
ComponentOne

Themes for WinForms

GrapeCity US

GrapeCity
201 South Highland Avenue, Suite 301
Pittsburgh, PA 15206

Tel: 1.800.858.2739 | 412.681.4343

Fax: 412.681.4384

Website: <https://www.grapecity.com/en/>

E-mail: us.sales@grapecity.com

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 \$2.5 (to cover postage and handling) to ComponentOne.

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

Copying and Distribution

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

Themes for WinForms Overview	3
HowToIncorporateComponentInProject	3
Run Time File Of Component	3
Help with WinForms Edition	3
Themes for WinForms Key Features	4
Themes for WinForms Quick Start	5
Applying a Built-In Theme to your Application	5-8
Creating a Custom Theme	8-9
Step 1 of 2: Creating a New Theme in the Theme Designer	9-10
Step 2 of 2: Using Themes in Visual Studio	10
ThemeController Overview	11
Themeable ComponentOne WinForms Controls	11-12
Themeable Microsoft Controls	12
Theme Concepts	12-13
Theme Sources	13
Theme Names	13-14
C1Themes and the Visual Style Property	14-15
Design-Time Support	15
C1ThemeController Context Menu	15-16
C1ThemeController Smart Tag	16-17
C1ThemeController Dialog Box	17-18
C1Theme Designer Application Overview	19-20
Theme Designer Ribbon Menu	20
File Tab	20-21
Home Tab	21-25
Edit Group	25-27
Find Group	27
Filter Group	27-28
Form Preview	28
Theme Preview	28-29
Theme Designer Dialog Boxes	29
New Theme Dialog Box	29-30
New Property Dialog Box	30-31
Adding New Property in Theme Tree	31-32

Navigating Property Items in Theme Tree	32-33
Searching and Filtering Property Items in Theme Tree	33-36
Specifying Theme Property by Reference	36-37

Themes for WinForms Overview

Themes for WinForms makes it easy to theme your WinForms controls with many built-in themes and an easy to use Theme designer to create your own themes or modify the existing ones. This product enables you to easily provide a consistent customized look to WinForms applications which use other ComponentOne WinForms products such as C1Ribbon, C1Scheduler, C1FlexGrid and others. Themes for WinForms includes one component, **C1ThemeController**, and a **Theme** designer.

The C1Themes product physically consists of two separate parts:

- C1.Win.C1Themes.4 assembly

This assembly provides the C1ThemeController component, which is used to load and apply themes to controls at design and runtime. Besides, the assembly contains (as embedded resources) a set of predefined themes which can be used as is, or customized to better serve your purposes. Projects using this assembly need to have a reference to C1.C1Zip.4.dll.

- C1ThemeDesigner application

This application can be used to create new, or customize existing themes (including those embedded into the C1.Win.C1Themes.4 assembly as mentioned above).

Getting Started

- [Themes for WinForms Quick Start](#)
- [ThemeController Overview](#)
- [C1Theme Designer Application Overview](#)

HowToIncorporateComponentInProject

Run Time File Of Component

Help with WinForms Edition

For information on installing **ComponentOne Studio WinForms Edition**, licensing, technical support, namespaces and creating a project with the control, please visit [Getting Started with WinForms Edition](#).

Themes for WinForms Key Features

C1Themes includes the following key features:

- C1Themes allow to provide a consistent customized look to your applications that use ComponentOne WinForms controls.
- The look of an application can be changed with a single click at design or run time.
- The look of your applications can be designed/modified by visual designers (not programmers) independently of actual coding, using the easy to use C1ThemeDesigner.
- Use one of the pre-built themes, or create your own either from scratch, or by modifying a pre-built theme.
- Themes can be deployed independently of applications, and used by different applications throughout your company as long as the applications are built using C1 WinForms controls.
- Within a theme, a common repository of properties (Base Theme Properties) can be referenced throughout the theme to enable creation of consistent and easily modifiable themes.

Themes for WinForms Quick Start


Themes for WinForms enables users to quickly create many prebuilt themes using the **C1ThemeDesigner** so they can easily apply those themes to any of their ComponentOne WinForms or Standard WinForms controls. C1Themes for WinForms includes a **C1ThemeController.Theme** property that enables users to get their theme created by the **C1ThemesDesigner** so they can apply it to all supporting ComponentOne WinForm or Standard WinForms controls or they can apply different themes to individual ComponentOne WinForms or Standard WinForms controls using the **C1ThemeController.Theme** property.

In this quick start you will learn how to apply a built-in theme to your application and you will also learn how to create or modify existing themes using the C1ThemeDesigner.

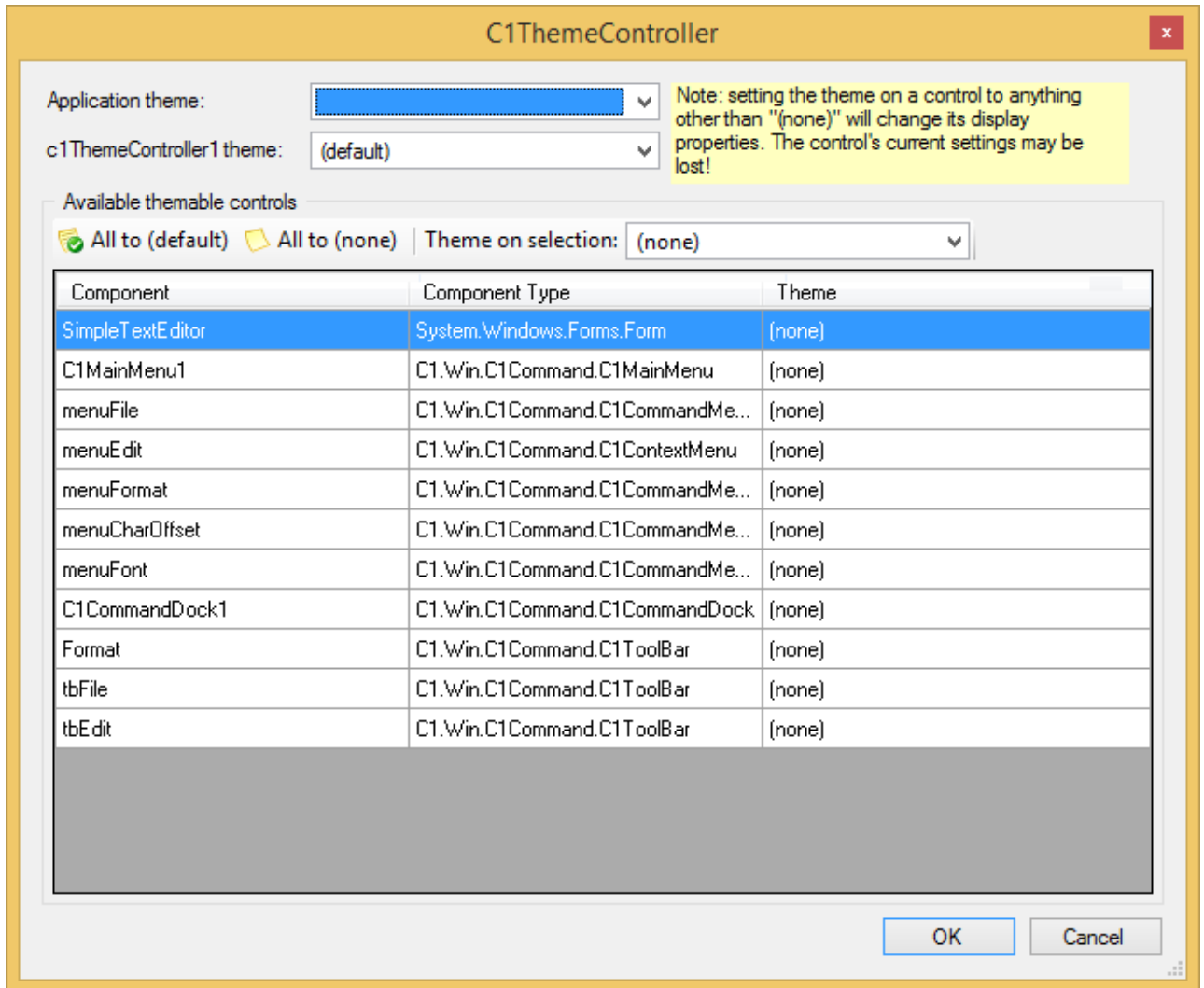
Applying a Built-In Theme to your Application

To simply theme the controls in your application use C1ThemeController like the following:

1. Open a form belonging to your application in the Visual Studio's forms designer. In this example we will open a ComponentOne WinForms sample, **SimpleTextEditor**.
2. From the designer's toolbox, drag a C1ThemeController and drop it on your form. (The C1ThemeController is provided by the C1.Win.C1Themes.4 assembly; if you don't see it in your toolbox you can add it manually).

 **Note:** Reference to C1.C1Zip.4.dll is required in a project that uses C1ThemeController.


A ThemeController dialog box appears.

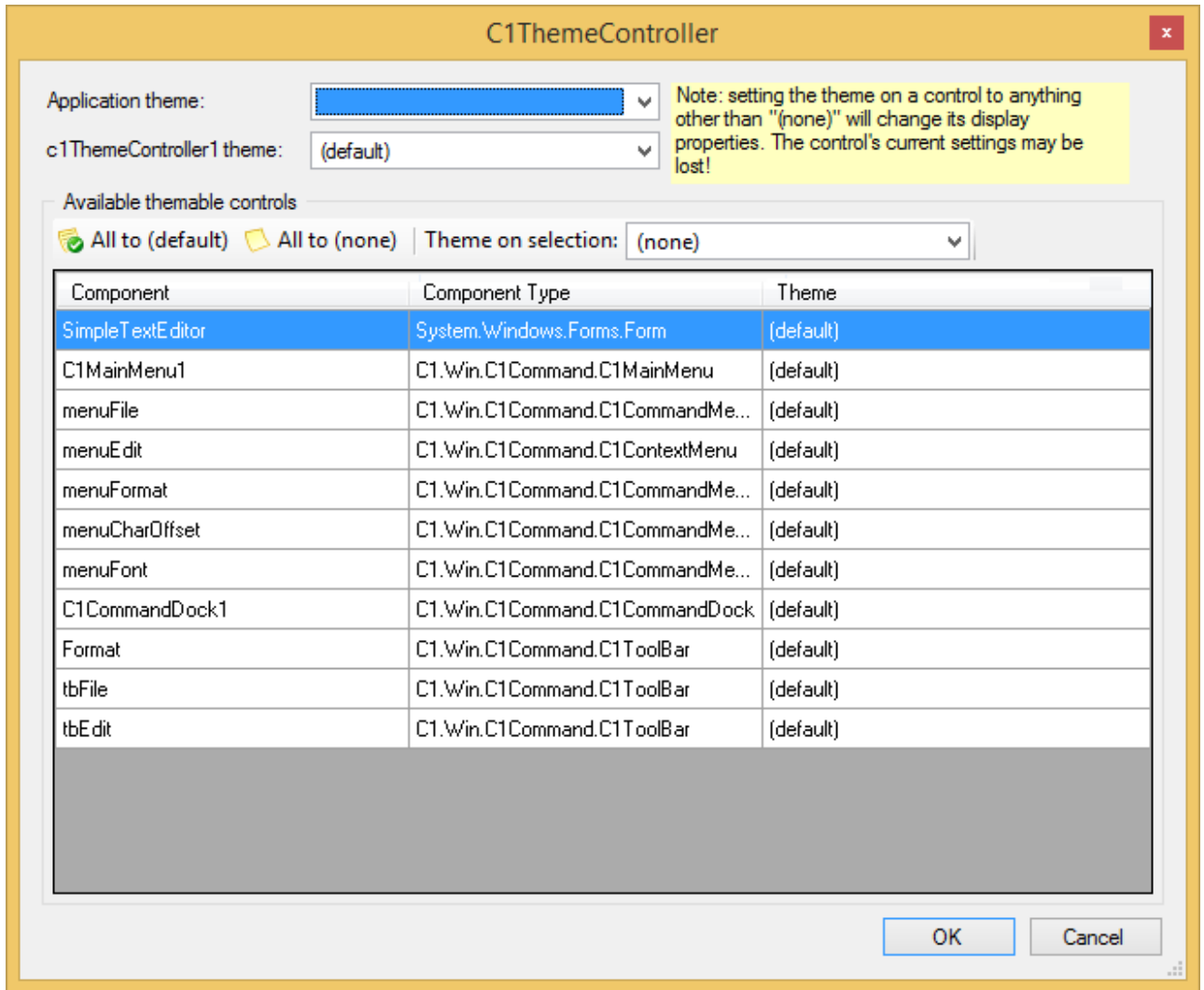


This enables you to select the application-wide default theme, the current controller's default theme, and themes for all supporting controls already on the form. Note: The same dialog is also available via the **Set Themes** action on the C1ThemeController tasks menu.

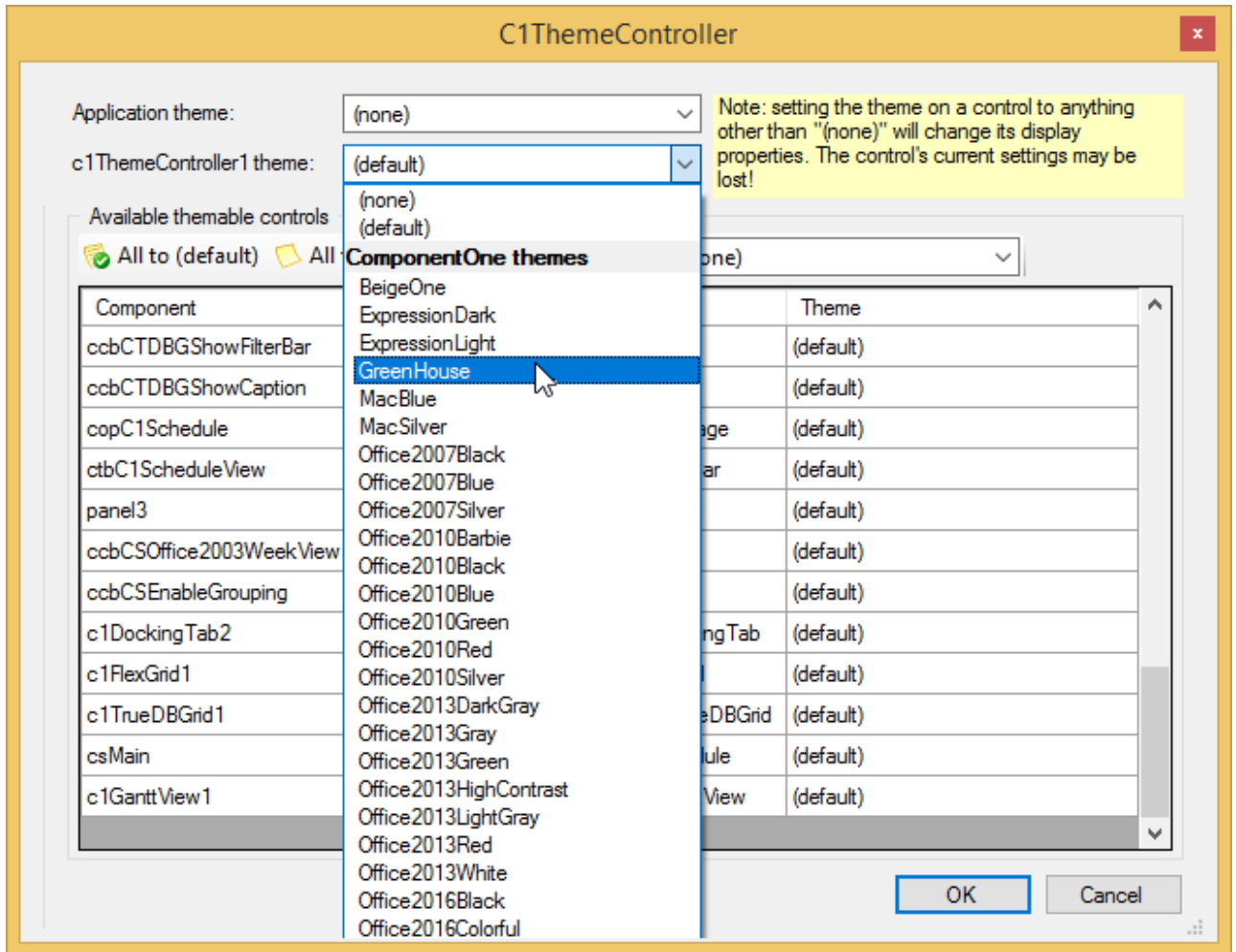
Initially in the dialog that pops up, the theme is specified as "(none)" for all supporting controls already on the form. This is done to prevent unintentional loss of property settings on those controls.

3. Click the **All to (default)** button in the dialog so that the default theme is set on all controls. If you have deliberately customized some of the controls already this will not need to be done.

 (When new controls are dropped on the form already containing a C1ThemeController component, theme is set to "(default)" on them automatically.)



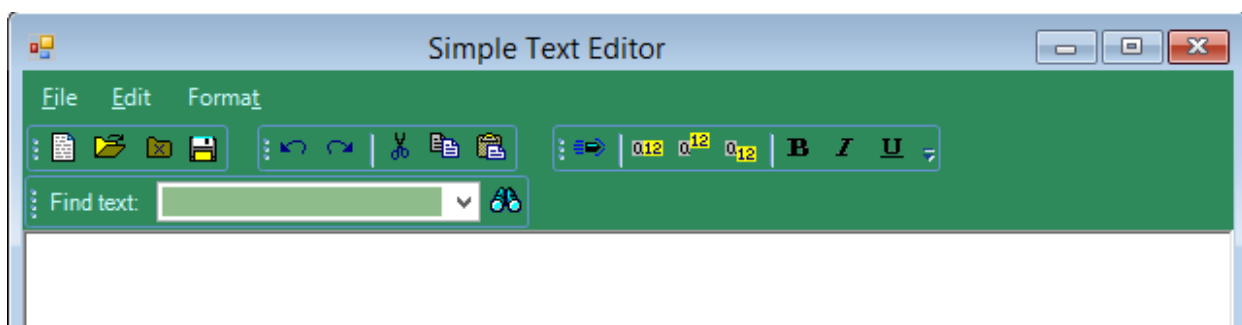
4. Select the default application theme from the list of available prebuilt themes. For example, This theme name will be automatically stored in the App.config file.



4. Leave the theme on the current theme controller ('c1ThemeController1 theme') at its default "(default)" value (which means that the application theme, **GreenHouse**, will be used).
5. Click the **All to (default)** button as mentioned above, to set the default theme on all supporting controls on the form.
6. Finally, click the **OK** button to close the dialog and apply the specified theme to the controls on the form.
7. If your application contains multiple forms, repeat the steps above (with the exception of specifying the default ApplicationTheme as it will already be pre-selected for you) for all forms in your application.

When you are done, the selected application theme will be applied to controls on your forms both at design and runtime.

The following image illustrates the GreenHouse theme applied to the SimpleTextEditor application:



Creating a Custom Theme

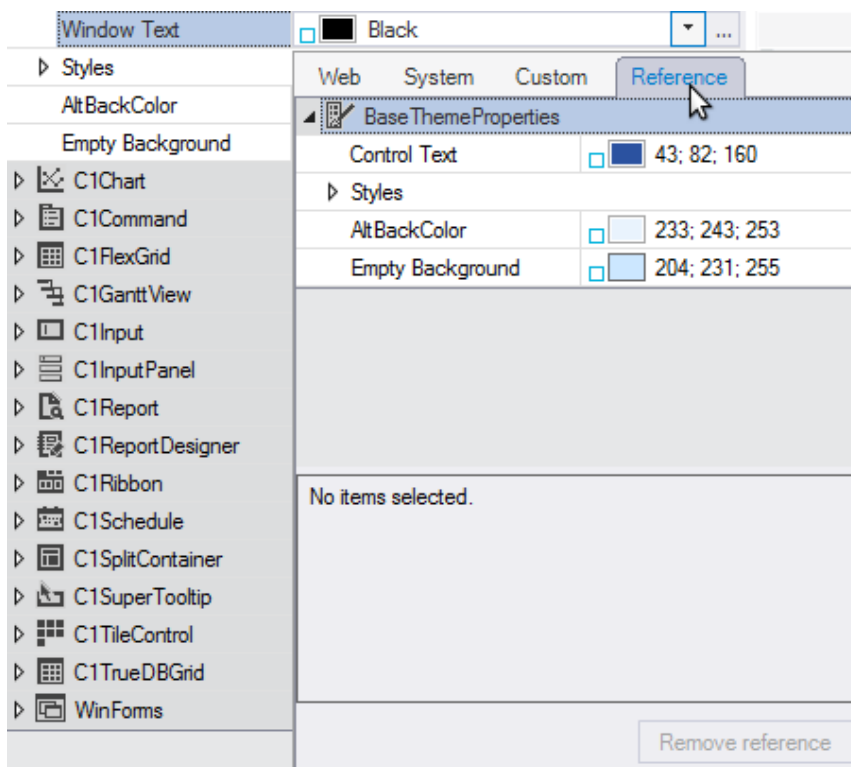
This QuickStart guide will walk you through creating a new theme based on a built-in theme template using the

C1ThemeDesigner. You will modify the built in theme template using the C1ThemeDesigner.

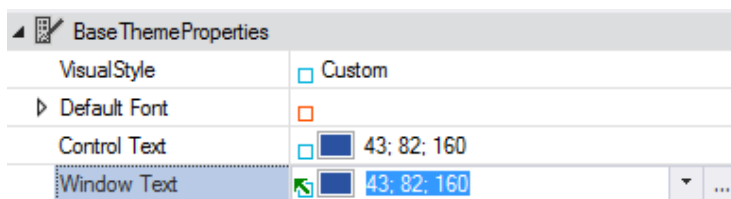
Step 1 of 2: Creating a New Theme in the Theme Designer

In this step, you will create a new theme based on the Standard built-in theme template. You will also modify some Theme Base Properties and add a reference to another theme base property. For more information see **C1ThemeDesigner Application Overview**.

1. Double-click on the C1ThemeDesigner from the **C:\Program Files\ComponentOne\WinForms Edition\bin\v4\Designer** folder.
2. Click on **Create New Theme** from the Topic Bar. The **New Theme** dialog box appears.
3. Select the Standard built-in theme template and enter BlueandPink for the theme Name and Click **OK**.
4. Select **Control Text** from the BaseThemeProperties in the Theme tree.
5. Click on the Control Text dropdown arrow and click the **Custom** tab.
6. Set the **BackColor** to **151; 45; 85**.
7. Click on the dropdown arrow next to **Window Text** and select the **Reference** tab.
8. Select the **Control Text** in the Reference tab.

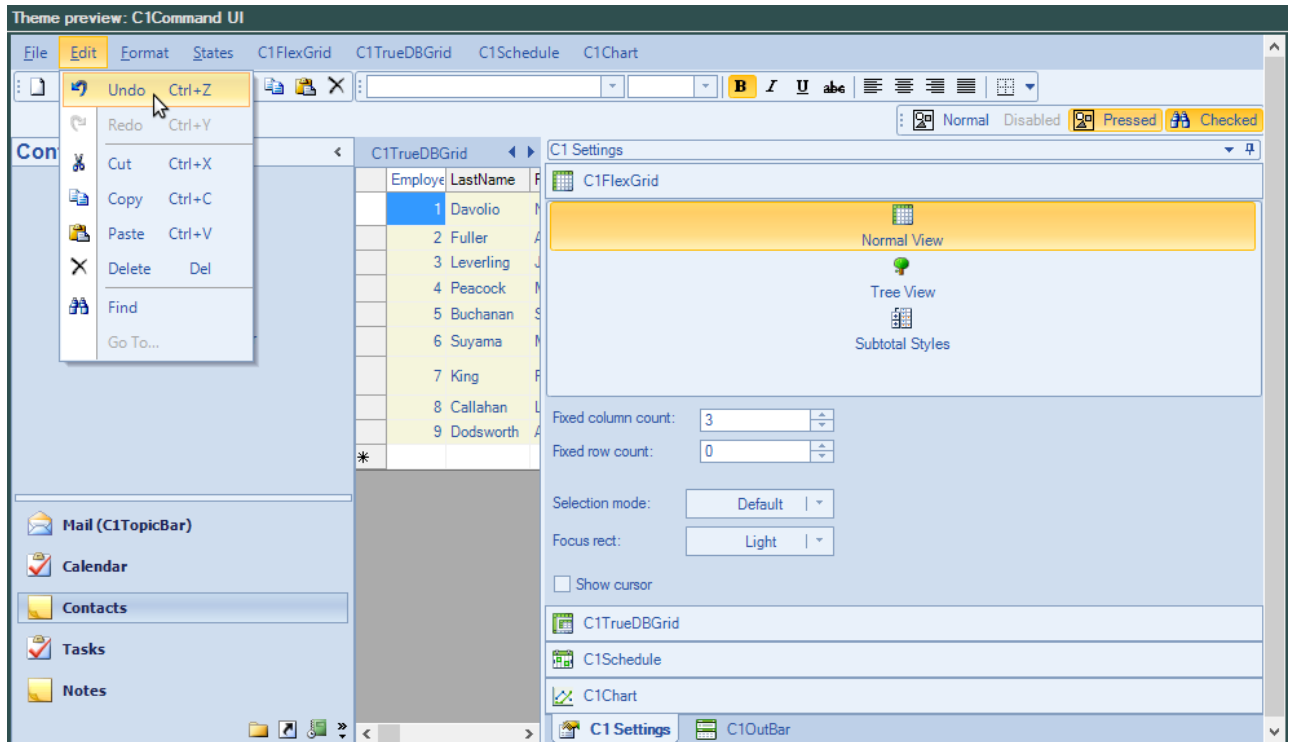


Notice the icon appears as a visual indicator that the **Window Text** property is referenced from the BaseThemeProperties **Control Text**.



9. Expand **C1Command| C1MainMenu| | Bar| BackColor**.
10. Click on the **BackColor** dropdown arrow and select the **Custom** tab.
11. Select the Light pink so the numbers appear as **255; 231; 242**.

Notice the new settings appear in the Preview pane.



12. Select **File| Save As** from the **File** menu and save and name your file. For example, **BlueandPink**. The new location appears at the top of the C1Theme Designer.

Step 2 of 2: Using Themes in Visual Studio

In this step, you'll add the custom theme to your application.

1. Open a form belonging to your application in the Visual Studio's forms designer. In this example we will open a ComponentOne WinForms sample, **SimpleTextEditor2**.
2. Add the **C1ThemeController** from the toolbox onto your form. Also, provide reference to C1.C1Zip.4.dll in your project.
3. Click on the C1ThemeController component's **Add theme...** action item. This will show a file open dialog. Select the theme file and click **OK**.
The selected theme will be added to your C# or VB project as an embedded resource in a **Themes** folder, and will be available for selection from the themes list provided by the C1ThemeController.
4. Run and observe your changes.

ThemeController Overview

C1ThemeController is a component that loads and manages visual themes and applies them to forms and controls. C1ThemeController can be used in any of the following ways:

- **Applies the same theme to all supporting ComponentOne controls in your application**
 - Use the [C1ThemeController.ApplicationTheme](#) property to accomplish this method.
- **Applies the same theme to all supporting ComponentOne controls on your form**
 - Use the [C1ThemeController.Theme](#) property to accomplish this method.
- **Applies a different theme to all supporting ComponentOne controls on your form**
 - Set the extender property 'Theme on c1ThemeController1' on individual controls on your form.

Themeable ComponentOne WinForms Controls

To effectively use Themes for WinForms you need the C1ThemeController component and the supported versions of the ComponentOne WinForms controls. The Theme designer application is not necessary for applying themes to your WinForms controls, but can be used to customize and/or create new themes.

Themes for WinForms supports the following **ComponentOne WinForms** controls:

ComponentOne WinForms Product	Controls
Chart for WinForms	C1Chart and C1Chart3D
FlexGrid for WinForms	C1FlexGrid
GanttView for WinForms	C1GanttView
Input for WinForms	C1Textbox, C1DateEdit, C1NumericEdit, C1DropDownControl, C1DbNavigator, C1CheckBox, C1Button, C1Label, C1PictureBox, C1SplitButton, C1ComboBox
InputPanel for WinForms	C1InputPanel
Menus and ToolBars for WinForms	C1DockingTab, C1MainMenu, C1ContextMenu, C1ToolBar, C1TopicBar, C1NavBar, C1OutBar, C1RadialMenu
Report for WinForms	C1Report, C1PrintPreview, C1PreviewPane, C1PreviewThumbnailView, C1PreviewOutlineView, C1PreviewTextSearchPanel, and C1PrintPreviewDialog. C1ReportDesigner
Ribbon for WinForms	C1Ribbon and C1StatusBar
Scheduler for WinForms	C1Schedule and C1Calendar

SplitContainer for WinForms	C1SplitContainer
SuperToolTip for WinForms	C1SuperToolTip, C1SuperErrorProvider, C1SuperLabel
TileControl for WinForms	C1TileControl
TrueDBGrid for WinForms	C1TrueDBGrid

Themeable Microsoft Controls

You can easily customize your applications by applying C1Themes even to the Microsoft controls used within the application. **C1Theme** in WPF supports following Microsoft controls:

- Button
- CheckBox
- CheckedListBox
- ComboBox
- DomainUpDown
- Form
- GroupBox
- Label
- LinkLabel
- ListBox
- ListView
- MaskedTextBox
- NumericUpDown
- Panel
- PropertyGrid
- RadioButton
- SplitContainer
- Splitter
- StatusStrip
- TextBox
- ToolStrip
- TrackBar
- TreeView
- UserControl

Theme Concepts

A theme is an XML document (with the **.c1theme** extension when saved in a file) that contains a consistent set of values for properties determining the look and feel of supporting controls. When applied to a form in an application (at runtime and/or design time, using the **C1ThemeController** component), the theme changes the look of controls on that form so that they all comply with the theme.

Internally, a theme is divided into sections corresponding to different ComponentOne products, so that each product only uses its own section when the theme is applied. This allows to provide a customized "themed" look for even the smallest and most specialized details of a control. Consistency within a theme is achieved via a mechanism of references, which enables properties such as colors and fonts in individual controls to reference a common repository accessible to all other sections of the theme, which is called the "Base Theme Properties". Properties in that repository

are not directly applied to controls' properties, but they can be referenced by the theme author when specifying the actual properties used by the controls. Additionally, a theme includes a section that is used by **C1ThemeController** to apply the theme to standard WinForms forms and controls such as buttons and labels.

Theme Sources

In an application, themes can come from the following sources:

- Standard C1 themes embedded as resources into the C1.Win.C1Themes.4 assembly;
- Themes which you create using the C1ThemeDesigner application, and add as embedded resources into your own assemblies;
- Themes which are loaded at runtime from .c1theme file(s).

The first two theme sources are available at design time via the C1ThemeController properties and designer actions. At runtime, you can also load themes in code from .c1theme files on disk using methods available on the C1ThemeController component.

For example the following code shows how to apply the built-in theme through code using the [Register](#) and [SetTheme](#) methods:

C#

```
//Register the theme file with the C1ThemeController;
C1.Win.C1Themes.C1ThemeController.RegisterTheme(@"C:\Users\Documents\Visual Studio
2010\Projects\ThemesBuild6\ThemesBuild6\NewOffice2013White.c1theme");
//Apply it to a control and use the theme name
this.c1ThemeController1.SetTheme(c1MainMenu1, "NewOffice2013White")
```

Visual Basic

```
'Register the theme file with the C1ThemeController
C1.Win.C1Themes.C1ThemeController.RegisterTheme("C:\Users\Documents\Visual Studio
2010\Projects\ThemesBuild6\ThemesBuild6\NewOffice2013White.c1theme")
'Apply it to a control, use the theme name, not the file name
Me.c1ThemeController1.SetTheme(someControl, "NewOffice2013White")
```

Theme Names

In **C1Themes**, themes are identified by their names. Names are case-sensitive, and must be unique within an application.

Themes for WinForms include the following regular theme names which are located in the **Themes** folder:

- BeigeOne.c1themez
- ExpressionDark.c1themez
- ExpressionLight.c1themez
- GreenHouse.c1themez
- MacBlue.c1themez

- MacSilver.c1themez
- Office2007Black.c1themez
- Office2007Blue.c1themez
- Office2007Silver.c1themez
- Office2010Barbie.c1themez
- Office2010Black.c1themez
- Office2010Blue.c1themez
- Office2010Green.c1themez
- Office2010Red.c1themez
- Office2010Silver.c1themez
- Office2013DarkGray.c1themez
- Office2013Gray.c1themez
- Office2013Green.c1themez
- Office2013HighContrast.c1themez
- Office2013LightGray.c1themez
- Office2013Red.c1themez
- Office2013White.c1themez
- Office2016Black.c1themez
- Office2016Colorful.c1themez
- Office2016DarkGray.c1themez
- Office2016White.c1themez
- RanierOrange.c1themez
- ShinyBlue.c1themez
- Violette.c1themez
- VS2013Blue.c1themez
- VS2013Dark.c1themez
- VS2013DarkSolar.c1themez
- VS2013Green.c1themez
- VS2013Light.c1themez
- VS2013Purple.c1themez
- VS2013Red.c1themez
- VS2013Tan.c1themez
- Windows8Blue.c1themez
- Windows8Brown.c1themez
- Windows8Gray.c1themez
- Windows8Green.c1themez
- Windows8Red.c1themez

Themes for WinForms include the following non-regular theme names:

- "(default)": specifies the default theme in the current context;
- "(none)": specifies that no theme should be applied.

C1Themes and the Visual Style Property

Many **ComponentOne WinForms** controls include a property called **VisualStyle**, of an enum type defined in the control assembly, but following a common naming pattern for the type and possible values. Typically, the enum type is called **VisualStyle**, with values such as **Office2010Blue**, **Office2010Black**, and so on. C1Themes are designed to provide a more powerful and flexible mechanism for adjusting the look of controls. Still there is obvious overlap between the two mechanisms, with the possibility of conflicts. The rules for dealing with this overlap are as follows:

- Theme sections for all C1 controls include a VisualStyle property that can be specified by the theme.
- By default and in all supplied themes, those properties are set to 'Custom' so that the VisualStyle does not interfere with applying other properties.

- Setting `VisualStyle` in a theme to anything other than `custom` sets the corresponding property on the target control and disables applying all other theme properties. (In the `C1ThemeDesigner`, this actually disables the rest of the theme tree for the control.)

While we recognize that backward compatibility or other considerations may require the use of `VisualStyle` rather than themes to customize the look of your application, we recommend that if possible you use `C1Themes` as they provide a more powerful and flexible mechanism for that. Support for visual styles in new controls will be phased out as themes will replace it

Design-Time Support

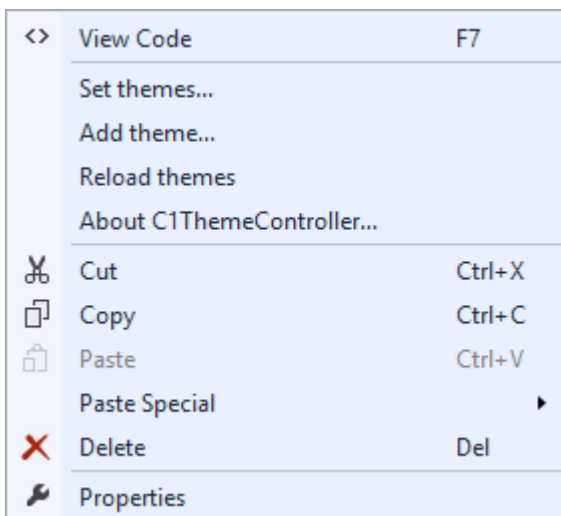
`C1ThemeController` includes a smart tag and dialog boxes to simplify design-time tasks when using the **C1ThemeController** component.

The following topics describe how to use `C1ThemeController`'s design-time environment to configure `C1ThemeController`.

C1ThemeController Context Menu

The **C1ThemeController** component provides a context menu for additional functionality to use at design time.

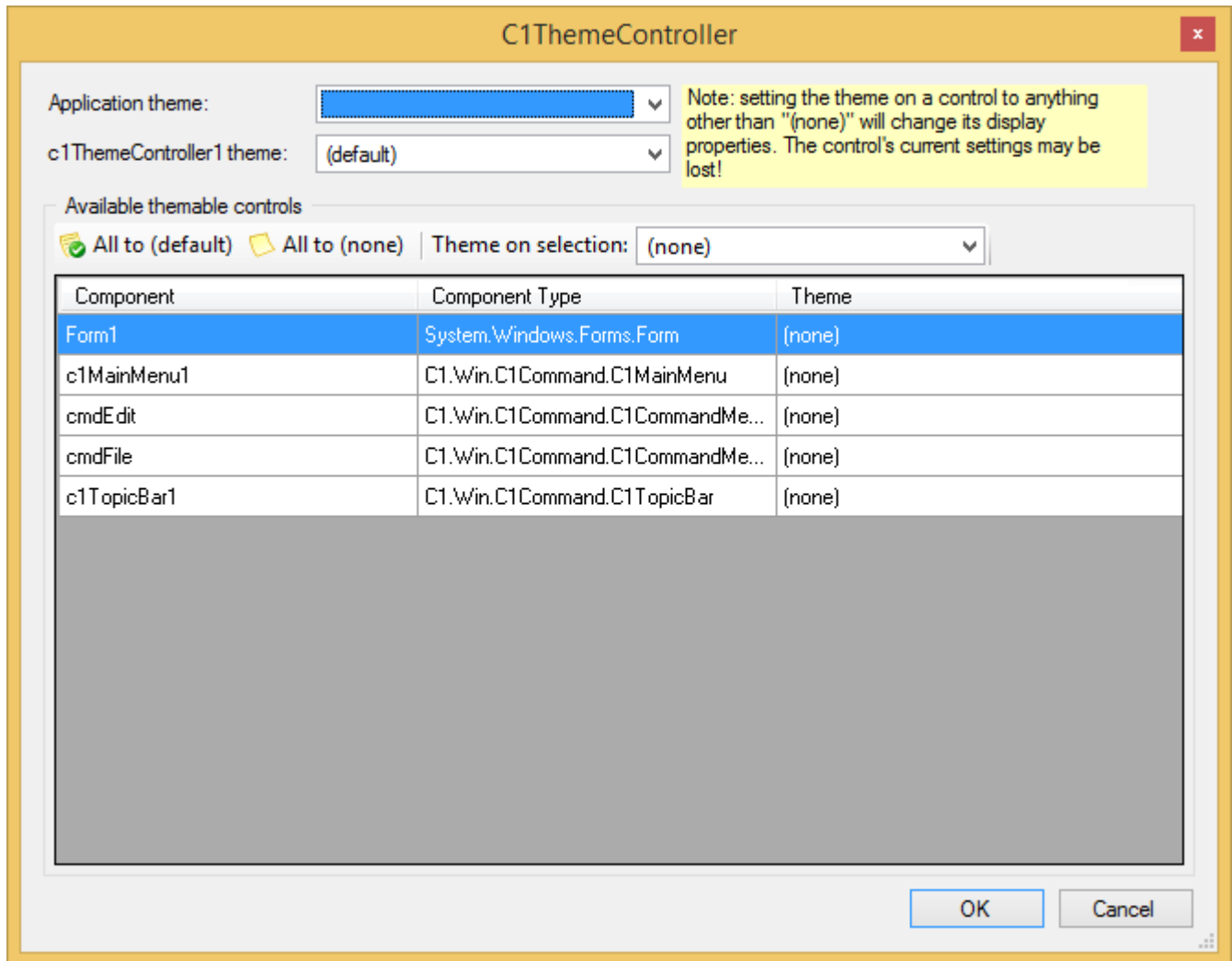
To access `C1ThemeController`'s context menu, right-click on the `C1ThemeController` control and the context menu for it appears like the following:



The `C1ThemeController` Context menu operates as follows:

Set themes..

Clicking **Set themes...** opens the **C1ThemeController** dialog box. For more information see [C1ThemeController Dialog Box](#).



Add theme...

Clicking **Add theme...** opens the **Select Theme** dialog box where you can browse to the location of your theme and add it to the C1ThemeController. Once you select your theme, it will be registered to your project and in Visual Studio a Themes folder will appear in the Solution Explorer. Once the theme has been added you apply it to the Theme or ApplicationTheme properties.

Reload themes

Clicking **Reload themes** reloads all themes used by the current application.

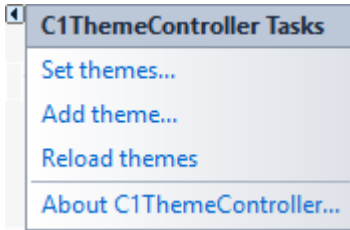
About C1ThemeController..

Clicking **About C1ThemeController** reveals the **About** dialog box. This dialog box displays the version number and licensing information for the **ThemeController** product.

C1ThemeController Smart Tag

In Visual Studio, the C1ThemeController component in Themes for WinForms includes a smart tag. A smart tag represents a short-cut tasks menu that provides the most commonly used properties in each control.

To access the **C1ThemeController Tasks** menu, click the smart tag in the upper right corner of the **C1ThemeController** component. This will open the **C1ThemeController Tasks** menu.



The **C1ThemeController Tasks** menu operates as follows:

- Set themes..

Clicking **Set themes...** opens the **C1ThemeController** dialog box. For more information see [C1Theme Controller Dialog Box](#).

- Add theme...

Clicking **Add theme...** opens the **Select Theme** dialog box where you can browse to the location of your theme and add it to the C1ThemeController. Once you select your theme, it will be registered to your project and in Visual Studio a Themes folder will appear in the Solution Explorer. Once the theme has been added you apply it to the Theme or ApplicationTheme properties.

- Reload themes

Clicking **Reload themes** reloads all themes used by the current application.

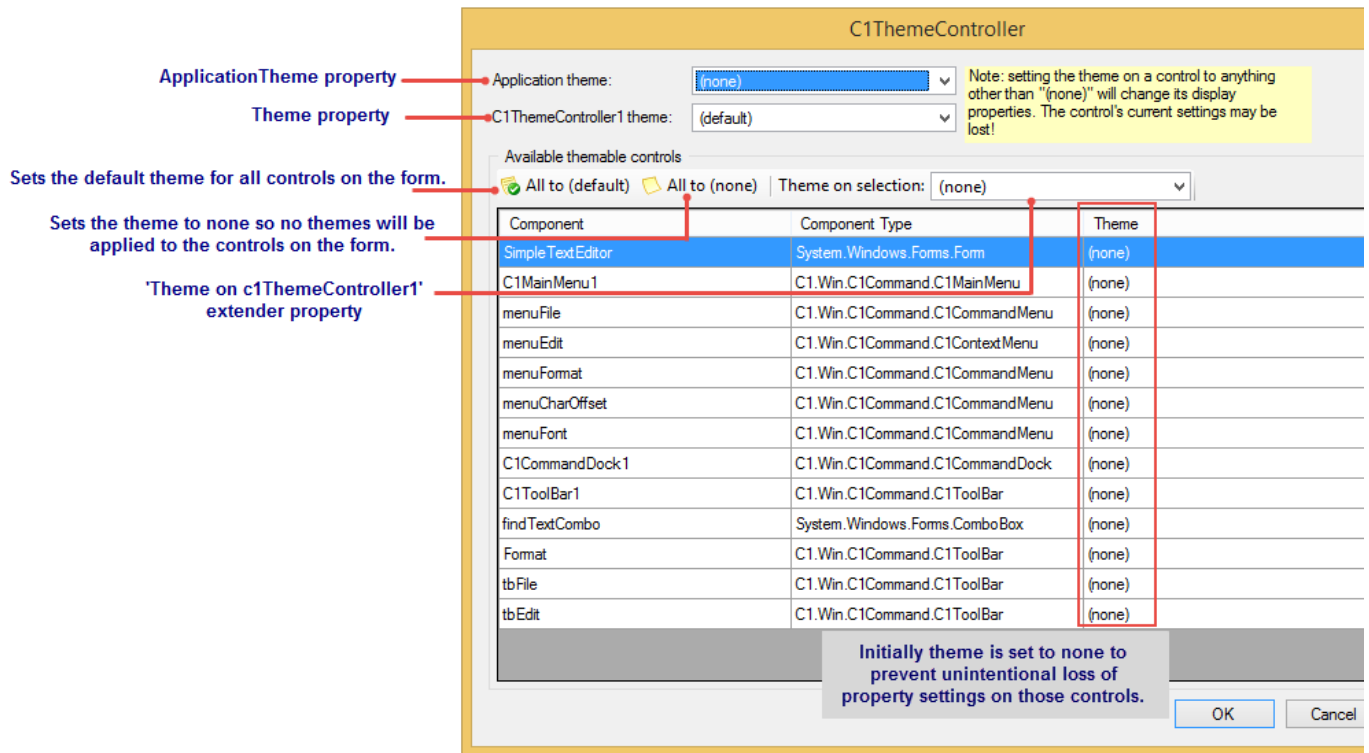
- About C1ThemeController..

Clicking **About C1ThemeController** reveals the **About** dialog box. This dialog box displays the version number and licensing information for the **ThemeController** product.

C1ThemeController Dialog Box

Once you drag and drop the C1ThemeController 2.0 component onto the component tray a C1ThemeController dialog box appears. The ThemeController dialog box is used to quickly to apply the theme to all themeable controls in the application, all themeable controls on the form, or different themes on different controls.

The following image illustrates the main properties used to theme your application:



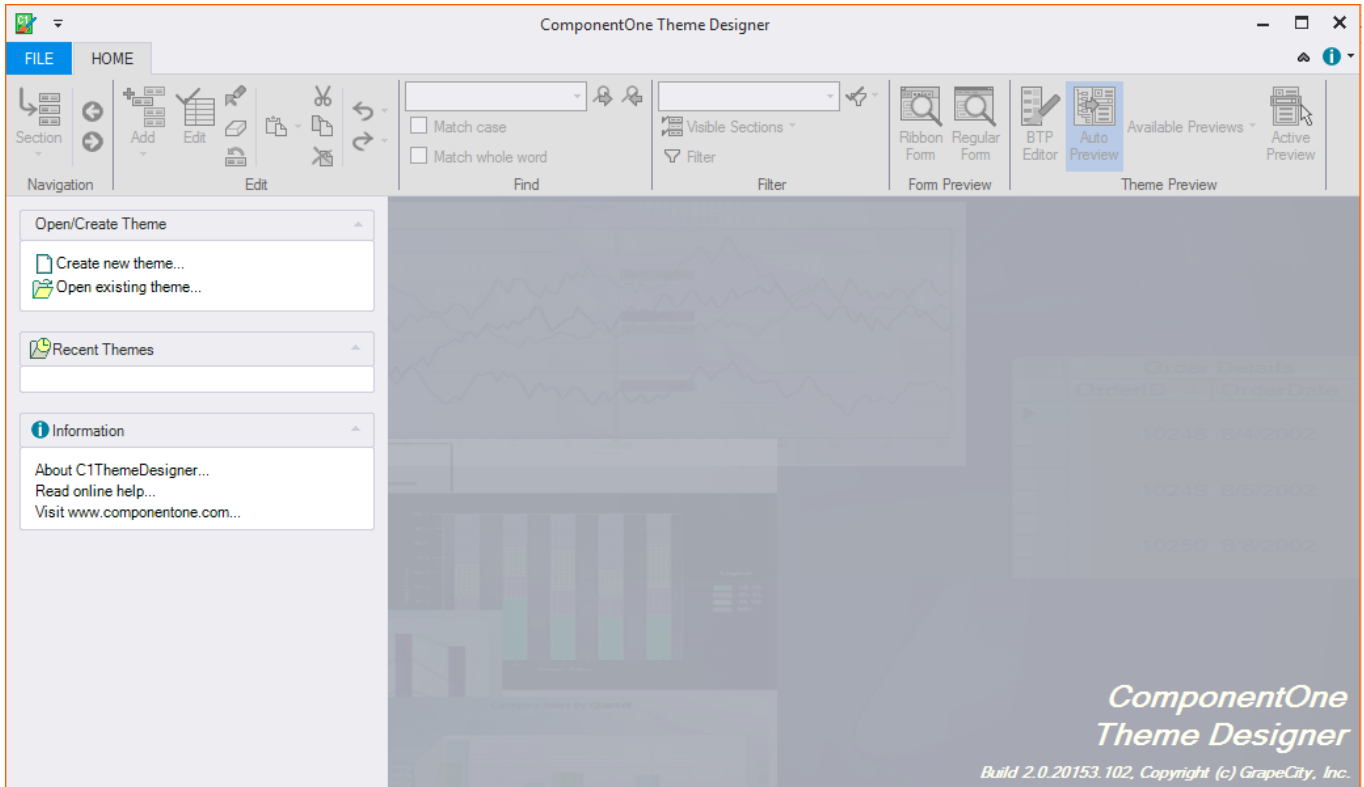
If you have themeable controls on your form before you add the C1ThemeController the dialog box lists all of the components on your form. Each control/component is initially set to none to prevent unintentional loss of property settings on those controls.

The following items in the C1ThemeController dialog box function as follows:

- Application theme dropdown listbox - Used to specify the application-wide (default) theme. The name of this theme is stored in the App.config file.
- C1ThemeController1 theme: dropdown listbox - Specifies the default theme for the current C1ThemeController. The value "(default)" indicates that the application-wide theme specified by ApplicationTheme will be used.
- Theme on selection: dropdown listbox - A 'Theme on c1ThemeController1' extender property is added to all supporting controls on the form. When set to a specific theme name, this property will apply that theme to the current control regardless of the theme controller's default theme. This allows to apply different themes to different controls on the same form if needed. This property supports two special values: 'default' and 'none'. The default value implies that the default theme specified for the C1ThemeController on the current form will be used for this control. 'None' implies no theme will be applied to this control.
- All to (default) button - Sets the default theme for all controls on the form.
- Add to (none) button - Sets the theme to none so no themes will be applied to the controls on the form.

C1Theme Designer Application Overview

Easily design new themes for any of the controls in **WinForms Edition** controls using the **C1ThemeDesigner** application. When the C1ThemeDesigner exe is opened, a TopicBar appears:



The Topic Bar includes the following groups:

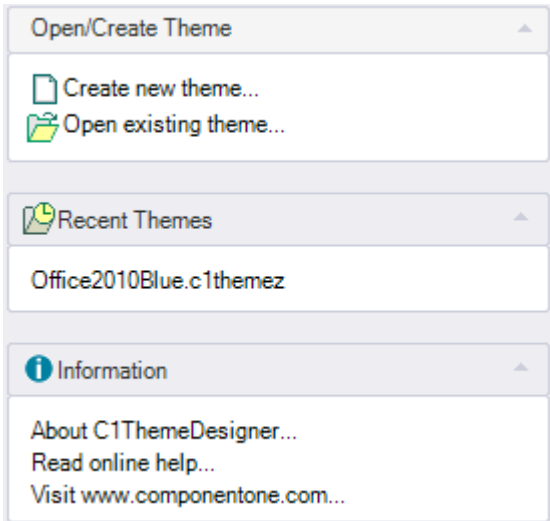
Open/Create Theme

In the **Open/Create Theme** group you can do either of the following:

- **Create new theme...** - Clicking this item opens the **New Theme** dialog box.
- **Open existing theme...** - Clicking this item opens the **Open Theme** dialog box where you last saved your recent theme. Select the theme and then click **Open**. The existing theme will be opened in the **Theme Designer**.

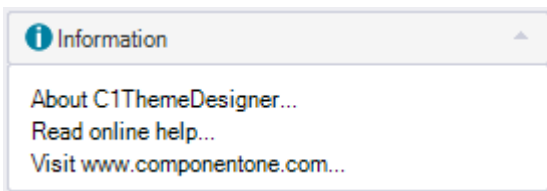
Recent Themes

The recent themes lists the themes you have recently created using the Theme designer.



Information

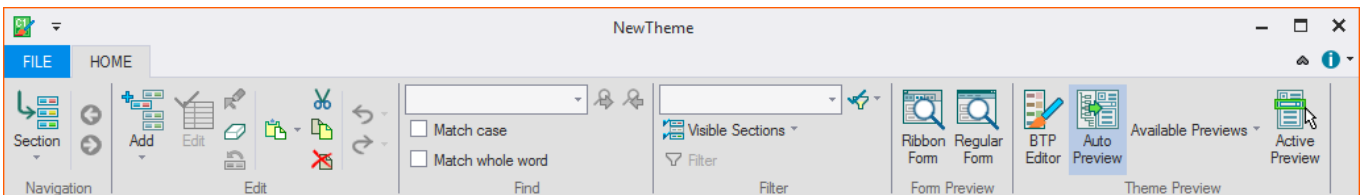
An information group box provides information about the product, C1ThemeDesigner, and a link to the online help.



The **ThemeDesigner** includes a Ribbon menu which consists of a **File** tab and a **Home** tab.

Theme Designer Ribbon Menu

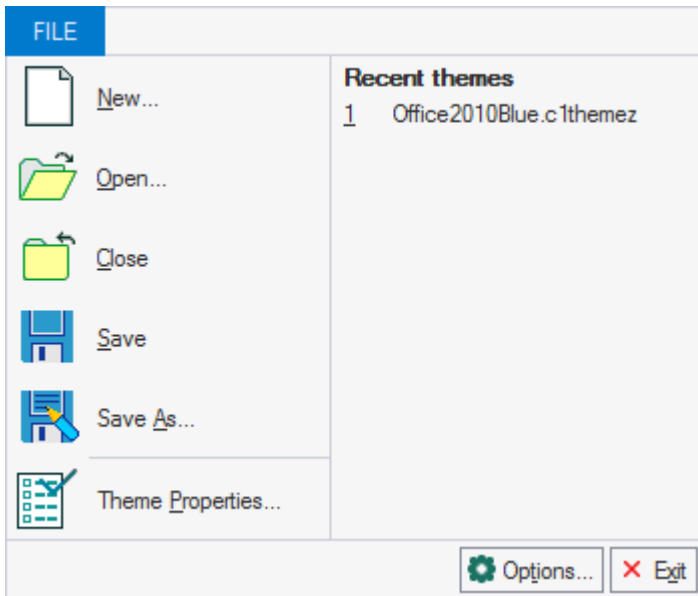
In this topic you will explore the different commands in C1Theme designer ribbon.



The Theme Designer has two tabs: **File** and **Home**.

File Tab

The File menu includes the following options: **New...**, **Open...**, **Close**, **Save**, **Save As** and Theme Properties.... Recent themes that were created appear to the right under **Recent themes**.



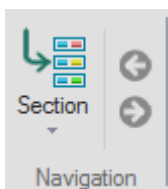
The **File** menu operates as follows:

Command Button	Description
New...	Opens the New Theme dialog box where you can create a new theme from scratch or base it on an existing theme file.
Open...	Opens the Open Theme dialog box where you can open your existing .c1theme file.
Close	Closes the current theme, but keeps the Theme designer open.
Save	Saves the current theme.
Save As...	Saves the current theme in a different file.
Theme Properties...	Opens the Edit theme dialog box where you can view or edit theme name and other properties.

Home Tab

The Home tab includes the **Navigation**, **Edit**, **Find**, **Filter**, and **Preview** groups.

Navigation Group



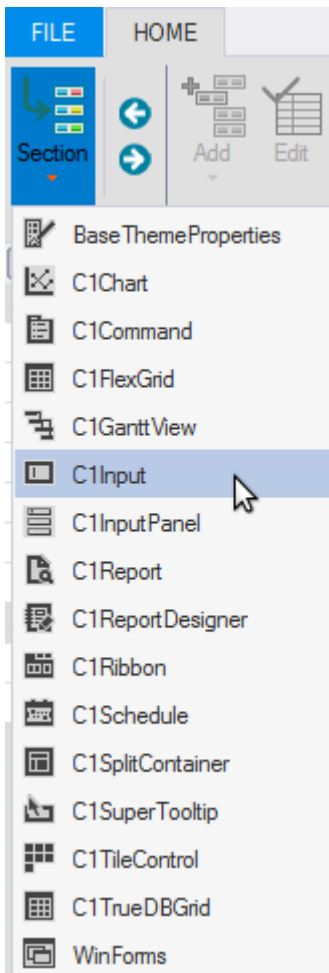
In the **Navigation** group you can quickly navigate the **Theme tree** by selecting the section you wish to edit from the **Section** menu. The Navigation group consists of the following command buttons:



Command Button	Description
Section	Go directly to the theme section for a specific product.
Back	Go back in history to the next position in the theme tree.
Forward	Go forward in history to the next position in the theme tree.

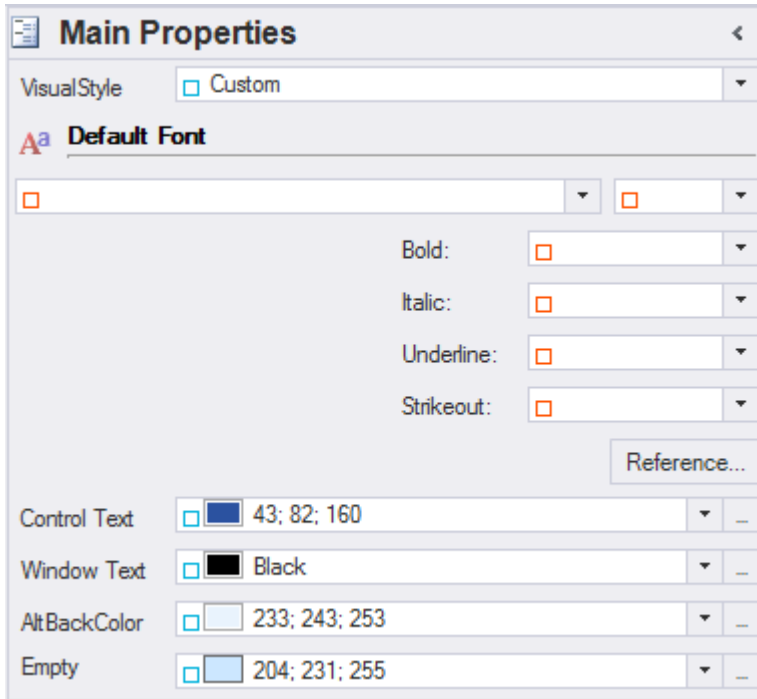
Section Menu

In the **Section** menu you can choose what section of the Theme tree you wish to edit such as Base theme properties, C1Command, C1FlexGrid, C1GanttView, C1Input, C1Preview, C1Schedule, C1SuperToolTip, C1TrueDBGrid, and WinForms.



Base theme properties Editor

When the Base theme properties section is selected the **Main Properties Navigation Bar** appears to the right of the **Theme tree**.



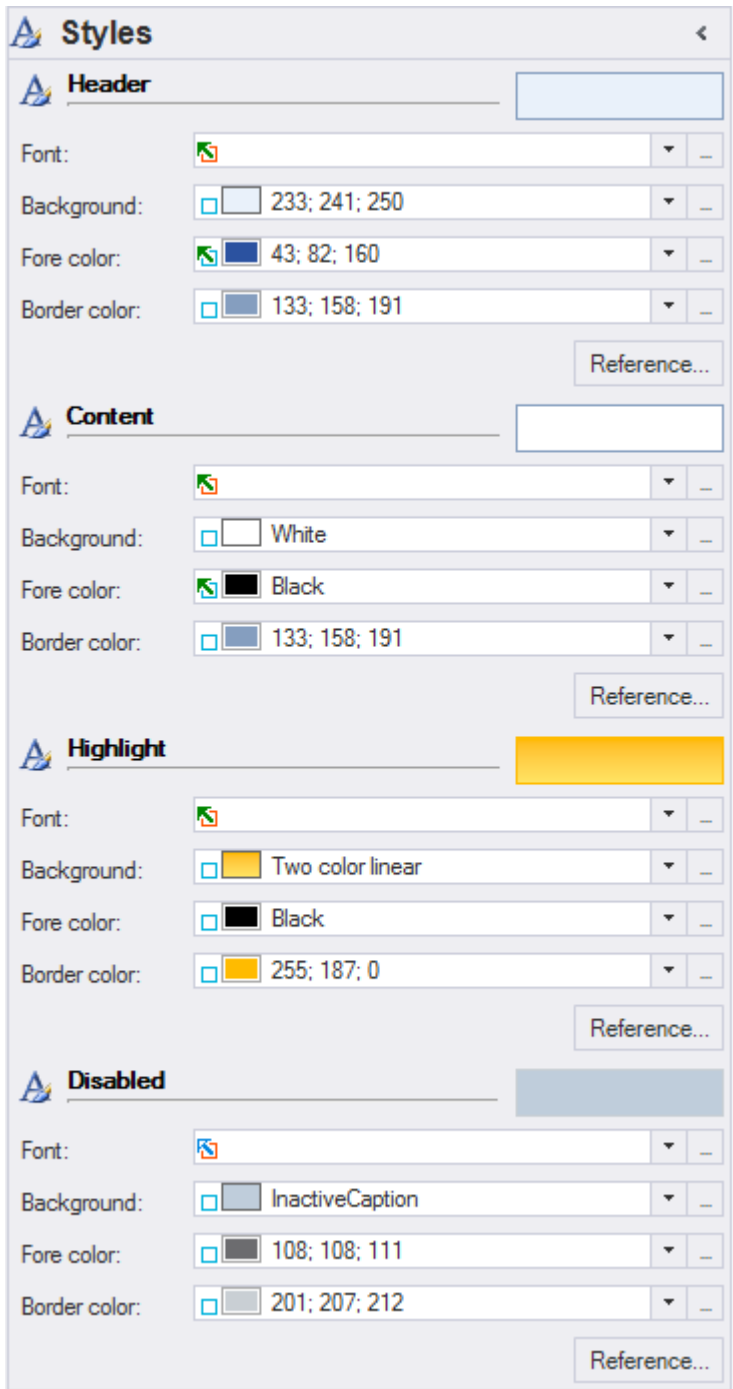
The Navigation Bar includes three categories: **Main Properties**, **Styles**, and **Clickable Styles**.

Main Properties

In the **Main Properties** category you can modify the **FontStyle** properties for the **Default Font**, **Control Text**, and **Window Text** items.

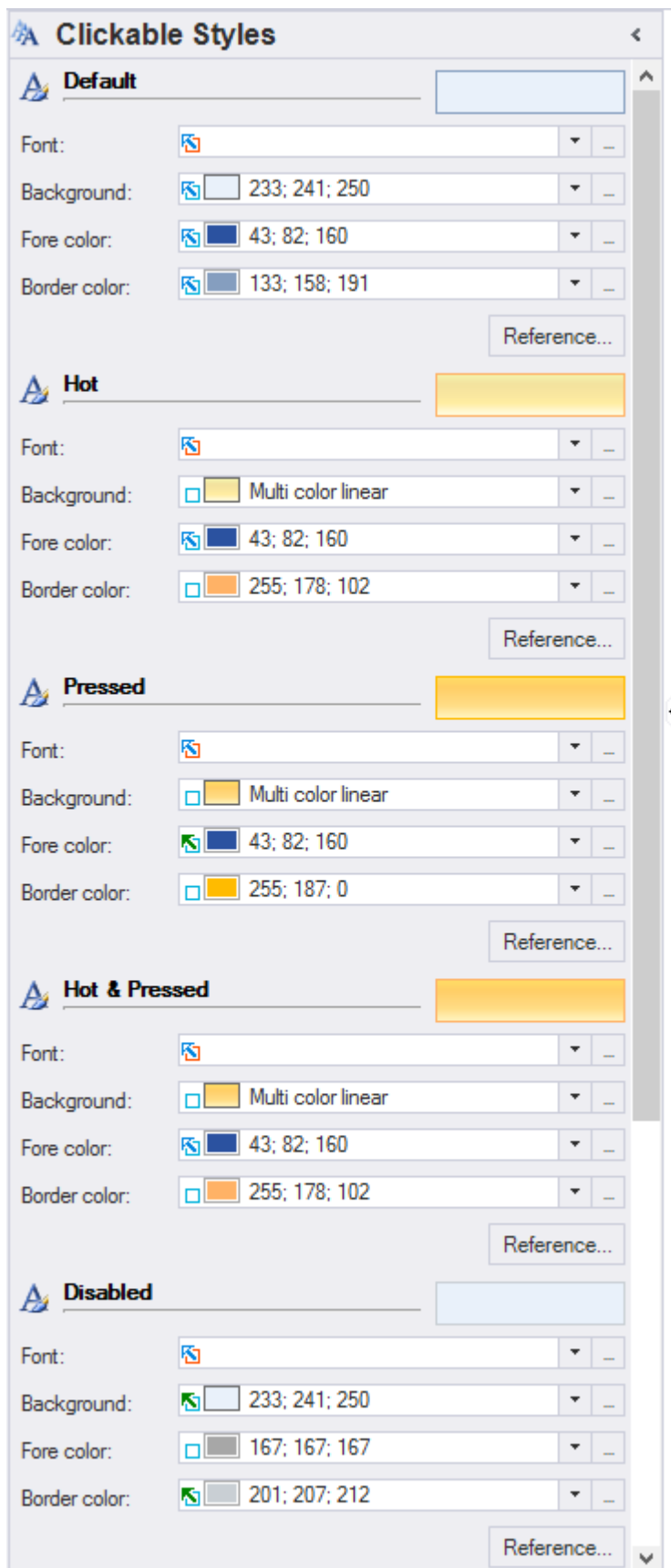
Styles

In the **Styles** button you can modify the **Font**, **Background**, **ForeColor**, and **BorderColor** properties for the **Header**, **Content**, **Highlighted text**, and **Disabled text items**.



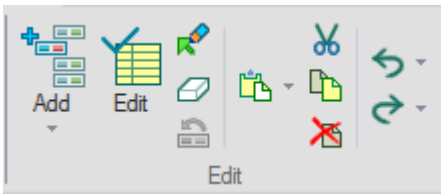
Clickable Styles

In the Clickable Styles category you can modify the Font, Background, ForeColor, and BorderColor properties for the Default, Hot, Pressed, and Hot & Pressed items.



Edit Group

The **Edit Group** in the Ribbon menu of the **C1ThemeDesigner** appears like the following:



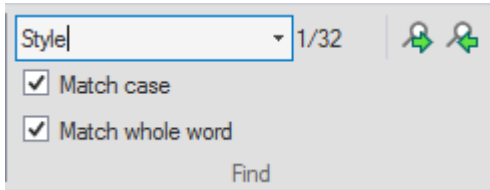
The following command buttons are included in the **Edit** group:

Command Button	Description
Add	The Add command is active when any of the items in the Base Theme Properties section are selected. Click on the Add button to open a submenu of command items. Below command adds the new property immediately below the selected property. Above command adds the new property immediately above the selected property. As Child add the new property as a child of the selected property.
Edit	Edits the attributes (name etc.) of the current Base Theme Properties tree item. The Edit command is active when any of the items in the Base Theme Properties section are selected. Select an item in the Base Theme Properties that you wish to edit and then click on the Edit button. Clicking on the Edit button opens the dialog box: <div data-bbox="357 1079 1350 1503" data-label="Image"> </div>
Reference	Clicking on the Reference command button opens the Edit Reference dialog box where you can edit, add, or remove the selected reference on the theme tree item.
Clear	Clears the property value on the selected theme tree item.
Reset	Resets the property value on the selected theme tree item.
Paste	Pastes the item into the Base Theme properties tree. Click on the Paste button to open a submenu of command items: Paste after, Paste before, Paste as child, and Paste value. The Paste after pastes the properties after the selected property. Paste before pastes the properties before the selected property. Paste as child pastes properties as child for the selected property. Paste value replaces the value of selected properties with data from the clipboard.
Cut	Cuts the selected Base Theme properties item(s) and puts them on the clipboard.
Copy	Copies the selected Base Theme tree properties item(s) and puts them on the clipboard.

Delete	Deletes the selected Base Theme tree properties item(s).
Undo	Undo the last edit.
Redo	Redo the last undone edit.

Find Group

The Find groupbox includes a Find textbox, Find Next command button, Find Previous command button, Match case checkbox, and Match whole word checkbox.



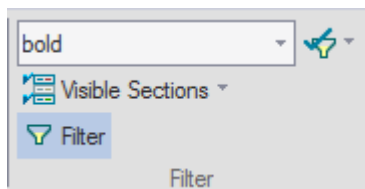
The items in the **Find** group operate as follows:

Item	Description
Find textbox	Finds the nodes with names matching the search string in the theme tree.
Find Next	Finds the next property with a matching name in the theme tree.
Find Previous	Find the previous property with a matching name in the theme tree.
Match case	Treats all search queries as case-sensitive.
Match whole word	Finds the exact word that you enter in the search textbox.

For more information see [Searching and Filtering Property Items in Theme Tree](#).

Filter Group

The Filter group appears on the Theme designer's Main ribbon menu.



The following table describes each item inside the Filter Group:

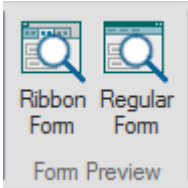
Item	Description
Name Filter textbox	Filters nodes visible in the theme tree by property names.
Visible Sections	Select which sections to show in the theme tree.

Filter button	Turn the property name in the theme tree on or off.
----------------------	---

For more information see [Searching and Filtering Property Items in Theme Tree](#).

Form Preview

The Form Preview groupbox appears on the Theme designer's Ribbon menu. Two buttons appear in the Form preview groupbox: Ribbon Form and Regular Form.

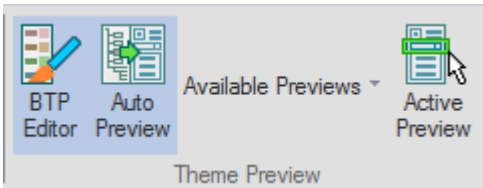


The following table describes each item inside the **Form Preview**:

Item	Description
Ribbon Form	Preview current theme applied to C1Controls (C1FlexGrid, C1Schedule, C1TrueDBGrid) in a C1RibbonForm. This opens in a new window.
Regular Form	Preview current theme applied to C1Controls (C1FlexGrid, C1Schedule, C1TrueDBGrid) in a WinForms with C1Command UI elements. This opens in a new window.

Theme Preview

The Form Preview groupbox appears on the Theme designer's Ribbon menu.



The following table describes each item inside the **Form Preview**:

Item	Description
BTP Editor	Preview current theme applied to C1Controls (C1FlexGrid, C1Schedule, C1TrueDBGrid) in a C1RibbonForm. This opens in a new window.
Auto Preview	Preview current theme applied to C1Controls (C1FlexGrid, C1Schedule, C1TrueDBGrid) in a WinForms with C1Command UI elements. This opens in a new window.
C1Command UI	Preview current theme applied to various C1Controls (C1TrueDBGrid, C1FlexGrid, C1Schedule) with C1Command menu and other UI elements.

C1Ribbon UI	Preview current theme applied to various C1Controls (C1TrueDBGrid, C1FlexGrid, C1Schedule) with a Ribbon-based UI.
C1Input (C1Command UI)	Preview current theme applied to various C1Input controls with C1Command menus and other UI elements.
C1Input (C1Ribbon UI)	Preview current theme applied to various C1Input controls with a C1Ribbon-based UI.
C1RadialMenu	Preview current theme applied to a C1RadialMenu.
C1GanttView	Preview current theme applied to a C1GanttView.
C1Report	Preview current theme applied to C1Report print preview controls.
C1SuperTooltip	Preview current theme applied to C1SuperTooltip and C1SuperLabel.
WinForms Controls	Preview current theme applied to standard winform controls.

Theme Designer Dialog Boxes

Theme Designer provides the following dialog box:

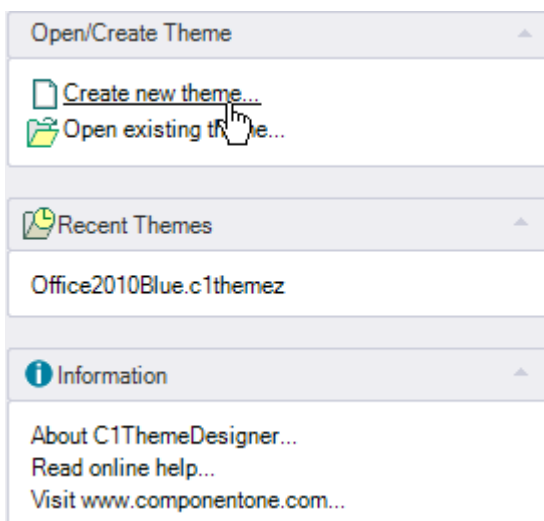
- [New Theme Dialog Box](#)
- [Add Property Dialog Box](#)

New Theme Dialog Box

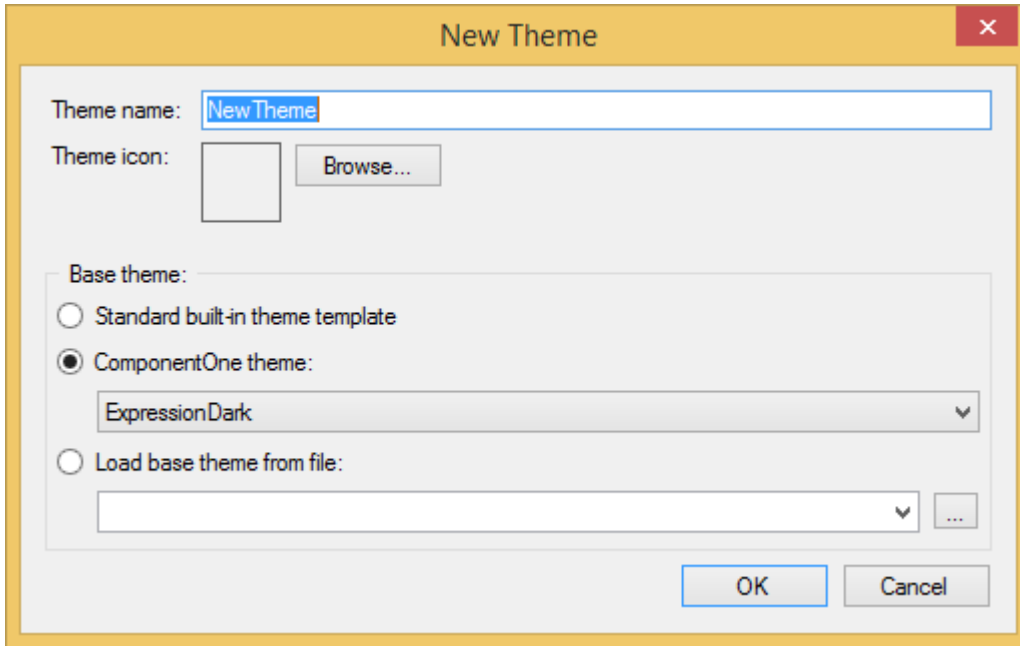
The New Theme dialog box can be used to create a new theme.

To Access the New Theme Dialog Box

Click on the **Create New Theme...** link in the Topic Bar:



The New Theme dialog box appears:

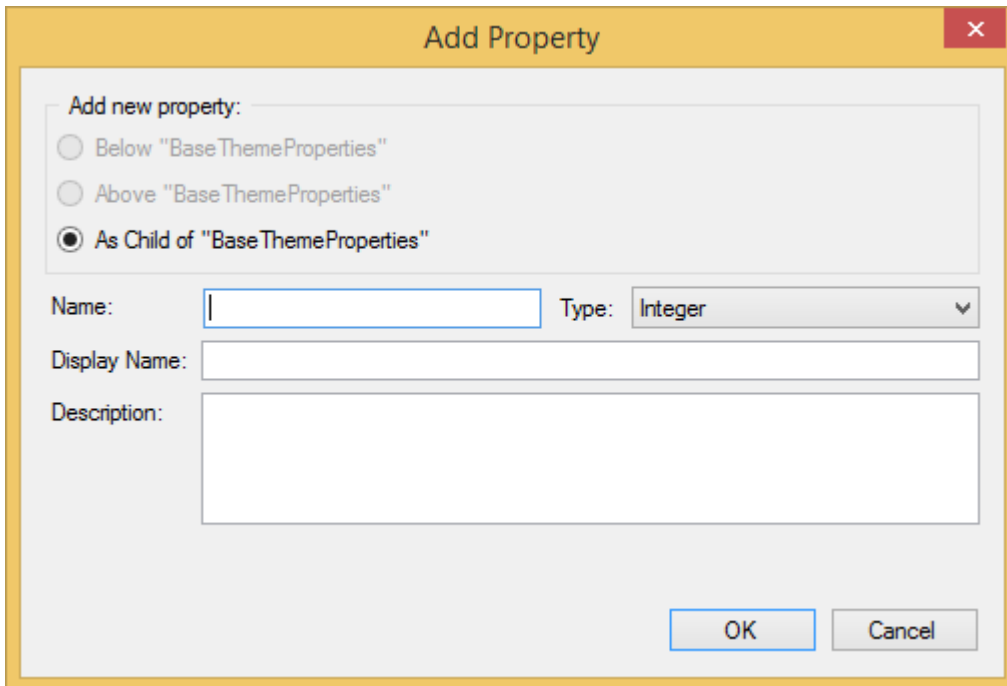


The **New Theme** dialog box consists of the following items:

- **Theme name:** - A textbox is provided for you to enter the name of your new theme.
- **Theme icon:** Initially an empty picturebox appears and is filled when you click the Browse button to open your theme icon.
- **Base Theme:** -Choose from one of the following three items to base your theme from:
 - Standard built-in theme template
 - ComponentOne theme – Select from one of the existing built-in themes.
 - Load base theme file: You can open an existing base theme you created to add it to your WinForms theme. When you enter a Theme name in the **Theme name:** textbox the new Theme name will appear at the top of the Theme Designer:

New Property Dialog Box

When you click on the **Add** command button from the Edit group, the **New Property** dialog box appears:



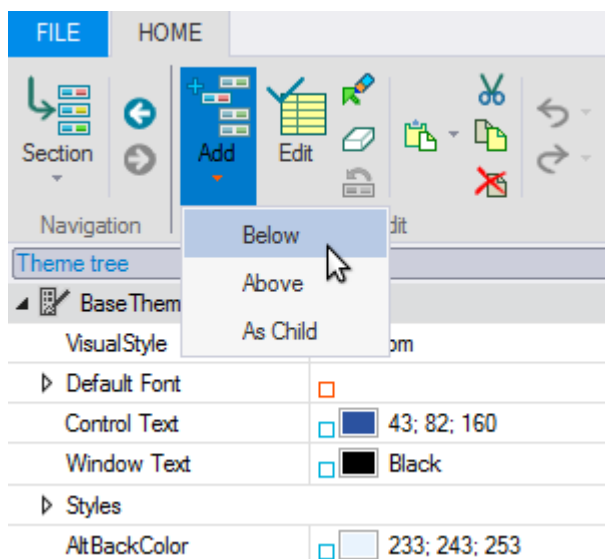
You can specify the name, property type, caption and description for the new property. For more information see [Adding New Property in Theme Tree](#).

Adding New Property in Theme Tree

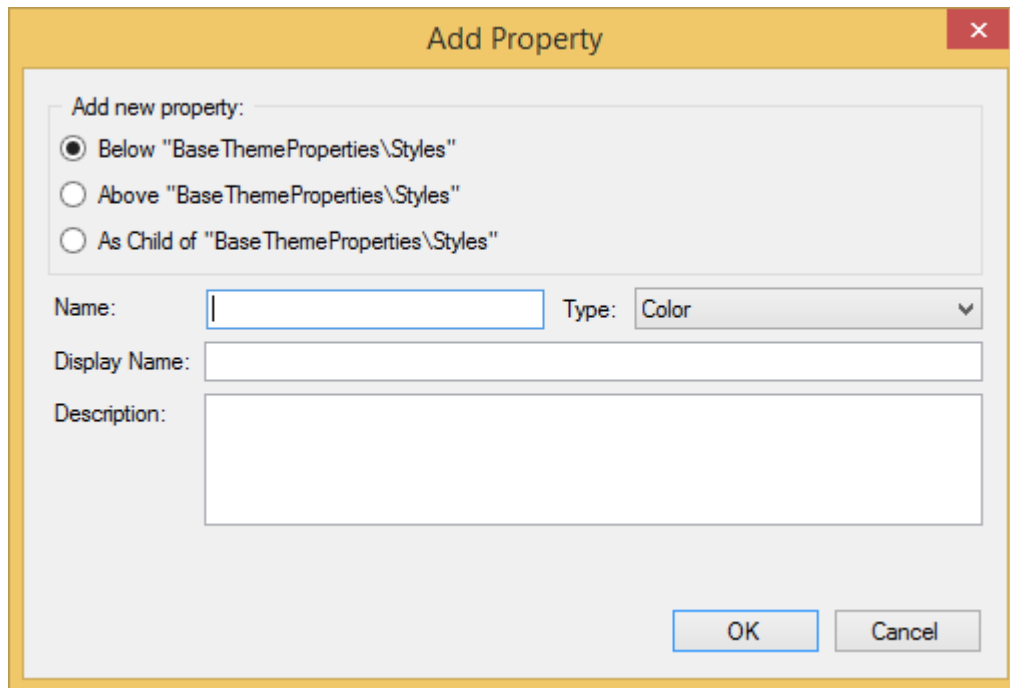
A new property can be inserted at a specific position in the Theme tree via the **Add** command button from the **Edit** group in the C1Theme Designer Ribbon menu.

To add a new property in the Theme Tree, complete the following:

1. Open the C1Theme designer and create a new theme based from any of the office themes such as **Office2010Blue**.
2. Select the **Styles** under the **BaseThemeProperties**.
3. Click **Below** from the **Add** dropdown menu in the **Edit** group.

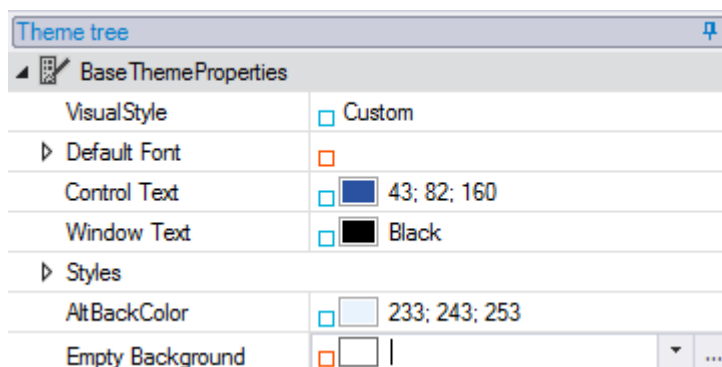


The **Add Property** dialog box appears:



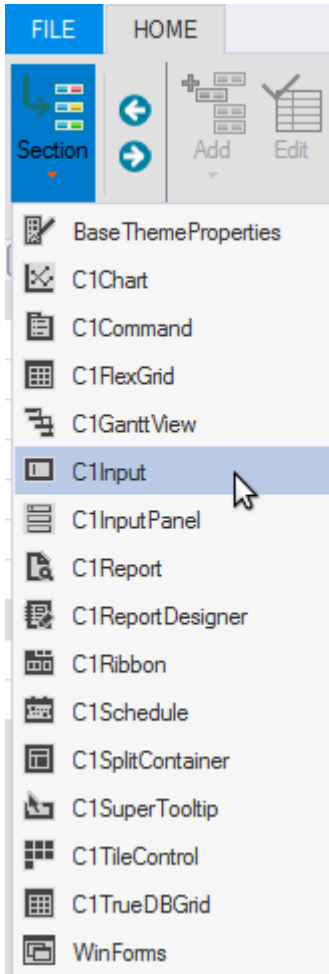
4. Enter Emptybackground in the **Name** textbox.
5. Select Background from the **Type** dropdown listbox.
6. In the **Description** richtextbox enter "Background for empty window areas".
7. Click **OK** in the **Add Property** dialog box.

The new property, Empty Background, appears below the Styles property in the BaseThemeProperties Theme Tree.



Navigating Property Items in Theme Tree

To quickly navigate to a section in the Theme tree, click on the dropdown arrow and select the section you wish to navigate in the Theme tree.



Clicking the **Back** button will move back to the last item selected in the Theme tree. Clicking the **Forward** button will move to the next item in the Theme tree.

Searching and Filtering Property Items in Theme Tree

The C1Theme Designer has a **Search** and **Filter** group. Use the **Search** textbox when you know what you want to find and use the **Filter** textbox when you don't know exactly what you want to find. Filtering is beneficial when you need to find common information.

Find Group

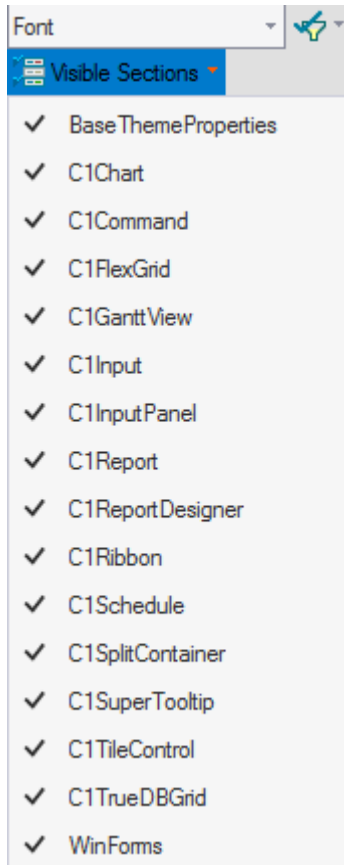
The following illustration labels each command inside the **Find** group.

To use the filter, complete the following:

Enter a property name like, **Font**, in the **Filter** search textbox.

Only the nodes in the Theme tree containing that text are shown.

Click the dropdown arrow in the **Visible sections** dropdown listbox and deselect all items except, **C1FlexGrid** and **C1Input**.



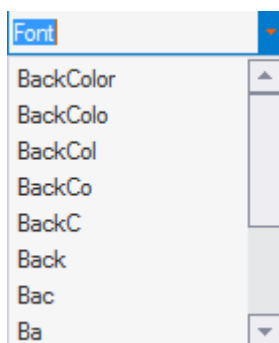
Click the **Filter** button in the **Filter** group.

This will expand the **C1FlexGrid** section with only the **Font** properties showing and the **C1Input** section with only the **Font** properties showing.

Note: When the Filter button is enabled the button will appear highlighted.

How to view past filters:

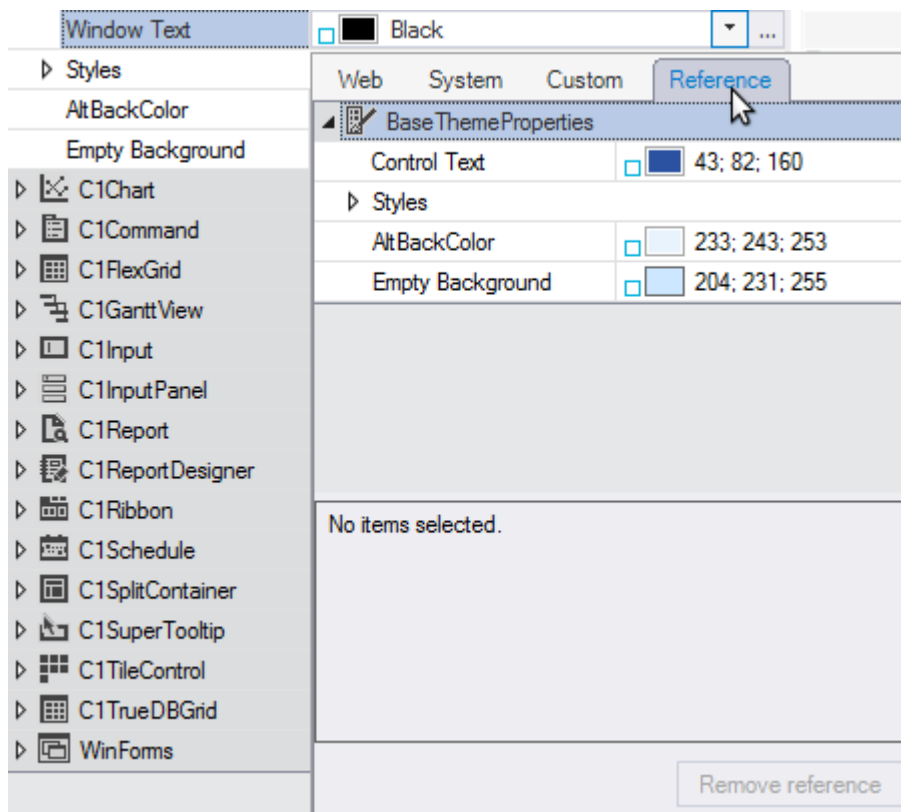
Past filters go into the history which can be brought up by clicking the dropdown in the Filter textbox. The following example shows the past filters:



Specifying Theme Property by Reference

To specify a theme property by reference, complete the following:

1. Double-click on the C1ThemeDesigner from the C:\Program Files\ComponentOne\WinForms Edition\bin\v4\Designer folder.
2. Click on **Create New Theme** from the Topic Bar. The **New Theme** dialog box appears.
3. Select the Standard built-in theme template and enter BlueandPink for the theme Name and Click **OK**.
4. Select **Control Text** from the BaseThemeProperties in the Theme tree.
5. Click on the Control Text dropdown arrow and click the **Custom** tab.
6. Set the **BackColor** to **151; 45; 85**.
7. Click on the dropdown arrow next to **Window Text** and select the **Reference** tab.
8. Select the **Control Text** in the Reference tab.



Notice the icon appears as a visual indicator that the **Window Text** property is referenced from the BaseThemeProperties **Control Text**.

