

ARTICLE

READER
LEVEL:



ColorDialog in C#

By [Maresh Chand](#) on Dec 21, 2000

A ColorDialog control is used to select a color from available colors and also define custom colors. This article demonstrates how to use a ColorDialog in Windows Forms and C#.

 147.7

 4

 0



 [Download Files:ColorDialogSample.zip](#)

[Download Aspose, it really helps](#)

Color Dialog

A ColorDialog control is used to select a color from available colors and also define custom colors. A typical Color Dialog looks like Figure 1 where you can see there is a list of basic solid colors and there is an option to create custom colors.



Figure 1

Creating a ColorDialog

We can create a ColorDialog control using a Forms designer at design-time or using the ColorDialog class in code at run-time (also known as dynamically). Unlike other Windows Forms controls, a ColorDialog does not have and not need visual properties like others. The only purpose of ColorDialog to display available colors, create custom colors and select a color from these colors. Once a color is selected, we need that color in our code so we can apply it on other controls.

Again, you can create a ColorDialog at design-time but it is easier to create a ColorDialog at run-time.

Design-time

To create a ColorDialog control at design-time, you simply drag and drop a ColorDialog control from Toolbox to a Form in Visual Studio. After you drag and drop a ColorDialog on a Form, the ColorDialog looks like Figure 2.

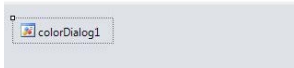


Figure 2

Adding a ColorDialog to a Form adds following two lines of code.

```
private System.Windows.Forms.ColorDialog colorDialog1;  
this.colorDialog1 = new System.Windows.Forms.ColorDialog();
```

Run-time

Creating a ColorDialog control at run-time is merely a work of creating an instance of ColorDialog class, set its properties and add ColorDialog class to the Form controls.

First step to create a dynamic ColorDialog is to create an instance of ColorDialog class. The following code snippet creates a ColorDialog control object.

```
ColorDialog colorDlg = new ColorDialog();
```

ShowDialog method of ColorDialog displays the ColorDialog. The following code snippet sets background color, foreground color, Text, Name, and Font properties of a ColorDialog.

```
colorDlg.ShowDialog();
```

Once the ShowDialog method is called, you can pick colors on the dialog.

Setting ColorDialog Properties

After you place a `ColorDialog` control on a Form, the next step is to set properties.

The easiest way to set properties is from the Properties Window. You can open Properties window by pressing F4 or right click on a control and select Properties menu item. The Properties window looks like Figure 3.

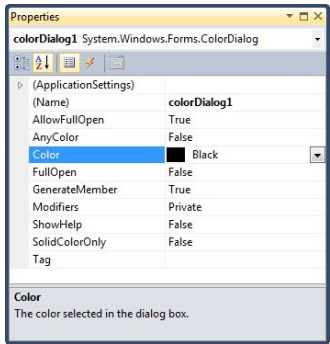


Figure 3

AllowFullOpen

If you look at Figure 1, you will see a button called Define Custom Colors on the ColorDialog. Clicking on this button opens the custom color editor area where you can define colors by setting RGB color values (between 0 to 255) and can also select a color from the color area as you can see in Figure 4.

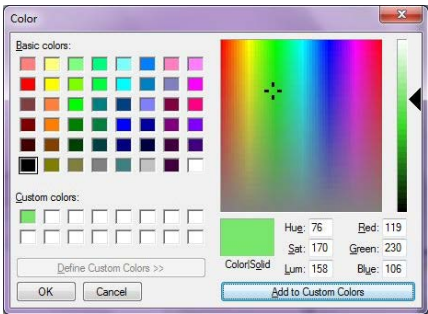


Figure 4

AllowFullOpen property makes sure that **Define Custom Color** option is enabled on a **ColorDialog**. If you wish to disable this option, you can set **AllowFullOpen** property to false and your **ColorDialog** will look like Figure 5.



Figure 5

The following code snippet sets the `AllowFullOpen` property to false.

```
colorDlg.AllowFullOpen = false;
```

Color, AnyColor, and SolidColorO

Color property is used to get and set the color selected by the user in a ColorDialog.

AnyColor is used to get and set whether a ColorDialog displays all available colors in the set of basic colors.

`SolidColorOnly` is used to get and set whether a `ColorDialog` restricts users to selecting solid colors only.

The following code snippet sets these properties.

```
colorDlg.AnyColor = true;
colorDlg.SolidColorOnly = false;
colorDlg.Color = Color.Red;
```

Using ColorDialog in Applications

Now let's create an application that will use a ColorDialog to set colors of bunch of controls. The Windows Forms application looks like Figure 6.

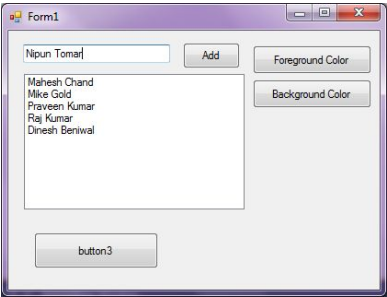


Figure 6

In Figure 6, we have a few Windows Forms controls and clicking on **Foreground Color** and **Background Color** buttons will let user select a color and set that color as foreground and background colors of the controls. After selecting foreground and background colors, the Form looks like Figure 7.

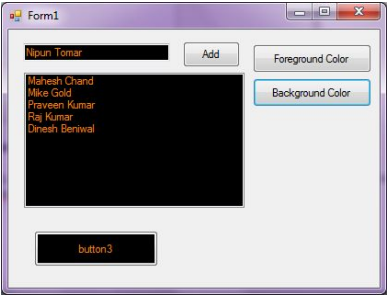


Figure 7

The following code snippet is the code for Foreground Color and Background Color buttons click event handlers.

```
private void ForegroundButton_Click(object sender, EventArgs e)
{
    ColorDialog colorDlg = new ColorDialog();
    colorDlg.AllowFullOpen = false;
    colorDlg.AnyColor = true;
    colorDlg.SolidColorOnly = false;
    colorDlg.Color = Color.Red;

    if (colorDlg.ShowDialog() == DialogResult.OK)
    {
        textBox1.ForeColor = colorDlg.Color;
        listBox1.ForeColor = colorDlg.Color;
        button3.ForeColor = colorDlg.Color;
    }
}

private void BackgroundButton_Click(object sender, EventArgs e)
{
    ColorDialog colorDlg = new ColorDialog();
    if (colorDlg.ShowDialog() == DialogResult.OK)
    {
        textBox1.BackColor = colorDlg.Color;
        listBox1.BackColor = colorDlg.Color;
        button3.BackColor = colorDlg.Color;
    }
}
```

Summary

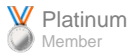
A ColorDialog control allows users to launch Windows Color Dialog and let them select a solid color or create a custom color from available colors. In this article, we discussed how to use a Windows Color Dialog and set its properties in a Windows Forms application.



Mahesh Chand

C# Corner founder, 10-times Microsoft MVP, technical author, and software architect. Mahesh authored his first book, A Programmer's Guide to ADO.NET in C# with APress at the age of 25. C# Corner is a memb... [Read more](#)

Personal Blog: <http://www.c-sharpcorner.com>



RELATED ARTICLES

- Introduction to Template Method Pattern
- Introduction to Memento Pattern
- C# Tutorial for Beginners: Part I
- ColorDialog : How to use?
- Drawing a Pie Chart in GDI+
- Tutorial: Working with Windows Forms - Part I

COMMENTS



Code and article updated.
Maresh Chand
•1 •47.4k •82.2m
Jun 28, 2010
0 0 Post Reply



Hi, Great Article, Thanks! Just one question: How do you position the Colordialog on the screen?
There are no Top or Left properties... Regards, Walter
Walter Kiess
•679 •20 •7.3k
Mar 08, 2012
0 0 Post Reply



Hi there and thanks for posting this, I was wondering is there a way to return the custom color in real time - as the user changes setting as opposed to having to wait for the OK button to be pressed ? - Thanks
James Cleary
•698 •1 •0
Jun 21, 2012
0 0 Post Reply



Niilce :)
Grabsi Amine
•637 •62 •3
Dec 16, 2012
0 0 Post Reply

