Animated gif

<https://stackoverflow.com/questions/32927243/c-sharp-wpf-gif-black-background>

**Resenii**

You might consider using a small lib for this. Thomas Levesque has created a WpfAnimatedGif which is available as a Nuget package. Just type the following in Package Manager Console:

Install-Package WpfAnimatedGif

PM> Install-Package WpfAnimatedGif

Installing 'WpfAnimatedGif 1.4.13'.

Successfully installed 'WpfAnimatedGif 1.4.13'.

Adding 'WpfAnimatedGif 1.4.13' to AnimatedGIF.

Successfully added 'WpfAnimatedGif 1.4.13' to AnimatedGIF.

Let's then define a small GUI to test this lib out:

<Window x:Class="AnimatedGIF.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="Hypnotize"

xmlns:wpfanimated="http://wpfanimatedgif.codeplex.com"

Height="350" Width="525">

<Grid>

<Image wpfanimated:ImageBehavior.AnimatedSource="a\_transparent\_gif.gif"

wpfanimated:ImageBehavior.RepeatBehavior="1"

wpfanimated:ImageBehavior.AnimateInDesignMode="False"

wpfanimated:ImageBehavior.AutoStart="True"

Height="300" />

</Grid>

</Window>

**Ovladani:**

Musim do VM dostat instanci mediaElementu

<Image x:Name="scaner"

private void Button\_Click(object sender, RoutedEventArgs e)

{

var controller = ImageBehavior.GetAnimationController(this.scaner);

controller.Pause();

}

[**https://github.com/XamlAnimatedGif/WpfAnimatedGif/wiki**](https://github.com/XamlAnimatedGif/WpfAnimatedGif/wiki)

**Documentation**

**Adding WPF Animated GIF to your project**

**Using NuGet**

*(Note:*[*NuGet*](http://nuget.codeplex.com/)*must already be installed)*

* Right-click on your project, and select *Manage Nuget Packages*
* Select the *NuGet official package source* repository in the *Online* category
* Type "wpfanimatedgif" in the search box
* Click the *Install* button on the only search result

**Manually**

* Download WPF Animated GIF from the [download page](http://wpfanimatedgif.codeplex.com/releases/)
* Unpack the zip file somewhere
* Right-click on your project, and select *Add Reference*
* Browse to the WpfAnimatedGif.dll file and select it

**XAML usage**

On the root element of your XAML file, declare the XML namespace for WPF Animated GIF:

<Window x:Class="WpfAnimatedGif.Demo.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:gif="http://wpfanimatedgif.codeplex.com">

Then, on your Image control, instead of setting the Source property, set the ImageBehavior.AnimatedSource attached property to the image you want:

<Image gif:ImageBehavior.AnimatedSource="Images/animated.gif" />

If you need to change the repeat behavior, just set the ImageBehavior.RepeatBehavior attached property:

<!-- Repeat 3 times -->

<Image gif:ImageBehavior.RepeatBehavior="3x"

gif:ImageBehavior.AnimatedSource="Images/animated.gif" />

<!-- Repeat for 10 seconds -->

<Image gif:ImageBehavior.RepeatBehavior="0:0:10"

gif:ImageBehavior.AnimatedSource="Images/animated.gif" />

<!-- Repeat forever -->

<Image gif:ImageBehavior.RepeatBehavior="Forever"

gif:ImageBehavior.AnimatedSource="Images/animated.gif" />

Valid values for this property include:

* Forever: the image animation will be repeated forever
* a fixed number of iterations
* a duration specified in days, minutes, seconds, and fractional seconds

The default value is 0x, which means it will use the repeat count from the GIF metadata (using the Netscape Application Block). If no repeat count is specified in the image metadata, the animation will run only once.

(see the [RepeatBehavior documentation on MSDN](http://msdn.microsoft.com/en-us/library/system.windows.media.animation.repeatbehavior) for more details)

Both properties can be set through a binding.

**Code usage**

You can set the image in code by using the ImageBehavior.SetAnimatedSource method:

var image = new BitmapImage();

image.BeginInit();

image.UriSource = new Uri(fileName);

image.EndInit();

ImageBehavior.SetAnimatedSource(img, image);

Similarly, you can set the repeat behavior with the ImageBehavior.SetRepeatBehavior method:

// Repeat 3 times

ImageBehavior.SetRepeatBehavior(img, new RepeatBehavior(3));

// Repeat for 10 seconds

ImageBehavior.SetRepeatBehavior(img, new RepeatBehavior(TimeSpan.FromSeconds(10)));

// Repeat forever

ImageBehavior.SetRepeatBehavior(img, RepeatBehavior.Forever);

**Animation preview in design mode**

By default the GIF image is not animated in the XAML designer, to avoid distracting attention. If you want to see the animation in design mode, just set the AnimateInDesignMode attached property to true on the root of the XAML file. It will affect all animated GIF images under the root element.

<Window x:Class="WpfAnimatedGif.Demo.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:gif="http://wpfanimatedgif.codeplex.com"

gif:ImageBehavior.AnimateInDesignMode="True">

**Animation completion notification**

If you need to be notified when the animation completes, you can subscribe to the AnimationCompleted attached event:

<Image gif:ImageBehavior.RepeatBehavior="3x"

gif:ImageBehavior.AnimatedSource="Images/animated.gif"

gif:ImageBehavior.AnimationCompleted="AnimationCompleted"/>

Note that the event obviously won't be raised if the repeat behavior is Forever.

**Manual control of the animation**

In order to control the animation, you need to get the ImageAnimationController for the Imagecontrol:

var controller = ImageBehavior.GetAnimationController(imageControl);

You can then use the controller to pause, resume, or seek the animation:

// Pause the animation

controller.Pause();

// Resume the animation (or restart it if it was completed)

controller.Play();

// Go to the last frame

controller.GotoFrame(controller.FrameCount - 1);

If you intend to control the animation manually, you probably don't want it to start automatically. In this case you can set the AutoStart attached property to false:

<Image Name="img"

gif:ImageBehavior.AnimatedSource="animated.gif"

gif:ImageBehavior.AutoStart="False" />

There is also a CurrentFrameChanged attached event that you can use to be notified when the current frame has changed (in order to show a progress bar, for instance).

Note: GetAnimationController will return null if the animation is not yet fully loaded; you can be notified of when the animation is done loading by subscribing to the AnimationLoaded attached event.

**OvladaniO**