

# How to load image to WPF in runtime?

Asked 8 years, 6 months ago   Active 18 days ago   Viewed 103k times

It seems like it's quite complicated to load an image in runtime to a WPF window.

17

```
Image image;
image = new Uri("Bilder/sas.png", UriKind.Relative);
?????.Source = new BitmapImage(image);
```



5

I'm trying this code, but I need some help to get it to work. I get some red lines below the code! I also wonder if I need to add some extra code inside the XAML code or is in enough with this:

```
<Image Height="200" HorizontalAlignment="Left" Margin="12,12,0,0" Name="image1"
Stretch="Fill" VerticalAlignment="Top" Width="350" />
```

Wonder because I have seen examples with sorces to the images inside the XAML tags.

EDIT:

I'm using this now:

```
var uri = new Uri("pack://application:,,,/sas.png");
var bitmap = new BitmapImage(uri);
image1.Source = bitmap;
```

The XAML:

```
<Grid Width="374">
    <Image Height="200" HorizontalAlignment="Left" Margin="12,12,0,0" Name="image1"
Stretch="Fill" VerticalAlignment="Top" Width="350" />
    <Button Content="Start" Height="23" HorizontalAlignment="Left" Margin="12,226,0,0"
Name="btnStart" VerticalAlignment="Top" Width="75" />
    <Button Content="Land" Height="23" HorizontalAlignment="Left" Margin="287,226,0,0"
Name="btnLand" VerticalAlignment="Top" Width="75" />
    <ComboBox Height="23" HorizontalAlignment="Left" Margin="110,226,0,0"
Name="cmbChangeRoute" VerticalAlignment="Top" Width="156" />
</Grid>
```

EDIT 2: My issue is solved, this code works fine:

```
BitmapImage image = new BitmapImage();
image.BeginInit();
image.UriSource = new Uri("pack://application:,,,/Resources/" + company + ".png");
image.EndInit();
```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Sign up



c# wpf

Share Improve this question Follow

edited Oct 26 '19 at 12:29



ASh

29.2k

8

46

70

asked Aug 9 '12 at 10:02



3D-kreativ

8,183

34

83

148

2 Those first two lines of code can't be right; a Uri is NOT an Image. – Erno Aug 9 '12 at 10:04


Thanks! But know it complain about the path! It's looking for the path at C: when it should be bin/debug!? – 3D-kreativ Aug 9 '12 at 11:07

## 2 Answers

Active Oldest Votes

▲ In WPF an image is typically loaded from a [Stream](#) or an [Uri](#).

39 [BitmapImage](#) supports both and an Uri can even be passed as constructor argument:

▼ 

```
var uri = new Uri("http://...");
var bitmap = new BitmapImage(uri);
```

If the image file is located in a local folder, you would have to use a `file://` Uri. You could create such a Uri from a path like this:

```
var path = Path.Combine(Environment.CurrentDirectory, "Bilder", "sas.png");
var uri = new Uri(path);
```

If the image file is an assembly resource, the Uri must follow the the [Pack Uri](#) scheme:

```
var uri = new Uri("pack://application:,,,/Bilder/sas.png");
```

In this case the Visual Studio Build Action for `sas.png` would have to be `Resource`.

Once you have created a `BitmapImage` and also have an [Image](#) control like in this XAML

```
<Image Name="image1" />
```

you would simply assign the `BitmapImage` to the `Source` property of that Image control:

```
image1.Source = bitmap;
```



108k

10

128

224

1 But what about the code in XAML?! Like this: `<Image Name="Image1" />` – 3D-kreativ Aug 9 '12 at 11:37

No I missed that! Well since I hade a folder like that in the bin/debug it should work? But know I'm trying to get it work from the resources and would preciate some help to get it to work? To start, I guess I dont need any extra in the XAML code, since the image already has a name: image1 ?? – 3D-kreativ Aug 9 '12 at 11:48

A folder in bin/Debug is not sufficient. It would have to be in your VS project, as I said. And the XAML `<Image Name="image1" ... />` creates a member variable named `image1` of type `Image`, which is a WPF control, not a bitmap/image. – Clemens Aug 9 '12 at 11:54

I'm using your code, but I only get an error about Uri prefix is not recognised !? – 3D-kreativ Aug 9 '12 at 12:11

3 There's no need for the `BeginInit` and `EndInit`. Simply use the `BitmapImage(Uri)` constructor. Otherwise that code is *in no way* different from what I answered here, except the the folder is called `Resources` , not `Bilder` . – Clemens Aug 9 '12 at 13:41



Make sure that your `sas.png` is marked as `Build Action: Content` and `Copy To Output Directory: Copy Always` in its Visual Studio Properties ...

4



I think the C# source code goes like this...



```
Image image = new Image();
image.Source = (new
ImageSourceConverter()).ConvertFromString("pack://application:,,,/Bilder/sas.png") as
ImageSource;
```

and XAML should be

```
<Image Height="200" HorizontalAlignment="Left" Margin="12,12,0,0"
Name="image1" Stretch="Fill" VerticalAlignment="Top"
Source=" ../Bilder/sas.png"
Width="350" />
```

## EDIT

Dynamically I think XAML would provide best way to load Images ...

```
<Image Source="{Binding Converter={StaticResource MyImageSourceConverter}}"
x:Name="MyImage"/>
```

where `image.DataContext` is string path.

```
MyImage.DataContext = "pack://application:,,,/Bilder/sas.png";
```



```

object parameter_, System.Globalization.CultureInfo culture_)
{
    return (new ImageSourceConverter()).ConvertFromString (value.ToString());
}

public object ConvertBack(object value, Type targetType,
object parameter, CultureInfo culture)
{
    throw new NotImplementedException();
}
}

```

Now as you set a different data context, `Image` would be automatically loaded at runtime.

Share Improve this answer Follow

edited Oct 26 '19 at 12:30

answered Aug 9 '12 at 10:09



ASh

29.2k

8

46

70



WPF-it

18.6k

6

51

68

Hmm, but if I want to load different images during runtime. When I create a object of the WPF window I want to pass different images, then I guess the XAML code would look different? I have saved all the images in a folder inside Bin/Debug, OK? – 3D-kreativ Aug 9 '12 at 10:19

Thanks! But isn't his way to complicated to just use some images!? – 3D-kreativ Aug 9 '12 at 10:50

1 :) Thats how we would do in WPF... if you dont want XAML then use the C# code I originally gave....  
`image.Source = (new ImageSourceConverter()).ConvertFromString("pack://application:,,,/Bilder/sas.png" ) as ImageSource;` – WPF-it Aug 9 '12 at 11:48

Thanks for the help, but I'm using the another code! See my second edit. – 3D-kreativ Aug 9 '12 at 13:09

Think you mean `value_.ToString()` – tonydev314 Jul 13 '20 at 10:24