

---

ComponentOne

# HtmlHost for Silverlight

Copyright © 1987-2011 ComponentOne LLC. All rights reserved.

*Corporate Headquarters*

**ComponentOne LLC**

201 South Highland Avenue

3<sup>rd</sup> Floor

Pittsburgh, PA 15206 • USA

**Internet:** [info@ComponentOne.com](mailto:info@ComponentOne.com)

**Web site:** <http://www.componentone.com>

**Sales**

E-mail: [sales@componentone.com](mailto: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 ComponentOne LLC. All other trademarks used herein are the properties of their respective owners.

**Warranty**

ComponentOne warrants that the original CD (or diskettes) are 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 CD (or disk) 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 a defective CD (or disk) by sending it and a check for \$25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original CD (or disks) 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. We are 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.

This manual was produced using ComponentOne Doc-To-Help™.

# Table of Contents

HtmlHost.....	1
HtmlHost for Silverlight Key Features.....	1
HtmlHost for Silverlight Quick Start .....	1
Step 1 of 3: Creating a Silverlight Application .....	2
Step 2 of 3: Adding Code to the Application .....	3
Step 3 of 3: Running the Application .....	4
Working with HtmlHost for Silverlight .....	6
Showing HTML Content .....	6
Populating C1HtmlHost.....	6
Windowless Mode.....	6
Frame Borders.....	7
Basic Properties .....	8
Basic Events.....	8
Expander for Silverlight Appearance Properties .....	8
HtmlHost for Silverlight Samples .....	9
HtmlHost for Silverlight Task-Based Help.....	9
Displaying an External Web Site .....	10
Displaying HTML Markup.....	10
Hiding Frame Borders.....	11



# HtmlHost

Render HTML and arbitrary URI content from within Silverlight using **ComponentOne HtmlHost™ for Silverlight**. The HTML host control (**C1HtmlHost**) provides a frame that can host arbitrary HTML content, and display content from arbitrary URIs or HTML text.



## Getting Started

Get started with the following topics:

- [Key Features](#) (page 1)
- [Quick Start](#) (page 1)
- [Task-Based Help](#) (page 9)

## HtmlHost for Silverlight Key Features

**ComponentOne HtmlHost for Silverlight** includes several key features, such as:

- **Inside Browser Support**  
You can see HTML content inside your browser.
- **Display HTML Content**  
Display existing HTML content within the Silverlight plug-in using the browser itself to provide accurate rendering and interaction.
- **Load HTML from URI**  
**HtmlHost for Silverlight** can load HTML content from any URI available, not just the application server.
- **Load HTML from Text**  
Display HTML content contained in a string.
- **Access the HTML Content**  
**HtmlHost for Silverlight** fires an event when the content is fully loaded and allows the developer to access the content through Silverlight's browser object model.
- **Silverlight Layout System Support**  
**HtmlHost for Silverlight** arranges its content following Silverlight's layout for the control.
- **Silverlight Toolkit Themes Support**  
Add style to your UI with built-in support for the most popular Microsoft Silverlight Toolkit themes, including ExpressionDark, ExpressionLight, WhistlerBlue, RainerOrange, ShinyBlue, and BureauBlack.

## HtmlHost for Silverlight Quick Start

The following quick start guide is intended to get you up and running with **ComponentOne HtmlHost for Silverlight**. In this quick start you'll create a Silverlight application that allows you to view a Web site in a Silverlight application using the **C1HtmlHost** control. You'll create a new project in Visual Studio, add and customize controls, and view the run-time interactions possible with the **C1HtmlHost** control.

## Step 1 of 3: Creating a Silverlight Application

In this step you'll create a Silverlight application in Visual Studio which will use **ComponentOne HtmlHost for Silverlight** to display a Web site. You'll create a new Silverlight project and add controls to your application.

To set up and add controls to your application, complete the following steps:

1. In Visual Studio, select **File | New | Project**.
2. In the **New Project** dialog box, select a language in the left pane, and in the templates list select **Silverlight Application**. Enter a **Name** for your project and click **OK**. The **New Silverlight Application** dialog box will appear.
3. Click **OK** to accept default settings, close the **New Silverlight Application** dialog box, and create your project. The **MainPage.xaml** file should open.
4. In the XAML window of the project, place the cursor between the `<Grid>` and `</Grid>` tags and click once.
5. Navigate to the Visual Studio Toolbox and double-click the **StackPanel** icon to add the panel to the page.
6. Update the `<StackPanel/>` tag in the **MainPage.xaml** file so it appears like the following:

```
<StackPanel Name="StackPanel1" Margin="10"
Orientation="Vertical"></StackPanel>
```

This markup adds a margin and sets the panel's orientation.

7. In the XAML window of the project, place the cursor between the `<StackPanel>` and `</StackPanel>` tags and click once – you will add controls to the **StackPanel**.
8. Navigate to the Visual Studio Toolbox and double-click the **C1HtmlHost** icon to add the control to the panel.
9. Update the `<c1ext:C1HtmlHost/>` tag in the **MainPage.xaml** file so it appears like the following:

```
<c1ext:C1HtmlHost Name="C1HtmlHost1" Margin="5"
SourceUri="http://www.componentone.com" />
```

This markup names the control, adds a margin, and sets the `SourceUri` property which sets the Web site or page that will initially appear in the `C1HtmlHost` control when the application is loaded.

10. Add the following markup below the `<c1ext:C1HtmlHost/>` tag in XAML view:

```
<StackPanel Name="StackPanel2" Orientation="Horizontal">
  <TextBlock Height="23" HorizontalAlignment="Left" Margin="5"
Name="TextBlock1" Text="Source URI:" VerticalAlignment="Top" />
  <TextBox Height="23" Name="TextBox1" Width="200" Margin="5"
Text="http://www.componentone.com" />
  <Button Content="Set Source URI" Height="23" Name="Button1"
Margin="5" />
</StackPanel>
```

The markup above adds a **StackPanel** that contains a **TextBlock**, **TextBox**, and **Button**. When the user enters a URL in the text box at run time and presses the button, the `C1HtmlHost` control will display the entered Web site. Your markup will now appear similar to the following:

```
<Grid x:Name="LayoutRoot" Background="White">
  <StackPanel Margin="10" Name="StackPanel1" Orientation="Vertical">
    <c1ext:C1HtmlHost Name="C1HtmlHost1" Margin="5"
SourceUri="http://www.componentone.com" c1:C1NagScreen.Nag="True" />
    <StackPanel Name="StackPanel2" Orientation="Horizontal">
      <TextBlock Height="23" HorizontalAlignment="Left"
Margin="5" Name="TextBlock1" Text="Source URI:" VerticalAlignment="Top"
/>
      <TextBox Height="23" Name="TextBox1" Width="200" Margin="5"
Text="http://www.componentone.com" />
    </StackPanel>
  </StackPanel>
</Grid>
```

```

        <Button Content="Set Source URI" Height="23" Name="Button1"
Margin="5" />
    </StackPanel>
</StackPanel>
</Grid>

```

11. Navigate to the Solution Explorer, and expand the **YourProject.Web** node (where *YourProject* is the name of the project) and double-click the **YourProjectTestPage.aspx** file (where *YourProject* is the name of the project) to open it.
12. In the .aspx file scroll down to the `<div id="silverlightControlHost">` tag and add the following parameter in the list of parameters between the `<object></object>` tags:

```

<param name="windowless" value="true" />

```
13. Save all your changes and return to the **MainPage.xaml** page.

### ✔ What You've Accomplished

You've successfully created and set up a Silverlight application and added controls, including a `C1HtmlHost` control, to the page. In the next step you'll add code to add functionality to your application.

## Step 2 of 3: Adding Code to the Application

In the last step you set up a Silverlight application, but if you run your application the button and text box currently do nothing. In this step you'll continue by adding code to add functionality to the application.

Complete the following steps:

1. Navigate to the Solution Explorer, right-click **MainPage.xaml** file, and select **View Code** to switch to Code view.
2. In Code view, add the following import statement to the top of the code page:
  - Visual Basic

```
Imports C1.Silverlight.Extended
```
  - C#

```
using C1.Silverlight.Extended;
```
3. Add the following event handler to the **MainPage.xaml.cs** file, below all the other methods in the **MainPage** class:

- Visual Basic

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles Button1.Click
    Me.C1HtmlHost1.SourceUri = New Uri(String.Format(TextBox1.Text))
End Sub

```

- C#

```

private void Button1_Click(object sender, RoutedEventArgs e)
{
    C1HtmlHost1.SourceUri = new Uri(string.Format(TextBox1.Text));
}

```

This code handles the button's **Click** event and customizes the `C1HtmlHost` control.

### ✔ What You've Accomplished

In this step you added code to add functionality to your application – now when a user enters a URL in the text box and clicks the button at run time, the `C1HtmlHost` control will display the selected Web site. In the next step you'll run your application and observe some of the run-time interactions possible with **ComponentOne HtmlHost for Silverlight**.

### Step 3 of 3: Running the Application

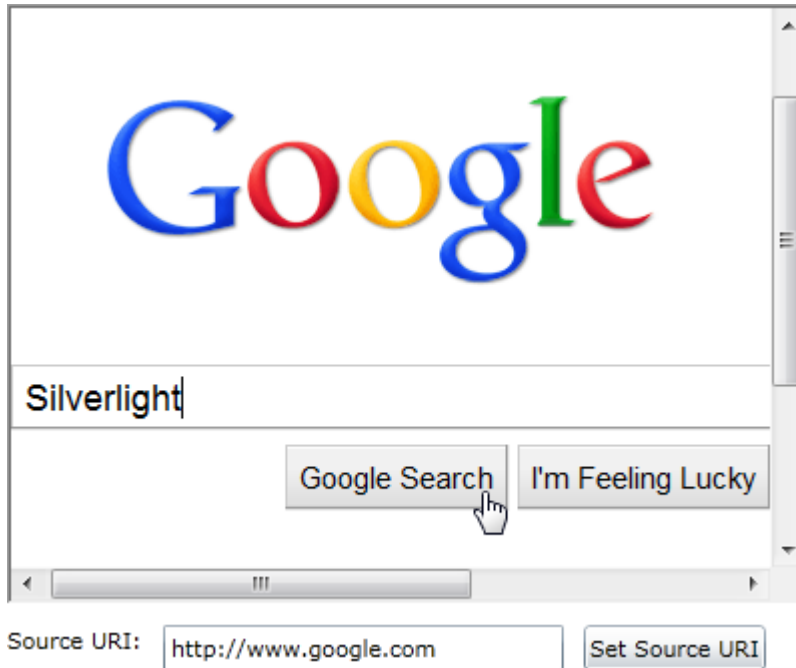
Now that you've created a Silverlight application and, set up the application, and added code to add functionality to the application, the only thing left to do is run your application. To observe your application's run-time interactions, complete the following steps:

1. Choose **Debug | Start Debugging** from the menu to run your application. The application will appear similar to the following image:



The application appears with the ComponentOne Web site loaded in the **C1HtmlHost** control.

2. Enter <http://www.google.com> in the text box and click the **Set Source URI** button. The **C1HtmlHost** control will display's Google's Web site.
3. Enter "Silverlight" in the text box and click the **Google Search** button:



A page will appear displaying search results.

4. Click a search result link. Notice that the control displays the page you choose:



5. Continue to click links and surf the Internet through the **C1HtmlControl** – notice that the page it displays functions as a page in a Web browser would.

### ✔ What You've Accomplished

Congratulations, you've completed the **HtmlHost for Silverlight** quick start! You've created a simple application that uses **HtmlHost for Silverlight** to display a selected Web site.

To learn more about the features and functionality of **ComponentOne HtmlHost for Silverlight**, see the [Working with HtmlHost for Silverlight](#) (page 6) topic. For examples of specific customizations, see the [HtmlHost for Silverlight Task-Based Help](#) (page 9) topic.

## Working with HtmlHost for Silverlight

**ComponentOne HtmlHost™ for Silverlight** provides a simple and reliable way to display a Website or HTML content within a Silverlight application. The following topics detail information useful in working with **HtmlHost for Silverlight**.

### Showing HTML Content

The **C1HtmlHost** control behaves like an HTML iFrame tag, and allows you to show HTML content within your Silverlight pages. This is a reversal of the typical scenario where Silverlight plug-ins are islands in the HTML that makes up the page. The **C1HtmlHost** control allows you to create HTML islands in your Silverlight pages.

This concept is useful because in many cases your Silverlight applications may want to display existing HTML content within the Silverlight plug-in itself (as opposed to controlling elements in the host page's DOM, which Silverlight also allows).

**ComponentOne HtmlHost for Silverlight** works using an iFrame. The C1HtmlHost control adds an iFrame through JavaScript and locates it over the Silverlight plugin in the exact location where the C1HtmlHost control is located. Then it listens to changes in the layout and updates the iFrame continually.

Note that some pages that you may want to display may not be viewable in the C1HtmlHost control. For example, the page you wish to be display may be customized so that is not displayable with in an iFrame object. At its essence the C1HtmlHost control is uses an iFrame object located on top of the Silverlight plugin, so this behavior is expected.

### Populating C1HtmlHost

The **C1HtmlHost** can be populated in two ways:

- Use the **SourceHtml** property to specify an HTML string that should be displayed in the control. This option is useful if your application builds or loads the HTML content.
- Use the **SourceUri** property to specify the URL that will be displayed in the control. This option is useful if your application needs to display content that is already available at a given URL.

The **C1HtmlHost** control has an important requirement. In order to display the HTML content within the Silverlight plug-in, the plug-in must have its **Windowless** property set to **True**. If you run the application now, you will see an error message that notes:

**Note:** To use this control the Silverlight plug-in must have its **Windowless** parameter set to **True**.

The control cannot modify the properties of the plug-in, the page author must do that instead. To make the necessary change, open the page that creates the plug-in and add a line to set the **Windowless** property to **True**. For details see [Windowless Mode](#) (page 6).

### Windowless Mode

**ComponentOne HtmlHost for Silverlight** requires that the Silverlight plug-in's **Windowless** parameter be set to **True**. In windowless mode, the Silverlight plug-in does not have its own rendering window. Instead, the plug-in content is displayed directly by the browser window. This enables Silverlight content to visually overlap and blend with HTML content if the plug-in and its content both specify background transparency.

To set the **Windowless** parameter, you would complete the following steps:

1. Create a Silverlight application that includes a **C1HtmlHost** control.
2. Navigate to the Solution Explorer, and expand the **YourProject.Web** node (where *YourProject* is the name of the project).
3. In the Solution Explorer, double-click the page in which the Silverlight plugin is declared. For example, double-click **YourProjectTestPage.aspx** or **YourProjectTestPage.html** file (where *YourProject* is the name of the project) to open the page.
4. In the page scroll down to the `<div id="silverlightControlHost">` tag and add the following parameter in the list of parameters between the `<object></object>` tags:  
`<param name="windowless" value="true" />`
5. Save your change and return to the **MainPage.xaml** page.

When the **Windowless** parameter is not set to **True**, you may receive a warning when running the application.

## Frame Borders

The C1HtmlHost control is displayed within a frame by default. By default, borders are displayed around the C1HtmlHost control. For example, in the following image a frame appears in the top and left borders of the C1HtmlHost control:



You may want to hide this frame border – for example, if you want the control to blend more seamlessly with your application. When the **FrameBorder** property is set to **False**, a frame will not appear around the C1HtmlHost control. For example, in the image below the frame is not visible:



You'll notice that when the frame is not visible, the C1HtmlHost control blends into the background of the application more. For an example, see [Hiding Frame Borders](#) (page 11).

## Basic Properties

**ComponentOne HtmlHost for Silverlight** includes several properties that allow you to set the functionality of the C1HtmlHost control. Some of the more important properties are listed below.

The following properties let you customize the C1HtmlHost control:

Property	Description
AllowTransparency	Gets or sets a value indicating whether transparency is allowed (only required for IE).
FrameBorder	Gets or sets whether the default browser frame border is displayed when using <a href="#">SourceUri</a> .
HtmlElement	Gets the <b>HtmlElement</b> hosted by this <a href="#">C1HtmlHost</a> control.
SourceHtml	Gets or set the HTML content as a string.
SourceUri	Gets or set a URI that will provide the content for the control.

## Basic Events

**ComponentOne HtmlHost for Silverlight** includes several events that allow you to set interaction and customize the control. Some of the more important events are listed below.

The following events let you customize the C1HtmlHost control:

Event	Description
UriLoaded	Fired when the content from a URI has been loaded.

## Expander for Silverlight Appearance Properties

**ComponentOne HtmlHost for Silverlight** includes several properties that allow you to customize the appearance of the control. You can change the color, border, and height of the control. The following topics describe some of these appearance properties.

### *Color Properties*

The following properties let you customize the colors used in the control itself:

Property	Description
<a href="#">Background</a>	Gets or sets a brush that describes the background of a control. This is a dependency property.

### *Alignment Properties*

The following properties let you customize the control's alignment:

Property	Description
<a href="#">HorizontalAlignment</a>	Gets or sets the horizontal alignment characteristics applied to this element when it is composed within a parent element, such as a panel or items control. This is a dependency property.
<a href="#">VerticalAlignment</a>	Gets or sets the vertical alignment characteristics

	applied to this element when it is composed within a parent element such as a panel or items control. This is a dependency property.
--	--

## Size Properties

The following properties let you customize the size of the **C1HtmlHost** control:

Property	Description
<a href="#">Height</a>	Gets or sets the suggested height of the element. This is a dependency property.
<a href="#">MaxHeight</a>	Gets or sets the maximum height constraint of the element. This is a dependency property.
<a href="#">MaxWidth</a>	Gets or sets the maximum width constraint of the element. This is a dependency property.
<a href="#">MinHeight</a>	Gets or sets the minimum height constraint of the element. This is a dependency property.
<a href="#">MinWidth</a>	Gets or sets the minimum width constraint of the element. This is a dependency property.
<a href="#">Width</a>	Gets or sets the width of the element. This is a dependency property.

## HtmlHost Styles

**ComponentOne HtmlHost for Silverlight's C1HtmlHost** control provides several style properties that you can use to change the appearance of the control. Some of the included styles are described in the table below:

Style	Description
<a href="#">Style</a>	Gets or sets the style used by this element when it is rendered. This is a dependency property.

## HtmlHost for Silverlight Samples

**ComponentOne HtmlHost for Silverlight** includes C# samples. By default samples are installed in the **Documents** or **My Documents** folder in the **ComponentOne Samples\Studio for Silverlight** folder.

The following sample is included:

Sample	Description
ControlExplorer	The <b>ControlExplorer</b> includes a page that demonstrates using the <b>C1HtmlHost</b> control to display an external Web site. This sample and is installed by default in the <b>ControlExplorer</b> folder in the samples directory.

## HtmlHost for Silverlight Task-Based Help

The following task-based help topics assume that you are familiar with Visual Studio and Expression Blend and know how to use the C1HtmlHost control in general. If you are unfamiliar with the **ComponentOne HtmlHost for Silverlight** product, please see the [HtmlHost for Silverlight Quick Start](#) (page 1) first.

Each topic in this section provides a solution for specific tasks using the **ComponentOne HtmlHost for Silverlight** product. Most task-based help topics also assume that you have created a new Silverlight project and added the C1HtmlHost control to the application.

## Displaying an External Web Site

Displaying an external Web site in **ComponentOne HtmlHost for Silverlight** is as simple as setting a single property. The `SourceUri` property determines the URI of the site displayed in the control. You can set the `SourceUri` property to any Web site – even an external site – to display.

### At Design Time

To set the `SourceUri` property in Expression Blend, complete the following steps:

1. Click the `C1HtmlHost` control once to select it.
2. Navigate to the Properties window and locate the **SourceUri** item.
3. Click in the text box next to the **SourceUri** item and enter "http://www.componentone.com".

### In XAML

For example, to set the `SourceUri` property add `SourceUri=""` to the `<c1:C1HtmlHost>` tag so that it appears similar to the following:

```
<c1ext:C1HtmlHost Name="c1HtmlHost1" SourceUri="http://www.componentone.com" />
```

### In Code

For example, to set the `SourceUri` property, add the following code to your project:

- Visual Basic  
`Me.C1HtmlHost1.SourceUri = "http://www.componentone.com"`
- C#  
`this.C1HtmlHost1.SourceUri = "http://www.componentone.com";`

### ✔ What You've Accomplished

You've set the `SourceUri` property and customized the `C1HtmlHost` control so that it displays the ComponentOne Web site at run time. Run the application, and observe the ComponentOne Web site appears in the `C1HtmlHost` control at run time.

## Displaying HTML Markup

**HtmlHost for Silverlight** supports displaying HTML markup. You set the `SourceHtml` property to HTML markup that will be translated at run time; for example see the steps below.

### At Design Time

To set the `SourceHtml` property, complete the following steps:

1. Click the `C1HtmlHost` control once to select it.
2. Navigate to the Properties window and locate the **SourceHtml** item.
3. Enter text in the text box next to the **SourceHtml** item; for example, enter "This is &lt;b&gt;HTML&lt;/b&gt; hosted in a Browser!".

### In XAML

For example, to set the `SourceHtml` property add `SourceHtml` to the `<c1:C1HtmlHost>` tag so that it appears similar to the following:

```
<c1ext:C1HtmlHost Name="c1HtmlHost1" SourceHtml="This is &lt;b&gt;HTML&lt;/b&gt; hosted in a Browser!"/>
```

### In Code

For example, to set the `SourceHtml` property, add the following code to your project:

- Visual Basic

```
Me.C1HtmlHost1.SourceHtml="This is &lt;b&gt;HTML&lt;/b&gt; hosted in a Browser!"
```

- C#

```
this.C1HtmlHost1.SourceHtml="This is &lt;b&gt;HTML&lt;/b&gt; hosted in a Browser!";
```

### ✔ What You've Accomplished

The C1HtmlHost control will now display HTML text at run time. Note that in the markup above the "&lt;" and "&gt;" were used for the "<" and ">" symbols. So, for example a tag to make the text bold, which would normally appear like "<b>" is written as "&lt;b&gt;".

## Hiding Frame Borders

By default, borders are displayed around the C1HtmlHost control. You may want to hide this frame border – for example, if you want the control to blend more seamlessly with your application. Hiding frame borders in **ComponentOne HtmlHost for Silverlight** is as simple as setting a single property. The **FrameBorder** property determines if a frame is displayed around the control.

### At Design Time

To set the **FrameBorder** property, complete the following steps:

1. Click the C1HtmlHost control once to select it.
2. Navigate to the Properties window and locate the **FrameBorder** item.
3. Clear the check box next to the **FrameBorder** item.

### In XAML

For example, to set the **FrameBorder** property add `FrameBorder="False"` to the `<c1:C1HtmlHost>` tag so that it appears similar to the following:

```
<c1ext:C1HtmlHost Name="c1HtmlHost1" FrameBorder="False" />
```

### In Code

For example, to set the **FrameBorder** property, add the following code to your project:

- Visual Basic

```
Me.C1HtmlHost1.FrameBorder = False
```

- C#

```
this.C1HtmlHost1.FrameBorder = false;
```

### ✔ What You've Accomplished

You've set the **FrameBorder** property so a frame does not appear around the control at run time. Run the application, and observe the C1HtmlHost control appears without a surrounding frame.