

Home

PUBLIC

Questions

Tags

Users

Companies

COLLECTIVES

Explore Collectives

TEAMS

Stack Overflow for Teams

Start collaborating and sharing organizational knowledge.

Free

Create a free team

Why Teams?

Load control style from separate file in wpf

Asked 11 years ago Modified 9 months ago Viewed 37k times

Ask Question

I have the following style added to my Windows.Resources

```
<Window.Resources>
...
<!--A Style that extends the previous TextBlock Style-->
<!--This is a "named style" with an x:key of TitleText-->
<Style BasedOn="{StaticResource {x:Type TextBlock}}"
    TargetType="TextBlock"
    x:Key="TitleText">
    <Setter Property="FontSize" Value="26"/>
    <Setter Property="Foreground">
        <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
            <LinearGradientBrush.GradientStops>
                <GradientStop Offset="0.0" Color="#990000" />
                <GradientStop Offset="1.0" Color="#8BFFFF" />
            </LinearGradientBrush.GradientStops>
        </LinearGradientBrush>
    </Setter.Value>
    </Setter>
</Style>
...
</Window.Resources>
```

I have a lot of these styles in my xaml code and I would like to save each component style to an extra file (not an external file), for example all the styles related to TextBlocks should be in a file called `TextBlockStyles.xaml`

How would I do this in wpf?

How do I link the style in my project ?

Thanks in advance

c# .net wpf

Share Improve this question Follow

edited Oct 11, 2022 at 13:59

Daniel Möller 84.6k ● 18 ● 192 ● 214

asked Jun 11, 2012 at 10:07

lebhero 1,371 ● 5 ● 18 ● 35

The Overflow Blog

- Why on earth are people paying for digital real estate? (Ep. 587)
- The Overflow #185: The hardest part of software is requirements

Featured on Meta

- Starting the Prompt Design Site: A New Home in our Stack Exchange Neighborhood
- Colors update: A more detailed look
- Temporary policy: Generative AI (e.g. ChatGPT) is banned
- Testing native, sponsored banner ads on Stack Overflow (starting July 6)

Linked

- 1 I need help changing a UI element's color with a text file in C#

Related

- 1 Loading Custom Styles in a WPF Custom Controls Library
- 6 Loading WPF Style from Resource File
- 2 Applying style to control programmatically
- 18 How to use styles from separate xaml files
- 0 wpf use elements declared in style
- 1 How to load style using xamlreader
- 2 How to access a Style outside the code

Load style library into WPF

- 0 Keep control resource consisting of multiple styles in a single place and use it from there in WPF app

Hot Network Questions

- Other than Will Riker and Deanna Troi, have we seen on-screen any commanding officers on starships who are married?
- Countering the Forcecage spell with reactions?
- Travelling from Frankfurt airport to Mainz with lot of luggage
- What 'Irish Oath' sounds like "Thoth"?
- How does the theory of evolution make it less likely that the world is designed?
- Is a CLA necessary?
- Short story about the best time to travel back to for each season, summer
- Book or novel with a man that exchanges his sword for an army
- Why do keywords have to be reserved words?
- Air that escapes from tire smells really bad
- Is confession legally privileged?
- Is the line between photosynthesis and chemisorption species specific?
- What's the difference between 'ultio' and 'vindicta'?
- Can the Secret Service arrest someone who uses an illegal drug inside of the White House?
- How to play the "Ped" symbol when there's no corresponding release symbol
- How did the IBM 360 detect memory errors?
- Is Password Hashing Bad?
- Were Patton's and/or other generals' vehicles prominently flagged with stars (and if so, why)?
- Table in landscape mode keeps going out of bounds
- In the movie Loper, why do assassins in the future use inaccurate weapons such as blunderbuss?
- Typo in cover letter of the journal name where my manuscript is currently under review
- Space elevator from Earth to Moon with multiple temporary anchors
- Where is the "flux in core" inside soldering wire? How does it change the soldering wire vs the pure element?
- Characters with only one possible next character

Question feed

Не нашли ответ? Задайте вопрос на Stack Overflow на русском.

You use merged resource dictionaries

In your app.xaml you would use

```
<Application.Resources>
<ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary
            Source="{Your.Assembly.Name;component/TextBlockStyles.xaml}" />
        ... other dictionaries here
    </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
</Application.Resources>
```

or directly into a UserControl would be

```
<UserControl.Resources>
<ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary
            Source="{Your.Assembly.Name;component/TextBlockStyles.xaml}" />
        ... other dictionaries here
    </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
</UserControl.Resources>
```

You can shorten `Source="..."` to just `Source="TextBlockStyles.xaml"` if the file is in the same assembly and in the root of the project, or alternatively `Source="Styles\TextBlockStyles.xaml"` if you put the resource dictionary into the folder `Styles`.

Share Improve this answer Follow

edited Jun 11, 2012 at 10:25

answered Jun 11, 2012 at 10:16

Phil 42.1k ● 9 ● 100 ● 100

5 So what does the file called `TextBlockStyles.xaml` look like? – The Muffin Man Oct 19, 2017 at 21:47

@TheMuffinMan: An example [can be found here](#). (It is called `Dictionary1.xaml` in this example.)

– knote Nov 2, 2017 at 22:09

Don't forget to add a reference to the dll the style is defined in. – dba Jul 26, 2018 at 11:51

Add a comment

Use case: you have a user control called `MyView.xaml` with a button. You want to style the button with an external XAML file.

In `MyView.xaml`:

```
<User Control ...namespaces...>
<UserControl.Resources>
    <ResourceDictionary>
        ...converters...

        <ResourceDictionary.MergedDictionaries>
            <ResourceDictionary Source="MyButton.xaml" />
        </ResourceDictionary.MergedDictionaries>
    </ResourceDictionary>
</UserControl.Resources>

...the rest of the control...
</UserControl>
```

In `MyButton.xaml`:

```
<ResourceDictionary
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:MSDNSample">

    <Style x:Key="FooButton" TargetType="{x:Type Button}">
        <Setter Property="Background" Value="Pink" />
    </Style>
</ResourceDictionary>
```

Back to `MyView.xaml` ("the rest of the control"):

```
<Button Style="{StaticResource FooButton}">
    Hello world
</Button>
```

Share Improve this answer Follow

edited May 23, 2018 at 18:32

answered May 23, 2018 at 17:25

Ben 54.5k ● 49 ● 178 ● 224

Add a comment

In **Solution Explorer** Right Click on your **Project** Select **Add** After that click on **Resource Dictionary**. Choose name and add to your project. Open **App.xaml** Add This Code in

Application Tag

```
<Application.Resources>
<ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary Source="YourStyle.xaml" />
    </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
</Application.Resources>
```

In `YourStyle.xaml`:

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr:namespace:APPNAME">

    <Style x:Key="ButtonStyle" TargetType="{x:Type Button}">
        <Setter Property="Background" Value="Pink" />
    </Style>
</ResourceDictionary>
```

Share Improve this answer Follow

answered Jul 26, 2018 at 11:47

MDRezaY 51 ● 1 ● 6

Add a comment

you are looking for the dynamic resources, well the best way is to load and merge the dictionary in the resources. application or either on control page. here is a good sample for it.

1 <http://blogs.msdn.com/b/wslpsdk/archive/2007/06/08/defining-and-using-shared-resources-in-a-custom-control-library.aspx>

```
<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

    <ResourceDictionary Source="Dictionary1.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>
```

this.Resources.MergedDictionaries.Add(Smyresourcedictionary);

Share Improve this answer Follow

answered Jun 11, 2012 at 12:12

JSJ 5,633 ● 3 ● 25 ● 31

Add a comment

Simply, go to your Window (example: `MainWindow.xaml`) where you want to include the resource from the outer file and use [MergedDictionaries](#) principle to refer to that file:

1

```
<Window x:Class="UseMergedResource.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:de="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="de">
    Title="MainWindow"
    Height="680"
    Width="680">
<Window.Resources>

<!-- DECLARING MERGED DICTIONARY -->
<ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary Source="Merged/BrushResources.xaml" />
    </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>

</Window.Resources>

<StackPanel>
    <Rectangle Width="200"
        Height="100"
        Fill="{StaticResource PrimaryBrush}" /> <!-- USAGE HERE -->
</StackPanel>
</Window>
```

From above `Merged/BrushResources.xaml` is the location of the resource file, which is located under the folder called `Merged`.

Now if you are wondering what should be the declaration syntax in the outer file, check this:

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <!-- Location for application brushes -->
    <SolidColorBrush x:Key="BorderBrush"
        Color="Orange" />
    <SolidColorBrush x:Key="HighLightBrush"
        Color="LightBlue" />
    <SolidColorBrush x:Key="PrimaryBrush"
        Color="Pink" />
    <SolidColorBrush x:Key="AccentBrush"
        Color="Yellow" />

</ResourceDictionary>
```

If you want to make the resource available through all application (visible in all your windows) then declare in the `App.xaml` resources section.

Share Improve this answer Follow

edited Mar 17, 2020 at 18:08

answered Mar 17, 2020 at 18:03

Arsen Khachatryan 7,884 ● 4 ● 42 ● 42

Note that if you have resources already in your `<Window.Resources>` section, they should go inside `<ResourceDictionary>`, e.g. just before the `</ResourceDictionary>` line. – Geoff Mar 22 at 23:43

Add a comment

Your Answer

B I

Sign up or log in

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

Post as a guest

Name

Email

Required, but never shown

By clicking "Post Your Answer", you agree to our [terms of service](#) and acknowledge that you have read and understand our [privacy policy](#) and [code of conduct](#).

Not the answer you're looking for? Browse other questions tagged [c#](#) [.net](#) [wpf](#) or [ask your own question](#).