

# Themes for WinForms

2022.04.20 更新

グレープシティ株式会社

## 目次

<u>はじめに</u>	3
<u>コンポーネントをプロジェクトに組み込む方法</u>	3-4
<u>コンポーネントのランタイムファイル</u>	4-5
<u>クイックスタート</u>	6
<u>アプリケーションへの組み込みテーマの適用</u>	6-8
<u>マテリアルテーマの適用</u>	8-12
<u>カスタムテーマの作成</u>	12
<u>手順1: テーマデザイナでの新しいテーマの作成</u>	12-13
<u>手順2: Visual Studio でのテーマの使用</u>	13
<u>カスタムコントロールテーマの作成</u>	13-17
<u>.NETでテーマを作成</u>	17-19
<u>ThemeController の概要</u>	20
<u>テーマ指定可能な ComponentOne WinForms コントロール</u>	20
<u>テーマ指定可能な Microsoftコントロール</u>	20-21
<u>テーマの概念</u>	21
<u>テーマソース</u>	21-22
<u>テーマ名</u>	22-24
<u>C1Themes とビジュアルスタイルプロパティ</u>	24-25
<u>設計時サポート</u>	25
<u>C1ThemeController のコンテキストメニュー</u>	25-26
<u>C1ThemeController スマートタグ</u>	26-27
<u>C1ThemeController ダイアログボックス</u>	27-28
<u>C1ThemeDesigner アプリケーションの概要</u>	29-31
<u>テーマデザイナのリボンメニュー</u>	31
<u>[ファイル]タブ</u>	31-32
<u>[メイン]タブ</u>	32-36
<u>[編集]グループ</u>	36-37
<u>[検索]グループ</u>	37-38
<u>[フィルタ]グループ</u>	38
<u>フォームプレビュー</u>	38-39
<u>テーマのプレビュー</u>	39-40

<u>テーマデザイナのダイアログボックス</u>	40-41
<u>[新しいテーマ]ダイアログボックス</u>	41-42
<u>[プロパティの追加]ダイアログボックス</u>	42
<u>テーマツリーへの新しいプロパティの追加</u>	42-44
<u>テーマツリーにあるプロパティ項目間の移動</u>	44
<u>テーマツリー内のプロパティ項目の検索とフィルタリング</u>	44-47
<u>参照によるテーマプロパティの指定</u>	47-48
<u>C1ThemePicker</u>	49-50
<u>C1ThemePicker 設計時サポート</u>	50

## はじめに

**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 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.dll.

- **C1.Win.C1Themes.Extended assembly**

This assembly provides the C1ThemePicker control, which can be dropped from the toolbox to the form. The C1ThemePicker control represents a drop-down list to select one of the registered themes and apply it to the assigned C1ThemeController.

- **C1ThemeDesigner application**

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

## 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.
- The user can add a Theme Combo Box to switch between multiple themes at run-time. To know more, refer the [C1ThemePicker](#) topic.

 Note: ComponentOne Themes for WinForms is compatible with both .NET and .NET Framework.

API References	
<a href="#">C1.Win.C1Themes .NET Framework Assembly</a>	<a href="#">C1.Win.Themes .NET Assembly</a>

## コンポーネントをプロジェクトに組み込む方法

Visual Studio では、ツールボックスにコンポーネントを追加しただけでは、プロジェクトにコンポーネントを追加したことにはなりません。プロジェクトの参照設定へ追加された時点ではコンポーネントが組み込まれます。

以下のいずれかの操作を行うとプロジェクトへコンポーネントが組み込まれます。

1. フォームにコンポーネントを配置する
2. ソリューションエクスプローラ上で参照の追加を行う

プロジェクトに組み込まれているコンポーネントの一覧は、ソリューションエクスプローラで確認できます。また、各コンポーネントが使用している DLL もソリューションエクスプローラに登録される場合があります。詳細については、Visual Studio の製品ヘルプを参照してください。

本製品で使用しているコンポーネントの一覧を以下に示します。

ファイル	内容
C1.Win.C1Themes.4.5.2.dll	本体アセンブリ(※)

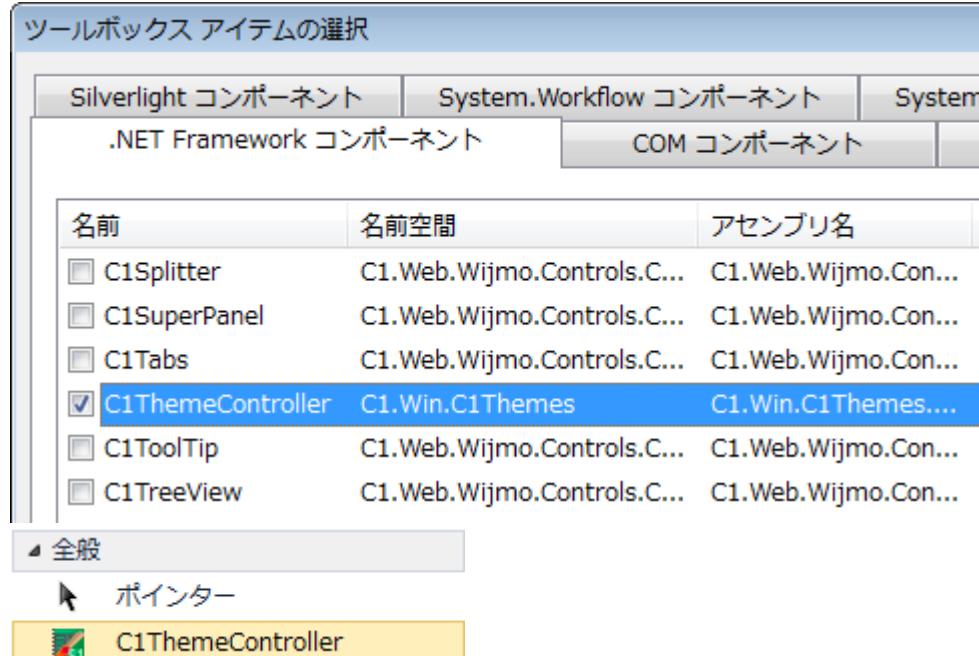
※ .NET Framework 4 以上でご利用いただけます。

### フォームにコンポーネントを配置する方法

以下に、C1Themes コントロールをツールボックスに追加し、フォームに配置する方法を示します。これにより、コンポーネントがプロジェクトに組み込まれます。

#### 配置手順

1. [ツール]メニューから[ツールボックス アイテムの選択]を選択します。
2. [ツールボックス アイテムの選択]ダイアログの「.NET Framework コンポーネント」タブを選択します。
3. 使用するコンポーネントのチェックボックスを ON にして〈OK〉ボタンをクリックしてください。ツールボックスに指定したコントロールのアイコンが表示されます。



4. ツールボックスから指定したコントロールのアイコンを選択してフォームに配置します。ソリューションエクスプローラの参照設定に指定したコントロールの名前空間が追加されます。

## コンポーネントのランタイムファイル

# Themes for WinForms

Themes for WinForms のランタイムファイルは、C:\Program Files\ComponentOne\WinForms\Bin\ フォルダにインストールされる次のファイルです。

ファイル	内容
C1.Win.C1Themes.4.5.2.dll	本体アセンブリ(※)

※ .NET Framework 4 以上でご利用いただけます。

 **注意:**本製品に含まれているファイルのうち、上記以外のファイルは配布できません。

## クイックスタート

Themes for WinForms の **C1ThemeDesigner** を使用して、さまざまな組み込みテーマをすばやく作成し、それらのテーマを任意の ComponentOne WinForms コントロールや標準 WinForms コントロールに簡単に適用できます。C1Themes for WinForms の **Theme** プロパティを使用すると、**C1ThemesDesigner** によって作成されたテーマを取得して、サポートされているすべての ComponentOne WinForm コントロールや標準 WinForms コントロールに適用できます。**C1ThemeController**.**Theme** プロパティを使用して、個々の ComponentOne WinForms コントロールや標準 WinForms コントロールに異なるテーマを適用することもできます。

このクイックスタートでは、アプリケーションに組み込みテーマを適用する方法と、C1ThemeDesigner を使用して既存のテーマを作成または変更する方法を習得します。

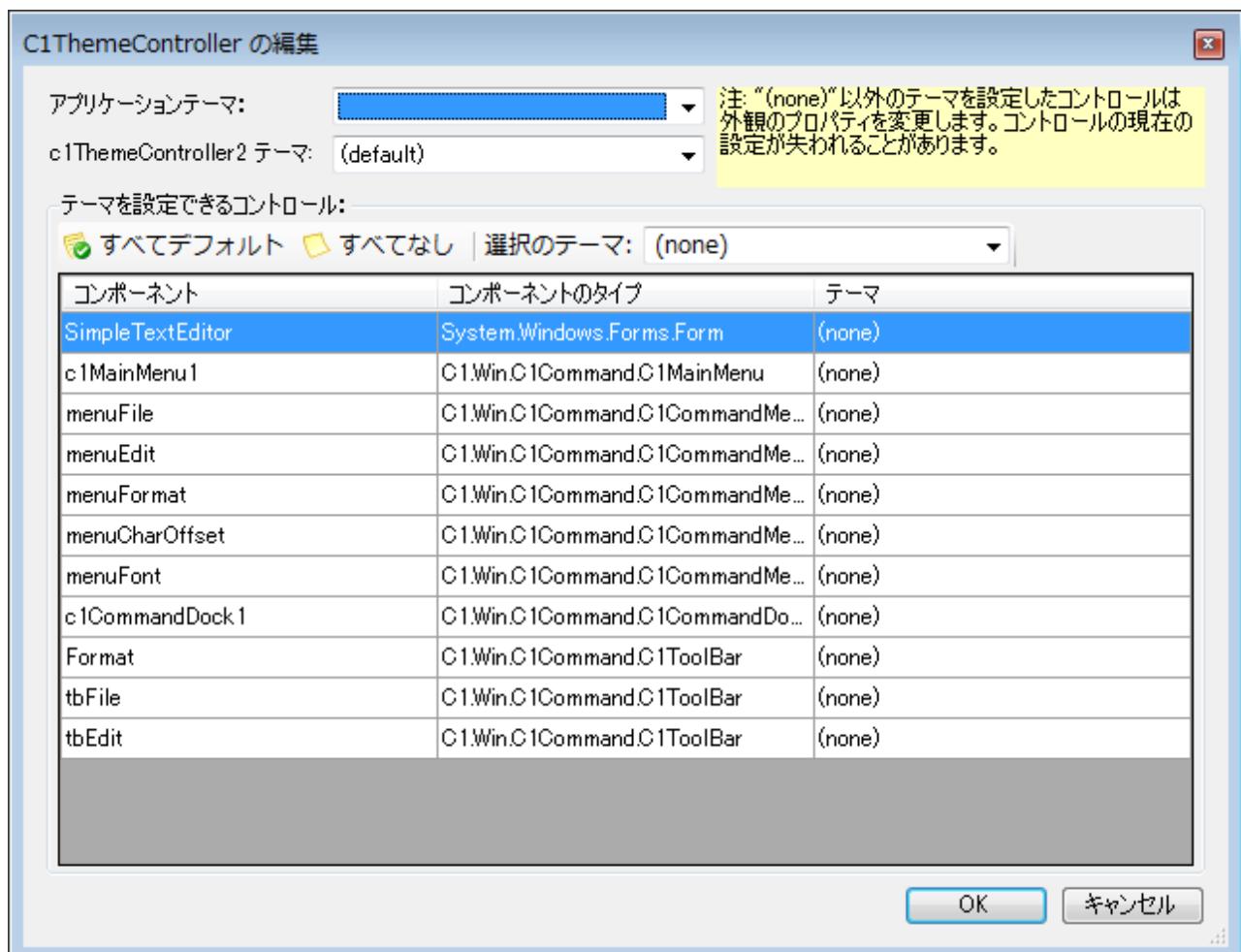
## アプリケーションへの組み込みテーマの適用

アプリケーション内のコントロールに簡単にテーマを適用するには、次のように C1ThemeController を使用します。

1. Visual Studio のフォームデザイナで、アプリケーションに属するフォームを開きます。この例では、ComponentOne WinForms サンプルの **SimpleTextEditor** を開きます。
2. デザイナのツールボックスから、C1ThemeController をドラッグしてフォームにドロップします。(C1ThemeController は C1.Win.C1Themes.4.5.2 アセンブリから提供されます。ツールボックスにない場合は、手作業で追加できます。)

**注意:** C1ThemeController を使用するプロジェクトでは C1.C1Zip.4.5.2 アセンブリを参照する必要があります。

ThemeController ダイアログボックスが表示されます。



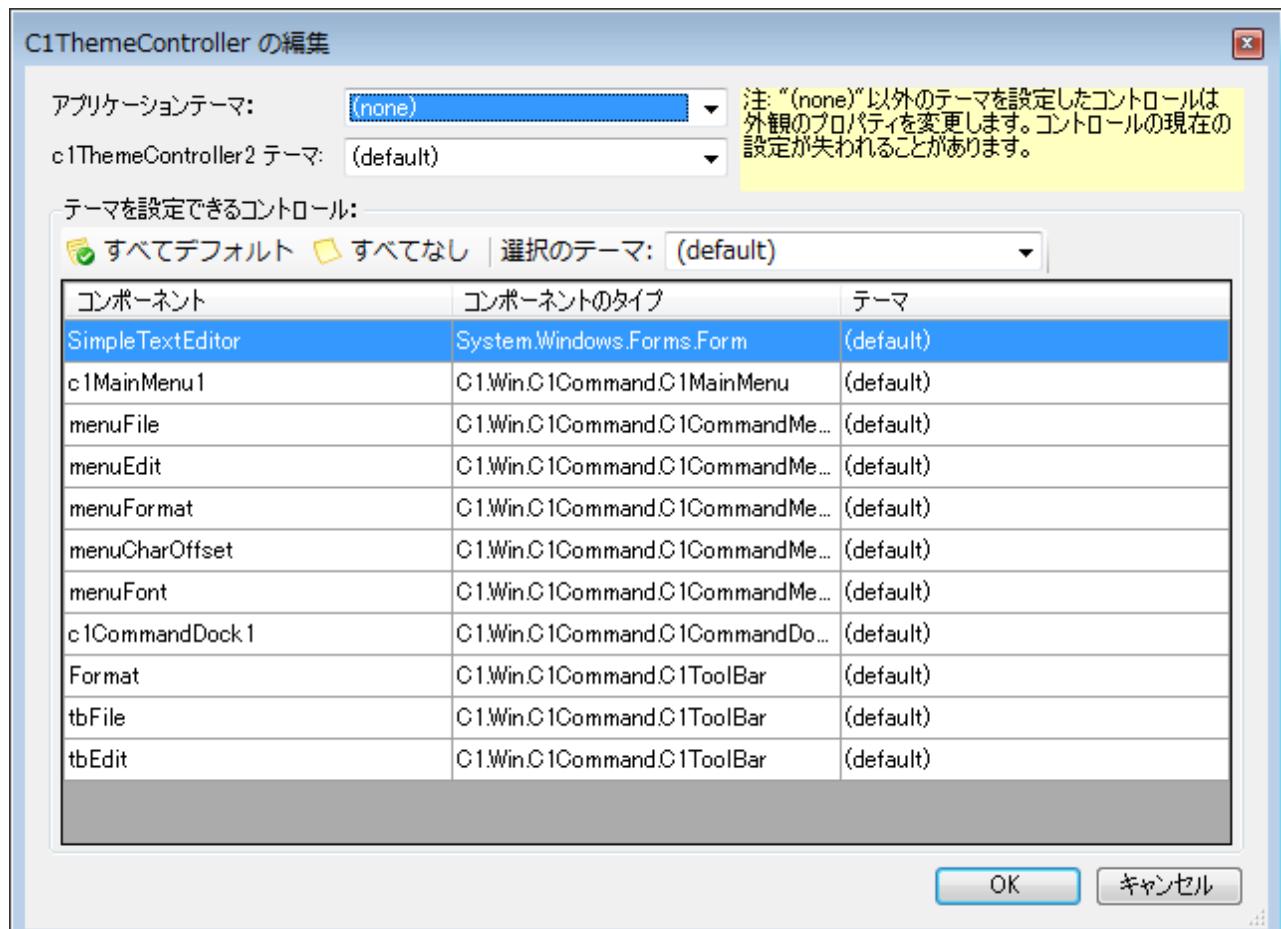
これで、アプリケーション全体のデフォルトテーマ、現在のコントローラのデフォルトテーマ、およびフォームに既にある

# Themes for WinForms

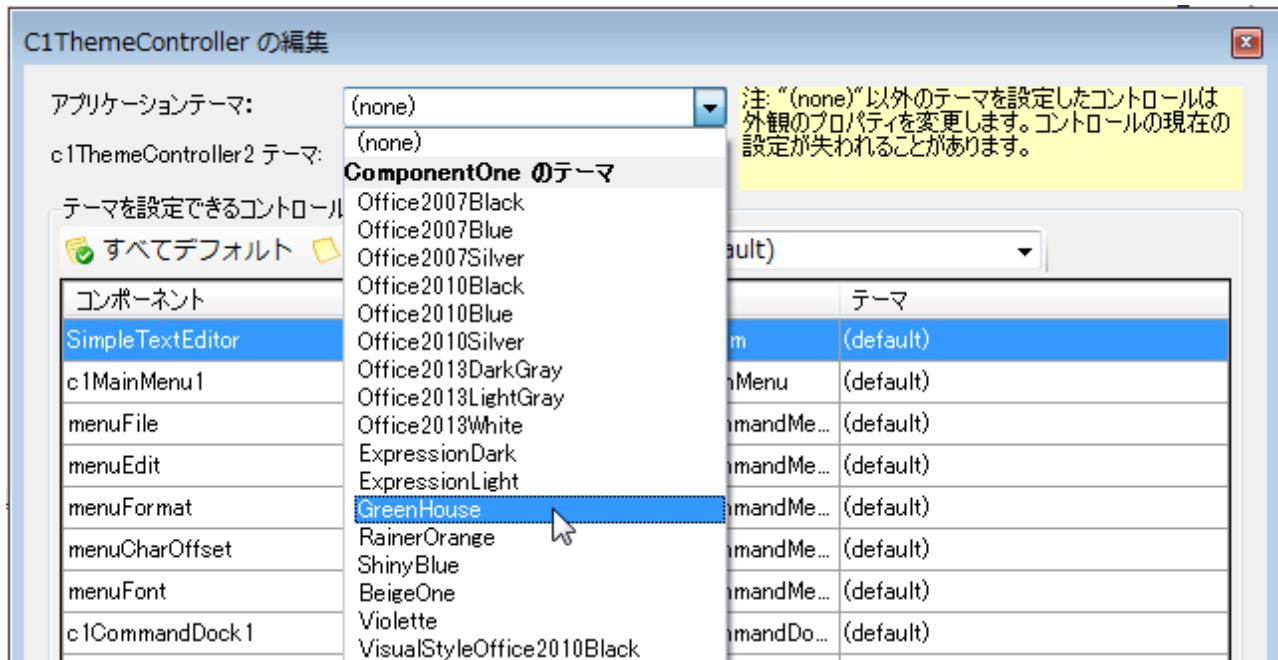
サポートされているすべてのコントロールに対するテーマを選択できます。メモ:このダイアログボックスは、[C1ThemeControllerのタスク]メニューの【テーマの設定】アクションから開くこともできます。最初にダイアログボックスが表示された際は、フォームに既にあるサポートされているすべてのコントロールに対するテーマが"(なし)"に設定されています。これは、それらのコントロールのプロパティ設定が意図せず失われないようにするためです。

3. ダイアログの【すべてを(デフォルト)に設定】ボタンをクリックして、すべてのコントロールにデフォルトのテーマを設定します。一部のコントロールが意図的にカスタマイズされている場合は、この操作を行う必要はありません。

 **注意:** C1ThemeController コンポーネントが既に含まれるフォームに新しいコントロールをドロップすると、それらのテーマは自動的に"(デフォルト)"に設定されます。



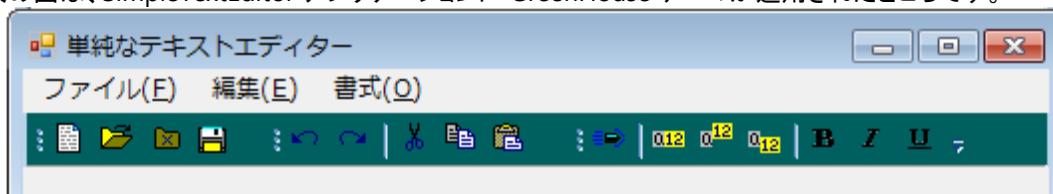
4. 有効な組み込みテーマのリストからデフォルトのアプリケーションテーマを選択します。このテーマ名は、App.config ファイルに自動的に保存されます。



5. 現在のテーマコントローラのテーマ('c1ThemeController1 テーマ')はデフォルト値 "デフォルト" のままにします。したがって、アプリケーションテーマ **GreenHouse** が使用されます。
6. 上で説明した [すべてを(デフォルト)に設定] ボタンをクリックして、フォームにあるサポートされているすべてのコントロールにデフォルトのテーマを設定します。
7. 最後に、[OK] ボタンをクリックしてダイアログを閉じ、フォーム内のコントロールに指定したテーマを適用します。
8. アプリケーションに複数のフォームがある場合は、アプリケーションのすべてのフォームに対して上の手順を繰り返します。ただし、デフォルトのアプリケーションテーマの指定は既に行われているので省略します。

完了したら、設計時と実行時のどちらの場合でも、選択したアプリケーションテーマがフォーム内のコントロールに適用されます。

次の図は、SimpleTextEditor アプリケーションに GreenHouse テーマが適用されたところです。



## マテリアルテーマの適用

To apply material theme in your application use the C1ThemeController as the following:

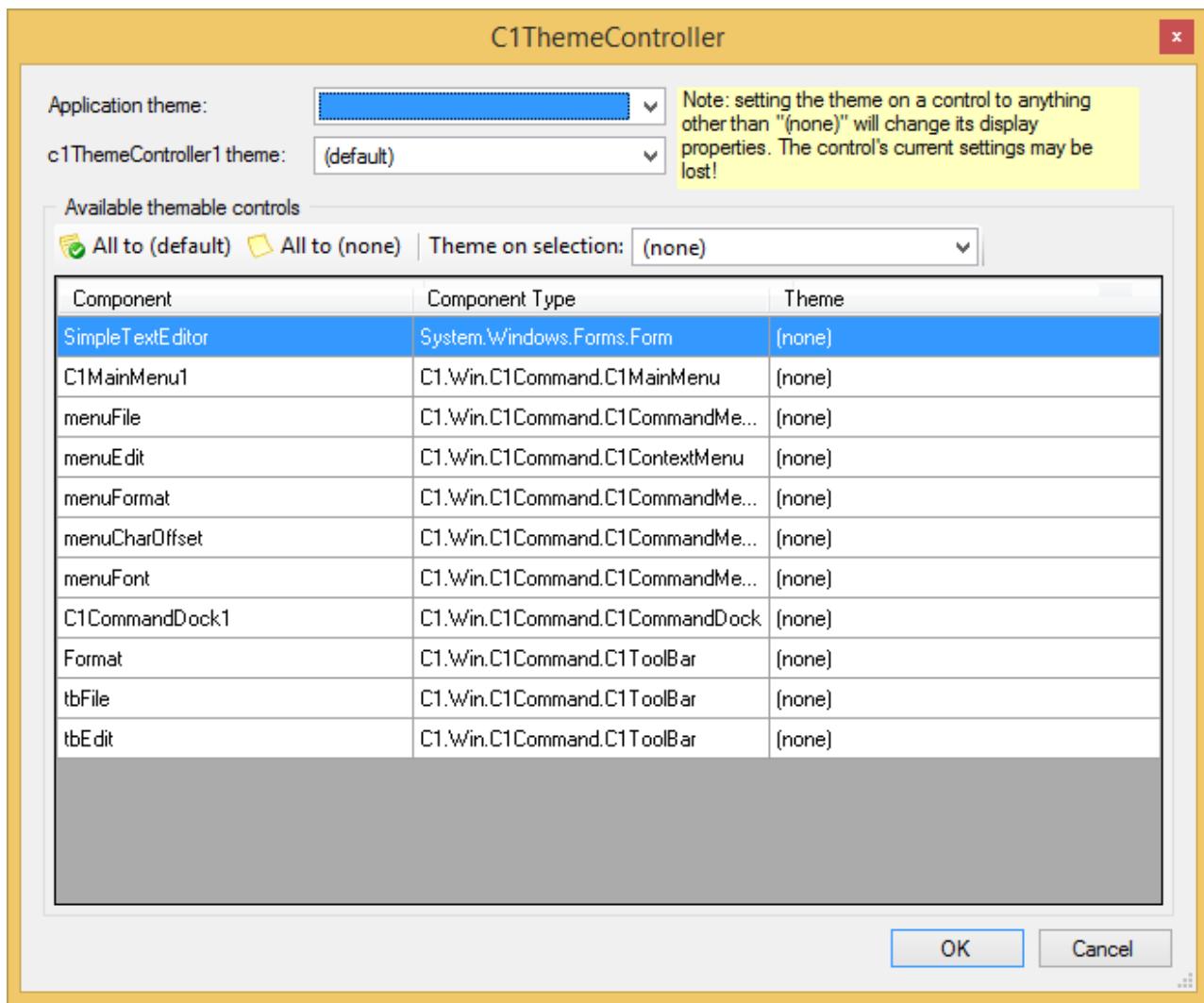
1. Open a form belonging to your application in the Visual Studio's forms designer. In this example we will open a ComponentOne WinForms sample, **SimpleTextEditor**.
2. From the designer's toolbox, drag a C1ThemeController and drop it on your form. (The C1ThemeController is provided by the C1.Win.C1Themes.4.5.2 assembly; if you don't see it in your toolbox you can add it manually).



**Note:** Reference to C1.C1Zip.4.5.2.dll is required in a project that uses C1ThemeController.

A ThemeController dialog box appears.

# Themes for WinForms

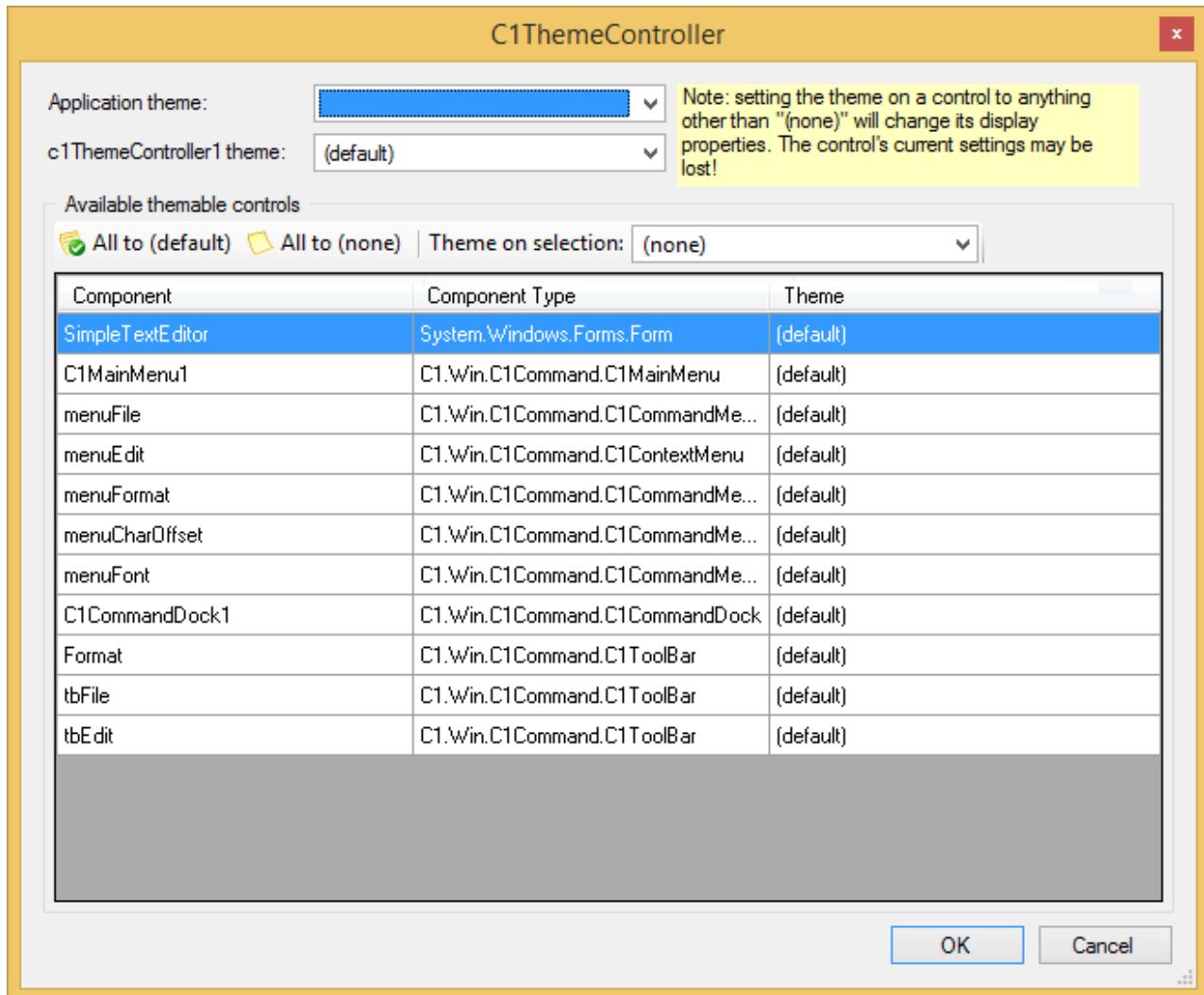


This enables you to select the application-wide default theme, the current controller's default theme, and themes for all supporting controls already on the form. Note: The same dialog is also available via the **Set Themes** action on the C1ThemeController tasks menu.

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.

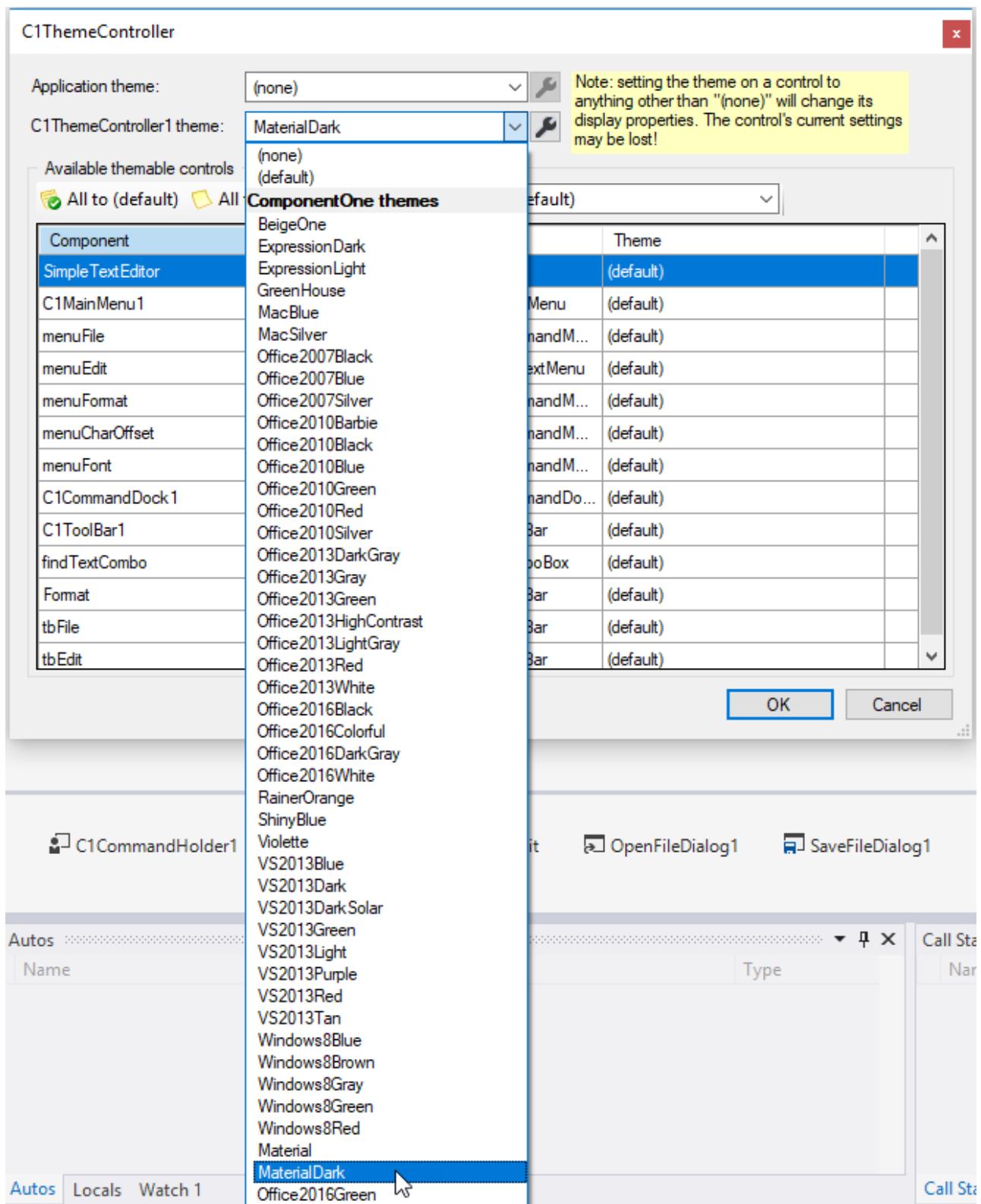
3. Click the **All to (default)** button in the dialog so that the default theme is set on all controls. If case you have deliberately customized some of the controls already this will not need to be done.

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



4. Select the material theme from the list of available prebuilt themes. In our case, we have selected **Material Dark** theme from the list as shown in the image below.

# Themes for WinForms



4. Click the **All to (default)** button as mentioned above, to set the default theme on all supporting controls on the form.
5. Finally, click the **OK** button to close the dialog and apply the specified theme to the controls on the form.
6. 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 MaterialDark theme applied to the SimpleTextEditor application.



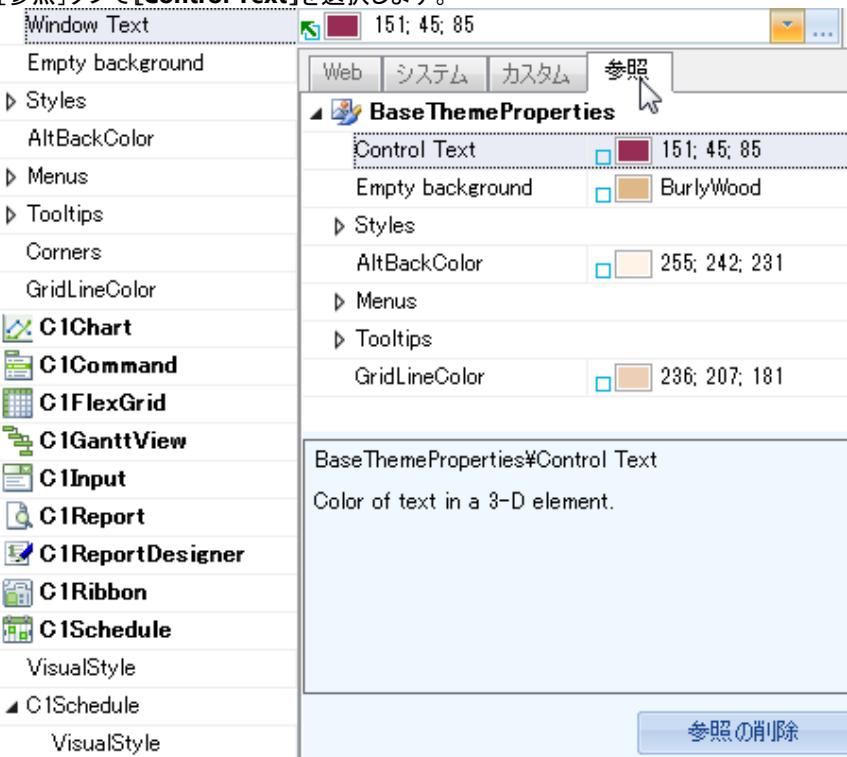
## カスタムテーマの作成

このクイックスタートガイドでは、**C1ThemeDesigner**を使用して、組み込みテーマテンプレートから新しいテーマを作成する手順について説明します。**C1ThemeDesigner**を使用して、組み込みテーマテンプレートを変更します。

### 手順1: テーマデザイナでの新しいテーマの作成

この手順では、標準の組み込みテーマテンプレートに基づいて新しいテーマを作成します。また、いくつかの基本テーマプロパティを変更し、別の基本テーマプロパティへの参照を追加します。詳細については、「[C1ThemeDesignerアプリケーションの概要](#)」を参照してください。

1. **C:\Program Files\ComponentOne\ComponentOne\Apps\v4** フォルダの **C1ThemeDesigner** をダブルクリックします。
2. トピックバーの【新しいテーマの作成】をクリックします。【新しいテーマ】ダイアログボックスが表示されます。
3. 標準の組み込みテーマテンプレートを選択し、テーマ名として「BlueandPink」と入力し、【OK】をクリックします。
4. テーマツリー内の【BaseThemeProperties】から【Control Text】を選択します。
5. 【Control Text】ドロップダウン矢印をクリックし、【カスタム】タブをクリックします。
6. **BackColor** を **151; 45; 85** に設定します。
7. 【Window Text】の横のドロップダウン矢印をクリックし、【参照】タブを選択します。
8. 【参照】タブで【Control Text】を選択します。



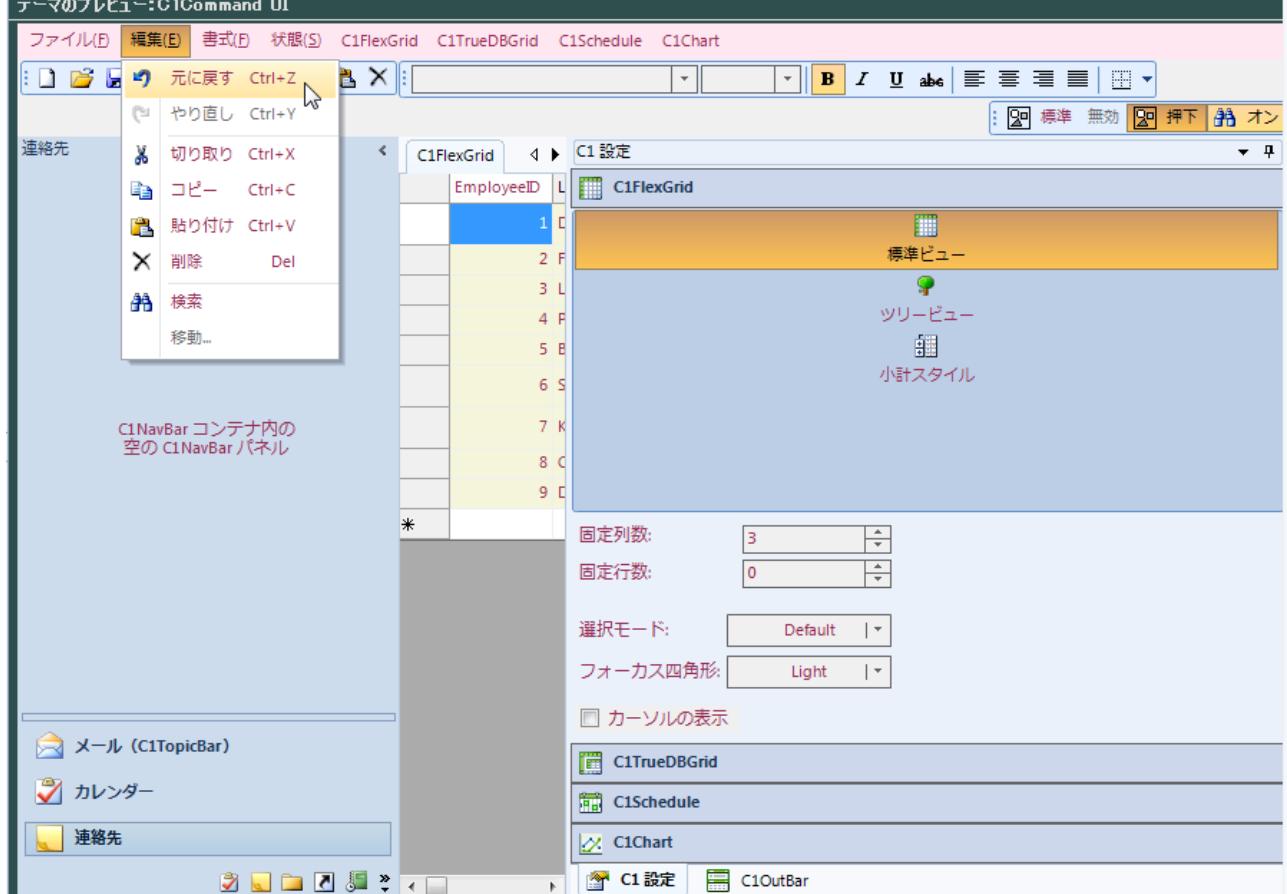
【Window Text】プロパティが【BaseThemeProperties】の【Control Text】から参照されることを示すアイコンがビジュアルリンクジェーティングとして表示されます。

# Themes for WinForms



9. **C1Command | C1MainMenu | Bar | BackColor** と展開します。
10. **BackColor** ドロップダウン矢印をクリックし、【カスタム】タブを選択します。
11. 明るいピンクを選択すると、**255; 231; 242** という数値が表示されます。

プレビューペインに新しい設定が表示されます。



12. 【ファイル】メニューから【ファイル】→【別名保存】を選択し、ファイルに名前を付けて保存します。たとえば、**BlueandPink** とします。C1ThemeDesignerの上部に新しい場所が表示されます。

## 手順2:Visual Studio でのテーマの使用

この手順では、アプリケーションにカスタムテーマを追加します。

1. Visual Studio のフォームデザイナーで、アプリケーションに属するフォームを開きます。この例では、ComponentOne WinForms サンプルの **SimpleTextEditor2** を開きます。
2. ツールボックスからフォームに **C1ThemeController** を追加します。そして、C1.C1Zip.2/4.dll への参照も設定してください。
3. C1ThemeController コンポーネントの【テーマの追加】アクション項目をクリックします。これで、ファイルを開くダイアログが表示されます。テーマファイルを選択し、[OK]をクリックします。  
選択したテーマが **Themes** フォルダ内の埋め込みリソースとして C# または VB プロジェクトに追加されます。このテーマは、C1ThemeController に表示されるテーマリストから選択できます。
4. プロジェクトを実行し、変更点を確認します。

## カスタムコントロールテーマの作成

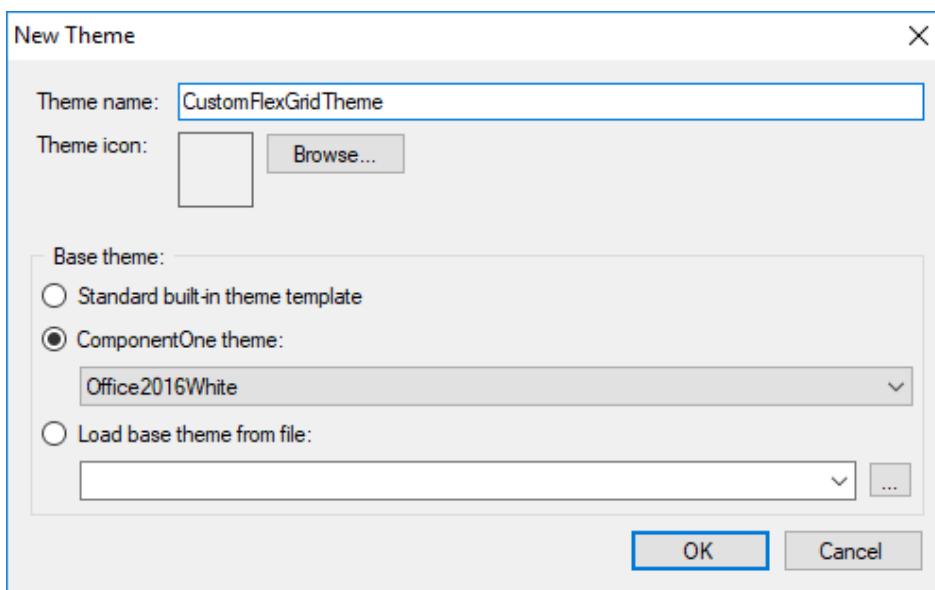
This QuickStart guide will walk you through creating a new theme for FlexGrid control based on a built-in theme template using the C1ThemeDesigner. The topic comprises of two steps:

- **Step 1: Create a New Theme using ThemeDesigner**
- **Step 2: Using Theme in Visual Studio**

### Step 1: Create a New Theme using ThemeDesigner

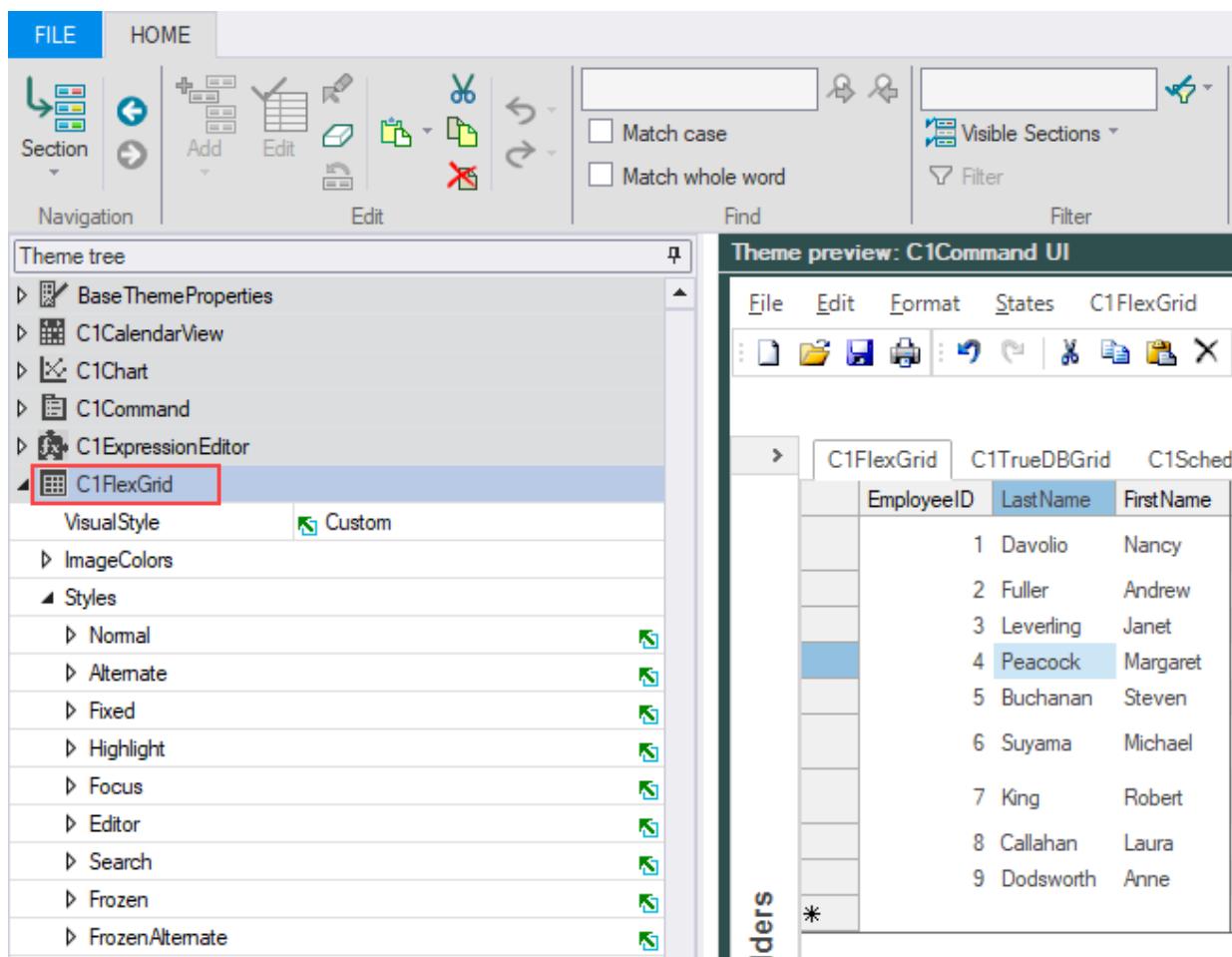
In this step, you will create a new theme for FlexGrid control based on the Standard built-in theme template. For more information, see [C1ThemeDesigner Application Overview](#).

1. Open the **C1ThemeDesigner** from the `C:\Program Files (x86)\ComponentOne\Apps\v4.0` folder.
2. Click on Create New Theme from the Topic Bar. The **New Theme** dialog box appears.
3. Enter a suitable name for the theme and select **Office2016White** from the ComponentOne theme dropdown.



4. Click the right-pointing arrow beside **C1FlexGrid** in the Theme tree pane and, then select **Styles**.

# Themes for WinForms



5. Expand the **Normal** section and set the following properties.

Property	Value
Background	White
Border > Color	212; 212; 212;
Border > Style	Flat
Border > Width	1
Font	Calibri; 11.25

6. Expand the **Alternate** section and set the following properties.

Property	Value
Background	White
Font	Calibri; 11.25

7. Expand the **Fixed** section and set the following properties.

Property	Value
Background	White
Border > Color	171; 171; 171
Border > Direction	Both

Border > Style	Flat
Border > Width	1
Font	Calibri; 11.25
ForeColor	ControlText

8. Expand the **Highlight** section and set the following properties.

Property	Value
Background	198; 198; 198
Border > Color	165; 165; 165
Border > Direction	Both
Border > Style	Flat
Border > Width	1
Font	Calibri; 11.25
ForeColor	68; 68; 68

9. Expand the **Focus** section and set the following properties.

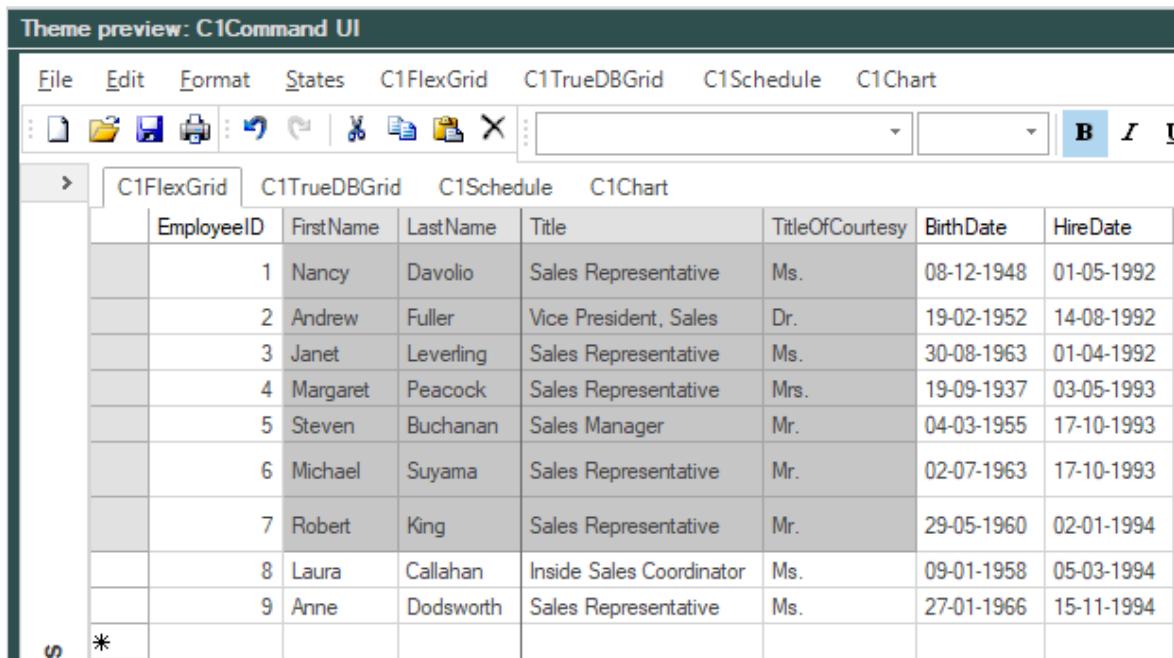
Property	Value
Background	White
Border > Color	33; 115; 70
Border > Direction	Both
Border > Style	Double
Border > Width	2

10. Expand the **SelectedColumnHeader** and set the following properties.

Property	Value
Background	225; 225; 225
ForeColor	Black

Notice that the changes made in the properties appear in the **Theme preview** pane.

# Themes for WinForms



11. Select **File | Save As** from the **File** menu to save the current theme in **.c1themez** file format.

## Step 2: Using Theme in Visual Studio

To add and use the custom control theme in your application, perform the following steps.

1. Open a form belonging to your FlexGrid application in the Visual Studio's forms designer.
2. Add the **C1ThemeController** from the toolbox onto your form. Make sure to 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.
4. Browse and select the theme file (**.c1themez**) in your system and then 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**.
5. Run the application and observe the theme applied on your control.

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder
1	Chai	1	1	10 boxes x 20 bags	18	39	
2	Chang	1	1	24 - 12 oz bottles	19	17	
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13	
4	Chef Anton's Ca	2	2	48 - 6 oz jars	22	53	
5	Chef Anton's Gu	2	2	36 boxes	21.35	0	
6	Grandma's Boys	3	2	12 - 8 oz jars	25	120	
7	Uncle Bob's Org	3	7	12 - 1 lb pkgs.	30	15	
8	Northwoods Cra	3	2	12 - 12 oz jars	40	6	
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	29	
10	Ikura	4	8	12 - 200 ml jars	31	31	
11	Queso Cabrales	5	4	1 kg pkg.	21	22	
12	Queso Manchego	5	4	10 - 500 g pkgs.	38	86	
13	Konbu	6	8	2 kg box	6	24	
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35	
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39	
16	Pavlova	7	3	32 - 500 g boxes	17.45	29	
17	Alice Mutton	7	6	20 - 1 kg tins	39	0	
18	Canned Tiger Prawn	7	9	16 kg pkgs.	225	47	

## .NETでテーマを作成

This topic showcases the implementation of a simple themes application which includes a C1Ribbon and ComboBox with a list of all the inbuilt themes.

To create a simple WinForms application in .NET 5 for C1Themes control, complete the following steps:

1. Initialize the C1Themes object.

- o C#

```
// C1Themeオブジェクトを初期化します
C1Theme theme;
```

2. Initialize the C1Themes class.

- o C#

```
// C1Themesクラスを初期化します
theme = new C1Theme();
```

3. Add the following code to the form load event:

- o C#

```
theme = null;
// C1ThemeControllerを追加します
cmb_list.Items.Clear();
string[] themes = C1ThemeController.GetThemes();
cmb_list.Items.Add("(No Theme)");
foreach (string theme in themes)
    cmb_list.Items.Add(theme);
cmb_list.SelectedIndex = 0;
```

4. Add a combobox with name cmb\_list and button with name btn\_apply.

5. Add the following code to the SelectedIndexChanged event of the ComboBox.

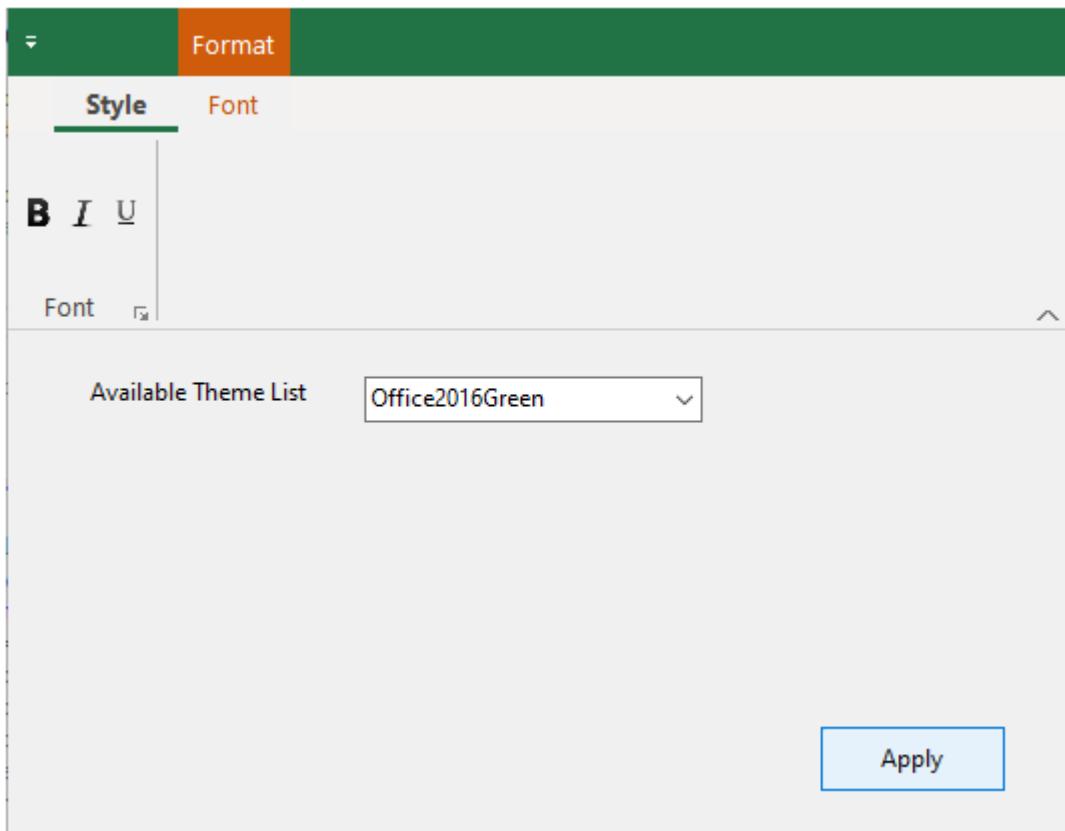
- o C#

```
private void cmb_list_SelectedIndexChanged(object sender, EventArgs e)
{
    C1Theme theme = null;
    try
    {
        theme = C1ThemeController.GetThemeByName(cmb_list.Text, false);
    }
    catch
    {
    }
    if (theme != null)
    {
        C1ThemeController.ApplyThemeToControlTree(c1Ribbon1, theme);
    }
}
```

The GetThemeByName method in C1ThemeController class retrieves a theme registered with the application, while the ApplyThemeToControlTree method recursively applies a theme to a control.

6. Run the code and observe the output.

# Themes for WinForms



Also, you can directly apply the specific theme using the following code:

C#

copyCode

```
var theme = C1ThemeController.GetThemeByName("Office2016Green", false);  
// フォーム上のすべてのコントロールにテーマを適用します  
C1ThemeController.ApplyThemeToControlTree(this, theme);
```

Note that in the above code, in place of C1Ribbon, you can use a control name specific to your requirement.

You can also check the product sample specific to the Ribbon implementation with the Themes control, at [Documents\ComponentOne Samples\WinForms\C1Themes\C1RibbonBasedApp](#).



**Note:** WinForms .NET 5 Edition does not include rich design-time support yet. We will enhance it in future releases.

## ThemeController の概要

C1ThemeController は、ComponentOne のビジュアルテーマをロードおよび管理し、それらをフォームやコントロールに適用するためのコンポーネントです。C1ThemeController は、以下の方法で使用できます。

- アプリケーション内のサポートされているすべての ComponentOne コントロールに同じテーマを適用する
  - それには、**C1ThemeController.ApplicationTheme** プロパティを使用します。
- フォーム内のサポートされているすべての ComponentOne コントロールに同じテーマを適用する
  - それには、**C1ThemeController.Theme** プロパティを使用します。
- フォーム内のサポートされているすべての ComponentOne コントロールに異なるテーマを適用する
  - それには、フォーム内の各コントロールの拡張プロパティ 'c1ThemeController1 のテーマ' を設定します。

## テーマ指定可能な ComponentOne WinForms コントロール

Themes for WinForms を効果的に使用するには、C1ThemeController コンポーネントと、サポートされているバージョンの ComponentOne WinForms コントロールが必要です。テーマデザイナアプリケーションは、WinForms コントロールにテーマを適用するためには必要ありませんが、新しいテーマをカスタマイズしたり作成する場合に使用できます。

**Themes for WinForms** は、以下の **ComponentOne WinForms** コントロールをサポートします。

ComponentOne WinForms 製品	コントロール名
<b>Chart for WinForms</b>	C1Chart および C1Chart3D
<b>FlexGrid for WinForms</b>	C1FlexGrid
<b>GanttView for WinForms</b>	C1GanttView
<b>Input for WinForms</b>	C1Textbox, C1DateEdit, C1NumericEdit, C1DropDownControl, C1DbNavigator, C1CheckBox, C1Button, C1Label, C1PictureBox, C1SplitButton, C1ComboBox
<b>InputPanel for WinForms</b>	C1InputPanel
<b>Menus and ToolBars for WinForms</b>	C1DockingTab, C1MainMenu, C1ContextMenu, C1ToolBar, C1TopicBar, C1NavBar, C1OutBar, C1RadialMenu
<b>Report for WinForms</b>	C1Report, C1PrintPreview, C1PreviewPane, C1PreviewThumbnailView, C1PreviewOutlineView, C1PreviewTextSearchPanel, and C1PrintPreviewDialog. C1ReportDesigner
<b>Ribbon for WinForms</b>	C1Ribbon and C1StatusBar
<b>Scheduler for WinForms</b>	C1Schedule and C1Calendar
<b>SplitContainer for WinForms</b>	C1SplitContainer
<b>SuperToolTip for WinForms</b>	C1SuperToolTip, C1SuperErrorProvider, C1SuperLabel
<b>TileControl for WinForms</b>	C1TileControl
<b>TrueDBGrid for WinForms</b>	C1TrueDBGrid

## テーマ指定可能な Microsoft コントロール

アプリケーション内で使用される Microsoft コントロールに **C1Themes** を適用して、アプリケーションをカスタマイズできます。

# Themes for WinForms

Themes for WPF は、以下の Microsoft コントロールをサポートします。

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

## テーマの概念

テーマは、サポートしているコントロールの外観を決定するプロパティに対して統一的に値を設定するための XML ドキュメントです(ファイルに保存すると **.c1theme** 拡張子が付く)。(C1ThemeController コンポーネントを使用して実行時や設計時に) アプリケーション内のフォームにテーマを適用すると、そのフォーム内のすべてのコントロールの外観がテーマに従って変更されます。

内部では、テーマはさまざまな ComponentOne 製品に対応するいくつかのセクションに分かれています。テーマが適用されると、各製品はそれぞれのセクションだけを使用します。これにより、どれほど小さく特殊なコントロールの細部にも、カスタマイズした「テーマ付きの」外観を適用できます。テーマの統一感は参照メカニズムによって実現されます。これは、個々のコントロールの色やフォントといったプロパティが、テーマの他のすべてのセクションからアクセス可能な共通のリポジトリを参照できるようにします。この共通リポジトリを「基本テーマプロパティ」と呼びます。このリポジトリ内のプロパティはコントロールのプロパティに直接適用されませんが、コントロールで使用される実際のプロパティの指定時に、テーマの作成者がそれらのプロパティを参照できます。さらに、テーマには、標準の WinForms フォームや、ボタン、ラベルなどのコントロールにテーマを適用するために C1ThemeController によって使用されるセクションがあります。

## テーマソース

アプリケーションでは、次の場所からテーマを取得できます。

- C1.Win.C1Themes.4.5.2 アセンブリにリソースとして埋め込まれた標準の C1 テーマ
- C1ThemeDesigner アプリケーションを使用して作成し、独自のアセンブリに埋め込みリソースとして追加したテーマ
- 実行時に **.c1theme** ファイルからロードされるテーマ

最初の2つのテーマソースは、設計時に C1ThemeController のプロパティとデザイナのアクションから取得できます。実行時には、C1ThemeController コンポーネントのメソッドを使用して、ディスク上の **.c1theme** ファイルからコードでテーマをロードすることもできます。

たとえば、次のコードは、**Register** メソッドと **SetTheme** メソッドを使用して、プログラムから組み込みテーマを適用する方法

を示します。

#### Visual Basic

```
'C1ThemeController を使用してテーマファイルを登録します
C1.Win.C1Themes.C1ThemeController.RegisterTheme("C:\Users\Documents\Visual Studio
2010\Projects\ThemesBuild6\ThemesBuild6\NewOffice2013White.c1theme")
'ファイル名ではなくテーマ名を使用してテーマをコントロールに適用します
Me.c1ThemeController1.SetTheme(someControl, "NewOffice2013White")
```

#### C#

```
//C1ThemeController を使用してテーマファイルを登録します
C1.Win.C1Themes.C1ThemeController.RegisterTheme(@"C:\Users\Documents\Visual Studio
2010\Projects\ThemesBuild6\ThemesBuild6\NewOffice2013White.c1theme");
//テーマ名を使用してテーマをコントロールに適用します
this.c1ThemeController1.SetTheme(c1MainMenu1, "NewOffice2013White")
```

## テーマ名

**C1Themes** では、テーマがそれぞれの名前で識別されます。名前は大文字と小文字が区別され、アプリケーション内で一意でなければなりません。

WinForms のテーマには、以下に示す.NET 5および.NET 4.5.2アセンブリのテーマ名が格納されています。

## .NET 5で利用可能なテーマ

- ExpressionDark
- ExpressionLight
- HighContrast
- MacBlue
- MacSilver
- Material
- Material Dark
- Office2016Black
- Office2016Colorful
- Office2016DarkGray
- Office2016Green
- Office2016White
  
- ExpressionDark
- ExpressionLight
- HighContrast
- MacBlue
- MacSilver
- Material
- Material Dark
- Office2016Black
- Office2016Colorful
- Office2016DarkGray
- Office2016Green
- Office2016White

## .NET 4.5.2で利用可能なテーマ

- BeigeOne
  - ExpressionDark
  - ExpressionLight
  - GreenHouse
  - MacBlue
  - MacSilver
  - Material
  - Material Dark
  - Office2007Black
  - Office2007Blue
  - Office2007Silver
  - Office2010Barbie
  - Office2010Black
  - Office2010Blue
  - Office2010Green
  - Office2010Red
  - Office2010Silver
  - Office2013DarkGray
  - Office2013Gray
  - Office2013Green
  - Office2013HighContrast
  - Office2013LightGray
  - Office2013Red
  - Office2013White
  - Office2016Black
  - Office2016Colorful
  - Office2016DarkGray
  - Office2016Green
  - Office2016White
  - RanierOrange
  - ShinyBlue
  - Violette
  - VS2013Blue
  - VS2013Dark
  - VS2013DarkSolar
  - VS2013Green
  - VS2013Light
  - VS2013Purple
  - VS2013Red
  - VS2013Tan
  - Windows8Blue
  - Windows8Brown
  - Windows8Gray
  - Windows8Green
  - Windows8Red
- 
- BeigeOne
  - ExpressionDark
  - ExpressionLight
  - GreenHouse
  - MacBlue
  - MacSilver
  - Material

- Material Dark
- Office2007Black
- Office2007Blue
- Office2007Silver
- Office2010Barbie
- Office2010Black
- Office2010Blue
- Office2010Green
- Office2010Red
- Office2010Silver
- Office2013DarkGray
- Office2013Gray
- Office2013Green
- Office2013HighContrast
- Office2013LightGray
- Office2013Red
- Office2013White
- Office2016Black
- Office2016Colorful
- Office2016DarkGray
- Office2016Green
- Office2016White
- RanierOrange
- ShinyBlue
- Violette
- VS2013Blue
- VS2013Dark
- VS2013DarkSolar
- VS2013Green
- VS2013Light
- VS2013Purple
- VS2013Red
- VS2013Tan
- Windows8Blue
- Windows8Brown
- Windows8Gray
- Windows8Green
- Windows8Red

Themes for WinForms では、以下に示す非標準のテーマ名も使用されます。

- "(デフォルト)": 現在のコンテキストのデフォルトテーマを指定します。
- "(なし)": テーマを適用しないことを指定します。

## C1Themes とビジュアルスタイルプロパティ

多くの ComponentOne WinForms コントロールには、ビジュアルスタイルと呼ばれるプロパティがあります。このプロパティは、コントロールアセンブリで定義される列挙型ですが、型と設定可能な値は共通の命名パターンに従っています。通常、この列挙型は **VisualStyle** という名前で、**Office2010Blue**、**Office2010Black** などの値が含まれます。C1Themes は、これまでより多機能で柔軟なメカニズムによってコントロールの外観を調整できるように設計されています。ただし、この2つのメカニズムの間には明らかな重複があり、これらが競合する可能性があります。この重複は、次の規則に基づいて処理されます。

- すべての C1 コントロールのテーマセクションには、そのテーマで指定できる VisualStyle プロパティが含まれます。
- 用意されているすべてのテーマのプロパティはデフォルトで 'Custom' に設定されます。これは、この VisualStyle が他のプロパティの適用を妨げないようにします。

# Themes for WinForms

- テーマの VisualStyle をカスタム以外に設定すると、ターゲットコントロールの対応するプロパティが設定され、他のすべてのテーマプロパティの適用が無効になります。(C1ThemeDesigner では、コントロールのテーマツリーの残りの部分が実際に無効になります。)

下位互換性などを考慮して、テーマではなく VisualStyle を使用してアプリケーションの外観のカスタマイズしなければならないこともあります。可能であれば、多機能で柔軟なメカニズムを備えた C1Themes の方を使用することをお勧めします。新しいコントロールでのビジュアルスタイルのサポートは段階的に廃止され、テーマに置き換えられる予定です。

## 設計時サポート

C1ThemeController に含まれるスマートタグといくつかのダイアログボックスを使用して、**C1ThemeController** コンポーネントを使用した設計時のタスクを簡略化できます。

以下のトピックでは、C1ThemeController の設計時環境を使用して C1ThemeController を構成する方法について説明します。

## C1ThemeController のコンテキストメニュー

**C1ThemeController** コンポーネントは、設計時に使用される機能のコンテキストメニューを提供します。

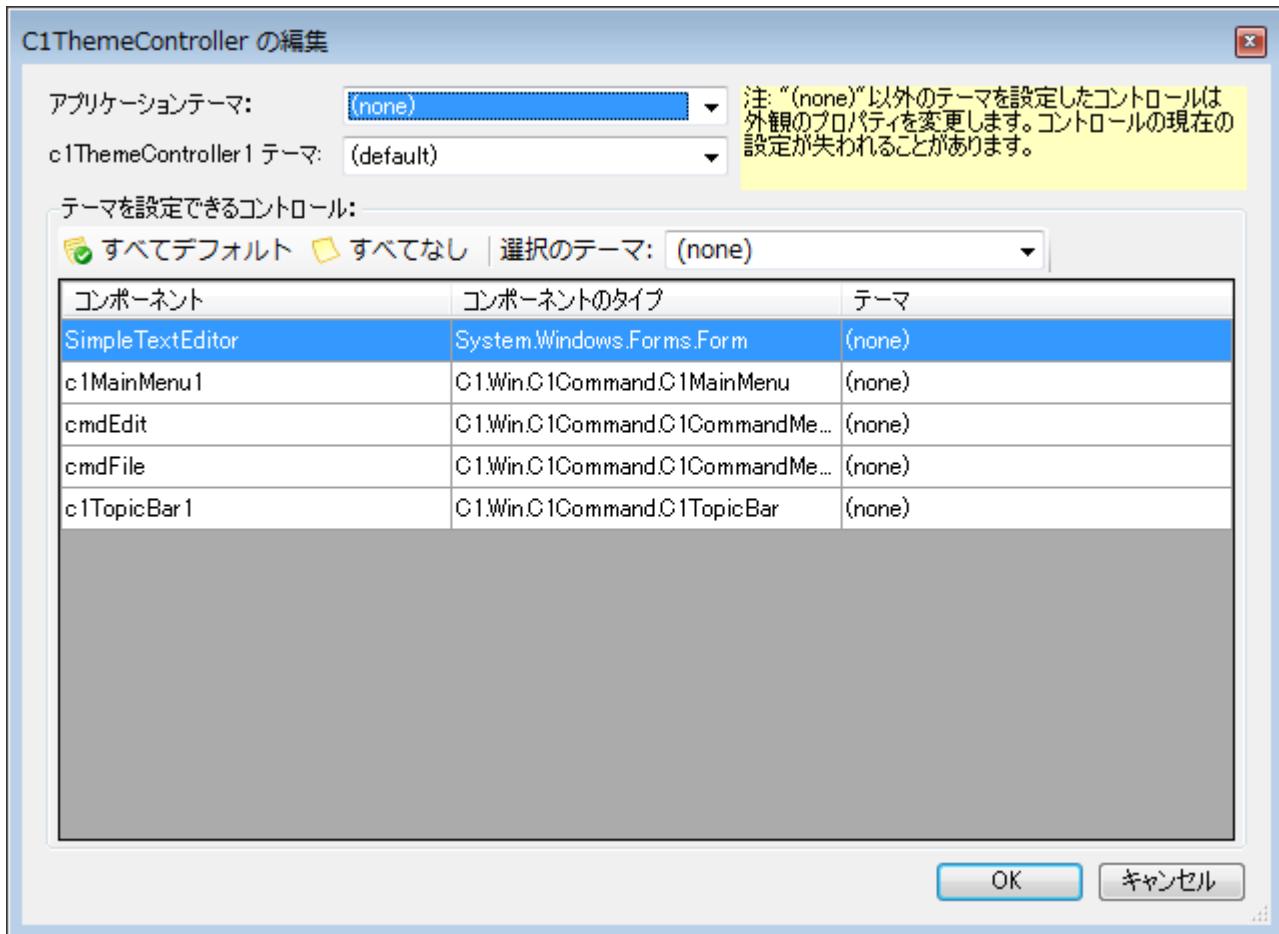
C1ThemeController のコンテキストメニューにアクセスするには、C1ThemeController コントロールを右クリックします。次のようなコンテキストメニューが表示されます。



C1ThemeController コンテキストメニューの機能は次のとおりです。

### テーマの設定

[テーマの設定]をクリックすると、[C1ThemeController]ダイアログボックスが開きます。詳細については、「[C1ThemeController ダイアログボックス](#)」を参照してください。



## テーマの追加

[テーマの追加]をクリックすると、[テーマの選択]ダイアログボックスが開きます。このダイアログボックスで、テーマの場所を参照し、それを C1ThemeController に追加できます。テーマを選択すると、それがプロジェクトに登録され、Visual Studio ではソリューションエクスプローラにテーマフォルダが表示されます。テーマを追加したら、Theme プロパティまたは ApplicationTheme プロパティにそれを適用します。

## テーマの再読み込み

[テーマの再読み込み]をクリックすると、現在のアプリケーションで使用されるすべてのテーマを再読み込みます。

## C1ThemeController のバージョン情報

[C1ThemeController のバージョン情報]をクリックすると、[ComponentOne のバージョン情報]ダイアログボックスが表示されます。このダイアログボックスには、ComponentOne ThemeController 製品のバージョン番号とライセンス情報が表示されます。

## C1ThemeController スマートタグ

Visual Studio では、Themes for WinForms 内の C1ThemeController コンポーネントにスマートタグが含まれます。スマートタグは、各コントロールで最もよく使用されるプロパティを提供するショートカットタスクメニューです。

[C1ThemeController のタスク]メニューにアクセスするには、C1ThemeController コンポーネントの右上隅にあるスマートタグをクリックします。[C1ThemeController のタスク]メニューが表示されます。



[C1ThemeController のタスク]メニューの機能は次のとおりです。

- テーマの設定

[テーマの設定]をクリックすると、[C1ThemeController]ダイアログボックスが開きます。詳細については、「[C1ThemeController ダイアログボックス](#)」を参照してください。

- テーマの追加

[テーマの追加]をクリックすると、[テーマの選択]ダイアログボックスが開きます。このダイアログボックスで、テーマの場所を参照し、それを C1ThemeController に追加できます。テーマを選択すると、それがプロジェクトに登録され、Visual Studio ではソリューションエクスプローラにテーマフォルダが表示されます。テーマを追加したら、Theme プロパティまたは ApplicationTheme プロパティにそれを適用します。

- テーマの読み込み

[テーマの再読み込み]をクリックすると、現在のアプリケーションで使用されるすべてのテーマを再読み込みます。

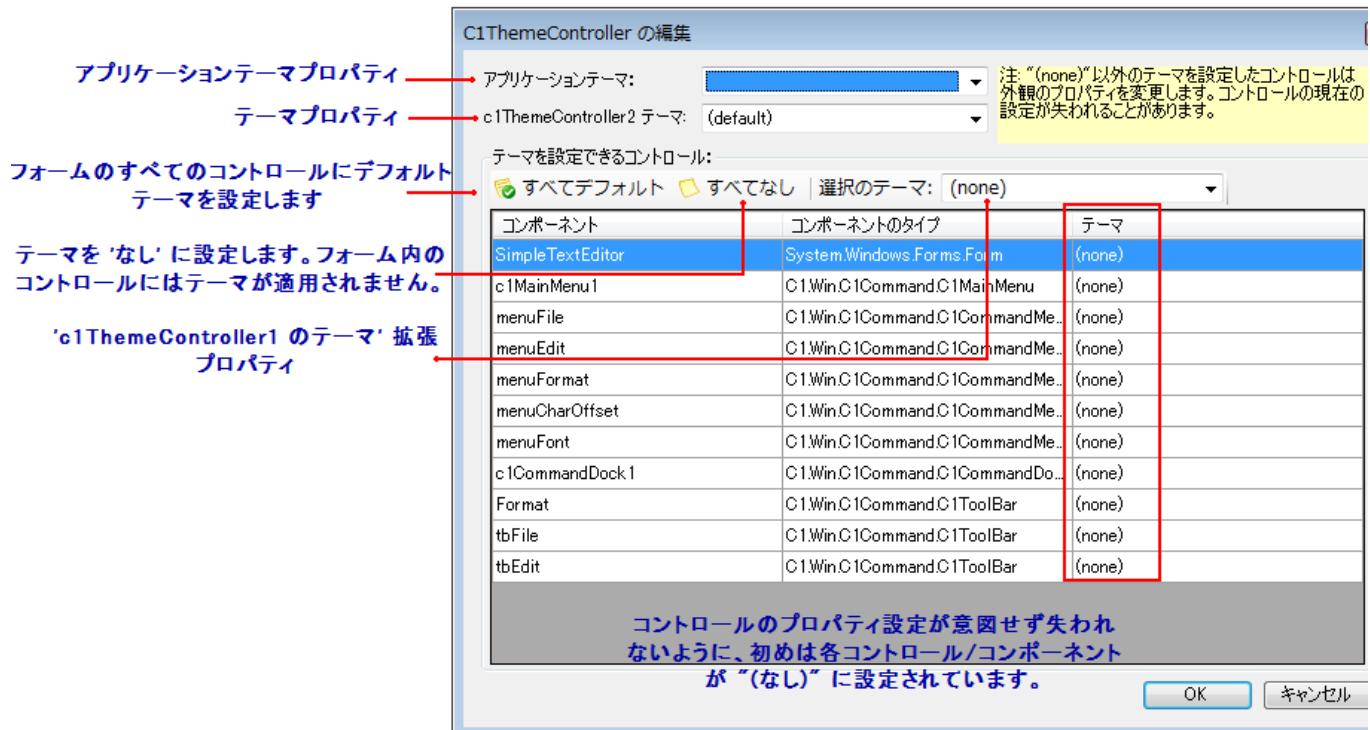
- C1ThemeController のバージョン情報

[C1ThemeController のバージョン情報]をクリックすると、[ComponentOne のバージョン情報]ダイアログボックスが表示されます。このダイアログボックスには、ComponentOne ThemeController 製品のバージョン番号とライセンス情報が表示されます。

## C1ThemeController ダイアログボックス

C1ThemeController 2.0 コンポーネントをコンポーネントトレイにドラッグ & ドロップすると、C1ThemeController ダイアログボックスが表示されます。ThemeController ダイアログボックスを使用して、アプリケーション内のテーマ指定可能なすべてのコントロールにすばやくテーマを適用できます。また、フォーム内のテーマ指定可能なすべてのコントロールにテーマを適用したり、いくつかのコントロールにそれぞれ異なるテーマを適用することもできます。

次の図に、アプリケーションにテーマを適用するために使用される主なプロパティを示します。



C1ThemeController を追加する前に、テーマ指定可能なコントロールがフォームにあった場合は、このダイアログボックスにフォーム内のすべてのコンポーネントがリストされます。コントロールのプロパティ設定が意図せず失われないように、初めは各コントロール/コンポーネントが "(なし)" に設定されています。

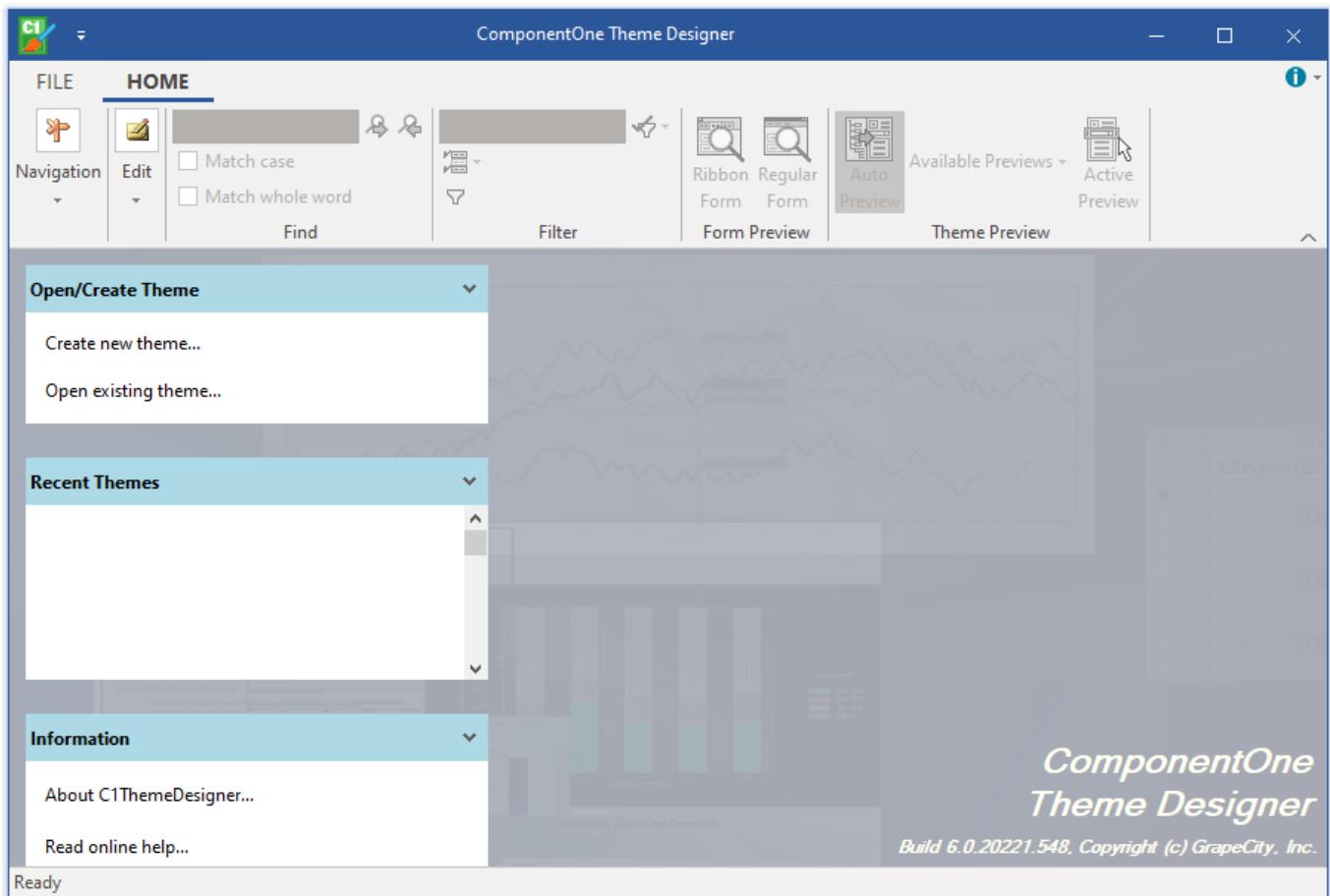
C1ThemeController ダイアログボックスの項目は、以下のように機能します。

- [アプリケーションテーマ] ドロップダウンリストボックス - アプリケーション全体の(デフォルト)テーマを指定するために使用されます。このテーマの名前は、App.config ファイルに保存されます。
- [C1ThemeController1 テーマ] ドロップダウンリストボックス - 現在の C1ThemeController のデフォルトテーマを指定します。値 "(デフォルト)" は、ApplicationTheme で指定されるアプリケーション全体のテーマが使用されることを示します。
- [選択されたテーマ] ドロップダウンリストボックス - フォーム内のサポートされているすべてのコントロールには、'c1ThemeController1 のテーマ' 拡張プロパティが追加されます。このプロパティを特定のテーマ名に設定すると、テーマコントローラのデフォルトテーマに関係なく、そのテーマが現在のコントロールに適用されます。これにより、必要に応じて、同じフォーム内の異なるコントロールに異なるテーマを適用できます。このプロパティは、'デフォルト' と 'なし' の2つの特別な値をサポートします。デフォルト値は、コントロールに現在のフォームの C1ThemeController に指定されたデフォルトテーマが使用されることを意味します。'なし' は、コントロールにテーマが適用されないことを意味します。
- [すべてを(デフォルト)に設定] ボタン - フォームのすべてのコントロールにデフォルトテーマを設定します。
- [すべてを(なし)に設定] ボタン - テーマを 'なし' に設定します。フォーム内のコントロールにはテーマが適用されません。

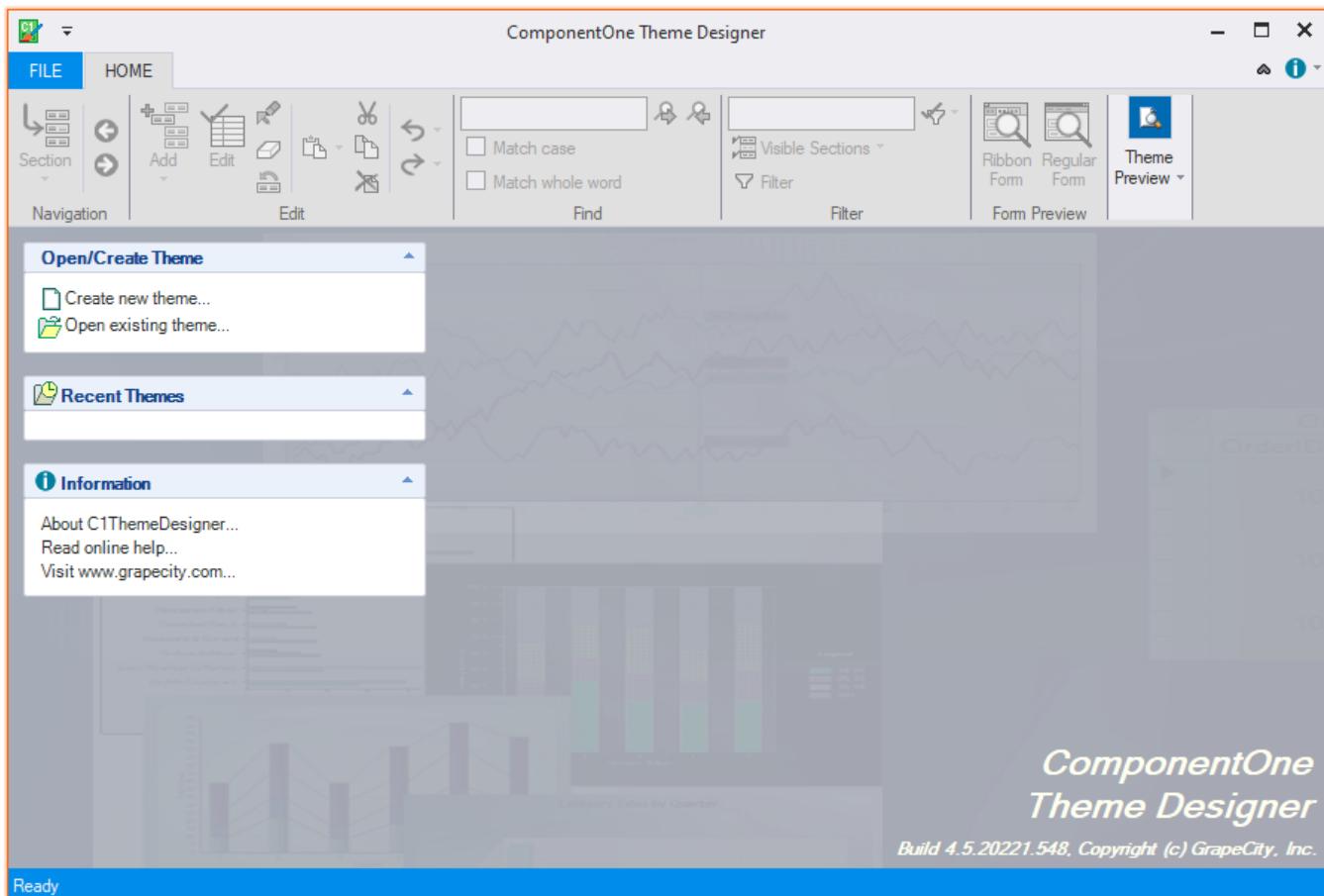
## C1ThemeDesigner アプリケーションの概要

ThemeDesigner アプリケーションを使用して、ComponentOne for WinForms コントロール内の任意のコントロールに対して新しいテーマを簡単に設計できます。C1ThemeDesigner.exe を開くと、トピックバーが表示されます。

### .NET



### .NET Framework



トピックバーには、以下のグループがあります。

### テーマを開く/作成

[テーマを開く/作成] グループでは、以下の作業を実行できます。

- **新しいテーマの作成** - この項目をクリックすると、[新しいテーマ] ダイアログボックスが表示されます。
- **既存のテーマを開く** - この項目をクリックすると、[テーマを開く] ダイアログボックスが表示され、最近使ったテーマを最後に保存した場所が開かれます。テーマを選択し、[開く] をクリックします。既存のテーマがテーマデザイナで開かれます。

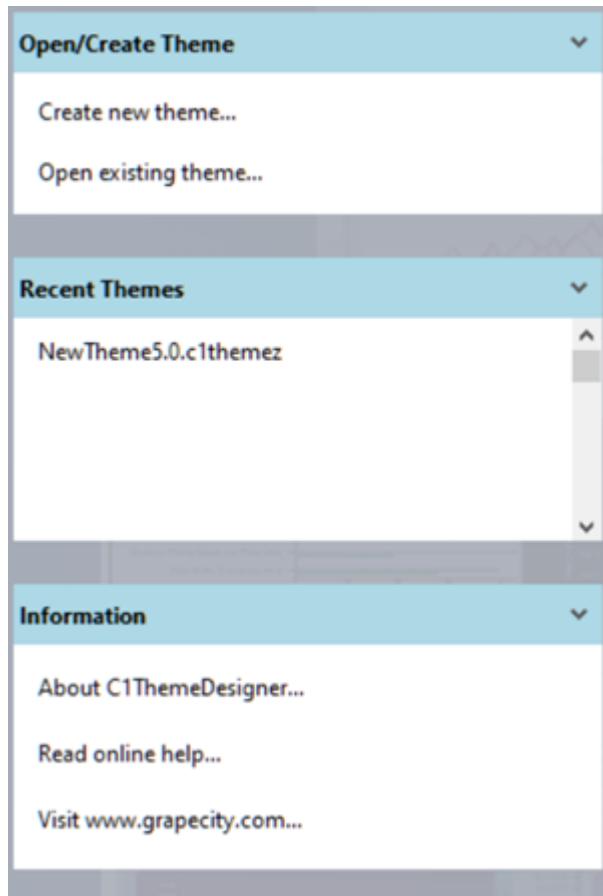
### 最近使ったテーマ

最近使ったテーマには、テーマデザイナを使用して最近作成したテーマがリストされます。

### 情報

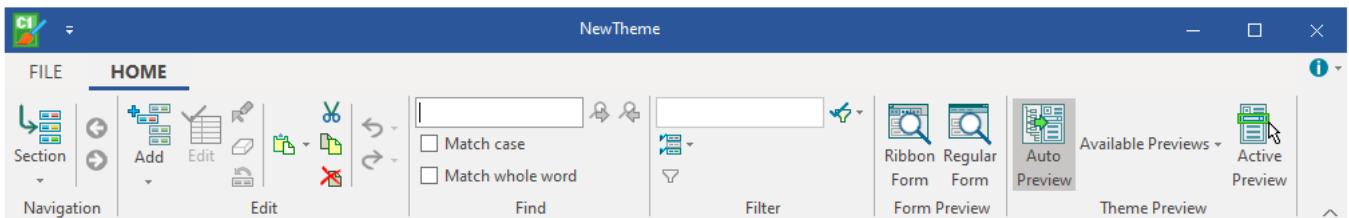
[情報] グループボックスは、製品、C1ThemeDesigner、およびオンラインヘルプへのリンクに関する情報を提供します。

# Themes for WinForms



## リボン

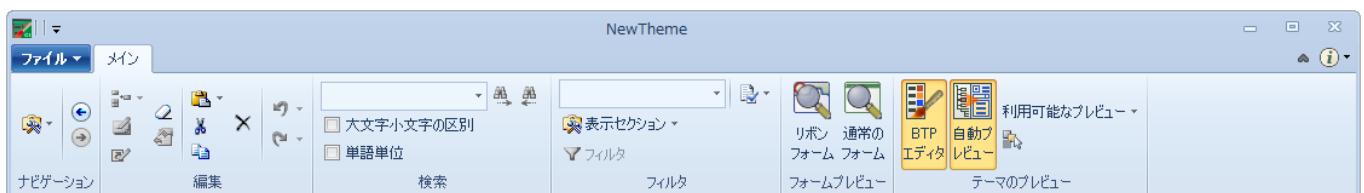
テーマデザイナには、[ファイル]タブと[メイン]タブから成るリボンメニューがあります。



さらに、ThemeDesignerはテーマをインポートすることができます。

## テーマデザイナのリボンメニュー

このトピックでは、C1ThemeDesigner リボンのさまざまなコマンドを試します。

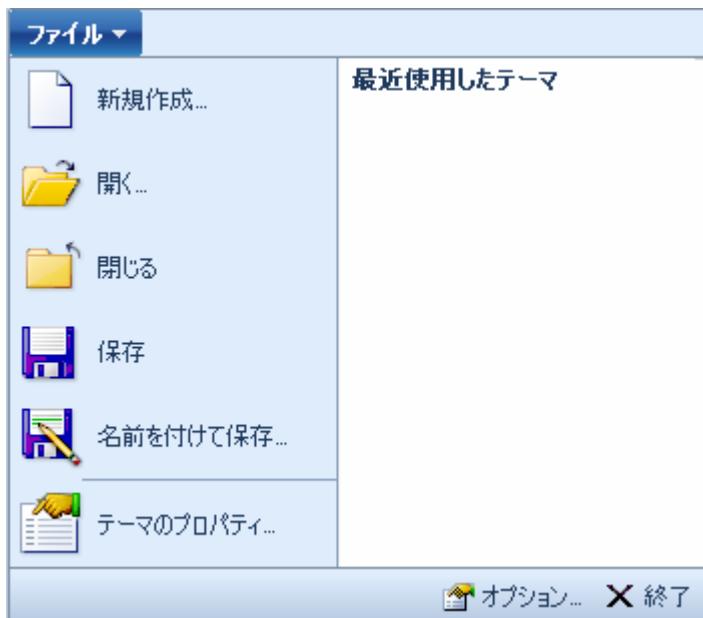


ComponentOne テーマデザイナには、[ファイル]と[メイン]の2つのタブがあります。

## [ファイル]タブ

[ファイル]メニューには、[新規作成]、[開く]、[閉じる]、[保存]、[別名保存]の各オプションがあります。作成して最近使った

テーマは、[最近使ったテーマ]の直下に表示されます。



[ファイル]メニューの機能は次のとおりです。

コマンドボタン	説明
新規作成	新しいテーマを作成するための[新しいテーマ]ダイアログボックスを開きます。新しいテーマは、一から作成することも、既存のテーマファイルに基づいて作成することもできます。
開く	既存の.c1themeファイルを開くための[テーマを開く]ダイアログボックスを開きます。
閉じる	テーマデザイナを開いたまま、現在のテーマを閉じます。
保存	現在のテーマを保存します。
別名保存	現在のテーマを別のファイルに保存します。
テーマのプロパティ	テーマ名などのプロパティを表示または編集するための[テーマの編集]ダイアログボックスを開きます。

## [メイン]タブ

[メイン]タブには、[ナビゲーション]、[編集]、[検索]、[フィルタ]、[プレビュー]の各グループがあります。

### [ナビゲーション]グループ



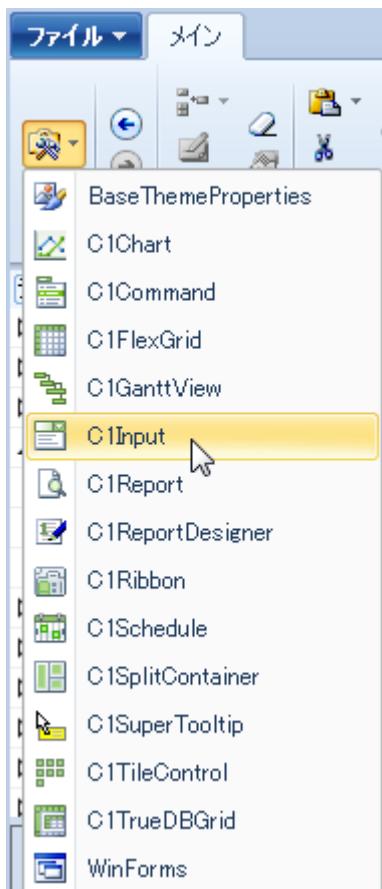
[ナビゲーション]グループでは、編集するセクションを[セクション]メニューから選択することで、テーマツリー内をすばやく移動できます。[ナビゲーション]グループには、次のコマンドボタンがあります。

# Themes for WinForms

コマンドボタン	説明
セクション	特定の製品のテーマセクションに直接移動します。
戻る	テーマツリー内で、履歴にある次の位置に戻ります。
進む	テーマツリー内で、履歴にある次の位置に進みます。

## [セクション]メニュー

[セクション]メニューでは、基本テーマプロパティ、C1Command、C1FlexGrid、C1GanttView、C1Input、C1Preview、C1Schedule、C1SuperToolTip、C1TrueDBGrid、WinForms など、編集するセクションをテーマツリーから選択できます。



## 基本テーマプロパティビュー

基本テーマプロパティセクションを選択すると、テーマツリーの右に[主要なプロパティ]ナビゲーションバーが表示されます。この機能は、.NET Frameworkでのみ使用できます。



このナビゲーションバーには、[主要なプロパティ]、[スタイル]、[クリック可能スタイル]の3つのカテゴリが含まれます。

### 主要なプロパティ

[主要なプロパティ]カテゴリでは、[デフォルトフォント]、[Control Text]、[Window Text]の各項目の FontStyle プロパティを変更できます。

### スタイル

[Styles]カテゴリでは、[ヘッダ]、[コンテンツ]、[強調表示文字]、[淡色表示文字]の各項目の Font、Background、ForeColor、BorderColor の各プロパティを変更できます。

# Themes for WinForms



## クリック可能スタイル

[Clickable Styles] カテゴリでは、[デフォルト]、[ホット]、[押下]、[ホット & 押下]の各項目の Font、Background、ForeColor、BorderColor の各プロパティを変更できます。

Clickable Styles	
<b>A</b>	<b>Default</b>
フォント:	<input type="button" value="参照..."/>
背景色:	<input type="button" value="参照..."/> 直線状マルチカラー
前景色:	<input type="button" value="参照..."/> 96; 64; 0
境界線色:	<input type="button" value="参照..."/> 102; 51; 0
	<input type="button" value="参照..."/>
<b>A</b>	<b>Hot</b>
フォント:	<input type="button" value="参照..."/>
背景色:	<input type="button" value="参照..."/> 直線状マルチカラー
前景色:	<input type="button" value="参照..."/> 96; 64; 0
境界線色:	<input type="button" value="参照..."/> 102; 51; 0
	<input type="button" value="参照..."/>
<b>A</b>	<b>Pressed</b>
フォント:	<input type="button" value="参照..."/>
背景色:	<input type="button" value="参照..."/> 直線状マルチカラー
前景色:	<input type="button" value="参照..."/> 96; 64; 0
境界線色:	<input type="button" value="参照..."/> 102; 51; 0
	<input type="button" value="参照..."/>
<b>A</b>	<b>Hot &amp; Pressed</b>
フォント:	<input type="button" value="参照..."/>
背景色:	<input type="button" value="参照..."/> 直線状マルチカラー
前景色:	<input type="button" value="参照..."/> 96; 64; 0
境界線色:	<input type="button" value="参照..."/> 102; 51; 0
	<input type="button" value="参照..."/>

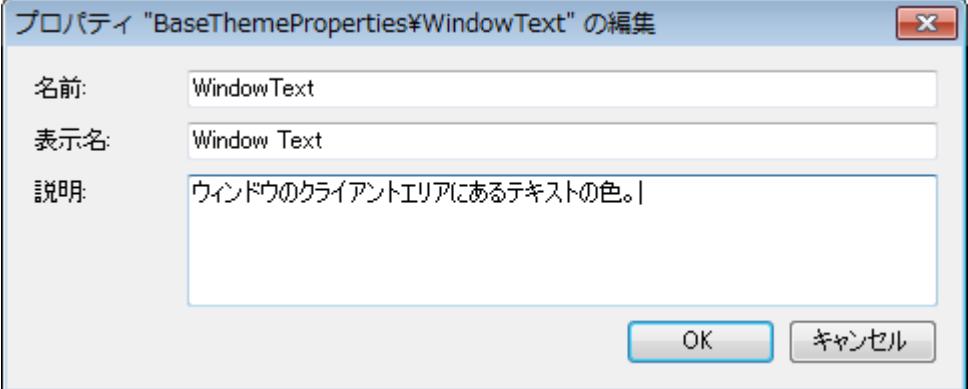
## [編集]グループ

**C1ThemeDesigner** のリボンメニューの【編集】グループは、次のように表示されます。



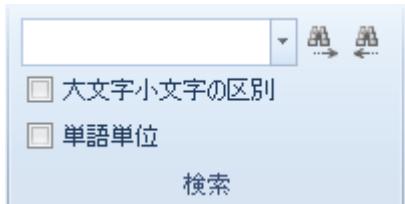
〔編集〕グループには、以下のコマンドボタンが含まれます。

# Themes for WinForms

コマンドボタン	説明
追加	基本テーマプロパティセクションの任意の項目を選択すると、[追加]コマンドが有効になります。コマンド項目のサブメニューを開くには、[追加]ボタンをクリックします。[下]コマンドは、選択されたプロパティのすぐ下に新しいプロパティを追加します。[上]コマンドは、選択されたプロパティのすぐ上に新しいプロパティを追加します。[子として]は、選択されたプロパティの子として新しいプロパティを追加します。
編集	現在の基本テーマプロパティツリー項目の属性(名前など)を編集します。基本テーマプロパティセクションの任意の項目を選択すると、[編集]コマンドが有効になります。編集する基本テーマプロパティの項目を選択してから、[編集]ボタンをクリックします。[編集]ボタンをクリックすると、次のダイアログボックスが開きます。 
参照	[参照]コマンドボタンをクリックすると、テーマツリー項目で選択した参照を編集、追加、削除するための【参照の編集】ダイアログボックスが開きます。
クリア	選択されたテーマツリー項目のプロパティ値をクリアします。
リセット	選択されたテーマツリー項目のプロパティ値をリセットします。
貼り付け	項目を基本テーマプロパティツリーに貼り付けます。[貼り付け]ボタンをクリックすると、[後に貼り付け]、[前に貼り付け]、[子として貼り付け]、[値を貼り付け]などのコマンド項目が入ったサブメニューが開きます。[後に貼り付け]は、選択されたプロパティの後にプロパティを貼り付けます。[前に貼り付け]は、選択されたプロパティの前にプロパティを貼り付けます。[子として貼り付け]は、選択されたプロパティの子としてプロパティを貼り付けます。[値を貼り付け]は、選択されたプロパティの値をクリップボードのデータに置き換えます。
切り取り	選択された基本テーマプロパティ項目を切り取り、クリップボードに入れます。
コピー	選択された基本テーマプロパティツリー項目をコピーし、クリップボードに入れます。
削除	選択された基本テーマプロパティツリー項目を削除します。
元に戻す	最後の編集を元に戻します。
やり直し	最後に元に戻した編集をやり直します。

## [検索]グループ

[検索]グループボックスには、検索テキストボックス、[次を検索]コマンドボタン、[前を検索]コマンドボタン、[大文字と小文字を区別する]チェックボックス、[単語単位]チェックボックスがあります。



[検索] グループの項目は、以下のように機能します。

項目	説明
検索テキストボックス	検索文字列に一致する名前を含むノードをテーマツリー内で検索します。
次を検索	一致する名前を含む次のプロパティをテーマツリー内で検索します。
前を検索	一致する名前を含む前のプロパティをテーマツリー内で検索します。
大文字と小文字を区別する	すべての検索クエリで大文字と小文字を区別します。
単語単位	検索テキストボックスに入力した単語と正確に一致する単語を検索します。

詳細については、「[テーマツリー内のプロパティ項目の検索とフィルタリング](#)」を参照してください。

## [フィルタ] グループ

[フィルタ] グループは、テーマデザイナの[メイン]リボンメニューに表示されます。



次の表で、[フィルタ] グループ内の各項目について説明します。

項目	説明
名前フィルタテキストボックス	テーマツリーに表示するノードをプロパティ名でフィルタリングします。
表示セクション	テーマツリーに表示するセクションを選択します。
[フィルタ] ボタン	テーマツリー内のプロパティ名をオンまたはオフにします。

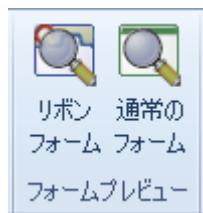
詳細については、「[テーマツリー内のプロパティ項目の検索とフィルタリング](#)」を参照してください。

## フォームプレビュー

[フォームプレビュー] グループボックスは、テーマデザイナのリボンメニューに表示されます。[フォームプレビュー] グループ

# Themes for WinForms

ボックスには、[リボンフォーム]と[通常のフォーム]の2つのボタンが表示されます。

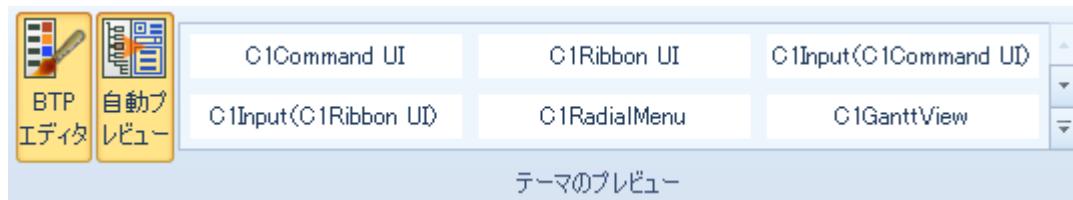


次の表で、[フォームプレビュー]内の各項目について説明します。

項目	説明
<b>リボンフォーム</b>	C1RibbonForm 内の C1Control(C1FlexGrid、C1Schedule、C1TrueDBGrid) に適用されている現在のテーマをプレビュー表示します。これは、新しいウィンドウで開きます。
<b>通常のフォーム</b>	C1Command UI 要素を含む WinForms 内の C1Control(C1FlexGrid、C1Schedule、C1TrueDBGrid) に適用されている現在のテーマをプレビュー表示します。これは、新しいウィンドウで開きます。

## テーマのプレビュー

[テーマプレビュー]グループボックスは、テーマデザイナのリボンメニューに表示されます。



次の表で、[テーマプレビュー]内の各項目について説明します。

項目	説明
<b>BTP エディタ</b>	C1RibbonForm 内の C1Control(C1FlexGrid、C1Schedule、C1TrueDBGrid) に適用されている現在のテーマをプレビュー表示します。これは、新しいウィンドウで開きます。
<b>自動プレビュー</b>	C1Command UI 要素を含む WinForms 内の C1Control(C1FlexGrid、C1Schedule、C1TrueDBGrid) に適用されている現在のテーマをプレビュー表示します。これは、新しいウィンドウで開きます。
<b>C1Command UI</b>	C1Command メニューなどの UI 要素を含む各種 C1Control(C1TrueDBGrid、C1FlexGrid、C1Schedule) に適用されている現在のテーマをプレビュー表示します。
<b>C1Ribbon UI</b>	リボンベースの UI を含む各種 C1Control(C1TrueDBGrid、C1FlexGrid、C1Schedule) に適用されている現在のテーマをプレビュー表示します。
<b>C1Input (C1Command UI)</b>	C1Command メニューなどの UI 要素を含む各種 C1Input コントロールに適用されている現在のテーマをプレビュー表示します。
<b>C1Input (C1Ribbon UI)</b>	C1Ribbon リボンベースの UI を含む各種 C1Input コントロールに適用されている現在のテーマをプレビュー表示します。

	ルに適用されている現在のテーマをプレビュー表示します。
<b>C1RadialMenu</b>	C1RadialMenu に適用されている現在のテーマをプレビュー表示します。
<b>C1GanttView</b>	C1GanttView に適用されている現在のテーマをプレビュー表示します。
<b>C1Report</b>	C1Report 印刷プレビューコントロールに適用されている現在のテーマをプレビュー表示します。
<b>C1SuperTooltip</b>	C1SuperTooltip と C1SuperLabel に適用されている現在のテーマをプレビュー表示します。
<b>WinForms コントロール</b>	標準の WinForms コントロールに適用されている現在のテーマをプレビュー表示します。
<b>CalendarView</b>	CalendarView に適用されている現在のテーマをプレビュー表示します。
<b>DashboardLayout</b>	DashboardLayout に適用されている現在のテーマをプレビュー表示します。
<b>DataFilter</b>	DataFilter に適用されている現在のテーマをプレビュー表示します。
<b>ExpressionEditor</b>	ExpressionEditor に適用されている現在のテーマをプレビュー表示します。
<b>FlexPivot</b>	FlexPivot に適用されている現在のテーマをプレビュー表示します。
<b>FlexViewer</b>	FlexViewer に適用されている現在のテーマをプレビュー表示します。
<b>Gauge</b>	Gauge に適用されている現在のテーマをプレビュー表示します。
<b>Map</b>	Map に適用されている現在のテーマをプレビュー表示します。
<b>Ribbon UI</b>	Ribbon UI に適用されている現在のテーマをプレビュー表示します。
<b>RulesManager</b>	RulesManager に適用されている現在のテーマをプレビュー表示します。
<b>ScrollBar</b>	ScrollBar に適用されている現在のテーマをプレビュー表示します。
<b>Sparkline</b>	Sparkline に適用されている現在のテーマをプレビュー表示します。
<b>TreeView</b>	TreeView に適用されている現在のテーマをプレビュー表示します。
<b>TileView</b>	TileView に適用されている現在のテーマをプレビュー表示します。この機能は、.NET Frameworkでのみ使用できます。

## テーマデザイナのダイアログボックス

ComponentOne テーマデザイナには、以下のダイアログボックスが用意されています。

- [新しいテーマ]ダイアログボックス
- [プロパティの追加]ダイアログボックス

## [新しいテーマ]ダイアログボックス

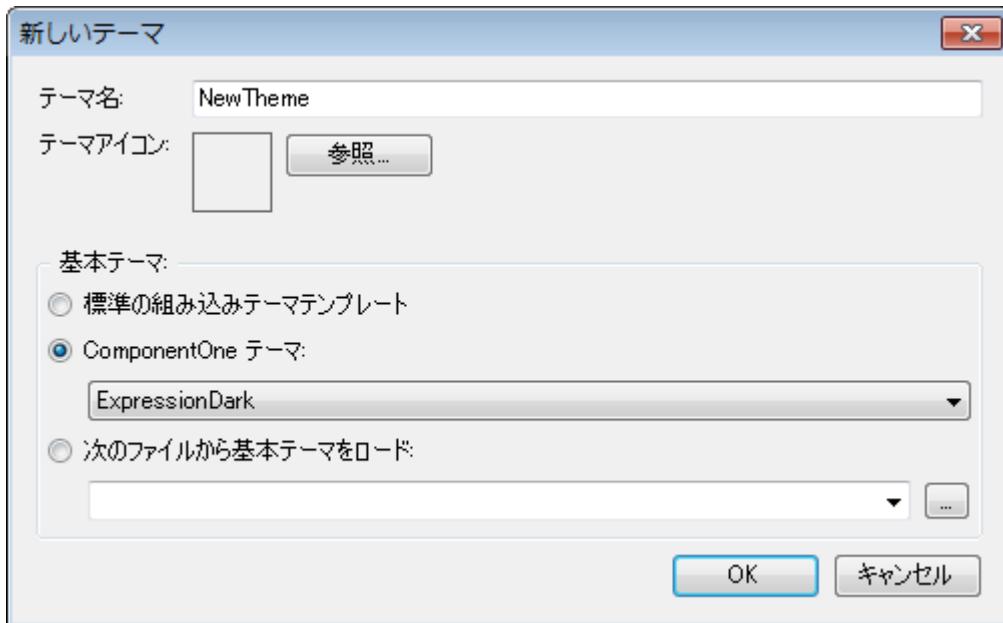
[新しいテーマ]ダイアログボックスを使用して、新しいテーマを作成できます。

### [新しいテーマ]ダイアログボックスにアクセスする

トピックバーの[新しいテーマの作成]リンクをクリックします。



[新しいテーマ]ダイアログボックスが表示されます。



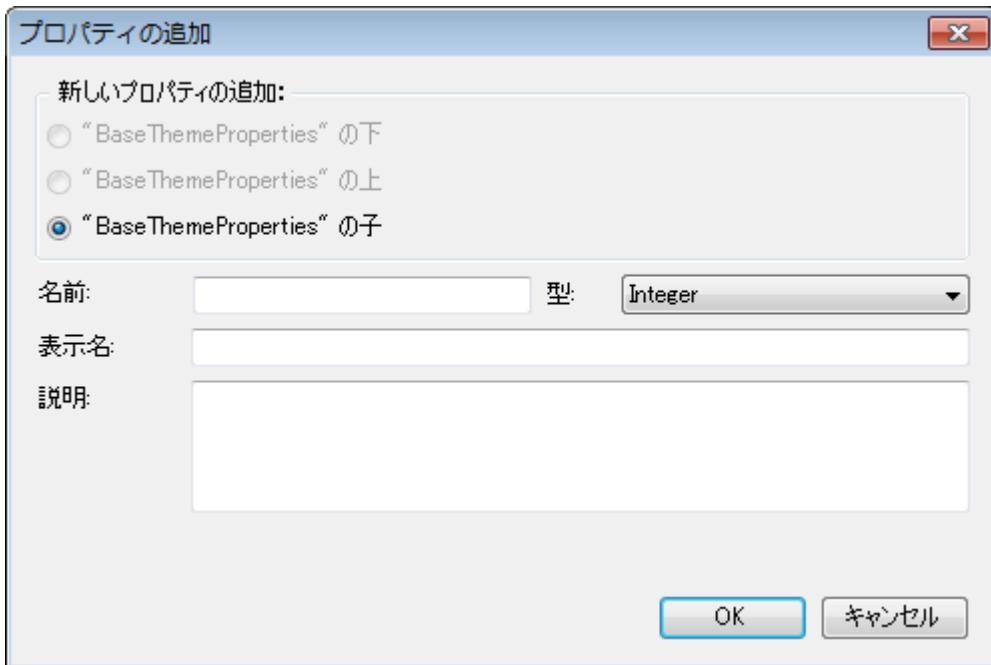
[新しいテーマ]ダイアログボックスは、以下の項目で構成されます。

- **テーマ名:**新しいテーマの名前を入力するためのテキストボックスです。
- **テーマアイコン:**初めは空のピクチャボックスが表示されますが、[参照]ボタンをクリックしてテーマアイコンを開くと、アイコンが表示されます。
- **基本テーマ:**作成するテーマの基礎となるテーマを以下の3つの項目から選択します。
  - 標準の組み込みテーマテンプレート

- ComponentOne テーマ – 既存の組み込みテーマから選択します。
- 基本テーマファイルのロード: 作成した既存の基本テーマを開いて、現在の WinForms テーマに追加できます。[テーマ名] テキストボックスにテーマ名を入力すると、テーマデザイナの上部にその新しい名前が表示されます。

## [プロパティの追加] ダイアログボックス

[編集] グループの [追加] コマンドボタンをクリックすると、[新しいプロパティ] ダイアログボックスが表示されます。



新しいプロパティには、名前、プロパティタイプ、キャプション、説明を指定できます。詳細については、「[テーマツリーへの新しいプロパティの追加](#)」を参照してください。

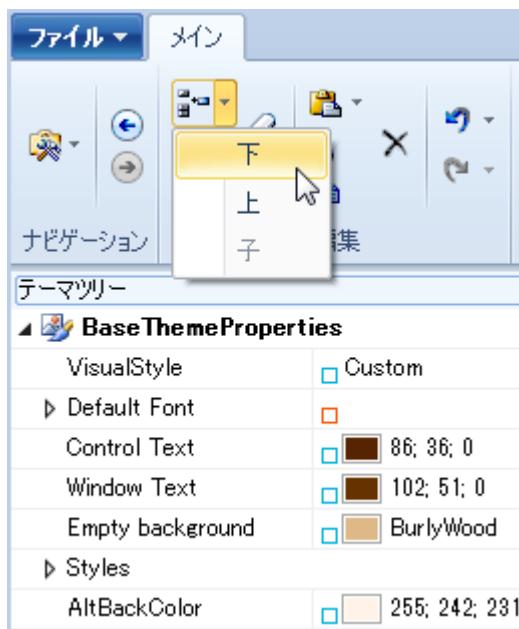
## テーマツリーへの新しいプロパティの追加

C1ThemeDesigner リボンメニューの [編集] グループの [追加] コマンドボタンから、テーマツリー内の指定の位置に新しいプロパティを挿入できます。

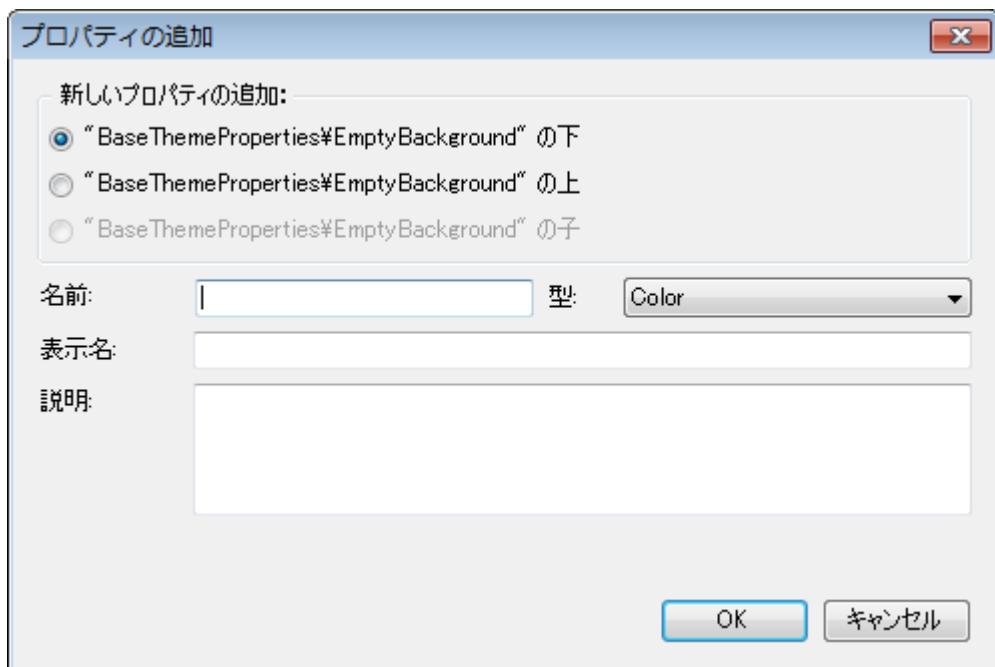
テーマツリーに新しいプロパティを追加するには、次の手順を実行します。

1. C1ThemeDesigner を開き、Office2010Blue などの任意の Office テーマに基づいて新しいテーマを作成します。
2. **BaseThemeProperties** の下の **Styles** を選択します。
3. [編集] グループの [追加] ドロップダウンメニューから [下] をクリックします。

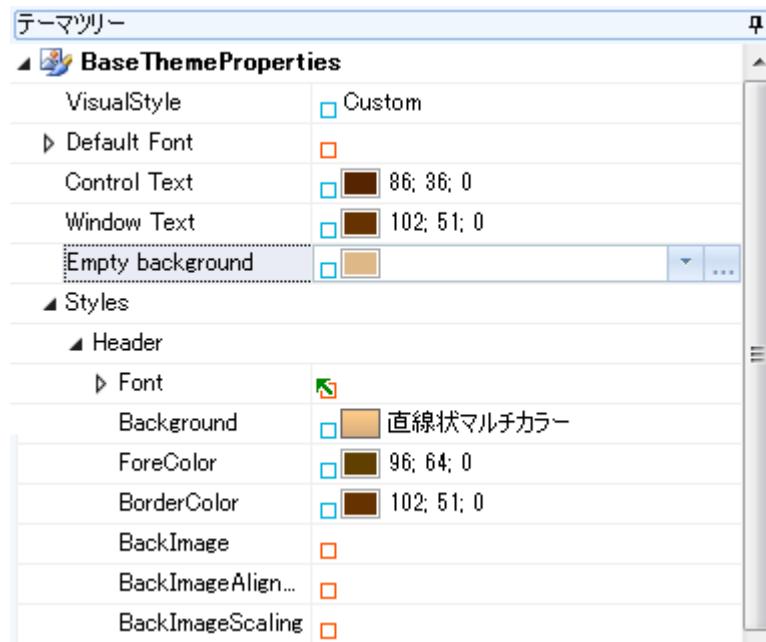
# Themes for WinForms



【プロパティの追加】ダイアログボックスが表示されます。

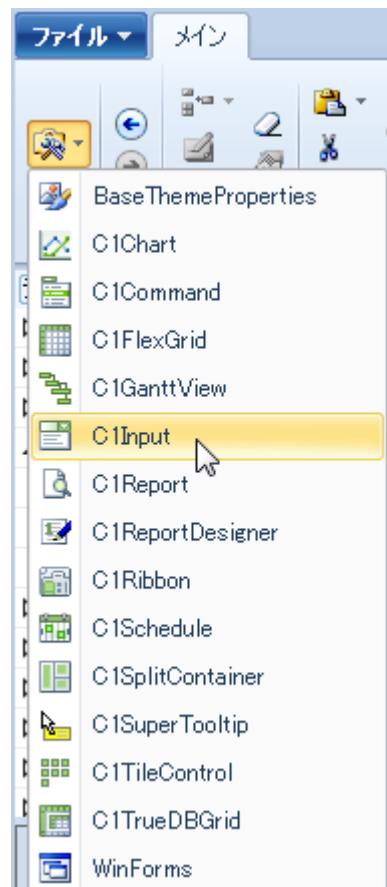


4. 【名前】テキストボックスに「Emptybackground」と入力します。
  5. 【タイプ】ドロップダウンリストボックスから Background を選択します。
  6. 【説明】リッチテキストボックスに、「空白ウィンドウエリアのため背景」と入力します。
  7. 【プロパティの追加】ダイアログボックスの【OK】をクリックします。
- 新しいプロパティ Emptybackground が BaseThemeProperties テーマツリーの Styles プロパティの下に表示されます。



## テーマツリーにあるプロパティ項目間の移動

テーマツリー内の特定のセクションにすばやく移動するには、ドロップダウン矢印をクリックし、テーマツリーにある移動先のセクションを選択します。



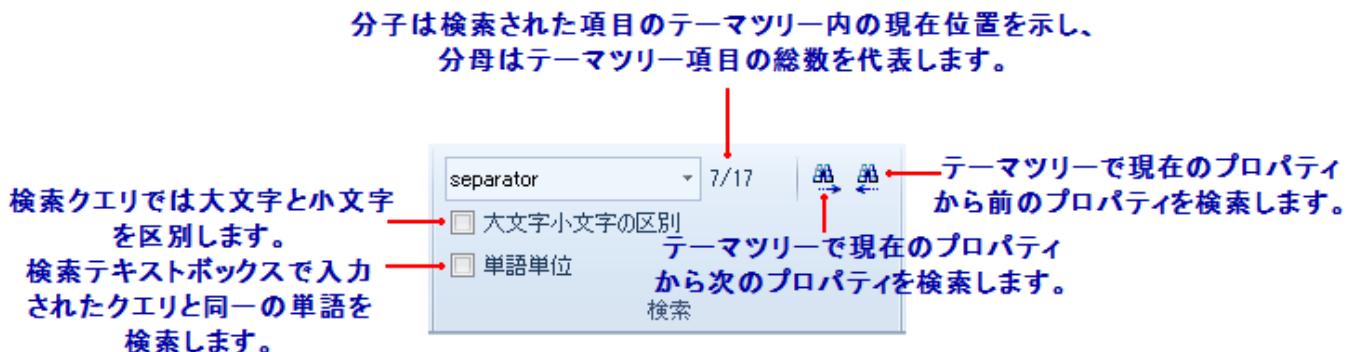
【戻る】ボタンをクリックすると、テーマツリー内で直前に選択された項目に戻ります。【進む】ボタンをクリックすると、テーマツリー内の次の項目に移動します。

## テーマツリー内のプロパティ項目の検索とフィルタリング

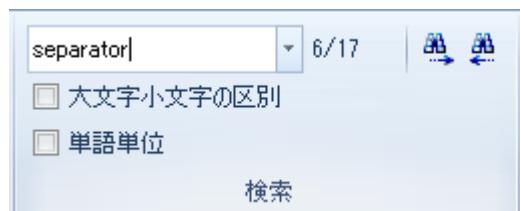
C1ThemeDesigner には、[検索] グループと[フィルタ] グループがあります。検索対象がわかっている場合は、[検索] テキストボックスを使用し、検索対象が正確にわからない場合は、[フィルタ] テキストボックスを使用します。一般的な情報を検索する場合は、フィルタリングが便利です。

### [検索] グループ

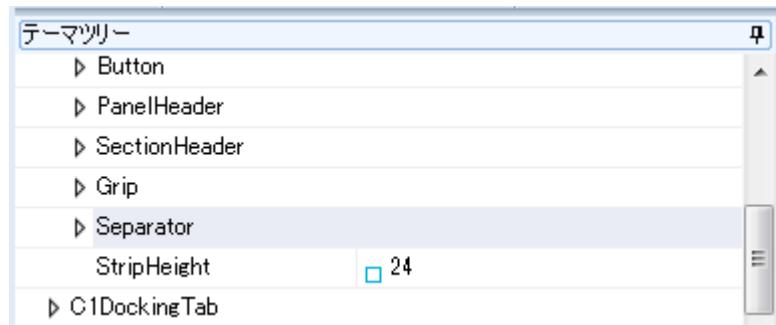
次の図では、[検索] グループ内の各コマンドにラベルを付けています。



検索テキストボックスに入力すると、最初の数文字が自動的に検索され、一致が見つかると、テーマツリーのノードまたは項目が強調表示されて、編集が可能になります。テーマツリーで強調表示された項目が編集対象のプロパティでない場合は、[次を検索] プロパティボタンまたは[前を検索] プロパティボタンをクリックします。

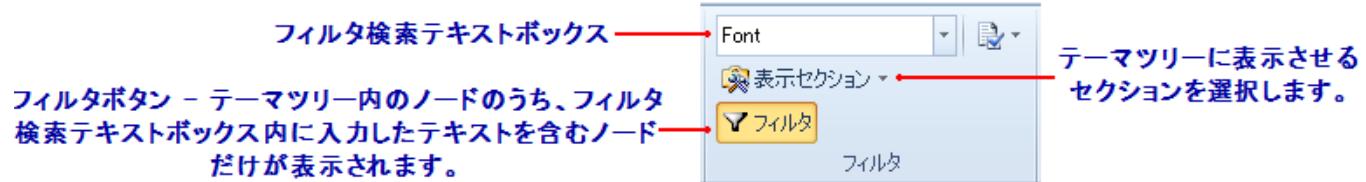


次の画像は、Theme ツリーで Separator 項目が検出されたところです。



### [フィルタ] グループ

次の図では、[フィルタ] グループ内の各コマンドにラベルを付けています。



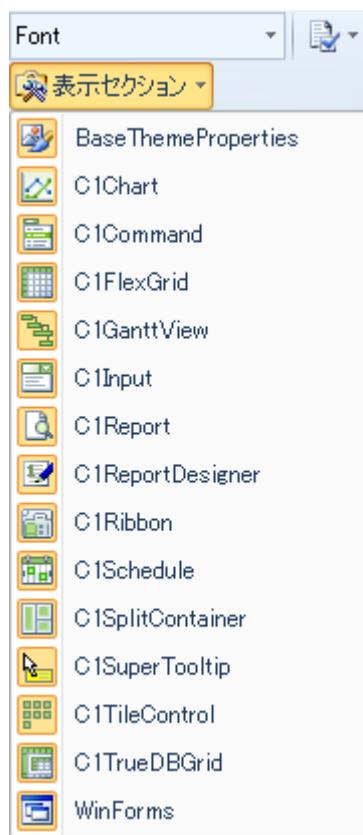
## フィルタの使用方法

フィルタを使用するには、次の手順に従います。

フィルタ検索テキストボックスに「Font」などのプロパティ名を入力します。

テーマツリー内のノードのうち、そのテキストを含むノードだけが表示されます。

【表示セクション】ドロップダウンリストボックスのドロップダウン矢印をクリックし、C1FlexGrid と C1Input 以外の項目の選択を解除します。



【フィルタ】グループの【フィルタ】ボタンをクリックします。

これで、C1FlexGrid セクションと C1Input セクションが、それぞれ Font プロパティだけが表示されて展開されます。

メモ: 【フィルタ】ボタンが有効な場合は、強調表示されます。

## 過去のフィルタの表示方法

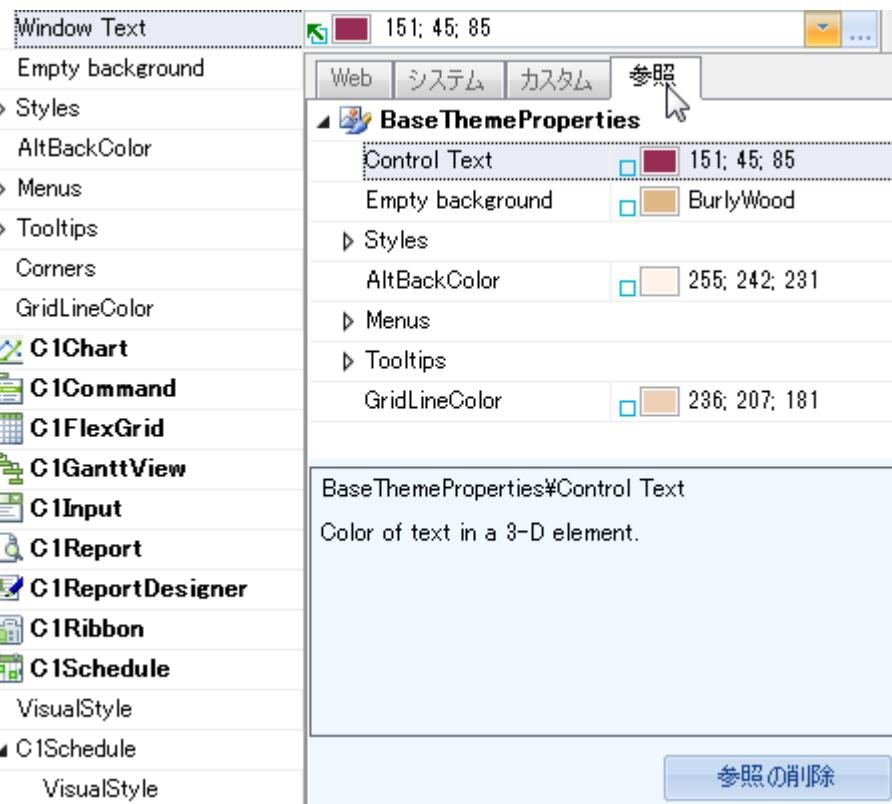
過去のフィルタは履歴に入れられ、フィルタ検索テキストボックスのドロップダウンをクリックすると表示されます。過去のフィルタの例を次に示します。



## 参照によるテーマプロパティの指定

参照によってテーマプロパティを指定するには、次の手順を実行します。

1. C:\Program Files\ComponentOne\WinForms\Bin\v2\Designer フォルダの C1ThemeDesigner をダブルクリックします。
2. トピックバーの【新しいテーマの作成】をクリックします。【新しいテーマ】ダイアログボックスが表示されます。
3. 標準の組み込みテーマテンプレートを選択し、テーマ名として「BlueandPink」と入力し、【OK】をクリックします。
4. テーマツリー内の【BaseThemeProperties】から【Control Text】を選択します。
5. 【Control Text】ドロップダウン矢印をクリックし、【カスタム】タブをクリックします。
6. **BackColor** を **151; 45; 85** に設定します。
7. 【Window Text】の横のドロップダウン矢印をクリックし、【参照】タブを選択します。
8. 【参照】タブで【Control Text】を選択します。



【Window Text】プロパティが【BaseThemeProperties】の【Control Text】から参照されることを示すアイコンがビジュアルインジケータとして表示されます。

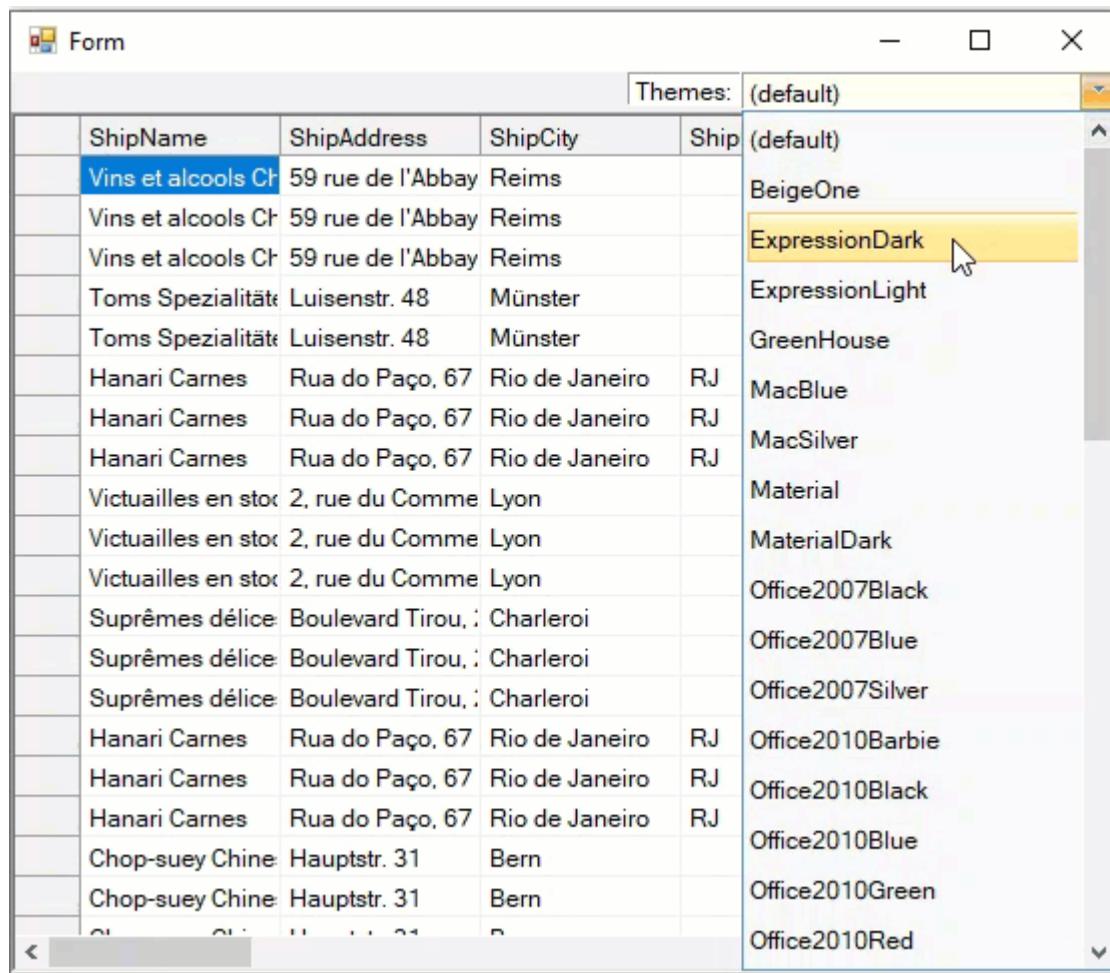


## C1ThemePicker

**C1ThemePicker** is a combo box, which can be used to select themes at runtime. The C1ThemePicker control is provided by the **C1.Win.C1Themes.Extended.4.5.2** assembly. C1ThemePicker can be simply dropped from the Toolbox and added to the form.

The C1ThemePicker control provides a drop-down list to select one of the registered themes and apply it to the assigned C1ThemeController in the form. C1ThemePicker makes it easy for the user to switch between different themes at runtime.

The GIF below depicts the **ThemePicker** control docked in a form with FlexGid.



**Note:** In order to work with the **C1ThemePicker** control, the following dependent assemblies should be added to your project:

- C1.Win.C1Themes.4.5.2
- C1.Win.C1Themes.Extended.4.5.2
- C1.Win.C1Input.4.5.2

## Filtering Themes

The user can also filter the list of available themes in ThemePicker. For example, the user can switch the application UI between Office 2016 themes using the **Filter** property of **C1ThemePicker** class.

- **Visual Basic**

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```

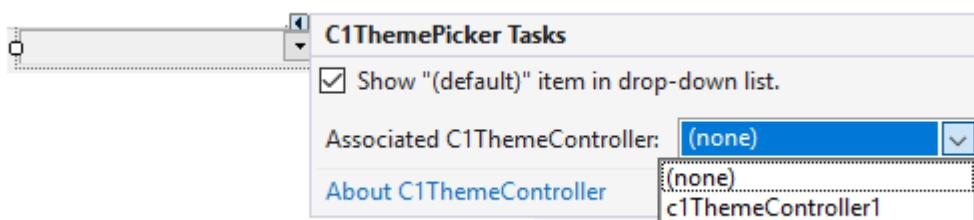
C1ThemePicker1.Filter = "2016"
End Sub

• C#
private void Form1_Load(object sender, EventArgs e)
{
    c1ThemePicker1.Filter = "2016";
}

```

## C1ThemePicker 設計時サポート

The user can customize the properties of the **C1ThemePicker** control such as its BackColor, ForeColor, Location, DropDownWidth etc. using the **Properties** window. You can access the **C1ThemePicker Tasks** menu by clicking the smart tag  in the upper-right corner of the C1ThemePicker control.



The **C1ThemePicker Tasks** menu can perform the following operations:

- **Show "(default)" item in the dropdown list**  
The user can select the checkbox to display the "default" theme in the drop-down list.
- **Associated C1ThemeController**  
The user can set the associated C1ThemeController to apply the selected theme.
- **About C1ThemePicker**  
The user can click this option and view a dialog box that displays the version number of C1ThemePicker and other online resources.