies all css/js references that wijmo control depends on.
	CDNPath	Content Delivery Network path.
	ClientID	(Inherited from System.Web.UI.Control)
	ClientIDMode	(Inherited from System.Web.UI.Control)
	Controls	(Inherited from System.Web.UI.Control)
	ControlStyle	(Inherited from System.Web.UI.WebControls.WebControl)
	ControlStyleCreated	(Inherited from System.Web.UI.WebControls.WebControl)
	CssClass	(Inherited from System.Web.UI.WebControls.WebControl)
	DisplayVisible	Gets or sets a value that indicates whether a server control is rendered as UI on the page.
	EnableCombinedJavaScripts	Enable JavaScripts files combined into one file. In order to combined enabled, you must register the WijmoHttpHandler in web.config.
	Enabled	Overridden. Gets or sets a value indicating whether the Web server control is enabled.
	EnableTheming	(Inherited from System.Web.UI.WebControls.WebControl)
	EnableViewState	(Inherited from System.Web.UI.Control)

	HasAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
	Height	(Inherited from System.Web.UI.WebControls.WebControl)
	ID	(Inherited from System.Web.UI.Control)
	NamingContainer	(Inherited from System.Web.UI.Control)
	Page	(Inherited from System.Web.UI.Control)
	Parent	(Inherited from System.Web.UI.Control)
	RenderingCompatibility	(Inherited from System.Web.UI.Control)
	Site	(Inherited from System.Web.UI.Control)
	SkinID	(Inherited from System.Web.UI.WebControls.WebControl)
	Style	(Inherited from System.Web.UI.WebControls.WebControl)
	SupportsDisabledAttribute	(Inherited from System.Web.UI.WebControls.WebControl)
	TabIndex	(Inherited from System.Web.UI.WebControls.WebControl)
	TemplateControl	(Inherited from System.Web.UI.Control)
	TemplateSourceDirectory	(Inherited from System.Web.UI.Control)
	Theme	Name of the theme that will be used to style the widgets. Available embedded themes include: aristo, midnight, and ui-lightness. Note that only one theme can be used for the whole page at one time.




		Please make sure that all widget extenders have the same Theme value.
	ThemeSwatch	A value that indicates the theme swatch of the control, this property only works when WijmoControlMode property is Mobile.
	ToolTip	(Inherited from System.Web.UI.WebControls.WebControl)
	UniqueID	(Inherited from System.Web.UI.Control)
	UseCDN	Determines whether the widget extender must load client resources from CDN (Content Delivery Network) path given by property CDNPath.
	ValidateRequestMode	(Inherited from System.Web.UI.Control)
	ViewStateMode	(Inherited from System.Web.UI.Control)
	Width	(Inherited from System.Web.UI.WebControls.WebControl)
	WijmoControlMode	A value that indicates mode of the control, whether it is a mobile or web control. Note that only one value can be used for the whole website or project.
	WijmoCssAdapter	Indicates the control applies the theme of JQuery UI or Bootstrap.

[Top](#)

Public Methods







	Name	Description
	ApplyStyle	(Inherited from System.Web.UI.WebControls.WebControl)

≡	ApplyStyleSheetSkin	(Inherited from System.Web.UI.Control)
≡	CopyBaseAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
≡	DataBind	(Inherited from System.Web.UI.Control)
≡	Dispose	(Inherited from System.Web.UI.Control)
≡	FindControl	(Inherited from System.Web.UI.Control)
≡	Focus	(Inherited from System.Web.UI.Control)
≡	GetPropertyValues<V>	Gets the property value by property name.
≡	GetRouteUrl	Overloaded. (Inherited from System.Web.UI.Control)
≡	GetScriptDescriptors	When overridden in a derived class, registers the WidgetDescriptor objects for the control.
≡	GetScriptReferences	When overridden in a derived class, registers an additional script libraries for the extender control.
≡	GetUniqueIDRelativeTo	(Inherited from System.Web.UI.Control)
≡	HasControls	(Inherited from System.Web.UI.Control)
≡	MergeStyle	(Inherited from System.Web.UI.WebControls.WebControl)
≡	RegisterDesignTimeStyleSheet	Register design time stylesheet
≡	RenderBeginTag	(Inherited from System.Web.UI.WebControls.WebControl)
≡	RenderControl	(Inherited from System.Web.UI.Control)
≡	RenderEndTag	(Inherited from System.Web.UI.WebControls.WebControl)
≡	ResolveClientUrl	(Inherited from System.Web.UI.Control)

	ResolveUrl	(Inherited from System.Web.UI.Control)
	SetPropertyValue<V>	Sets the property value by property name.
	SetRenderMethodDelegate	(Inherited from System.Web.UI.Control)

[Top](#)

Public Events

	Name	Description
	DataBinding	(Inherited from System.Web.UI.Control)
	Disposed	(Inherited from System.Web.UI.Control)
	Init	(Inherited from System.Web.UI.Control)
	Load	(Inherited from System.Web.UI.Control)
	PreRender	(Inherited from System.Web.UI.Control)
	Unload	(Inherited from System.Web.UI.Control)

[Top](#)

See Also


Reference

[C1TargetControlBase Class](#)[C1.Web.Wijmo.Controls Namespace](#)




Methods

For a list of all members of this type, see [C1TargetControlBase members](#).

Public Methods

	Name	Description
	ApplyStyle	(Inherited from System.Web.UI.WebControls.WebControl)

⇒	ApplyStyleSheetSkin	(Inherited from System.Web.UI.Control)
⇒	CopyBaseAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
⇒	DataBind	(Inherited from System.Web.UI.Control)
⇒	Dispose	(Inherited from System.Web.UI.Control)
⇒	FindControl	(Inherited from System.Web.UI.Control)
⇒	Focus	(Inherited from System.Web.UI.Control)
⇒	GetPropertyValues<V>	Gets the property value by property name.
⇒	GetRouteUrl	Overloaded. (Inherited from System.Web.UI.Control)
⇒	GetScriptDescriptors	When overridden in a derived class, registers the WidgetDescriptor objects for the control.
⇒	GetScriptReferences	When overridden in a derived class, registers an additional script libraries for the extender control.
⇒	GetUniqueIDRelativeTo	(Inherited from System.Web.UI.Control)
⇒	HasControls	(Inherited from System.Web.UI.Control)
⇒	MergeStyle	(Inherited from System.Web.UI.WebControls.WebControl)
⇒	RegisterDesignTimeStyleSheet	Register design time stylesheet
⇒	RenderBeginTag	(Inherited from System.Web.UI.WebControls.WebControl)
⇒	RenderControl	(Inherited from System.Web.UI.Control)
⇒	RenderEndTag	(Inherited from System.Web.UI.WebControls.WebControl)
⇒	ResolveClientUrl	(Inherited from System.Web.UI.Control)

	ResolveUrl	(Inherited from System.Web.UI.Control)
	SetPropertyValue<V>	Sets the property value by property name.
	SetRenderMethodDelegate	(Inherited from System.Web.UI.Control)

[Top](#)

See Also

Reference

[C1TargetControlBase Class](#)[C1.Web.Wijmo.Controls Namespace](#)

GetPropertyValue<V> Method

Gets the property value by property name.

Syntax

Visual Basic (Declaration)	
<pre>Public Function GetPropertyValue(Of V)(_ ByVal <i>propertyName</i> As System.String, _ ByVal <i>nullValue</i> As V _) As V</pre>	
C#	
<pre>public V GetPropertyValue<V>(System.string <i>propertyName</i>, V <i>nullValue</i>)</pre>	

Parameters

propertyName

Name of the property.

nullValue

The null value.

Type Parameters

V

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

GetScriptDescriptors Method

When overridden in a derived class, registers the WidgetDescriptor objects for the control.

Syntax

Visual Basic (Declaration)	
<pre>Public Overridable Function GetScriptDescriptors() As System.Collections.Generic.IEnumerable(Of ScriptDescriptor)</pre>	
C#	
<pre>public virtual System.Collections.Generic.IEnumerable<ScriptDescriptor> GetScriptDescriptors()</pre>	

Return Value

An enumeration of WidgetDescriptor objects.

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

GetScriptReferences Method

When overridden in a derived class, registers an additional script libraries for the extender control.

Syntax

Visual Basic (Declaration)	
<pre>Public Overridable Function GetScriptReferences() As System.Collections.Generic.IEnumerable(Of ScriptReference)</pre>	
C#	

```
public virtual System.Collections.Generic.IEnumerable<ScriptReference>
GetScriptReferences()
```

Return Value

An object that implements the **System.Collections.IEnumerable** interface and that contains ECMAScript (JavaScript) files that have been registered as embedded resources.

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

RegisterDesignTimeStyleSheet Method

The **System.Web.UI.HtmlTextWriter** object that receives the server control content.

resource name of stylesheet file

Register design time stylesheet

Syntax

Visual Basic (Declaration)

```
Public Sub RegisterDesignTimeStyleSheet( _
    ByVal writer As System.Web.UI.HtmlTextWriter, _
    ByVal cssResourceName As System.String _
)
```

C#

```
public void RegisterDesignTimeStyleSheet(
    System.Web.UI.HtmlTextWriter writer,
    System.string cssResourceName
)
```

Parameters

writer

The **System.Web.UI.HtmlTextWriter** object that receives the server control content.

cssResourceName

resource name of stylesheet file

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

SetPropertyValue<V> Method

Sets the property value by property name.

Syntax

Visual Basic (Declaration)

```
Public Sub SetPropertyValue(Of V)( _  
    ByVal propertyName As System.String, _  
    ByVal value As V _  
)
```

C#

```
public void SetPropertyValue<V>(   
    System.string propertyName,  
    V value  
)
```

Parameters

propertyName

Name of the property.

value

The value.

Type Parameters

V

See Also

Reference














[C1TargetControlBase Class](#)
















[C1TargetControlBase Members](#)













Properties

For a list of all members of this type, see [C1TargetControlBase members](#).

Public Properties

	Name	Description
	AccessKey	(Inherited from System.Web.UI.WebControls.WebControl)
	AppRelativeTemplateSourceDirectory	(Inherited from System.Web.UI.Control)
	Attributes	(Inherited from System.Web.UI.WebControls.WebControl)
	CDNDependencyPaths	A string array value specifies all css/js references that wijmo control depends on.
	CDNPath	Content Delivery Network path.
	ClientID	(Inherited from System.Web.UI.Control)
	ClientIDMode	(Inherited from System.Web.UI.Control)
	Controls	(Inherited from System.Web.UI.Control)
	ControlStyle	(Inherited from System.Web.UI.WebControls.WebControl)
	ControlStyleCreated	(Inherited from System.Web.UI.WebControls.WebControl)
	CssClass	(Inherited from System.Web.UI.WebControls.WebControl)
	DisplayVisible	Gets or sets a value that indicates whether a server control is rendered as UI on the page.
	EnableCombinedJavaScripts	Enable JavaScripts files combined into one file. In order to combined enabled, you must register the WijmoHttpHandler in web.config.

	Enabled	Overridden. Gets or sets a value indicating whether the Web server control is enabled.
	EnableTheming	(Inherited from System.Web.UI.WebControls.WebControl)
	EnableViewState	(Inherited from System.Web.UI.Control)
	HasAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
	Height	(Inherited from System.Web.UI.WebControls.WebControl)
	ID	(Inherited from System.Web.UI.Control)
	NamingContainer	(Inherited from System.Web.UI.Control)
	Page	(Inherited from System.Web.UI.Control)
	Parent	(Inherited from System.Web.UI.Control)
	RenderingCompatibility	(Inherited from System.Web.UI.Control)
	Site	(Inherited from System.Web.UI.Control)
	SkinID	(Inherited from System.Web.UI.WebControls.WebControl)
	Style	(Inherited from System.Web.UI.WebControls.WebControl)
	SupportsDisabledAttribute	(Inherited from System.Web.UI.WebControls.WebControl)
	TabIndex	(Inherited from System.Web.UI.WebControls.WebControl)

	TemplateControl	(Inherited from System.Web.UI.Control)
	TemplateSourceDirectory	(Inherited from System.Web.UI.Control)
	Theme	Name of the theme that will be used to style the widgets. Available embedded themes include: aristo, midnight, and ui-lightness. Note that only one theme can be used for the whole page at one time. Please make sure that all widget extenders have the same Theme value.
	ThemeSwatch	A value that indicates the theme swatch of the control, this property only works when WijmoControlMode property is Mobile.
	ToolTip	(Inherited from System.Web.UI.WebControls.WebControl)
	UniqueID	(Inherited from System.Web.UI.Control)
	UseCDN	Determines whether the widget extender must load client resources from CDN (Content Delivery Network) path given by property CDNPath.
	ValidateRequestMode	(Inherited from System.Web.UI.Control)
	ViewStateMode	(Inherited from System.Web.UI.Control)
	Width	(Inherited from System.Web.UI.WebControls.WebControl)
	WijmoControlMode	A value that indicates mode of the control, whether it is a mobile or web control. Note that only one value can be used for the whole website or project.
	WijmoCssAdapter	Indicates the control applies the theme of JQuery UI or Bootstrap.

[Top](#)

See Also

Reference

[C1TargetControlBase Class](#)

[C1.Web.Wijmo.Controls Namespace](#)

CDNDependencyPaths Property

A string array value specifies all css/js references that wijmo control depends on.

Syntax

Visual Basic (Declaration)

```
<C1CategoryAttribute("Behavior")>
<System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)>
<C1DescriptionAttribute(Key="C1Base.CDNDependencyPaths", Description="A
string array value specifies all css/js references that wijmo control depends
on.")>
<System.ComponentModel.DefaultValueAttribute()>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Hidden)>
Public Overridable Property CDNDependencyPaths As System.String()
```

C#

```
[C1Category("Behavior")]
[System.ComponentModel.RefreshProperties(RefreshProperties.All)]
[C1Description(Key="C1Base.CDNDependencyPaths", Description="A string array
value specifies all css/js references that wijmo control depends on.")]
[System.ComponentModel.DefaultValue()]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Hidden)]
public virtual System.string[] CDNDependencyPaths {get; set;}
```

Remarks

Using this property, user can specify the CDN dependencies(such as jQuery, jQuery UI, jQuery mobile, bootstrap). If some dependencies are set in this property, wijmo controls will find dependencies path in this property. If found, use the path to load dependency, otherwise, use the default CDN path. For example, if user wants to specify the dependencies of jQuery and jQuery UI, he can set the value like this: ["http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.11.1.min.js", "http://ajax.aspnetcdn.com/ajax/jquery.ui/1.11.1/jquery-ui.js"]

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

CDNPath Property

Content Delivery Network path.

Syntax

Visual Basic (Declaration)

```
<C1CategoryAttribute("Behavior")>
<System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)>
<C1DescriptionAttribute(Key="C1Base.CDNPath", Description="Content Delivery
Network path.")>
<System.ComponentModel.DefaultValueAttribute()>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Hidden)>
Public Overridable Property CDNPath As System.String
```

C#

```
[C1Category("Behavior")]
[System.ComponentModel.RefreshProperties(RefreshProperties.All)]
[C1Description(Key="C1Base.CDNPath", Description="Content Delivery Network
path.")]
[System.ComponentModel.DefaultValue()]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Hidden)]
public virtual System.string CDNPath {get; set;}
```

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

DisplayVisible Property

Gets or sets a value that indicates whether a server control is rendered as UI on the page.

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
<C1DescriptionAttribute(Key="C1Controls.DisplayVisible", Description="Gets or
sets a value that indicates whether a server control is rendered as UI on the
page.")>
<C1CategoryAttribute("Behavior")>
<LayoutAttribute(LayoutType.Behavior)>
Public Property DisplayVisible As System.Boolean
```

C#

```
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
[C1Description(Key="C1Controls.DisplayVisible", Description="Gets or sets a
value that indicates whether a server control is rendered as UI on the
page.")]
[C1Category("Behavior")]
[Layout(LayoutType.Behavior)]
public System.bool DisplayVisible {get; set;}
```

Remarks

Note that a server control is created and invisible if DisplayVisible is set to false.

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

EnableCombinedJavaScripts Property

Enable JavaScripts files combined into one file. In order to combined enabled, you must register the WijmoHttpHandler in web.config.

Syntax

Visual Basic (Declaration)

```
<C1DescriptionAttribute(Key="C1Base.EnableCombinedJavaScripts",
Description="Enable JavaScripts files combined into one file.")>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeria
```

```

lizationVisibility.Hidden)>
<C1CategoryAttribute("Behavior")>
<System.ComponentModel.DefaultValueAttribute()>
Public Overridable Property EnableCombinedJavaScripts As System.Boolean

```

C#

```

[C1Description(Key="C1Base.EnableCombinedJavaScripts", Description="Enable
JavaScripts files combined into one file.")]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Hidden)]
[C1Category("Behavior")]
[System.ComponentModel.DefaultValue()]
public virtual System.bool EnableCombinedJavaScripts {get; set;}

```

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

Enabled Property

Gets or sets a value indicating whether the Web server control is enabled.

Syntax

Visual Basic (Declaration)

```

<WidgetOptionAttribute()>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Base.Enabled", Description="Indicates whether
control is enabled.")>
<C1CategoryAttribute("Behavior")>
Public Overrides Property Enabled As System.Boolean

```

C#

```

[WidgetOption()]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Base.Enabled", Description="Indicates whether control
is enabled.")]
[C1Category("Behavior")]
public override System.bool Enabled {get; set;}

```

Property Value

true if control is enabled; otherwise, false. The default is true.

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

Theme Property

Name of the theme that will be used to style the widgets. Available embedded themes include: aristo, midnight, and ui-lightness. Note that only one theme can be used for the whole page at one time. Please make sure that all widget extenders have the same Theme value.

Syntax

Visual Basic (Declaration)	
<pre> <TypeConverterAttribute("C1.Web.Wijmo.Controls.WijmoThemeNameConverter, C1.Web.Wijmo.Controls.4, Version=4.0.20151.207, Culture=neutral, PublicKeyToken=9b75583953471eea"> <System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali zationVisibility.Hidden)> <C1DescriptionAttribute(Key="C1Base.Theme", Description="Name of the theme that will be used to style widgets. Available themes: aristo / midnight / ui- lightness. Note, only one theme can be used for the whole page at one time. Please, make sure that all widget extenders have the same Theme value.")> <C1CategoryAttribute("Behavior")> <System.ComponentModel.DefaultValueAttribute()> <System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)> Public Overridable Property Theme As System.String </pre>	
C#	
<pre> [TypeConverter("C1.Web.Wijmo.Controls.WijmoThemeNameConverter, C1.Web.Wijmo.Controls.4, Version=4.0.20151.207, Culture=neutral, PublicKeyToken=9b75583953471eea")] [System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV isibility.Hidden)] [C1Description(Key="C1Base.Theme", Description="Name of the theme that will be used to style widgets. Available themes: aristo / midnight / ui-lightness. Note, only one theme can be used for the whole page at one time. Please, make sure that all widget extenders have the same Theme value.")] [C1Category("Behavior")] </pre>	

```
[System.ComponentModel.DefaultValue()]
[System.ComponentModel.RefreshProperties(RefreshProperties.All)]
public virtual System.string Theme {get; set;}
```

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

ThemeSwatch Property

A value that indicates the theme swatch of the control, this property only works when WijmoControlMode property is Mobile.

Syntax

Visual Basic (Declaration)

```
<C1CategoryAttribute("Appearance")>
<WidgetOptionAttribute()>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Hidden)>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Base.ThemeSwatch", Description="A value that indicates the theme swatch of the control, this property only works when WijmoControlMode property is Mobile.")>
<TypeConverterAttribute("C1.Web.Wijmo.Controls.WijmoThemeSwatchConverter, C1.Web.Wijmo.Controls.4, Version=4.0.20151.207, Culture=neutral, PublicKeyToken=9b75583953471eea")>
Public Overridable Property ThemeSwatch As System.String
```

C#

```
[C1Category("Appearance")]
[WidgetOption()]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Base.ThemeSwatch", Description="A value that indicates the theme swatch of the control, this property only works when WijmoControlMode property is Mobile.")]
[TypeConverter("C1.Web.Wijmo.Controls.WijmoThemeSwatchConverter, C1.Web.Wijmo.Controls.4, Version=4.0.20151.207, Culture=neutral,
```



```

PublicKeyToken=9b75583953471eea"")]
public virtual System.string ThemeSwatch {get; set;}

```

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

UseCDN Property

Determines whether the widget extender must load client resources from CDN (Content Delivery Network) path given by property CDNPath.

Syntax

Visual Basic (Declaration)	
<pre> <C1DescriptionAttribute(Key="C1Base.UseCDN", Description="Determines whether this extender loads client script references from CDN.")> <System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)> <C1CategoryAttribute("Behavior")> <System.ComponentModel.DefaultValueAttribute()> <System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali zationVisibility.Hidden)> Public Overridable Property UseCDN As System.Boolean </pre>	
C#	
<pre> [C1Description(Key="C1Base.UseCDN", Description="Determines whether this extender loads client script references from CDN.")] [System.ComponentModel.RefreshProperties(RefreshProperties.All)] [C1Category("Behavior")] [System.ComponentModel.DefaultValue()] [System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV isibility.Hidden)] public virtual System.bool UseCDN {get; set;} </pre>	

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

WijmoControlMode Property

A value that indicates mode of the control, whether it is a mobile or web control. Note that only one value can be used for the whole website or project.

Syntax

Visual Basic (Declaration)

```
<C1CategoryAttribute("Behavior")>
<C1DescriptionAttribute(Key="C1Base.WijmoControlMode", Description="A value
that indicates mode of the control, whether it is a mobile or web control.
Note that only one value can be used for the whole website or project.")>
<System.ComponentModel.DefaultValueAttribute()>
<System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Hidden)>
Public Overridable Property WijmoControlMode As WijmoControlMode
```

C#

```
[C1Category("Behavior")]
[C1Description(Key="C1Base.WijmoControlMode", Description="A value that
indicates mode of the control, whether it is a mobile or web control. Note
that only one value can be used for the whole website or project.")]
[System.ComponentModel.DefaultValue()]
[System.ComponentModel.RefreshProperties(RefreshProperties.All)]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Hidden)]
public virtual WijmoControlMode WijmoControlMode {get; set;}
```

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

WijmoCssAdapter Property

Indicates the control applies the theme of JQuery UI or Bootstrap.

Syntax

Visual Basic (Declaration)

```

<System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)>
<C1CategoryAttribute("Behavior")>
<C1DescriptionAttribute(Key="C1Base.WijmoCssAdapter", Description="Indicates
the control applies the theme of JQuery UI or Bootstrap.")>
<System.ComponentModel.DefaultValueAttribute()>
<TypeConverterAttribute("C1.Web.Wijmo.Controls.WijmoCssAdapterConverter,
C1.Web.Wijmo.Controls.4, Version=4.0.20151.207, Culture=neutral,
PublicKeyToken=9b75583953471eea")>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Hidden)>
Public Overridable Property WijmoCssAdapter As System.String

```

C#

```

[System.ComponentModel.RefreshProperties(RefreshProperties.All)]
[C1Category("Behavior")]
[C1Description(Key="C1Base.WijmoCssAdapter", Description="Indicates the
control applies the theme of JQuery UI or Bootstrap.")]
[System.ComponentModel.DefaultValue()]
[TypeConverter("C1.Web.Wijmo.Controls.WijmoCssAdapterConverter,
C1.Web.Wijmo.Controls.4, Version=4.0.20151.207, Culture=neutral,
PublicKeyToken=9b75583953471eea")]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Hidden)]
public virtual System.string WijmoCssAdapter {get; set;}

```

See Also

Reference

[C1TargetControlBase Class](#)

[C1TargetControlBase Members](#)

C1TargetControlHelper

Object Model

C1TargetControlHelper

Syntax

Visual Basic (Declaration)

```
Friend Class C1TargetControlHelper
```

C#	
internal class C1TargetControlHelper	

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1TargetControlHelper

[C1.Web.Wijmo.Controls.C1TargetControlHelper<T>](#)

See Also

Reference

[C1TargetControlHelper Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Overview

Object Model

C1TargetControlHelper

Syntax

Visual Basic (Declaration)	
Friend Class C1TargetControlHelper	
C#	
internal class C1TargetControlHelper	

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1TargetControlHelper

[C1.Web.Wijmo.Controls.C1TargetControlHelper<T>](#)

See Also

Reference


[C1TargetControlHelper Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Members


The following tables list the members exposed by [C1TargetControlHelper](#).

Public Constructors

	Name	Description
	C1TargetControlHelper Constructor	

[Top](#)

Public Methods

	Name	Description
	ResolveEmbeddedResourceUrl	

[Top](#)

See Also

Reference

[C1TargetControlHelper Class](#)

[C1.Web.Wijmo.Controls Namespace](#)

C1TargetControlHelper Constructor

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public C1TargetControlHelper()	

See Also

Reference


[C1TargetControlHelper Class](#)

[C1TargetControlHelper Members](#)

Methods

For a list of all members of this type, see [C1TargetControlHelper members](#).

Public Methods

	Name	Description
	ResolveEmbeddedResourceUrl	

[Top](#)

See Also

Reference

[C1TargetControlHelper Class](#)
[C1.Web.Wijmo.Controls Namespace](#)

ResolveEmbeddedResourceUrl Method

Syntax

Visual Basic (Declaration)	
<pre>Public Shared Function ResolveEmbeddedResourceUrl(_ ByVal resourceName As System.String, _ ByVal page As System.Web.UI.Page, _ ByVal combinedScript As System.Boolean _) As System.String</pre>	
C#	
<pre>public static System.string ResolveEmbeddedResourceUrl(System.string resourceName, System.Web.UI.Page page, System.bool combinedScript)</pre>	

Parameters

resourceName

page

combinedScript

See Also

Reference

[C1TargetControlHelper Class](#)

[C1TargetControlHelper Members](#)

C1TargetControlHelper<T>

Object Model

C1TargetControlHelper<T>

Syntax

Visual Basic (Declaration)

```
Friend Class C1TargetControlHelper(Of T As  
{System.Web.UI.WebControls.WebControl, IC1TargetControl})  
    Inherits C1TargetControlHelper
```

C#

```
internal class C1TargetControlHelper<T> : C1TargetControlHelper  
where T: System.Web.UI.WebControls.WebControl, IC1TargetControl
```

Type Parameters

T

Inheritance Hierarchy

System.Object

[C1.Web.Wijmo.Controls.C1TargetControlHelper](#)

C1.Web.Wijmo.Controls.C1TargetControlHelper<T>

See Also

Reference

[C1TargetControlHelper<T> Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Overview

Object Model

C1TargetControlHelper<T>

Syntax

Visual Basic (Declaration)	
<pre>Friend Class C1TargetControlHelper(Of T As {System.Web.UI.WebControls.WebControl, IC1TargetControl}) Inherits C1TargetControlHelper</pre>	
C#	
<pre>internal class C1TargetControlHelper<T> : C1TargetControlHelper where T: System.Web.UI.WebControls.WebControl, IC1TargetControl</pre>	

Type Parameters

T

Inheritance Hierarchy

System.Object

[C1.Web.Wijmo.Controls.C1TargetControlHelper](#)

C1.Web.Wijmo.Controls.C1TargetControlHelper<T>

See Also

Reference


[C1TargetControlHelper<T> Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Members

The following tables list the members exposed by [C1TargetControlHelper<T>](#).


Public Constructors

	Name	Description
	C1TargetControlHelper<T> Constructor	

[Top](#)













Public Properties

	Name	Description
--	------	-------------

	Control	
	JsonSerializableHelper	
	Page	
	ScriptManager	

[Top](#)

Public Methods

	Name	Description
	EnsureEnabledState	
	GetScriptDescriptors	Overloaded. When overridden in a derived class, registers the WidgetDescriptor objects for the control.
	RegisterDesignTimeStyleSheet	Register design-time CSS stylesheet.
	RegisterDesignTimeStyleSheets	
	RegisterIncludes	
	RegisterOnSubmitStatement	
	RegisterRunTimeStylesheets	
	RegisterScriptDescriptors	
	RegisterScriptDescriptorsWithoutScriptManager	
	RegisterScriptDescriptorsWithScriptManager	
	RenderJsonDataField	
	ShowAbout	Show about message box.

[Top](#)

See Also

Reference

[C1TargetControlHelper<T> Class](#)[C1.Web.Wijmo.Controls Namespace](#)

C1TargetControlHelper<T> Constructor

Syntax

Visual Basic (Declaration)

```
Public Function New( _  
    ByVal control As T _  
)
```

C#

```
public C1TargetControlHelper<T>(  
    T control  
)
```

Parameters

control

See Also


Reference












[C1TargetControlHelper<T> Class](#)[C1TargetControlHelper<T> Members](#)

Methods

For a list of all members of this type, see [C1TargetControlHelper<T> members](#).

Public Methods

	Name	Description
	EnsureEnabledState	

	GetScriptDescriptors	Overloaded. When overridden in a derived class, registers the WidgetDescriptor objects for the control.
	RegisterDesignTimeStyleSheet	Register design-time CSS stylesheet.
	RegisterDesignTimeStyleSheets	
	RegisterIncludes	
	RegisterOnSubmitStatement	
	RegisterRunTimeStylesheets	
	RegisterScriptDescriptors	
	RegisterScriptDescriptorsWithoutScriptManager	
	RegisterScriptDescriptorsWithScriptManager	
	RenderJsonDataField	
	ShowAbout	Show about message box.

[Top](#)

See Also

Reference

[C1TargetControlHelper<T> Class](#)[C1.Web.Wijmo.Controls Namespace](#)

EnsureEnabledState Method

Syntax

Visual Basic (Declaration)	
<code>Public Sub EnsureEnabledState()</code>	
C#	

```
public void EnsureEnabledState()
```

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

GetScriptDescriptors Method

When overridden in a derived class, registers the WidgetDescriptor objects for the control.

Overload List

Overload	Description
GetScriptDescriptors()	When overridden in a derived class, registers the WidgetDescriptor objects for the control.
GetScriptDescriptors(String)	When overridden in a derived class, registers the WidgetDescriptor objects for the control.

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

GetScriptDescriptors() Method

When overridden in a derived class, registers the WidgetDescriptor objects for the control.

Syntax

Visual Basic (Declaration)	
Public Overloads Function GetScriptDescriptors() As System.Collections.Generic.IEnumerable(Of ScriptDescriptor)	
C#	
public System.Collections.Generic.IEnumerable<ScriptDescriptor> GetScriptDescriptors()	

Return Value

An enumeration of WidgetDescriptor objects.

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

[Overload List](#)

GetScriptDescriptors(String) Method

When overridden in a derived class, registers the WidgetDescriptor objects for the control.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Function GetScriptDescriptors(_ ByVal <i>targetSelector</i> As System.String _) As System.Collections.Generic.IEnumerable(Of ScriptDescriptor)</pre>	
C#	
<pre>public System.Collections.Generic.IEnumerable<ScriptDescriptor> GetScriptDescriptors(System.string <i>targetSelector</i>)</pre>	

Parameters

targetSelector

The target selector.

Return Value

An enumeration of WidgetDescriptor objects.

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

[Overload List](#)

RegisterDesignTimeStyleSheet Method

Register design-time CSS stylesheet.

Syntax

Visual Basic (Declaration)

```
Public Sub RegisterDesignTimeStyleSheet( _  
    ByVal writer As System.Web.UI.HtmlTextWriter, _  
    ByVal cssResourceName As System.String _  
)
```

C#

```
public void RegisterDesignTimeStyleSheet(  
    System.Web.UI.HtmlTextWriter writer,  
    System.string cssResourceName  
)
```

Parameters

writer

cssResourceName

CSS resource name.

Remarks

The *cssResourceName* is the link to an resource element.

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RegisterDesignTimeStyleSheets Method

Syntax

Visual Basic (Declaration)

```
Public Sub RegisterDesignTimeStyleSheets( _  
    ByVal writer As System.Web.UI.HtmlTextWriter _  
)
```

C#

```
public void RegisterDesignTimeStyleSheets(  
    System.Web.UI.HtmlTextWriter writer  
)
```

Parameters

writer

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RegisterIncludes Method

Syntax

Visual Basic (Declaration)

```
Public Sub RegisterIncludes()
```

C#

```
public void RegisterIncludes()
```

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RegisterOnSubmitStatement Method

Syntax

Visual Basic (Declaration)

```
Public Sub RegisterOnSubmitStatement( _  
    Optional ByVal callback As System.String _  
)
```

C#

```
public void RegisterOnSubmitStatement(  
    System.string callback
```

```
)
```

Parameters

callback

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RegisterRunTimeStylesheets Method

Syntax

Visual Basic (Declaration)	
Public Sub RegisterRunTimeStylesheets()	
C#	
public void RegisterRunTimeStylesheets()	

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RegisterScriptDescriptors Method

Syntax

Visual Basic (Declaration)	
Public Sub RegisterScriptDescriptors()	
C#	
public void RegisterScriptDescriptors()	

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RegisterScriptDescriptorsWithoutScriptManager Method

Syntax

Visual Basic (Declaration)	
Public Sub RegisterScriptDescriptorsWithoutScriptManager()	
C#	
public void RegisterScriptDescriptorsWithoutScriptManager()	

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RegisterScriptDescriptorsWithScriptManager Method

Syntax

Visual Basic (Declaration)	
Public Sub RegisterScriptDescriptorsWithScriptManager()	
C#	
public void RegisterScriptDescriptorsWithScriptManager()	

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

RenderJsonDataField Method

Syntax

Visual Basic (Declaration)	
Public Sub RenderJsonDataField(_ ByVal <i>writer</i> As System.Web.UI.HtmlTextWriter _)	

C#	
<pre>public void RenderJsonDataField(System.Web.UI.HtmlTextWriter writer)</pre>	

Parameters

writer

See Also

Reference

- [C1TargetControlHelper<T> Class](#)
- [C1TargetControlHelper<T> Members](#)

ShowAbout Method

Show about message box.

Syntax

Visual Basic (Declaration)	
<pre>Public Sub ShowAbout()</pre>	
C#	
<pre>public void ShowAbout()</pre>	

See Also

Reference




- [C1TargetControlHelper<T> Class](#)
- [C1TargetControlHelper<T> Members](#)

Properties

For a list of all members of this type, see [C1TargetControlHelper<T> members](#).

Public Properties

	Name	Description
	Control	

	JsonSerializableHelper	
	Page	
	ScriptManager	

[Top](#)

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1.Web.Wijmo.Controls Namespace](#)

Control Property

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property Control As T	
C#	
public T Control {get;}	

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

JsonSerializableHelper Property

Syntax

Visual Basic (Declaration)	
Public ReadOnly Property JsonSerializableHelper As JsonSerializableHelper	
C#	
public JsonSerializableHelper JsonSerializableHelper {get;}	

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

Page Property

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property Page As System.Web.UI.Page
```

C#

```
public System.Web.UI.Page Page {get;}
```

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

ScriptManager Property

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property ScriptManager As System.Web.UI.ScriptManager
```

C#

```
public System.Web.UI.ScriptManager ScriptManager {get;}
```

See Also

Reference

[C1TargetControlHelper<T> Class](#)

[C1TargetControlHelper<T> Members](#)

WijChartNavigator

Object Model

WijChartNavigator

Syntax

Visual Basic (Declaration)	
<pre><WidgetDependenciesAttribute(System.Object[])> Friend Class WijChartNavigator Inherits WijChartCore</pre>	
C#	
<pre>[WidgetDependencies(System.Object[])] internal class WijChartNavigator : WijChartCore</pre>	

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.WijChartNavigator

See Also

Reference

[WijChartNavigator Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Overview

Object Model

WijChartNavigator

Syntax

Visual Basic (Declaration)	
<pre><WidgetDependenciesAttribute(System.Object[])> Friend Class WijChartNavigator Inherits WijChartCore</pre>	
C#	
<pre>[WidgetDependencies(System.Object[])] internal class WijChartNavigator : WijChartCore</pre>	

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.WijChartNavigator

See Also

Reference


[WijChartNavigator Members](#)

[C1.Web.Wijmo.Controls Namespace](#)

Members

The following tables list the members exposed by [WijChartNavigator](#).

Public Constructors

	Name	Description
	WijChartNavigator Constructor	

[Top](#)

See Also

Reference

[WijChartNavigator Class](#)

[C1.Web.Wijmo.Controls Namespace](#)

WijChartNavigator Constructor

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public WijChartNavigator()	

See Also

Reference

[WijChartNavigator Class](#)





[WijChartNavigator Members](#)

C1.Web.Wijmo.Controls.C1Dialog Namespace

Overview

[Inheritance Hierarchy](#)

Classes

	Class	Description
	C1Dialog	Represents a dialog window.
	DialogButton	Represents a WijDialogButton of the WijDialog.
	DialogCaptionButton	Represents a DialogCaptionButton of the WijDialog.
	DialogCaptionButtons	Represents all DialogCaptionButtons of the WijDialog (include: Pin, Maximize, Minimize, Refresh, Toggle, Close).

See Also

Reference

[C1.Web.Wijmo.Controls.4 Assembly](#)

Classes

C1Dialog

Represents a dialog window.

Object Model

C1Dialog

Syntax

Visual Basic (Declaration)

```
<System.Drawing.ToolboxBitmapAttribute(>>
```

```
<System.Web.UI.ToolboxDataAttribute("<{0}:C1Dialog
runat="server"></{0}:C1Dialog>")>
<System.Web.UI.ParseChildrenAttribute(True)>
<System.ComponentModel.LicenseProviderAttribute()>
<WidgetDependenciesAttribute(System.Object[])>
<System.ComponentModel.DesignerAttribute(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner",
DesignerTypeName="C1.Web.Wijmo.Controls.Design.C1Dialog.C1DialogDesigner,
C1.Web.Wijmo.Controls.Design.4, Version=4.0.20151.207, Culture=neutral,
PublicKeyToken=9b75583953471eea")>
```

Public Class C1Dialog

Inherits [C1.Web.Wijmo.Controls.C1TargetControlBase](#)

C#

```
[System.Drawing.ToolboxBitmap()]
[System.Web.UI.ToolboxData("<{0}:C1Dialog runat="server"></{0}:C1Dialog>")]
[System.Web.UI.ParseChildren(true)]
[System.ComponentModel.LicenseProvider()]
[WidgetDependencies(System.Object[])]
[System.ComponentModel.Designer(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner",
DesignerTypeName="C1.Web.Wijmo.Controls.Design.C1Dialog.C1DialogDesigner,
C1.Web.Wijmo.Controls.Design.4, Version=4.0.20151.207, Culture=neutral,
PublicKeyToken=9b75583953471eea")]
public class C1Dialog : C1.Web.Wijmo.Controls.C1TargetControlBase
```

Inheritance Hierarchy

System.Object

System.Web.UI.Control

System.Web.UI.WebControls.WebControl

[C1.Web.Wijmo.Controls.C1TargetControlBase](#)

C1.Web.Wijmo.Controls.C1Dialog.C1Dialog

See Also

Reference

[C1Dialog Members](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Overview

Represents a dialog window.

Object Model

C1Dialog

Syntax

Visual Basic (Declaration)

```
<System.Drawing.ToolboxBitmapAttribute()>
<System.Web.UI.ToolboxDataAttribute("<{0}:C1Dialog
runat="server"></{0}:C1Dialog">)>
<System.Web.UI.ParseChildrenAttribute(True)>
<System.ComponentModel.LicenseProviderAttribute()>
<WidgetDependenciesAttribute(System.Object[])>
<System.ComponentModel.DesignerAttribute(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner",
DesignerTypeName="C1.Web.Wijmo.Controls.Design.C1Dialog.C1DialogDesigner,
C1.Web.Wijmo.Controls.Design.4, Version=4.0.20151.207, Culture=neutral,
PublicKeyToken=9b75583953471eea">
Public Class C1Dialog
    Inherits C1.Web.Wijmo.Controls.C1TargetControlBase
```

C#

```
[System.Drawing.ToolboxBitmap()]
[System.Web.UI.ToolboxData("<{0}:C1Dialog runat="server"></{0}:C1Dialog">)]
[System.Web.UI.ParseChildren(true)]
[System.ComponentModel.LicenseProvider()]
[WidgetDependencies(System.Object[])]
[System.ComponentModel.Designer(DesignerBaseTypeName="System.ComponentModel.Design.IDesigner",
DesignerTypeName="C1.Web.Wijmo.Controls.Design.C1Dialog.C1DialogDesigner,
C1.Web.Wijmo.Controls.Design.4, Version=4.0.20151.207, Culture=neutral,
PublicKeyToken=9b75583953471eea")]
public class C1Dialog : C1.Web.Wijmo.Controls.C1TargetControlBase
```

Inheritance Hierarchy

```
System.Object
  System.Web.UI.Control
    System.Web.UI.WebControls.WebControl
      C1.Web.Wijmo.Controls.C1TargetControlBase
        C1.Web.Wijmo.Controls.C1Dialog.C1Dialog
```

See Also

Reference


[C1Dialog Members](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Members








The following tables list the members exposed by [C1Dialog](#).













Public Constructors














	Name	Description
	C1Dialog Constructor	Initializes a new instance of the C1Dialog class.













[Top](#)













Public Properties












	Name	Description
	AccessKey	(Inherited from System.Web.UI.WebControls.WebControl)
	AppendTo	A jQuery selector that indicate which element the dialog (and overlay, if modal) should be appended to.
	AppRelativeTemplateSourceDirectory	(Inherited from System.Web.UI.Control)
	Attributes	(Inherited from System.Web.UI.WebControls.WebControl)
	AutoExpand	Auto Expand window.
	Buttons	Gets or sets which buttons should be displayed on the dialog.
	CaptionButtons	Specifies visibility, click event, and icon for the










		caption buttons on the dialog.
	CDNDependencyPaths	A string array value specifies all css/js references that wijmo control depends on. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	CDNPath	Content Delivery Network path. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ClientID	(Inherited from System.Web.UI.Control)
	ClientIDMode	(Inherited from System.Web.UI.Control)
	CloseOnEscape	Specifies whether the dialog should close when it has focus and the user presses the escape (ESC) key.
	CloseText	Specifies the text for the close button. Note that the close text is visibly hidden when using a standard theme.
	CollapsingAnimation	A value determines the settings of the animation effect to be used when the wijdialog is collapsed.
	Content	Gets or sets the template for the content area of the C1Dialog control.
	ContentUrl	A URL string specifies the URL for the iframe element inside wijdialog.
	Controls	(Inherited from System.Web.UI.Control)
	ControlStyle	(Inherited from System.Web.UI.WebControls.WebControl)
	ControlStyleCreated	(Inherited from System.Web.UI.WebControls.WebControl)

	CssClass	(Inherited from System.Web.UI.WebControls.WebControl)
	DisplayVisible	Gets or sets a value that indicates whether a server control is rendered as UI on the page. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	Draggable	If set to true, the dialog will be draggable by the titlebar.
	EnableCombinedJavaScripts	Enable JavaScripts files combined into one file. In order to combined enabled, you must register the WijmoHttpHandler in web.config. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	Enabled	Overridden. Gets or sets if dialog is Enabled.
	EnableTheming	(Inherited from System.Web.UI.WebControls.WebControl)
	EnableViewState	(Inherited from System.Web.UI.Control)
	ExpandingAnimation	A value determines the settings of the animation effect to be used when the wijdialog is expanded.
	HasAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
	Height	Overridden.
	Hide	The effect to be used when the dialog is closed.
	ID	(Inherited from System.Web.UI.Control)
	MaintainStatesOnPostBack	A value indicates whether maintain the states when post back.

	MaxHeight	The maximum height to which the dialog can be resized, in pixels.
	MaxWidth	The maximum width to which the dialog can be resized, in pixels.
	MinHeight	The minimum height to which the dialog can be resized, in pixels.
	MinWidth	The minimum width to which the dialog can be resized, in pixels.
	Modal	If set to true, the dialog will have modal behavior; other items on the page will be disabled (i.e. cannot be interacted with). Modal dialogs create an overlay below the dialog but above other page elements.
	NamingContainer	(Inherited from System.Web.UI.Control)
	OnClientBeforeClose	The name of the function which will be called before close.
	OnClientClose	The name of the function which will be called when the dialog is closed.
	OnClientCreate	The name of the function which will be called when the dialog is created.
	OnClientDrag	The name of the function which will be called when dialog is dragged.
	OnClientDragStart	The name of the function which will be called when dialog begin to be dragged.
	OnClientDragStop	The name of the function which will be called after the dialog has been dragged.



	OnClientFocus	The name of the function which will be called when dialog is focused.
	OnClientOpen	The name of the function which will be called when dialog is opened.
	OnClientResize	The name of the function which will be called when the dialog is resized.
	OnClientResizeStart	The name of the function which will be called when the dialog begin to be resized.
	OnClientResizeStop	The name of the function which will be called after the dialog has been resized.
	Page	(Inherited from System.Web.UI.Control)
	Parent	(Inherited from System.Web.UI.Control)
	Position	Specifies where the dialog should be displayed. Possible values: 1) a single string representing position within viewport: 'center', 'left', 'right', 'top', 'bottom'. 2) a string with array format containing an x,y coordinate pair in pixel offset from left, top corner of viewport (e.g. "[350,100]") 3) a string with array format containing x,y position string values (e.g. "['right','top']" for top right corner).
	RenderingCompatibility	(Inherited from System.Web.UI.Control)
	Resizable	If set to true, the dialog will be resizable.
	Show	The effect to be used when the dialog is opened.
	ShowOnLoad	When autoOpen is true the dialog will open automatically when dialog is called.

	Site	(Inherited from System.Web.UI.Control)
	SkinID	(Inherited from System.Web.UI.WebControls.WebControl)
	Stack	Specifies whether the dialog will stack on top of other dialogs. This will cause the dialog to move to the front of other dialogs when it gains focus.
	Style	(Inherited from System.Web.UI.WebControls.WebControl)
	SupportsDisabledAttribute	(Inherited from System.Web.UI.WebControls.WebControl)
	TabIndex	(Inherited from System.Web.UI.WebControls.WebControl)
	TemplateControl	(Inherited from System.Web.UI.Control)
	TemplateSourceDirectory	(Inherited from System.Web.UI.Control)
	Theme	Name of the theme that will be used to style the widgets. Available embedded themes include: aristo, midnight, and ui-lightness. Note that only one theme can be used for the whole page at one time. Please make sure that all widget extenders have the same Theme value. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ThemeSwatch	A value that indicates the theme swatch of the control, this property only works when WijmoControlMode property is Mobile. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	Title	The Title of the Dialog.







	ToolTip	(Inherited from System.Web.UI.WebControls.WebControl)
	UniqueID	(Inherited from System.Web.UI.Control)
	UseCDN	Determines whether the widget extender must load client resources from CDN (Content Delivery Network) path given by property CDNPath. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ValidateRequestMode	(Inherited from System.Web.UI.Control)
	ViewStateMode	(Inherited from System.Web.UI.Control)
	Width	Overridden.
	WijmoControlMode	A value that indicates mode of the control, whether it is a mobile or web control. Note that only one value can be used for the whole website or project. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	WijmoCssAdapter	Indicates the control applies the theme of JQuery UI or Bootstrap. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ZIndex	The starting z-index for the dialog.

[Top](#)

Public Methods







	Name	Description
	ApplyStyle	(Inherited from System.Web.UI.WebControls.WebControl)
	ApplyStyleSheetSkin	(Inherited from System.Web.UI.Control)

≡	CopyBaseAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
≡	DataBind	(Inherited from System.Web.UI.Control)
≡	Dispose	(Inherited from System.Web.UI.Control)
≡	FindControl	(Inherited from System.Web.UI.Control)
≡	Focus	(Inherited from System.Web.UI.Control)
≡	GetPropertyValues<V>	Gets the property value by property name. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	GetRouteUrl	Overloaded. (Inherited from System.Web.UI.Control)
≡	GetScriptDescriptors	When overridden in a derived class, registers the WidgetDescriptor objects for the control. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	GetScriptReferences	When overridden in a derived class, registers an additional script libraries for the extender control. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	GetUniqueIDRelativeTo	(Inherited from System.Web.UI.Control)
≡	HasControls	(Inherited from System.Web.UI.Control)
≡	LoadLayout	Overloaded. Loads control layout properties from the file.
≡	MergeStyle	(Inherited from System.Web.UI.WebControls.WebControl)
≡	RegisterDesignTimeStyleSheet	Register design time stylesheet (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	RenderBeginTag	(Inherited from System.Web.UI.WebControls.WebControl)
≡	RenderControl	(Inherited from System.Web.UI.Control)

	RenderEndTag	(Inherited from System.Web.UI.WebControls.WebControl)
	ResolveClientUrl	(Inherited from System.Web.UI.Control)
	ResolveUrl	(Inherited from System.Web.UI.Control)
	SaveLayout	Overloaded. Saves the control layout properties to the file.
	SetPropertyValue<V>	Sets the property value by property name. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	SetRenderMethodDelegate	(Inherited from System.Web.UI.Control)

[Top](#)

Public Events

	Name	Description
	DataBinding	(Inherited from System.Web.UI.Control)
	Disposed	(Inherited from System.Web.UI.Control)
	Init	(Inherited from System.Web.UI.Control)
	Load	(Inherited from System.Web.UI.Control)
	PreRender	(Inherited from System.Web.UI.Control)
	Unload	(Inherited from System.Web.UI.Control)

[Top](#)

See Also

Reference

[C1Dialog Class](#)[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

C1Dialog Constructor

Initializes a new instance of the [C1Dialog](#) class.

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public C1Dialog()	

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

[Overload List](#)

Methods

For a list of all members of this type, see [C1Dialog members](#).

Public Methods

	Name	Description
≡	ApplyStyle	(Inherited from System.Web.UI.WebControls.WebControl)
≡	ApplyStyleSheetSkin	(Inherited from System.Web.UI.Control)
≡	CopyBaseAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
≡	DataBind	(Inherited from System.Web.UI.Control)
≡	Dispose	(Inherited from System.Web.UI.Control)
≡	FindControl	(Inherited from System.Web.UI.Control)
≡	Focus	(Inherited from System.Web.UI.Control)
≡	GetPropertyValue<V>	Gets the property value by property name. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)

≡	GetRouteUrl	Overloaded. (Inherited from System.Web.UI.Control)
≡	GetScriptDescriptors	When overridden in a derived class, registers the WidgetDescriptor objects for the control. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	GetScriptReferences	When overridden in a derived class, registers an additional script libraries for the extender control. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	GetUniqueIDRelativeTo	(Inherited from System.Web.UI.Control)
≡	HasControls	(Inherited from System.Web.UI.Control)
≡	LoadLayout	Overloaded. Loads control layout properties from the file.
≡	MergeStyle	(Inherited from System.Web.UI.WebControls.WebControl)
≡	RegisterDesignTimeStyleSheet	Register design time stylesheet (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	RenderBeginTag	(Inherited from System.Web.UI.WebControls.WebControl)
≡	RenderControl	(Inherited from System.Web.UI.Control)
≡	RenderEndTag	(Inherited from System.Web.UI.WebControls.WebControl)
≡	ResolveClientUrl	(Inherited from System.Web.UI.Control)
≡	ResolveUrl	(Inherited from System.Web.UI.Control)
≡	SaveLayout	Overloaded. Saves the control layout properties to the file.
≡	SetPropertyValue<V>	Sets the property value by property name. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
≡	SetRenderMethodDelegate	(Inherited from System.Web.UI.Control)

[Top](#)

See Also

Reference

[C1Dialog Class](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

LoadLayout Method

Loads control layout properties from the file.

Overload List

Overload	Description
LoadLayout(String)	Loads control layout properties from the file.
LoadLayout(Stream)	Load control layout properties from the stream.
LoadLayout(String,LayoutType)	Loads control layout properties with specified types from the file.
LoadLayout(Stream,LayoutType)	Loads the control layout properties with specified types from the stream.

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

LoadLayout(String) Method

Loads control layout properties from the file.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub LoadLayout(_ ByVal <i>filename</i> As System.String _)</pre>	
C#	

```
public void LoadLayout(  
    System.string filename  
)
```

Parameters

filename

The file where the values of layout properties will be loaded.

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

[Overload List](#)

LoadLayout(Stream) Method

Load control layout properties from the stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub LoadLayout( _  
    ByVal stream As System.IO.Stream _  
)
```

C#

```
public void LoadLayout(  
    System.IO.Stream stream  
)
```

Parameters

stream

The stream where the values of layout properties will be loaded.

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

[Overload List](#)

LoadLayout(String,LayoutType) Method

The file where the values of layout properties will be loaded.

The layout types to load.

Loads control layout properties with specified types from the file.

Syntax

Visual Basic (Declaration)	
<pre>Public Overloads Sub LoadLayout(_ ByVal <i>filename</i> As System.String, _ ByVal <i>layoutTypes</i> As LayoutType _)</pre>	
C#	
<pre>public void LoadLayout(System.string <i>filename</i>, LayoutType <i>layoutTypes</i>)</pre>	

Parameters

filename

The file where the values of layout properties will be loaded.

layoutTypes

The layout types to load.

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

[Overload List](#)

LoadLayout(Stream,LayoutType) Method

The stream where the values of the layout properties will be loaded.

The layout types to load.

Loads the control layout properties with specified types from the stream.

Syntax

Visual Basic (Declaration)

```
Public Overloads Sub LoadLayout( _  
    ByVal stream As System.IO.Stream, _  
    ByVal layoutTypes As LayoutType _  
)
```

C#

```
public void LoadLayout(  
    System.IO.Stream stream,  
    LayoutType layoutTypes  
)
```

Parameters

stream

The stream where the values of the layout properties will be loaded.

layoutTypes

The layout types to load.

See Also**Reference**

[C1Dialog Class](#)

[C1Dialog Members](#)

[Overload List](#)

SaveLayout Method

Saves the control layout properties to the file.

Overload List

Overload	Description
SaveLayout(String)	Saves the control layout properties to the file.
SaveLayout(Stream)	Saves control layout properties to the stream.

See Also**Reference**

[C1Dialog Class](#)[C1Dialog Members](#)

SaveLayout(String) Method

Saves the control layout properties to the file.

Syntax

Visual Basic (Declaration)	
Public Overloads Sub SaveLayout(_ ByVal <i>filename</i> As System.String _)	
C#	
public void SaveLayout(System.string <i>filename</i>)	

Parameters

filename

The file where the values of the layout properties will be saved.

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)[Overload List](#)

SaveLayout(Stream) Method

Saves control layout properties to the stream.

Syntax

Visual Basic (Declaration)	
Public Overloads Sub SaveLayout(_ ByVal <i>stream</i> As System.IO.Stream _)	
C#	

```
public void SaveLayout(
    System.IO.Stream stream
)
```

Parameters

stream

The stream where the values of layout properties will be saved.

See Also







Reference














[C1Dialog Class](#)
[C1Dialog Members](#)
[Overload List](#)














Properties













For a list of all members of this type, see [C1Dialog members](#).













Public Properties











	Name	Description
	AccessKey	(Inherited from System.Web.UI.WebControls.WebControl)
	AppendTo	A jQuery selector that indicate which element the dialog (and overlay, if modal) should be appended to.
	AppRelativeTemplateSourceDirectory	(Inherited from System.Web.UI.Control)
	Attributes	(Inherited from System.Web.UI.WebControls.WebControl)
	AutoExpand	Auto Expand window.
	Buttons	Gets or sets which buttons should be displayed on the dialog.











	CaptionButtons	Specifies visibility, click event, and icon for the caption buttons on the dialog.
	CDNDependencyPaths	A string array value specifies all css/js references that wijmo control depends on. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	CDNPath	Content Delivery Network path. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ClientID	(Inherited from System.Web.UI.Control)
	ClientIDMode	(Inherited from System.Web.UI.Control)
	CloseOnEscape	Specifies whether the dialog should close when it has focus and the user presses the escape (ESC) key.
	CloseText	Specifies the text for the close button. Note that the close text is visibly hidden when using a standard theme.
	CollapsingAnimation	A value determines the settings of the animation effect to be used when the wijdialog is collapsed.
	Content	Gets or sets the template for the content area of the C1Dialog control.
	ContentUrl	A URL string specifies the URL for the iframe element inside wijdialog.
	Controls	(Inherited from System.Web.UI.Control)
	ControlStyle	(Inherited from System.Web.UI.WebControls.WebControl)
	ControlStyleCreated	(Inherited from

		System.Web.UI.WebControls.WebControl)
	CssClass	(Inherited from System.Web.UI.WebControls.WebControl)
	DisplayVisible	Gets or sets a value that indicates whether a server control is rendered as UI on the page. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	Draggable	If set to true, the dialog will be draggable by the titlebar.
	EnableCombinedJavaScripts	Enable JavaScripts files combined into one file. In order to combined enabled, you must register the WijmoHttpHandler in web.config. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	Enabled	Overridden. Gets or sets if dialog is Enabled.
	EnableTheming	(Inherited from System.Web.UI.WebControls.WebControl)
	EnableViewState	(Inherited from System.Web.UI.Control)
	ExpandingAnimation	A value determines the settings of the animation effect to be used when the wijdialog is expanded.
	HasAttributes	(Inherited from System.Web.UI.WebControls.WebControl)
	Height	Overridden.
	Hide	The effect to be used when the dialog is closed.
	ID	(Inherited from System.Web.UI.Control)
	MaintainStatesOnPostBack	A value indicates whether maintain the states when

		post back.
	MaxHeight	The maximum height to which the dialog can be resized, in pixels.
	MaxWidth	The maximum width to which the dialog can be resized, in pixels.
	MinHeight	The minimum height to which the dialog can be resized, in pixels.
	MinWidth	The minimum width to which the dialog can be resized, in pixels.
	Modal	If set to true, the dialog will have modal behavior; other items on the page will be disabled (i.e. cannot be interacted with). Modal dialogs create an overlay below the dialog but above other page elements.
	NamingContainer	(Inherited from System.Web.UI.Control)
	OnClientBeforeClose	The name of the function which will be called before close.
	OnClientClose	The name of the function which will be called when the dialog is closed.
	OnClientCreate	The name of the function which will be called when the dialog is created.
	OnClientDrag	The name of the function which will be called when dialog is dragged.
	OnClientDragStart	The name of the function which will be called when dialog begin to be dragged.
	OnClientDragStop	The name of the function which will be called after

		the dialog has been dragged.
	OnClientFocus	The name of the function which will be called when dialog is focused.
	OnClientOpen	The name of the function which will be called when dialog is opened.
	OnClientResize	The name of the function which will be called when the dialog is resized.
	OnClientResizeStart	The name of the function which will be called when the dialog begin to be resized.
	OnClientResizeStop	The name of the function which will be called after the dialog has been resized.
	Page	(Inherited from System.Web.UI.Control)
	Parent	(Inherited from System.Web.UI.Control)
	Position	Specifies where the dialog should be displayed. Possible values: 1) a single string representing position within viewport: 'center', 'left', 'right', 'top', 'bottom'. 2) a string with array format containing an x,y coordinate pair in pixel offset from left, top corner of viewport (e.g. "[350,100]") 3) a string with array format containing x,y position string values (e.g. "['right','top']" for top right corner).
	RenderingCompatibility	(Inherited from System.Web.UI.Control)
	Resizable	If set to true, the dialog will be resizable.
	Show	The effect to be used when the dialog is opened.
	ShowOnLoad	When autoOpen is true the dialog will open

		automatically when dialog is called.
	Site	(Inherited from System.Web.UI.Control)
	SkinID	(Inherited from System.Web.UI.WebControls.WebControl)
	Stack	Specifies whether the dialog will stack on top of other dialogs. This will cause the dialog to move to the front of other dialogs when it gains focus.
	Style	(Inherited from System.Web.UI.WebControls.WebControl)
	SupportsDisabledAttribute	(Inherited from System.Web.UI.WebControls.WebControl)
	TabIndex	(Inherited from System.Web.UI.WebControls.WebControl)
	TemplateControl	(Inherited from System.Web.UI.Control)
	TemplateSourceDirectory	(Inherited from System.Web.UI.Control)
	Theme	Name of the theme that will be used to style the widgets. Available embedded themes include: aristo, midnight, and ui-lightness. Note that only one theme can be used for the whole page at one time. Please make sure that all widget extenders have the same Theme value. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ThemeSwatch	A value that indicates the theme swatch of the control, this property only works when WijmoControlMode property is Mobile. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)

	Title	The Title of the Dialog.
	ToolTip	(Inherited from <code>System.Web.UI.WebControls.WebControl</code>)
	UniqueID	(Inherited from <code>System.Web.UI.Control</code>)
	UseCDN	Determines whether the widget extender must load client resources from CDN (Content Delivery Network) path given by property <code>CDNPath</code> . (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ValidateRequestMode	(Inherited from <code>System.Web.UI.Control</code>)
	ViewStateMode	(Inherited from <code>System.Web.UI.Control</code>)
	Width	Overridden.
	WijmoControlMode	A value that indicates mode of the control, whether it is a mobile or web control. Note that only one value can be used for the whole website or project. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	WijmoCssAdapter	Indicates the control applies the theme of JQuery UI or Bootstrap. (Inherited from C1.Web.Wijmo.Controls.C1TargetControlBase)
	ZIndex	The starting z-index for the dialog.

[Top](#)

See Also

Reference

[C1Dialog Class](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

AppendTo Property

A jQuery selector that indicate which element the dialog (and overlay, if modal) should be appended to.

Syntax

Visual Basic (Declaration)

```
<WidgetOptionAttribute()>
<C1CategoryAttribute("Behavior")>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.AppendTo", Description="A jQuery
selector that indicate which element the dialog (and overlay, if modal)
should be appended to.")>
Public Property AppendTo As System.String
```

C#

```
[WidgetOption()]
[C1Category("Behavior")]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.AppendTo", Description="A jQuery selector that
indicate which element the dialog (and overlay, if modal) should be appended
to.")]
public System.string AppendTo {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

AutoExpand Property

Auto Expand window.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
```

```
<C1DescriptionAttribute(Key="C1Dialog.AutoExpand", Description="A value
indicates whether to set the option height to auto.")>
<WidgetOptionAttribute()>
Public Property AutoExpand As System.Boolean
```

C#

```
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.AutoExpand", Description="A value indicates
whether to set the option height to auto.")]
[WidgetOption()]
public System.bool AutoExpand {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Buttons Property

Gets or sets which buttons should be displayed on the dialog.

Syntax

Visual Basic (Declaration)

```
<WidgetOptionAttribute()>
<CollectionItemTypeAttribute(C1.Web.Wijmo.Controls.C1Dialog.DialogButton)>
<LayoutAttribute(LayoutType.Behavior)>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<System.Web.UI.PersistenceModeAttribute(PersistenceMode.InnerProperty)>
<C1DescriptionAttribute(Key="C1Dialog.Buttons", Description="Gets or sets
which buttons should be displayed on the dialog.")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeria
lizationVisibility.Content)>
<C1CategoryAttribute("Behavior")>
Public ReadOnly Property Buttons As System.Collections.Generic.List(Of
DialogButton)
```

C#

```

[WidgetOption()]
[CollectionItemType(C1.Web.Wijmo.Controls.C1Dialog.DialogButton)]
[Layout(LayoutType.Behavior)]
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[System.Web.UI.Persistencemode(Persistencemode.InnerProperty)]
[C1Description(Key="C1Dialog.Buttons", Description="Gets or sets which
buttons should be displayed on the dialog.")]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
[C1Category("Behavior")]
public System.Collections.Generic.List<DialogButton> Buttons {get;}

```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

CaptionButtons Property

Specifies visibility, click event, and icon for the caption buttons on the dialog.

Syntax

Visual Basic (Declaration)

```

<WidgetOptionAttribute()>
<System.Web.UI.PersistencemodeAttribute(Persistencemode.InnerProperty)>
<C1CategoryAttribute("Behavior")>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeria
lizationVisibility.Content)>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButtons", Description="Gets a
CaptionButtons object that provides access to C1Dialog title items like an
icon, buttons, and so on.")>
Public Property CaptionButtons As DialogCaptionButtons

```

C#

```
[WidgetOption()]
[System.Web.UI.PersistenceMode(PersistenceMode.InnerProperty)]
[C1Category("Behavior")]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
[System.ComponentModel.NotifyParentProperty(true)]
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[C1Description(Key="C1Dialog.CaptionButtons", Description="Gets a
CaptionButtons object that provides access to C1Dialog title items like an
icon, buttons, and so on.")]
public DialogCaptionButtons CaptionButtons {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

CloseOnEscape Property

Specifies whether the dialog should close when it has focus and the user presses the escape (ESC) key.

Syntax

Visual Basic (Declaration)

```
<C1DescriptionAttribute(Key="C1Dialog.CloseOnEscape", Description="Gets or
sets a value indicate whether to close activated dialog on pressing esc
button.")>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<C1CategoryAttribute("Behavior")>
Public Property CloseOnEscape As System.Boolean
```

C#

```
[C1Description(Key="C1Dialog.CloseOnEscape", Description="Gets or sets a
value indicate whether to close activated dialog on pressing esc button.")]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
```

```
[Layout(LayoutType.Behavior)]
[C1Category("Behavior")]
public System.bool CloseOnEscape {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

CloseText Property

Specifies the text for the close button. Note that the close text is visibly hidden when using a standard theme.

Syntax

Visual Basic (Declaration)

```
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Dialog.CloseText", Description="Specifies the
text for the close button.Note that the close text is visibly hidden when
using a standard theme.")>
<C1CategoryAttribute("Behavior")>
<System.ComponentModel.DefaultValueAttribute()>
Public Property CloseText As System.String
```

C#

```
[WidgetOption()]
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.CloseText", Description="Specifies the text for
the close button.Note that the close text is visibly hidden when using a
standard theme.")]
[C1Category("Behavior")]
[System.ComponentModel.DefaultValue()]
public System.string CloseText {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

CollapsingAnimation Property

A value determines the settings of the animation effect to be used when the wijdialog is collapsed.

Syntax

Visual Basic (Declaration)

```
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<C1DescriptionAttribute(Key="C1Dialog.CollapsingAnimation", Description="Gets
or sets the animation effect when the dialog is collapsed.")>
<C1CategoryAttribute("Behavior")>
<System.Web.UI.PersistencemodeAttribute(PersistenceMode.InnerProperty)>
<LayoutAttribute(LayoutType.Behavior)>
<WidgetOptionAttribute()>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Content)>
Public Property CollapsingAnimation As Animation
```

C#

```
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[C1Description(Key="C1Dialog.CollapsingAnimation", Description="Gets or sets
the animation effect when the dialog is collapsed.")]
[C1Category("Behavior")]
[System.Web.UI.Persistencemode(PersistenceMode.InnerProperty)]
[Layout(LayoutType.Behavior)]
[WidgetOption()]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
public Animation CollapsingAnimation {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Content Property

Gets or sets the template for the content area of the [C1Dialog](#) control.

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.BrowsableAttribute(False)>
<C1DescriptionAttribute(Key="C1Dialog.Content", Description="Gets or sets the
template for the content area of the C1Dialog control.")>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Content)>
<System.Web.UI.TemplateInstanceAttribute(TemplateInstance.Single)>
<System.ComponentModel.DefaultValueAttribute()>
<System.Web.UI.PersistenceModeAttribute(PersistenceMode.InnerProperty)>
<LayoutAttribute(LayoutType.Behavior)>
Public Property Content As System.Web.UI.ITemplate
```

C#

```
[System.ComponentModel.Browsable(false)]
[C1Description(Key="C1Dialog.Content", Description="Gets or sets the template
for the content area of the C1Dialog control.")]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
[System.Web.UI.TemplateInstance(TemplateInstance.Single)]
[System.ComponentModel.DefaultValue()]
[System.Web.UI.PersistenceMode(PersistenceMode.InnerProperty)]
[Layout(LayoutType.Behavior)]
public System.Web.UI.ITemplate Content {get; set;}
```

Remarks

Use the `ContentTemplate` property to control the contents of the dialog window. To specify a template for the dialog, reset the [ContentUrl](#) property, place the `<ContentTemplate>` tags between the opening and closing tags of the control. You can then list the contents of the template between the opening and closing `<ContentTemplate>` tags.

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

C1Window

ContentUrl Property

A URL string specifies the URL for the iframe element inside wijdialog.

Syntax

Visual Basic (Declaration)

```
<WidgetOptionAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.ContentUrl", Description="Gets or sets
URL to the content of the dialog window.")>
<LayoutAttribute(LayoutType.Behavior)>
<C1CategoryAttribute("Behavior")>
<System.ComponentModel.DefaultValueAttribute()>
<System.Web.UI.UrlPropertyAttribute()>
Public Property ContentUrl As System.String
```

C#

```
[WidgetOption()]
[C1Description(Key="C1Dialog.ContentUrl", Description="Gets or sets URL to
the content of the dialog window.")]
[Layout(LayoutType.Behavior)]
[C1Category("Behavior")]
[System.ComponentModel.DefaultValue()]
[System.Web.UI.UrlProperty()]
public System.string ContentUrl {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Draggable Property

If set to true, the dialog will be draggable by the titlebar.

Syntax

Visual Basic (Declaration)

```
<C1CategoryAttribute("Behavior")>
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
```



```
<C1DescriptionAttribute(Key="C1Dialog.Draggable", Description="Gets or sets a
value that determines whether the dialog window may be moved.")>
<System.ComponentModel.DefaultValueAttribute()>
Public Property Draggable As System.Boolean
```

C#

```
[C1Category("Behavior")]
[WidgetOption()]
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.Draggable", Description="Gets or sets a value
that determines whether the dialog window may be moved.")]
[System.ComponentModel.DefaultValue()]
public System.bool Draggable {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Enabled Property

Gets or sets if dialog is Enabled.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Base.Enabled", Description="Indicates whether
control is enabled.")>
<C1CategoryAttribute("Behavior")>
Public Overrides Property Enabled As System.Boolean
```

C#

```
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Base.Enabled", Description="Indicates whether control
is enabled.")]
[C1Category("Behavior")]
public override System.bool Enabled {get; set;}
```

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)

ExpandingAnimation Property

A value determines the settings of the animation effect to be used when the wijdialog is expanded.

Syntax

Visual Basic (Declaration)	
<pre> <System.ComponentModel.NotifyParentPropertyAttribute(True)> <C1DescriptionAttribute(Key="C1Dialog.ExpandingAnimation", Description="Gets or sets the animation effect when the dialog is expanded.")> <System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Content)> <System.Web.UI.PersistencemodeAttribute(Persistencemode.InnerProperty)> <TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter, System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")> <WidgetOptionAttribute()> <LayoutAttribute(LayoutType.Behavior)> Public Property ExpandingAnimation As Animation </pre>	
C#	
<pre> [System.ComponentModel.NotifyParentProperty(true)] [C1Description(Key="C1Dialog.ExpandingAnimation", Description="Gets or sets the animation effect when the dialog is expanded.")] [System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationVisibility.Content)] [System.Web.UI.Persistencemode(Persistencemode.InnerProperty)] [TypeConverter("System.ComponentModel.ExpandableObjectConverter, System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")] [WidgetOption()] [Layout(LayoutType.Behavior)] public Animation ExpandingAnimation {get; set;} </pre>	

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)

Height Property

Visual Basic (Declaration)

```
<System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)>
<LayoutAttribute(LayoutType.Sizes)>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
Public Overrides Property Height As System.Web.UI.WebControls.Unit
```

C#

```
[System.ComponentModel.RefreshProperties(RefreshProperties.All)]
[Layout(LayoutType.Sizes)]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
public override System.Web.UI.WebControls.Unit Height {get; set;}
```

Hide Property

The effect to be used when the dialog is closed.

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.Hide", Description="The effect to be
used when the dialog is closesd.")>
<C1CategoryAttribute("Behavior")>
<LayoutAttribute(LayoutType.Behavior)>
Public Property Hide As System.String
```

C#

```
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
[C1Description(Key="C1Dialog.Hide", Description="The effect to be used when
the dialog is closesd.")]
[C1Category("Behavior")]
[Layout(LayoutType.Behavior)]
public System.string Hide {get; set;}
```

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)

MaintainStatesOnPostback Property

A value indicates whether maintain the states when post back.

Syntax

Visual Basic (Declaration)

```

<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute(>
<C1DescriptionAttribute(Key="C1Dialog.MaintainVisibilityOnPostback",
Description="A value indicates whether maintain the open state when post
back.")>
<WidgetOptionAttribute(>
Public Property MaintainStatesOnPostback As System.Boolean

```

C#

```

[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.MaintainVisibilityOnPostback", Description="A
value indicates whether maintain the open state when post back.")]
[WidgetOption()]
public System.bool MaintainStatesOnPostback {get; set;}

```

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)

MaxHeight Property

The maximum height to which the dialog can be resized, in pixels.

Syntax

Visual Basic (Declaration)

```

<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute(>
<C1DescriptionAttribute(Key="C1Dialog.MaxHeight", Description="The maximum
height to which the dialog can be resized, in pixels.")>

```

```
<C1CategoryAttribute("Behavior")>
<WidgetOptionAttribute()>
Public Property MaxHeight As System.Integer
```

C#

```
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.MaxHeight", Description="The maximum height to
which the dialog can be resized, in pixels.")]
[C1Category("Behavior")]
[WidgetOption()]
public System.int MaxHeight {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

MaxWidth Property

The maximum width to which the dialog can be resized, in pixels.

Syntax

Visual Basic (Declaration)

```
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.MaxWidth", Description="The maximum
width to which the dialog can be resized, in pixels.")>
<C1CategoryAttribute("Behavior")>
Public Property MaxWidth As System.Integer
```

C#

```
[WidgetOption()]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.MaxWidth", Description="The maximum width to
which the dialog can be resized, in pixels.")]
[C1Category("Behavior")]
public System.int MaxWidth {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

MinHeight Property

The minimum height to which the dialog can be resized, in pixels.

Syntax

Visual Basic (Declaration)

```
<WidgetOptionAttribute()>
<C1CategoryAttribute("Behavior")>
<C1DescriptionAttribute(Key="C1Dialog.MinHeight", Description="The minimum
height to which the dialog can be resized, in pixels.")>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
Public Property MinHeight As System.Integer
```

C#

```
[WidgetOption()]
[C1Category("Behavior")]
[C1Description(Key="C1Dialog.MinHeight", Description="The minimum height to
which the dialog can be resized, in pixels.")]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
public System.int MinHeight {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

MinWidth Property

The minimum width to which the dialog can be resized, in pixels.

Syntax

Visual Basic (Declaration)

```

<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.MinWidth", Description="The minimum
width to which the dialog can be resized, in pixels.")>
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<C1CategoryAttribute("Behavior")>
Public Property MinWidth As System.Integer

```

C#

```

[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.MinWidth", Description="The minimum width to
which the dialog can be resized, in pixels.")]
[WidgetOption()]
[Layout(LayoutType.Behavior)]
[C1Category("Behavior")]
public System.int MinWidth {get; set;}

```

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)

Modal Property

If set to true, the dialog will have modal behavior; other items on the page will be disabled (i.e. cannot be interacted with). Modal dialogs create an overlay below the dialog but above other page elements.

Syntax

Visual Basic (Declaration)

```

<WidgetOptionAttribute()>
<System.ComponentModel.DefaultValueAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<C1CategoryAttribute("Behavior")>
<C1DescriptionAttribute(Key="C1Dialog.Modal", Description="Gets or sets a
value that determines whether the a modal window is visible on the page after
the page is loaded.")>
Public Property Modal As System.Boolean

```

C#

```
[WidgetOption()]
[System.ComponentModel.DefaultValue()]
[Layout(LayoutType.Behavior)]
[C1Category("Behavior")]
[C1Description(Key="C1Dialog.Modal", Description="Gets or sets a value that
determines whether the a modal window is visible on the page after the page
is loaded.")]
public System.bool Modal {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientBeforeClose Property

The name of the function which will be called before close.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<WidgetOptionNameAttribute("beforeClose")>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.OnClientBeforeClose", Description="Gets
or sets a name of the client side function that should be fired on client
side before dialog is hidden.")>
<C1CategoryAttribute("Client-Side Events")>
<WidgetEventAttribute("ev, index")>
Public Property OnClientBeforeClose As System.String
```

C#

```
[Layout(LayoutType.Behavior)]
[WidgetOptionName("beforeClose")]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.OnClientBeforeClose", Description="Gets or sets
a name of the client side function that should be fired on client side before
dialog is hidden.")]
[C1Category("Client-Side Events")]
```



```
[WidgetEvent("ev, index")]
public System.string OnClientBeforeClose {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientClose Property

The name of the function which will be called when the dialog is closed.

Syntax

Visual Basic (Declaration)

```
<C1CategoryAttribute("Client-Side Events")>
<WidgetEventAttribute("ev, index")>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.OnClientClose", Description="Gets or
sets a name of the client side function that should be fired on client side
after dialog is hidden.")>
<WidgetOptionNameAttribute("close")>
Public Property OnClientClose As System.String
```

C#

```
[C1Category("Client-Side Events")]
[WidgetEvent("ev, index")]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.OnClientClose", Description="Gets or sets a name
of the client side function that should be fired on client side after dialog
is hidden.")]
[WidgetOptionName("close")]
public System.string OnClientClose {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientCreate Property

The name of the function which will be called when the dialog is created.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Dialog.OnClientCreate", Description="Gets or
sets a name of the client side function that should be fired on client side
after dialog is created.")>
<C1CategoryAttribute("Client-Side Events")>
<WidgetEventAttribute("ev, index")>
<WidgetOptionNameAttribute("create")>
<System.ComponentModel.DefaultValueAttribute()>
Public Property OnClientCreate As System.String
```

C#

```
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.OnClientCreate", Description="Gets or sets a
name of the client side function that should be fired on client side after
dialog is created.")]
[C1Category("Client-Side Events")]
[WidgetEvent("ev, index")]
[WidgetOptionName("create")]
[System.ComponentModel.DefaultValue()]
public System.string OnClientCreate {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientDrag Property

The name of the function which will be called when dialog is dragged.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
```

```
<C1DescriptionAttribute(Key="C1Dialog.OnClientDrag", Description="Gets or
sets a name of the client side function that should be fired on client side
when dialog is being dragged.")>
<C1CategoryAttribute("Client-Side Events")>
<WidgetEventAttribute("ev, index")>
<WidgetOptionNameAttribute("drag")>
<System.ComponentModel.DefaultValueAttribute()>
Public Property OnClientDrag As System.String
```

C#

```
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.OnClientDrag", Description="Gets or sets a name
of the client side function that should be fired on client side when dialog
is being dragged.")]
[C1Category("Client-Side Events")]
[WidgetEvent("ev, index")]
[WidgetOptionName("drag")]
[System.ComponentModel.DefaultValue()]
public System.string OnClientDrag {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientDragStart Property

The name of the function which will be called when dialog begin to be dragged.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Dialog.OnClientDragStart", Description="Gets
or sets a name of the client side function that should be fired on client
side when dialog enters dragging mode.")>
<C1CategoryAttribute("Client-Side Events")>
<WidgetEventAttribute("ev, index")>
<WidgetOptionNameAttribute("dragStart")>
<System.ComponentModel.DefaultValueAttribute()>
Public Property OnClientDragStart As System.String
```

C#

```
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.OnClientDragStart", Description="Gets or sets a
name of the client side function that should be fired on client side when
dialog enters dragging mode.")]
[C1Category("Client-Side Events")]
[WidgetEvent("ev, index")]
[WidgetOptionName("dragStart")]
[System.ComponentModel.DefaultValue()]
public System.string OnClientDragStart {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientDragStop Property

The name of the function which will be called after the dialog has been dragged.

Syntax

Visual Basic (Declaration)

```
<WidgetEventAttribute("ev, index")>
<C1CategoryAttribute("Client-Side Events")>
<WidgetOptionNameAttribute("dragStop")>
<System.ComponentModel.DefaultValueAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Dialog.OnClientDragStop", Description="Gets or
sets a name of the client side function that should be fired on client side
when dialog exits dragging mode.")>
Public Property OnClientDragStop As System.String
```

C#

```
[WidgetEvent("ev, index")]
[C1Category("Client-Side Events")]
[WidgetOptionName("dragStop")]
[System.ComponentModel.DefaultValue()]
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.OnClientDragStop", Description="Gets or sets a
```

```
name of the client side function that should be fired on client side when
dialog exits dragging mode."}]
public System.string OnClientDragStop {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientFocus Property

The name of the function which will be called when dialog is focused.

Syntax

Visual Basic (Declaration)

```
<WidgetEventAttribute("ev, index")>
<WidgetOptionNameAttribute("focus")>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.OnClientFocus", Description="Gets or
sets a name of the client side function that should be fired on client side
after dialog is activated.")>
<LayoutAttribute(LayoutType.Behavior)>
<C1CategoryAttribute("Client-Side Events")>
Public Property OnClientFocus As System.String
```

C#

```
[WidgetEvent("ev, index")]
[WidgetOptionName("focus")]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.OnClientFocus", Description="Gets or sets a name
of the client side function that should be fired on client side after dialog
is activated.")]
[Layout(LayoutType.Behavior)]
[C1Category("Client-Side Events")]
public System.string OnClientFocus {get; set;}
```

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)

OnClientOpen Property

The name of the function which will be called when dialog is opened.

Syntax

Visual Basic (Declaration)

```

<LayoutAttribute(LayoutType.Behavior)>
<WidgetEventAttribute("ev, index")>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionNameAttribute("open")>
<C1DescriptionAttribute(Key="C1Dialog.OnClientOpen", Description="Gets or
sets a name of the client side function that should be fired on client side
after dialog is shown.")>
<C1CategoryAttribute("Client-Side Events")>
Public Property OnClientOpen As System.String

```

C#

```

[Layout(LayoutType.Behavior)]
[WidgetEvent("ev, index")]
[System.ComponentModel.DefaultValue()]
[WidgetOptionName("open")]
[C1Description(Key="C1Dialog.OnClientOpen", Description="Gets or sets a name
of the client side function that should be fired on client side after dialog
is shown.")]
[C1Category("Client-Side Events")]
public System.string OnClientOpen {get; set;}

```

See Also

Reference

[C1Dialog Class](#)[C1Dialog Members](#)

OnClientResize Property

The name of the function which will be called when the dialog is resized.

Syntax

Visual Basic (Declaration)

```

<C1DescriptionAttribute(Key="C1Dialog.OnClientResize", Description="Gets or
sets a name of the client side function that should be fired on client side
when dialog is being resized.")>
<LayoutAttribute(LayoutType.Behavior)>
<C1CategoryAttribute("Client-Side Events")>
<WidgetEventAttribute("ev, index")>
<WidgetOptionNameAttribute("resize")>
<System.ComponentModel.DefaultValueAttribute()>
Public Property OnClientResize As System.String

```

C#

```

[C1Description(Key="C1Dialog.OnClientResize", Description="Gets or sets a
name of the client side function that should be fired on client side when
dialog is being resized.")]
[Layout(LayoutType.Behavior)]
[C1Category("Client-Side Events")]
[WidgetEvent("ev, index")]
[WidgetOptionName("resize")]
[System.ComponentModel.DefaultValue()]
public System.string OnClientResize {get; set;}

```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientResizeStart Property

The name of the function which will be called when the dialog begin to be resized.

Syntax

Visual Basic (Declaration)

```

<WidgetOptionNameAttribute("resizeStart")>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.OnClientResizeStart", Description="Gets
or sets a name of the client side function that should be fired on client
side when dialog enters resizing mode.")>

```

```
<C1CategoryAttribute("Client-Side Events")>
<WidgetEventAttribute("ev, index")>
Public Property OnClientResizeStart As System.String
```

C#

```
[WidgetOptionName("resizeStart")]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.OnClientResizeStart", Description="Gets or sets
a name of the client side function that should be fired on client side when
dialog enters resizing mode.")]
[C1Category("Client-Side Events")]
[WidgetEvent("ev, index")]
public System.string OnClientResizeStart {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

OnClientResizeStop Property

The name of the function which will be called after the dialog has been resized.

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DefaultValueAttribute()>
<WidgetEventAttribute("ev, index")>
<WidgetOptionNameAttribute("resizeStop")>
<C1CategoryAttribute("Client-Side Events")>
<C1DescriptionAttribute(Key="C1Dialog.OnClientResizeStop", Description="Gets
or sets a name of the client side function that should be fired on client
side when dialog exits resizing mode.")>
<LayoutAttribute(LayoutType.Behavior)>
Public Property OnClientResizeStop As System.String
```

C#

```
[System.ComponentModel.DefaultValue()]
[WidgetEvent("ev, index")]
[WidgetOptionName("resizeStop")]
```



```
[C1Category("Client-Side Events")]
[C1Description(Key="C1Dialog.OnClientResizeStop", Description="Gets or sets a
name of the client side function that should be fired on client side when
dialog exits resizing mode.")]
[Layout(LayoutType.Behavior)]
public System.string OnClientResizeStop {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Position Property

Specifies where the dialog should be displayed. Possible values: 1) a single string representing position within viewport: 'center', 'left', 'right', 'top', 'bottom'. 2) a string with array format containing an x,y coordinate pair in pixel offset from left, top corner of viewport (e.g. "[350,100]") 3) a string with array format containing x,y position string values (e.g. "[right,top]" for top right corner).

Syntax

Visual Basic (Declaration)

```
<C1DescriptionAttribute(Key="C1Dialog.Position", Description="Specifies where
the dialog should be displayed. Possible values:
1) a single string representing position within viewport: 'center', 'left',
'right', 'top', 'bottom'.
2) a string with array format containing an x,y coordinate pair in pixel
offset from left, top corner of viewport (e.g. "[350,100]")
3) a string with array format containing x,y position string values (e.g.
"[right,top]" for top right corner).")>
<System.ComponentModel.DefaultValueAttribute()>
<LayoutAttribute(LayoutType.Appearance)>
<C1CategoryAttribute("Appearance")>
<WidgetOptionAttribute()>
Public Property Position As System.String
```

C#

```
[C1Description(Key="C1Dialog.Position", Description="Specifies where the
dialog should be displayed. Possible values:
1) a single string representing position within viewport: 'center', 'left',
```

```
'right', 'top', 'bottom'.
2) a string with array format containing an x,y coordinate pair in pixel
offset from left, top corner of viewport (e.g. "[350,100]")
3) a string with array format containing x,y position string values (e.g.
"['right','top']" for top right corner)."]
[System.ComponentModel.DefaultValue()]
[Layout(LayoutType.Appearance)]
[C1Category("Appearance")]
[WidgetOption()]
public System.string Position {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Resizable Property

If set to true, the dialog will be resizable.

Syntax

Visual Basic (Declaration)

```
<C1DescriptionAttribute(Key="C1Dialog.Resizable", Description="Gets or sets a
value that determines whether the dialog window may be resized.")>
<System.ComponentModel.DefaultValueAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<C1CategoryAttribute("Behavior")>
<WidgetOptionAttribute()>
Public Property Resizable As System.Boolean
```

C#

```
[C1Description(Key="C1Dialog.Resizable", Description="Gets or sets a value
that determines whether the dialog window may be resized.")]
[System.ComponentModel.DefaultValue()]
[Layout(LayoutType.Behavior)]
[C1Category("Behavior")]
[WidgetOption()]
public System.bool Resizable {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Show Property

The effect to be used when the dialog is opened.

Syntax

Visual Basic (Declaration)

```
<C1CategoryAttribute("Behavior")>
<C1DescriptionAttribute(Key="C1Dialog.Show", Description="The effect to be
used when the dialog is opened.")>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
Public Property Show As System.String
```

C#

```
[C1Category("Behavior")]
[C1Description(Key="C1Dialog.Show", Description="The effect to be used when
the dialog is opened.")]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
[Layout(LayoutType.Behavior)]
public System.string Show {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

ShowOnLoad Property

When autoOpen is true the dialog will open automatically when dialog is called.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
```

```

<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.ShowOnLoad", Description="Gets or sets
a value that determines whether the dialog window is visible on the page
after the page is loaded.")>
<WidgetOptionNameAttribute("AutoOpen")>
<C1CategoryAttribute("Behavior")>
Public Property ShowOnLoad As System.Boolean

```

C#

```

[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
[C1Description(Key="C1Dialog.ShowOnLoad", Description="Gets or sets a value
that determines whether the dialog window is visible on the page after the
page is loaded.")]
[WidgetOptionName("AutoOpen")]
[C1Category("Behavior")]
public System.bool ShowOnLoad {get; set;}

```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Stack Property

Specifies whether the dialog will stack on top of other dialogs. This will cause the dialog to move to the front of other dialogs when it gains focus.

Syntax

Visual Basic (Declaration)

```

<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.Stack", Description="Specifies whether
the dialog will stack on top of other dialogs. This will cause the dialog to
move to the front of other dialogs when it gains focus.")>
<C1CategoryAttribute("Behavior")>
Public Property Stack As System.Boolean

```

C#

```
[WidgetOption()]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.Stack", Description="Specifies whether the
dialog will stack on top of other dialogs. This will cause the dialog to move
to the front of other dialogs when it gains focus.")]
[C1Category("Behavior")]
public System.bool Stack {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Title Property

The Title of the Dialog.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Dialog.Title", Description="Gets or sets title
which displayed on the caption bar of the C1Dialog window.")>
<C1CategoryAttribute("Behavior")>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
Public Property Title As System.String
```

C#

```
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.Title", Description="Gets or sets title which
displayed on the caption bar of the C1Dialog window.")]
[C1Category("Behavior")]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
public System.string Title {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

Width Property

Visual Basic (Declaration)

```
<System.ComponentModel.RefreshPropertiesAttribute(RefreshProperties.All)>
<LayoutAttribute(LayoutType.Sizes)>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
Public Overrides Property Width As System.Web.UI.WebControls.Unit
```

C#

```
[System.ComponentModel.RefreshProperties(RefreshProperties.All)]
[Layout(LayoutType.Sizes)]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
public override System.Web.UI.WebControls.Unit Width {get; set;}
```

ZIndex Property

The starting z-index for the dialog.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Dialog.ZIndex", Description="The starting z-
index for the dialog.")>
<C1CategoryAttribute("Behavior")>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
Public Property ZIndex As System.Integer
```

C#

```
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.ZIndex", Description="The starting z-index for
the dialog.")]
[C1Category("Behavior")]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
public System.int ZIndex {get; set;}
```

See Also

Reference

[C1Dialog Class](#)

[C1Dialog Members](#)

DialogButton

Represents a WijDialogButton of the WijDialog.

Object Model

DialogButton

Syntax

Visual Basic (Declaration)

```
Public Class DialogButton
    Inherits C1.Web.Wijmo.Controls.Settings
```

C#

```
public class DialogButton : C1.Web.Wijmo.Controls.Settings
```

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1Dialog.DialogButton

See Also

Reference

[DialogButton Members](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Overview

Represents a WijDialogButton of the WijDialog.

Object Model

DialogButton

Syntax

Visual Basic (Declaration)	
<pre>Public Class DialogButton Inherits C1.Web.Wijmo.Controls.Settings</pre>	
C#	
<pre>public class DialogButton : C1.Web.Wijmo.Controls.Settings</pre>	

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1Dialog.DialogButton

See Also

Reference


[DialogButton Members](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Members



The following tables list the members exposed by [DialogButton](#).

Public Constructors

	Name	Description
	DialogButton Constructor	Overloaded.

[Top](#)

Public Properties

	Name	Description
	OnClickClick	Specify a function gets called when you click the button.
	Text	Set a text on the button

[Top](#)

See Also

Reference

[DialogButton Class](#)[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

DialogButton Constructor

Overload List

Overload	Description
DialogButton Constructor(String,String)	Initializes a new instance of the DialogButton class.
DialogButton Constructor()	Initializes a new instance of the DialogButton class.

See Also

Reference

[DialogButton Class](#)[DialogButton Members](#)

DialogButton Constructor(String,String)

Initializes a new instance of the [DialogButton](#) class.

Syntax

Visual Basic (Declaration)	
<pre>Public Function New(_ ByVal text As System.String, _ ByVal function As System.String _)</pre>	
C#	
<pre>public DialogButton(System.string text, System.string function)</pre>	

Parameters

text

function

Specify a function gets called when you click the button.

See Also

Reference

[DialogButton Class](#)

[DialogButton Members](#)

[Overload List](#)

DialogButton Constructor()

Initializes a new instance of the [DialogButton](#) class.

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public DialogButton()	

See Also

Reference

[DialogButton Class](#)



[DialogButton Members](#)

[Overload List](#)

Properties

For a list of all members of this type, see [DialogButton members](#).

Public Properties

	Name	Description
	OnClickClick	Specify a function gets called when you click the button.
	Text	Set a text on the button

[Top](#)

See Also

Reference

[DialogButton Class](#)[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

OnClientClick Property

Specify a function gets called when you click the button.

Syntax

Visual Basic (Declaration)

```

<WidgetOptionNameAttribute("click")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<WidgetEventAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.Button.OnClientClick",
Description="Specify a function gets called when you click the button.")>
Public Property OnClientClick As System.String

```

C#

```

[WidgetOptionName("click")]
[System.ComponentModel.NotifyParentProperty(true)]
[WidgetEvent()]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[C1Description(Key="C1Dialog.Button.OnClientClick", Description="Specify a
function gets called when you click the button.")]
public System.string OnClientClick {get; set;}

```

See Also

Reference

[DialogButton Class](#)[DialogButton Members](#)

Text Property

Set a text on the button

Syntax

Visual Basic (Declaration)

```

<LayoutAttribute(LayoutType.Behavior)>

```

```
<C1DescriptionAttribute(Key="C1Dialog.Button.Text", Description="Set the text
on the button.")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<System.ComponentModel.DefaultValueAttribute(>
<WidgetOptionAttribute(>
Public Property Text As System.String
```

C#

```
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.Button.Text", Description="Set the text on the
button.")]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
public System.string Text {get; set;}
```

See Also

Reference

[DialogButton Class](#)

[DialogButton Members](#)

DialogCaptionButton

Represents a DialogCaptionButton of the WijDialog.

Object Model

DialogCaptionButton

Syntax

Visual Basic (Declaration)

```
Public Class DialogCaptionButton
    Inherits C1.Web.Wijmo.Controls.Settings
```

C#

```
public class DialogCaptionButton : C1.Web.Wijmo.Controls.Settings
```

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1Dialog.DialogCaptionButton

See Also

Reference

[DialogCaptionButton Members](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Overview

Represents a DialogCaptionButton of the WijDialog.

Object Model

DialogCaptionButton

Syntax

Visual Basic (Declaration)

```
Public Class DialogCaptionButton
    Inherits C1.Web.Wijmo.Controls.Settings
```

C#

```
public class DialogCaptionButton : C1.Web.Wijmo.Controls.Settings
```

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1Dialog.DialogCaptionButton

See Also

Reference


[DialogCaptionButton Members](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Members





The following tables list the members exposed by [DialogCaptionButton](#).

Public Constructors

	Name	Description
	DialogCaptionButton Constructor	Overloaded.

[Top](#)

Public Properties

	Name	Description
	IconClassOff	Set the toggle icon of button
	IconClassOn	Set the icon of button
	OnClickClientClick	Specify a function gets called when you click the button.
	Visible	Set visibility of button

[Top](#)

See Also

Reference

[DialogCaptionButton Class](#)[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

DialogCaptionButton Constructor

Overload List

Overload	Description
DialogCaptionButton Constructor()	Initializes a new instance of the DialogCaptionButton class.
DialogCaptionButton Constructor(String,String)	Initializes a new instance of the DialogCaptionButton class.

See Also

Reference

[DialogCaptionButton Class](#)[DialogCaptionButton Members](#)

DialogCaptionButton Constructor()

Initializes a new instance of the [DialogCaptionButton](#) class.

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public DialogCaptionButton()	

See Also

Reference

[DialogCaptionButton Class](#)[DialogCaptionButton Members](#)[Overload List](#)

DialogCaptionButton Constructor(String,String)

Initializes a new instance of the [DialogCaptionButton](#) class.

Syntax

Visual Basic (Declaration)	
Public Function New(_ ByVal <i>iconClassOn</i> As System.String, _ ByVal <i>iconClassOff</i> As System.String _)	
C#	
public DialogCaptionButton(System.string <i>iconClassOn</i> , System.string <i>iconClassOff</i>)	

Parameters

iconClassOn

An css class of the button

iconClassOff

An css class of the toggle icon of button

See Also





Reference

[DialogCaptionButton Class](#)
[DialogCaptionButton Members](#)
[Overload List](#)

Properties

For a list of all members of this type, see [DialogCaptionButton members](#).

Public Properties

	Name	Description
	IconClassOff	Set the toggle icon of button
	IconClassOn	Set the icon of button
	OnClickClientClick	Specify a function gets called when you click the button.
	Visible	Set visibility of button

[Top](#)

See Also

Reference

[DialogCaptionButton Class](#)
[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

IconClassOff Property

Set the toggle icon of button

Syntax

Visual Basic (Declaration)


```

<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.DefaultValueAttribute()>
<WidgetOptionAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.IconClassOff",
Description="Set the toggle icon class of the button.")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
Public Property IconClassOff As System.String

```

C#

```

[Layout(LayoutType.Behavior)]
[System.ComponentModel.DefaultValue()]
[WidgetOption()]
[C1Description(Key="C1Dialog.CaptionButton.IconClassOff", Description="Set
the toggle icon class of the button.")]
[System.ComponentModel.NotifyParentProperty(true)]
public System.string IconClassOff {get; set;}

```

See Also

Reference

[DialogCaptionButton Class](#)

[DialogCaptionButton Members](#)

IconClassOn Property

Set the icon of button

Syntax

Visual Basic (Declaration)

```

<WidgetOptionAttribute()>
<System.ComponentModel.DefaultValueAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.IconClassOn",
Description="Set the icon class of caption button.")>
Public Property IconClassOn As System.String

```

C#

```

[WidgetOption()]
[System.ComponentModel.DefaultValue()]
[Layout(LayoutType.Behavior)]

```

```
[System.ComponentModel.NotifyParentProperty(true)]
[C1Description(Key="C1Dialog.CaptionButton.IconClassOn", Description="Set the
icon class of caption button.")]
public System.string IconClassOn {get; set;}
```

See Also

Reference

[DialogCaptionButton Class](#)

[DialogCaptionButton Members](#)

OnClickClient Property

Specify a function gets called when you click the button.

Syntax

Visual Basic (Declaration)

```
<LayoutAttribute(LayoutType.Behavior)>
<WidgetEventAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.OnClientClick",
Description="Specify a function gets called when you click the button.")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<System.ComponentModel.DefaultValueAttribute()>
Public Property OnClientClick As System.String
```

C#

```
[Layout(LayoutType.Behavior)]
[WidgetEvent()]
[C1Description(Key="C1Dialog.CaptionButton.OnClientClick",
Description="Specify a function gets called when you click the button.")]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DefaultValue()]
public System.string OnClientClick {get; set;}
```

See Also

Reference

[DialogCaptionButton Class](#)

[DialogCaptionButton Members](#)

Visible Property

Set visibility of button

Syntax

Visual Basic (Declaration)

```
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.Visible",
Description="Sets the visibility of the caption button.")>
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<System.ComponentModel.DefaultValueAttribute()>
Public Property Visible As System.Boolean
```

C#

```
[C1Description(Key="C1Dialog.CaptionButton.Visible", Description="Sets the
visibility of the caption button.")]
[WidgetOption()]
[Layout(LayoutType.Behavior)]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DefaultValue()]
public System.bool Visible {get; set;}
```

See Also

Reference

[DialogCaptionButton Class](#)

[DialogCaptionButton Members](#)

DialogCaptionButtons

Represents all DialogCaptionButtons of the WijDialog (include: Pin, Maximize, Minimize, Refresh, Toggle, Close).

Object Model

DialogCaptionButtons

Syntax

Visual Basic (Declaration)

```
Public Class DialogCaptionButtons
    Inherits C1.Web.Wijmo.Controls.Settings
```

C#

```
public class DialogCaptionButtons : C1.Web.Wijmo.Controls.Settings
```

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1Dialog.DialogCaptionButtons

See Also

Reference

[DialogCaptionButtons Members](#)[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Overview

Represents all DialogCaptionButtons of the WijDialog (include: Pin, Maximize, Minimize, Refresh, Toggle, Close).

Object Model

DialogCaptionButtons

Syntax

Visual Basic (Declaration)

```
Public Class DialogCaptionButtons  
    Inherits C1.Web.Wijmo.Controls.Settings
```

C#

```
public class DialogCaptionButtons : C1.Web.Wijmo.Controls.Settings
```

Inheritance Hierarchy

System.Object

C1.Web.Wijmo.Controls.C1Dialog.DialogCaptionButtons

See Also


Reference

[DialogCaptionButtons Members](#)[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Members







The following tables list the members exposed by [DialogCaptionButtons](#).

Public Constructors

	Name	Description
	DialogCaptionButtons Constructor	Initializes a new instance of the DialogCaptionButtons class.

[Top](#)

Public Properties

	Name	Description
	Close	Set close button on the title of dialog
	Maximize	Set maximize button on the title of dialog
	Minimize	Set minimize button on the title of dialog
	Pin	Set pin button on the title of dialog
	Refresh	Set refresh button on the title of dialog
	Toggle	Set toggle button on the title of dialog

[Top](#)

See Also

Reference

[DialogCaptionButtons Class](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

DialogCaptionButtons Constructor

Initializes a new instance of the [DialogCaptionButtons](#) class.

Syntax

Visual Basic (Declaration)	
Public Function New()	
C#	
public DialogCaptionButtons()	

See Also

Reference







[DialogCaptionButtons Class](#)

[DialogCaptionButtons Members](#)

Properties

For a list of all members of this type, see [DialogCaptionButtons members](#).

Public Properties

	Name	Description
	Close	Set close button on the title of dialog
	Maximize	Set maximize button on the title of dialog
	Minimize	Set minimize button on the title of dialog
	Pin	Set pin button on the title of dialog
	Refresh	Set refresh button on the title of dialog
	Toggle	Set toggle button on the title of dialog

[Top](#)

See Also

Reference

[DialogCaptionButtons Class](#)

[C1.Web.Wijmo.Controls.C1Dialog Namespace](#)

Close Property

Set close button on the title of dialog

Syntax

Visual Basic (Declaration)

```
<WidgetOptionAttribute()>
<LayoutAttribute(LayoutType.Behavior)>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.Close",
Description="Button that closes the C1Dialog.")>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Content)>
<System.Web.UI.PersistencemodeAttribute(Persistencemode.InnerProperty)>
Public Property Close As DialogCaptionButton
```

C#

```
[WidgetOption()]
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.CaptionButton.Close", Description="Button that
closes the C1Dialog.")]
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
[System.Web.UI.Persistencemode(Persistencemode.InnerProperty)]
public DialogCaptionButton Close {get; set;}
```

See Also

Reference

[DialogCaptionButtons Class](#)

[DialogCaptionButtons Members](#)

Maximize Property

Set maximize button on the title of dialog

Syntax

Visual Basic (Declaration)

```

<System.Web.UI.PersistencemodeAttribute(Persistencemode.InnerProperty)>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.Maximize",
Description="Button that resizes the C1Window to fit all available space in
the browser window.")>
<LayoutAttribute(LayoutType.Behavior)>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Content)>
<WidgetOptionAttribute()>
Public Property Maximize As DialogCaptionButton

```

C#

```

[System.Web.UI.Persistencemode(Persistencemode.InnerProperty)]
[C1Description(Key="C1Dialog.CaptionButton.Maximize", Description="Button
that resizes the C1Window to fit all available space in the browser
window.")]
[Layout(LayoutType.Behavior)]
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
[WidgetOption()]
public DialogCaptionButton Maximize {get; set;}

```

See Also

Reference

[DialogCaptionButtons Class](#)

[DialogCaptionButtons Members](#)

Minimize Property

Set minimize button on the title of dialog

Syntax

Visual Basic (Declaration)


```

<WidgetOptionAttribute(>>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<LayoutAttribute(LayoutType.Behavior)>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.Minimize",
Description="Minimizing and normalizing caption button.")>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeriali
zationVisibility.Content)>
<System.Web.UI.PersistenceModeAttribute(PersistenceMode.InnerProperty)>
Public Property Minimize As DialogCaptionButton

```

C#

```

[WidgetOption()]
[System.ComponentModel.NotifyParentProperty(true)]
[Layout(LayoutType.Behavior)]
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[C1Description(Key="C1Dialog.CaptionButton.Minimize", Description="Minimizing
and normalizing caption button.")]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
[System.Web.UI.PersistenceMode(PersistenceMode.InnerProperty)]
public DialogCaptionButton Minimize {get; set;}

```

See Also

Reference

[DialogCaptionButtons Class](#)

[DialogCaptionButtons Members](#)

Pin Property

Set pin button on the title of dialog

Syntax

Visual Basic (Declaration)

```

<LayoutAttribute(LayoutType.Behavior)>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>

```

```
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Content)>
<System.Web.UI.PersistencemodeAttribute(Persistencemode.InnerProperty)>
<WidgetOptionAttribute()>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.PinButton",
Description="Button that pins the dialog to it's current location.")>
Public Property Pin As DialogCaptionButton
```

C#

```
[Layout(LayoutType.Behavior)]
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[System.ComponentModel.NotifyParentProperty(true)]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
[System.Web.UI.Persistencemode(Persistencemode.InnerProperty)]
[WidgetOption()]
[C1Description(Key="C1Dialog.CaptionButton.PinButton", Description="Button
that pins the dialog to it's current location.")]
public DialogCaptionButton Pin {get; set;}
```

See Also

Reference

[DialogCaptionButtons Class](#)

[DialogCaptionButtons Members](#)

Refresh Property

Set refresh button on the title of dialog

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<WidgetOptionAttribute()>
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Content)>
<System.Web.UI.PersistencemodeAttribute(Persistencemode.InnerProperty)>
<LayoutAttribute(LayoutType.Behavior)>
```

```
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.Refresh",
Description="Button that reloads the C1Dialog content set by the
C1Dialog.ContentUrl property.")>
```

Public Property Refresh As [DialogCaptionButton](#)

C#

```
[System.ComponentModel.NotifyParentProperty(true)]
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]
[WidgetOption()]
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV
isibility.Content)]
[System.Web.UI.PersistenceMode(PersistenceMode.InnerProperty)]
[Layout(LayoutType.Behavior)]
[C1Description(Key="C1Dialog.CaptionButton.Refresh", Description="Button that
reloads the C1Dialog content set by the C1Dialog.ContentUrl property.")]
public DialogCaptionButton Refresh {get; set;}
```

See Also

Reference

[DialogCaptionButtons Class](#)

[DialogCaptionButtons Members](#)

Toggle Property

Set toggle button on the title of dialog

Syntax

Visual Basic (Declaration)

```
<System.ComponentModel.DesignerSerializationVisibilityAttribute(DesignerSeria
lizationVisibility.Content)>
<C1DescriptionAttribute(Key="C1Dialog.CaptionButton.Toggle",
Description="Button that collapse or expand the dialog.")>
<TypeConverterAttribute("System.ComponentModel.ExpandableObjectConverter,
System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")>
<LayoutAttribute(LayoutType.Behavior)>
<System.ComponentModel.NotifyParentPropertyAttribute(True)>
<WidgetOptionAttribute()>
<System.Web.UI.PersistenceModeAttribute(PersistenceMode.InnerProperty)>
Public Property Toggle As DialogCaptionButton
```

C#

```
[System.ComponentModel.DesignerSerializationVisibility(DesignerSerializationV  
isibility.Content)]  
[C1Description(Key="C1Dialog.CaptionButton.Toggle", Description="Button that  
collapse or expand the dialog.")]  
[TypeConverter("System.ComponentModel.ExpandableObjectConverter, System,  
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")]  
[Layout(LayoutType.Behavior)]  
[System.ComponentModel.NotifyParentProperty(true)]  
[WidgetOption()]  
[System.Web.UI.PersistenceMode(PersistenceMode.InnerProperty)]  
public DialogCaptionButton Toggle {get; set;}
```

See Also

Reference

[DialogCaptionButtons Class](#)

[DialogCaptionButtons Members](#)