https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/graphics-multimedia/animation-overview

**Animation Overview**

## Introducing Animations

Animation is an illusion that is created by quickly cycling through a series of images, each slightly different from the last.

Animation on a computer is similar. For example, a program that makes a drawing of a rectangle fade out of view might work as follows.

 The program creates a timer.

 The program checks the timer at set intervals to see how much time has elapsed.

 Each time the program checks the timer, it computes the current opacity value for the rectangle based on how much time has elapsed.

 The program then updates the rectangle with the new value and redraws it.

Prior to WPF, Microsoft Windows developers had to create and manage their own timing systems or use special custom libraries. WPF includes an efficient timing system that is deeply integrated into the WPF framework

WPF handles all the behind-the-scenes work of managing a timing system and redrawing the screen efficiently. It provides timing classes that enable you to focus on the effects you want to create, instead of the mechanics of achieving those effects

## WPF Property Animation System

in WPF, you animate objects by applying animation to their individual properties.

For a property to have animation capabilities, it must meet the following three requirements:

 It must be a dependency property.

 It must belong to a class that inherits from [DependencyObject](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.dependencyobject) and implements the [IAnimatable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ianimatable) interface.

 There must be a compatible animation type available. (If WPF does not provide one, you can create your own. See the [Custom Animations Overview](https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/graphics-multimedia/custom-animations-overview).)

## Example: Make an Element Fade In and Out of View

1. create a [Rectangle](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.rectangle) element
2. Create a DoubleAnimation

Because the [Opacity](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.opacity) property is of type [Double](https://docs.microsoft.com/zh-cn/dotnet/api/system.double), you need an animation that produces double values. A [DoubleAnimation](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimation) is one such animation. A [DoubleAnimation](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimation) creates a transition between two double values

1. Create a Storyboard

To apply an animation to an object, you create a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) and use the [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) and [TargetProperty](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetproperty) attached properties to specify the object and property to animate

<Storyboard>

<DoubleAnimation

From="1.0" To="0.0" Duration="0:0:1"

AutoReverse="True" RepeatBehavior="Forever" />

</Storyboard>

The [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) has to know where to apply the animation. Use the [Storyboard.TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) attached property to specify the object to animate.

1. Associate the Storyboard with a Trigger
   1. The easiest way to apply and start a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) in XAML is to use an event trigger.
   2. Create a [BeginStoryboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.beginstoryboard) object and associate your storyboard with it. A [BeginStoryboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.beginstoryboard) is a type of [TriggerAction](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.triggeraction) that applies and starts a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard).
   3. Create an [EventTrigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger) and add the [BeginStoryboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.beginstoryboard) to its [Actions](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger.actions) collection
   4. Add the [EventTrigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger) to the [Triggers](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement.triggers) collection of the Rectangle.

4 (Code): Associate the Storyboard with an Event Handler

myRectangle.Loaded += new RoutedEventHandler(myRectangleLoaded);

private void myRectangleLoaded(object sender, RoutedEventArgs e)

{

myStoryboard.Begin(this);

}

## Animation Types

Because animations generate property values, different animation types exist for different property types

they follow a strict naming convention that makes it easy to differentiate between them:

* <*Type*>Animation

Known as a "From/To/By" or "basic" animation, these animate between a starting and destination value, or by adding an offset value to its starting value

* <Type>AnimationUsingKeyFrames

Key frame animations are more powerful than From/To/By animations because you can specify any number of target values and even control their interpolation method.

<Type>AnimationUsingPath

Path animations enable you to use a geometric path in order to produce animated values.

### Animations Are Timelines

an animation is a [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline), it also represents a segment of time. An animation also calculates output values as it progresses through its specified segment of time (or [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.duration)).

Three frequently used timing properties are [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.duration), [AutoReverse](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.autoreverse), and [RepeatBehavior](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.repeatbehavior).

### Applying and Starting Storyboards

To start a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) in code, you can use an [EventTrigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger) or use the [Begin](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.begin) method of the [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) class.

# Animation and Timing System Overview

## Timelines and Clocks

a [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline), doesn't do anything other than just describe a segment of time. It's the timeline's [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) object that does the real work. an animation is a type of [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) that produces output values.

an animation class describes how output values should be calculated, but it’s the [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) that was created for the animation that drives the animation output and applies it to properties.

When you animate by using a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) or the [BeginAnimation](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animatable.beginanimation) method, clocks are automatically created for your timelines and animations

You can also create a [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) explicitly by using the [CreateClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.createclock) method of your [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline).

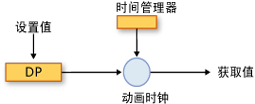
If the [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) contains child timelines, it creates [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) objects for them as well. The resulting [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) objects are arranged in trees that match the structure of the [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) objects tree from which they are created.

## Clocks and the Time Manager

it’s the time manager that manages the [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.mediaplayer.clock) objects created for your timelines.. A time manager is automatically created for each WPF application and is invisible to the application developer. The time manager is the root of a tree of [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.mediaplayer.clock) objects and controls the flow of time in that tree

The time manager "ticks" many times per second;

The following illustration shows the relationship between the time manager, and [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock), and an animated dependency property.



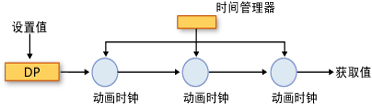
When the time manager ticks, it updates the time of every [Active](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockstate#System_Windows_Media_Animation_ClockState_Active) [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) in the application. If the [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) is an [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock), it uses the [GetCurrentValue](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline.getcurrentvalue) method of the [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) from which it was created to calculate its current output value

#### Clock Groups

A [ClockGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockgroup) is created for timelines that group other timelines, such as the [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) class, which groups animations and other timelines.

#### Composition

It's possible to associate multiple clocks with a single property, in which case each clock uses the output value of the preceding clock as its base value.



## Current Values and Base Values of Properties

An animatable property can have two values: a base value and a current value. When you set property using its CLR accessor or the [SetValue](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.dependencyobject.setvalue) method, you set its base value. When a property is not animated, its base and current values are the same.

When you animate a property, the [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock) sets the property's current value. Retrieving the property's value through its CLR accessor or the [GetValue](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.dependencyobject.getvalue) method returns the output of the [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock) when the [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock) is [Active](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockstate#System_Windows_Media_Animation_ClockState_Active) or [Filling](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockstate#System_Windows_Media_Animation_ClockState_Filling). You can retrieve the property's base value by using the [GetAnimationBaseValue](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ianimatable.getanimationbasevalue) method.

# From/To/By Animations Overview

A From/To/By animation is a type of [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) that creates a transition between a starting value and an ending value

## From/To/By Animation Types

use the following naming convention:

<Type> Animation

Where <Type> is the type of value that the class animates.

## Target Values

you can also specify only a starting value, a destination value, or an offset value. In these cases, the animation obtains the missing target value from the property that is being animated

**Starting Value**

If you specify only the [From](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimation.from) property, the animation transitions from that value to the base value of the animated property

**Ending Value**

If you use the [To](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimation.to) property by itself, the animation obtains its starting value from the property that is being animated or from the output of another animation that is applied to the same property

**Offset Value**

If you specify only the [By](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimation.by) property, the animation adds the offset value to the base value of the property or to the output of another animation.

# Key-Frame Animations Overview

while a From/To/By animation creates a transition between two values, a single key-frame animation can create transitions among any number of target values

A key-frame animation's target values are described using key frames objects

When the animation runs, it transitions between the frames you specified.

An animation's interpolation method defines how it transitions from one value to the next. There are three types of interpolations: discrete, linear, and splined.

<DoubleAnimationUsingKeyFrames

Storyboard.TargetName="MyAnimatedTranslateTransform"

Storyboard.TargetProperty="X"

Duration="0:0:10">

<LinearDoubleKeyFrame Value="0" KeyTime="0:0:0" />

<LinearDoubleKeyFrame Value="350" KeyTime="0:0:2" />

<LinearDoubleKeyFrame Value="50" KeyTime="0:0:7" />

<LinearDoubleKeyFrame Value="200" KeyTime="0:0:8" />

</DoubleAnimationUsingKeyFrames>

## Key-Frame Animation Types

aming convention:

<Type> AnimationUsingKeyFrames

Where <Type> is the type of value that the class animates

## Target Values (key frames) and Key Times

Key frame types adhere to the following naming convention:

<InterpolationMethod><Type> KeyFrame

Where <InterpolationMethod> is the interpolation method the key frame uses and <Type> is the type of value that the class animates

Every key frame type provides these two properties.

* The [Value](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.value) property specifies the target value for that key-frame.
* The [KeyTime](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.keytime) property specifies when (within the animation's [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.duration)) a key frame's [Value](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.value) is reached.

When a key frame animation begins, iterates through its key frames in the order defined by their [KeyTime](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.keytime) properties.

If there is no key frame at time 0, the animation creates a transition between the target property's current value and the [Value](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.value) of the first key frame; otherwise, the animation's output value becomes the value of the first key frame.

The animation creates a transition between the [Value](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.value) of the first and second key frames using the interpolation method specified by the second key frame. The transition starts at the first key frame's [KeyTime](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.keytime) and ends when the second key frame's [KeyTime](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.ikeyframe.keytime) is reached.

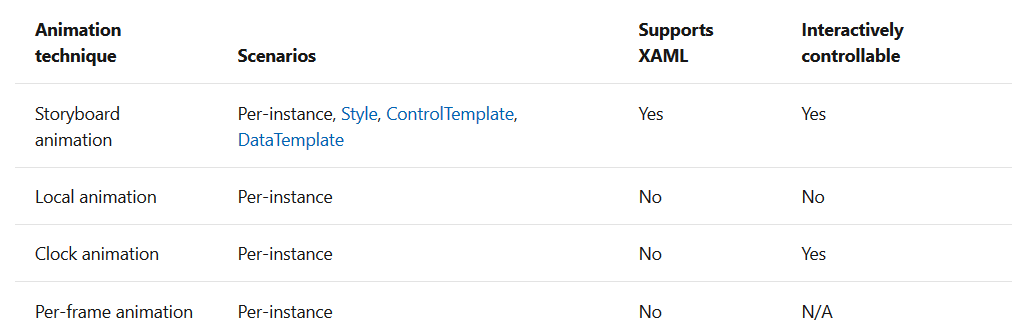
If the animation's [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.duration) is [Automatic](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.duration.automatic) or its [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.duration) is equal to the time of the last key frame, the animation ends. Otherwise, if the animation's [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.duration) is greater than the key time of the last key frame, the animation holds the key frame value until it reaches the end of its [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.duration).

# Property Animation Techniques Overview

## Different Ways to Animate

Because there are many different scenarios for animating properties, WPF provides several approaches for animating properties.

For each approach, the following table indicates whether it can be used per-instance, in styles, in control templates scenarios



## Storyboard Animations

For an object to be animated by a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard), it must be a [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) or [FrameworkContentElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkcontentelement), or it must be used to set a [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) or [FrameworkContentElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkcontentelement). For more details

A [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) is a special type of container [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) that provides targeting information for the animations it contains. To animate with a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard), you complete the following three steps.

 Declare a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) and one or more animations.

 Use the [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) and [TargetProperty](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetproperty) attached properties to specify the target object and property of each animation.

 (Code only) Define a [NameScope](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.namescope) for a [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) or [FrameworkContentElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkcontentelement). Register the names of the objects to animate with that [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) or [FrameworkContentElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkcontentelement).

 Begin the [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard).

There are two ways to begin a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard): you can use the [Begin](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.begin) method provided by the [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) class, or you can use a [BeginStoryboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.beginstoryboard) action

A [BeginStoryboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.beginstoryboard) action can be used in an [EventTrigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger), property [Trigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.trigger), or a [DataTrigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.datatrigger).

Local animations provide a convenient way to animate a dependency property of any [Animatable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animatable) object

Unlike a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) animation, a local animation can animate an object that isn't associated with a [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) or a [FrameworkContentElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkcontentelement). You also don't have to define a [NameScope](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.namescope) for this type of animation.

 Create an [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) object.

 Use the [BeginAnimation](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animatable.beginanimation) method of the object that you want to animate to apply the [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) to the property that you specify.

// Apply the animation to the brush's Color property.

myBrush.BeginAnimation(SolidColorBrush.ColorProperty, myColorAnimation);

aButton.Background = myBrush;

## Clock Animations

Use [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.mediaplayer.clock) objects when you want to animate without using a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) and you want to create complex timing trees or interactively control animations after they start

To apply a single [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) to a property, you complete the following steps.

1. Create an [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) object.
2. Use the [CreateClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline.createclock) method of the [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) to create an [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock).
3. Use the [ApplyAnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animatable.applyanimationclock) method of the object that you want to animate to apply the [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock) to the property you specify.

// Create a clock the for the animation.

AnimationClock myClock = myAnimation.CreateClock();

// Associate the clock the ScaleX and

// ScaleY properties of the button's

// ScaleTransform.

myScaleTransform.ApplyAnimationClock(

ScaleTransform.ScaleXProperty, myClock);

To create a timing tree and use it animate properties, you complete the following steps.

1. Use [ParallelTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.paralleltimeline) and [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) objects to create the timing tree.
2. Use the [CreateClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timelinegroup.createclock) of the root [ParallelTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.paralleltimeline) to create a [ClockGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockgroup).
3. Iterate through the [Children](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockgroup.children) of the [ClockGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockgroup) . For each [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock) child, use the [ApplyAnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animatable.applyanimationclock) method of the object that you want to animate to apply the [AnimationClock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationclock) to the property you specify

# Storyboards Overview

## What Is a Storyboard?

Other timeline classes（Container timelines） are provided to help you organize sets of timelines, and to apply timelines to properties.，making it easy to organize and control complex timing behaviors 。Container timelines derive from the [TimelineGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timelinegroup) class, and include [ParallelTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.paralleltimeline) and [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard).

## Where Can You Use a Storyboard?

A [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) can be used to animate dependency properties of animatable classes .However, because storyboarding is a framework-level feature, the object must belong to the [NameScope](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.namescope) of a [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) or a [FrameworkContentElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkcontentelement).

## Targeting Framework Elements, Framework Content Elements, and Freezables

<https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/graphics-multimedia/bitmap-effects-overview>

The previous section mentioned that, for an animation to find its target, it must know the target's name and the property to animate

For the [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.setter.targetname) property to work, the targeted object must have a name. Assigning a name to a [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) or a [FrameworkContentElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkcontentelement) in XAML is different than assigning a name to a [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) object.

To enable the targeting of a framework element or a framework content element in XAML, you set its [Name](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement.name) property. In code, you also need to use the [RegisterName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.namescope.registername) method to register the element's name with the element for which you've created a [NameScope](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.namescope).

Rectangle myRectangle = new Rectangle();

myRectangle.Name = "MyRectangle";

// Create a name scope for the page.

NameScope.SetNameScope(this, new NameScope());

this.RegisterName(myRectangle.Name, myRectangle);

Storyboard.SetTargetName(myDoubleAnimation, myRectangle.Name);

Storyboard.SetTargetProperty(myDoubleAnimation,

new PropertyPath(Rectangle.WidthProperty));

[Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) types are those classes that inherit from the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) class. Examples of [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) include [SolidColorBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.solidcolorbrush), [RotateTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.rotatetransform), and [GradientStop](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.gradientstop).

To enable the targeting of a [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) by an animation in XAML, you use the [x:Name Directive](https://docs.microsoft.com/zh-cn/dotnet/framework/xaml-services/x-name-directive) to assign it a name. In code, you use the [RegisterName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.namescope.registername) method to register its name with the element for which you've created a [NameScope](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.namescope).

<SolidColorBrush x:Name="MySolidColorBrush" Color="Blue" />

[Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) objects use name scopes to resolve the [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) property. For more information about WPF name scopes, see [WPF XAML Namescopes](https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/advanced/wpf-xaml-namescopes). If the [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) property is omitted, the animation targets the element on which it is defined, or, in the case of styles, the styled element.

## Indirect Targeting

There are times a [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) can't be targeted directly by an animation, such as when the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) is declared as a resource or used to set a property value in a style

Instead of setting the [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) property with the name of the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable), you give it the name of the element to which the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) "belongs."

To animate the brush, you would set the animation's [TargetProperty](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetproperty) with a chain of properties that starts at the property of the framework element or framework content element the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) was used to set and ends with the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) property to animate.

<ColorAnimation

Storyboard.TargetName="Rectangle01"

Storyboard.TargetProperty="Fill.Color"

From="Blue" To="AliceBlue" Duration="0:0:1" />

DependencyProperty[] propertyChain =

new DependencyProperty[]

{Rectangle.FillProperty, SolidColorBrush.ColorProperty};

string thePath = "(0).(1)";

PropertyPath myPropertyPath = new PropertyPath(thePath, propertyChain);

Storyboard.SetTargetProperty(myColorAnimation, myPropertyPath);

Sometimes you need to target a freezable contained in a collection or array.

To target a freezable contained in a collection, you use the following path syntax.

|  |
| --- |
| *ElementPropertyName* .Children[ *CollectionIndex* ]. *FreezablePropertyName* |

<TransformGroup x:Key="MyTransformGroupResource"

x:Shared="False">

<ScaleTransform />

<RotateTransform />

</TransformGroup>

<DoubleAnimation

Storyboard.TargetName="Rectangle02"

Storyboard.TargetProperty="RenderTransform.Children[1].Angle"

From="0" To="360" Duration="0:0:1" />

## Interactively Controlling a Storyboard in XAML

If you give the [BeginStoryboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.beginstoryboard) a name by specifying its [Name](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.beginstoryboard.name) property, you make it a controllable storyboard. You can then interactively control the storyboard after it's started

<EventTrigger RoutedEvent="Button.Click" SourceName="BeginButton">

<BeginStoryboard Name="MyBeginStoryboard">

<Storyboard>

<DoubleAnimation

Storyboard.TargetName="MyRectangle"

Storyboard.TargetProperty="(Rectangle.Opacity)"

From="1.0" To="0.0" Duration="0:0:5" />

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent="Button.Click" SourceName="PauseButton">

<PauseStoryboard BeginStoryboardName="MyBeginStoryboard" />

</EventTrigger>

<EventTrigger RoutedEvent="Button.Click" SourceName="ResumeButton">

<ResumeStoryboard BeginStoryboardName="MyBeginStoryboard" />

</EventTrigger>

## Interactively Controlling a Storyboard by Using Code

The following list shows the methods that can be used to manipulate a [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) after it has started:

* [Pause](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.pause)
* [Resume](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.resume)
* [Seek](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.seek)
* [SkipToFill](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.skiptofill)
* [Stop](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.stop)
* [Remove](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.remove)

## Animate in a Style

You can use [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) objects to define animations in a [Style](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.style).

You don't specify a [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname); the [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) always targets the element to which the [Style](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.style) is applied. To target [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) objects, you must use indirect targeting.

 You can't specify a [SourceName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger.sourcename) for an [EventTrigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger) or a [Trigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.trigger).

 In XAML, you can't declare event handlers for [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) or animation events.

## Animate in a ControlTemplate

You can use [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) objects to define animations in a [ControlTemplate](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.controltemplate).

The [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) may only refer to child objects of the [ControlTemplate](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.controltemplate). If [TargetName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard.targetname) is not specified, the animation targets the element to which the [ControlTemplate](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.controltemplate) is applied.

 The [SourceName](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger.sourcename) for an [EventTrigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.eventtrigger) or a [Trigger](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.trigger) may only refer to child objects of the [ControlTemplate](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.controltemplate).

 In XAML, you can't declare event handlers for [Storyboard](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.storyboard) or animation events.

# Timing Behaviors Overview

## Timeline Types

https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/graphics-multimedia/timing-behaviors-overview

A [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) represents a segment of time. It provides properties that enable you to specify the length of that segment, when it should start, how many times it will repeat, how fast time progresses in that segment, and more.

Classes that inherit from the timeline class provide additional functionality, such as animation and media playback. WPF provides the following [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) types.

## Properties that Control the Length of a Timeline

### The Duration Property

### The RepeatBehavior Property

### The AutoReverse Property

<DoubleAnimation

Storyboard.TargetName="MyRectangle" Storyboard.TargetProperty="Width"

From="0" To="100" Duration="0:0:5"

RepeatBehavior="2"

AutoReverse="True" />

As a result, the [DoubleAnimation](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimation) plays for 20 seconds: forward for five seconds, backwards for five seconds, forward for 5 seconds again, and then backwards for five seconds.

## The BeginTime Property

The [BeginTime](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.begintime) property enables you to specify when a timeline starts. A timeline's begin time is relative to its parent timeline

You may also set a timeline's begin time to null, which prevents the timeline from starting. In WPF, you specify null using the [x:Null Markup Extension](https://docs.microsoft.com/zh-cn/dotnet/framework/xaml-services/x-null-markup-extension).

Note that the begin time is not applied each time a timeline repeats because of its [RepeatBehavior](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.repeatbehavior) setting.

## The FillBehavior Property

When a [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) reaches the end of its total active duration, the [FillBehavior](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.fillbehavior) property specifies whether it stops or holds its last value

# Bitmap Effects Overview

Bitmap effects enable designers and developers to apply visual effects to rendered Windows Presentation Foundation (WPF) content

Bitmap effects ([BitmapEffect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.effects.bitmapeffect) object) are simple pixel processing operations. A bitmap effect takes a [BitmapSource](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imaging.bitmapsource) as an input and produces a new [BitmapSource](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imaging.bitmapsource) after applying the effect

effects can be set as properties on live [Visual](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.visual) objects, such as a [Button](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.button) or [TextBox](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.textbox). The pixel processing is applied and rendered at run-time. In this case, at the time of rendering, a [Visual](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.visual) is automatically converted to its [BitmapSource](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imaging.bitmapsource) equivalent and is fed as input to the [BitmapEffect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.effects.bitmapeffect). The output replaces the [Visual](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.visual) object's default rendering behavior. This is why [BitmapEffect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.effects.bitmapeffect) objects force visuals to render in software only i.e. no hardware acceleration on visuals when effects are applied

## How to Apply an Effect

[BitmapEffect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.effects.bitmapeffect) is a property on [Visual](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.visual).

[BitmapEffect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.bitmapeffect) can be set to a single [BitmapEffect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.effects.bitmapeffect) object or multiple effects can be chained by using the [BitmapEffectGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.effects.bitmapeffectgroup) object.

<Button Width="200">You Can't Read This!

<Button.BitmapEffect>

<!-- <BitmapEffectGroup> would go here if you wanted to apply more

then one effect to the Button. However, in this example only

one effect is being applied so BitmapEffectGroup does not need

to be included. -->

<!-- The larger the Radius, the more blurring. The default range is 20.

In addition, the KernelType is set to a box kernel. A box kernel

creates less disruption (less blur) then the default Gaussian kernel. -->

<BlurBitmapEffect Radius="10" KernelType="Box" />

</Button.BitmapEffect>

</Button>

# WPF Brushes Overview

Everything visible on your screen is visible because it was painted by a brush.

## Painting with a Brush

A [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush) "paints" an area with its output

Most visual objects enable you to specify how they are painted. The following table lists some common objects and properties with which you can use a [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush).

| **Class** | **Brush properties** |
| --- | --- |
| [Border](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.border) | [BorderBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.border.borderbrush), [Background](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.border.background) |
| [Control](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.control) | [Background](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.control.background), [Foreground](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.control.foreground) |
| [Panel](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.panel) | [Background](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.panel.background) |
| [Pen](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pen) | [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pen.brush) |
| [Shape](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape) | [Fill](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape.fill), [Stroke](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape.stroke) |
| [TextBlock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.textblock) | [Background](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.textblock.background) |

## Paint with a Solid Color

A [SolidColorBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.solidcolorbrush) paints an area with a solid [Color](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.color). There are a variety of ways to specify the [Color](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.solidcolorbrush.color) of a [SolidColorBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.solidcolorbrush): for example, you can specify its alpha, red, blue, and green channels or use one of the predefined color provided by the [Colors](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.colors) class.

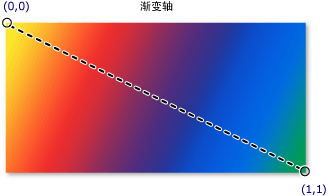
## Paint with a Linear Gradient

A [LinearGradientBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush) paints an area with a linear gradient. You use [GradientStop](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.gradientstop) objects to specify the colors in the gradient and their positions.

### The Gradient Axis

By manipulating the brush's [StartPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.startpoint) and [EndPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.endpoint), you can create horizontal and vertical gradients, reverse the gradient direction, condense the gradient spread, and more.

The default [StartPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.startpoint) of a [LinearGradientBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush) is (0,0), and its default [EndPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.endpoint) is (1,1), which creates a diagonal gradient starting at the upper-left corner and extending to the lower-right corner of the area being painted.



The following example shows how to create a horizontal gradient by specifying the brush's [StartPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.startpoint) and [EndPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.endpoint).

<Rectangle Width="200" Height="100">

<Rectangle.Fill>

<LinearGradientBrush StartPoint="0,0.5" EndPoint="1,0.5">

<GradientStop Color="Yellow" Offset="0.0" />

<GradientStop Color="Red" Offset="0.25" />

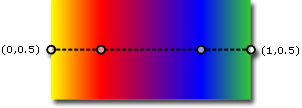
<GradientStop Color="Blue" Offset="0.75" />

<GradientStop Color="LimeGreen" Offset="1.0" />

</LinearGradientBrush>

</Rectangle.Fill>

</Rectangle>



## Paint with a Radial Gradient

<https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/graphics-multimedia/painting-with-solid-colors-and-gradients-overview#radial-gradients>

## Paint with an Image

An [ImageBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagebrush) paints an area with a [ImageSource](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagesource).

ImageBrush myBrush = new ImageBrush();

myBrush.ImageSource =

new BitmapImage(new Uri(@"sampleImages\pinkcherries.jpg", UriKind.Relative));

exampleRectangle.Fill = myBrush;

<Rectangle Width="75" Height="75">

<Rectangle.Fill>

<ImageBrush ImageSource="sampleImages\pinkcherries.jpg" />

</Rectangle.Fill>

</Rectangle>

## Paint with a Drawing

A [DrawingBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawingbrush) paints an area with a [Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawing). A [Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawing) can contain shapes, images, text, and media. Shapes inside a drawing brush may themselves be painted with a solid color, gradient, image, or even another [DrawingBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawingbrush).

A [Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawing) object describes visible content, such as a shape, bitmap, video, or a line of text. Different types of drawings describe different types of content. The following is a list of the different types of drawing objects.

* [GeometryDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing) – Draws a shape.
* [ImageDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagedrawing) – Draws an image.
* [GlyphRunDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.glyphrundrawing) – Draws text.
* [VideoDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.videodrawing) – Plays an audio or video file.
* [DrawingGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawinggroup) – Draws other drawings. Use a drawing group to combine other drawings into a single composite drawing.

## Paint with a Visual

A [VisualBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.visualbrush) paints an area with a [Visual](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.visual) object.

## Paint using Predefined and System Brushes

For convenience, Windows Presentation Foundation (WPF) provides a set of predefined and system brushes that you can use to paint objects.

For a list of available predefined brushes, see the [Brushes](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brushes) class

For a list of available system brushes, see the [SystemColors](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.systemcolors) class.

A system brush is a [SolidColorBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.solidcolorbrush) object that paints an area with the specified system color.

You can use system brushes as either a static or a dynamic resource.

## Common Brush Features

[Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush) objects provide an [Opacity](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.opacity) property

if a brush has an opacity value of 0.5 and a color used in the brush also has an opacity value of 0.5, the output color has an opacity value of 0.25.

It's more efficient to change the opacity value of a brush than it is to change the opacity of an entire element using its [UIElement.Opacity](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.opacity) property.

You can rotate, scale, skew, and translate a brush's content by using its [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.transform) or [RelativeTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.relativetransform) properties.

# Transforms Overview

## What Is a Transform?

A [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) defines how to map, or transform, points from one coordinate space to another coordinate space. This mapping is described by a transformation [Matrix](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.matrix), which is a collection of three rows with three columns of [Double](https://docs.microsoft.com/zh-cn/dotnet/api/system.double) values.

By manipulating matrix values, you can rotate, scale, skew, and move (translate) an object.

Although Windows Presentation Foundation (WPF) enables you to directly manipulate matrix values, it also provides several [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) classes that enable you to transform an object without knowing how the underlying matrix structure is configured.

## Transform Classes

Windows Presentation Foundation (WPF) provides the following 2-D [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) classes for common transformation operations

[RotateTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.rotatetransform)

[ScaleTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.scaletransform)

[SkewTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.skewtransform)

[TranslateTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.translatetransform)

[AngleX](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.skewtransform.anglex) skews x-axis values relative to the original coordinate system

For creating more complex transformations, Windows Presentation Foundation (WPF) provides the following two classes:

[TransformGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transformgroup)

Groups multiple [TransformGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transformgroup) objects into a single [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) that you can then apply to transform properties.

[MatrixTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.matrixtransform)

Creates custom transformations that are not provided by the other [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) classes. When you use a [MatrixTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.matrixtransform), you manipulate a Matrix directly.

## Common Transformation Properties

One way to transform an object is to declare the appropriate [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) type and apply it to the transformation property of the object. Different types of objects have different types of transformation properties

| **Type** | **Transformation properties** |
| --- | --- |
| [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush) | [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.transform), [RelativeTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.relativetransform) |
| [ContainerVisual](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.containervisual) | [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.containervisual.transform) |
| [DrawingGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawinggroup) | [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawinggroup.transform) |
| [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) | [RenderTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.rendertransform), [LayoutTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement.layouttransform) |
| [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) | [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry.transform) |
| [TextEffect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.texteffect) | [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.texteffect.transform) |
| [UIElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement) | [RenderTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.rendertransform) |

## Transformations and Coordinate Systems

When you transform an object, you do not just transform the object, you transform coordinate space in which that object exists. By default, a transform is centered at the origin of the target object's coordinate system: (0,0).

## Transforming a FrameworkElement

To apply transformations to a [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement), create a [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) and apply it to one of the two properties that the [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement) class provides:

* [LayoutTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement.layouttransform) – A transform that is applied before the layout pass. After the transform is applied, the layout system processes the transformed size and position of the element.
* [RenderTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.rendertransform) – A transform that modifies the appearance of the element but is applied after the layout pass is complete. By using the [RenderTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.rendertransform) property instead of the [LayoutTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement.layouttransform) property, you can obtain performance benefits.

The value of the [RenderTransformOrigin](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.rendertransformorigin) property is relative to the size of the button. As a result, the rotation is applied to the center of the button

<Button Content="Rotated Button"

RenderTransformOrigin="0.5,0.5">

<Button.RenderTransform>

<RotateTransform Angle="45" />

</Button.RenderTransform>

</Button>

## Animating Transformations

Because they inherit from the [Animatable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animatable) class, the [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) classes can be animated.

## Freezable Features

Because it inherits from the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) class, [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform) objects can be declared as [resources](https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/advanced/xaml-resources), shared among multiple objects, made read-only to improve performance,

# Brush Transformation Overview

The Brush class provides two transformation properties: [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.transform) and [RelativeTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.relativetransform). The properties enable you to rotate, scale, skew, and translate a brush's contents

## Differences between the Transform and RelativeTransform Properties

When you apply a transform to a brush's [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.transform) property, you need to know the size of the painted area if you want to transform the brush contents about its center

使用brush的[Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.transform) 属性时，先将brush的content输出到paint area，然后对paint area做变换

When you apply a transform to a brush's [RelativeTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.relativetransform) property, that transform is applied to the brush before its output is mapped to the painted area

The following list describes the order in which a brush’s contents are processed and transformed.

1. Process the brush’s contents. For a [GradientBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.gradientbrush), this means determining the gradient area. For a [TileBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.tilebrush), the [Viewbox](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.tilebrush.viewbox) is mapped to the [Viewport](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.tilebrush.viewport). This becomes the brush’s output.
2. Project the brush’s output onto the 1 x 1 transformation rectangle.
3. Apply the brush’s [RelativeTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.relativetransform), if it has one.
4. Project the transformed output onto the area to paint.
5. Apply the brush’s [Transform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.transform), if it has one.

Because the [RelativeTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush.relativetransform) is applied while the brush’s output is mapped to a 1 x 1 rectangle，if you used a [RotateTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.rotatetransform) to rotate the brush's output 45 degrees about its center, you'd give the [RotateTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.rotatetransform) a [CenterX](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.rotatetransform.centerx) of 0.5 and a [CenterY](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.rotatetransform.centery) of 0.5.

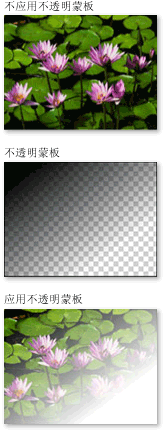
未完

# Opacity Masks Overview

To create an opacity mask, you apply a [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush) to the [OpacityMask](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.opacitymask) property of an element or [Visual](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.visual).

## Creating Visual Effects with Opacity Masks

The alpha channel of each of the brush's pixels are then used to determine the resulting opacity of the element or visual's corresponding pixels; the actual color of the brush is ignored.



## Creating an Opacity Mask

To create an opacity mask, you create a [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush) and apply it to the [OpacityMask](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.opacitymask) property of an element or visual。You can use any type of [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush) as an opacity mask.

Because all of the colors in the [Colors](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.colors) class, except [Transparent](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.colors.transparent), are fully opaque, they can be used to simply define a starting color for a gradient opacity mask.

For additional control over alpha values when defining an opacity mask, you can specify the alpha channel of colors using ARGB hexadecimal notation in markup or using the [Color.FromScRgb](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.color.fromscrgb) method.

## Specifying Gradient Stops for an Opacity Mask

### Specifying Color Opacity in "XAML"

In Extensible Application Markup Language (XAML), you use ARGB hexadecimal notation to specify the opacity of individual colors. ARGB hexadecimal notation uses the following syntax:

# **aa** *rrggbb*

The aa in the previous line represents a two-digit hexadecimal value used to specify the opacity of the color.

## Using an Image as an Opacity Mask

To use an image as an opacity mask, use an [ImageBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagebrush) to contain the image

When creating an image to be used as an opacity mask, save the image in a format that supports multiple levels of transparency, such as Portable Network Graphics (PNG)

<Image

Height="150"

Width="200"

Source="sampleImages/Waterlilies.jpg"

HorizontalAlignment="Left"

Margin="10"

Grid.Column="2" Grid.Row="1">

<Image.OpacityMask>

<ImageBrush ImageSource="sampleImages/tornedges.png"/>

</Image.OpacityMask>

</Image>

## Painting an Area with a Solid Color

### Using a SolidColorBrush in "XAML"

<!-- This button's background is painted with a red SolidColorBrush,

described using a named color. -->

<Button Background="Red">A Button</Button>

<!-- This button's background is painted with a red SolidColorBrush,

described using hexadecimal notation. -->

<Button Background="#FFFF0000">A Button</Button>

<!-- Both of these buttons' backgrounds are painted with red

SolidColorBrush objects, described using object element

syntax. -->

<Button>A Button

<Button.Background>

<SolidColorBrush Color="Red" />

</Button.Background>

</Button>

### Painting with a SolidColorBrush in Code

myButton.Background = Brushes.Red;

SolidColorBrush mySolidColorBrush = new SolidColorBrush();

mySolidColorBrush.Color = Colors.Red;

myButton.Background = mySolidColorBrush;

The static [FromArgb](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.color.fromargb) enables you to specify the color's alpha, red, green, and blue values.

SolidColorBrush mySolidColorBrush = new SolidColorBrush();

mySolidColorBrush.Color =

Color.FromArgb(

255, // Specifies the transparency of the color.

255, // Specifies the amount of red.

0, // specifies the amount of green.

0); // Specifies the amount of blue.

myButton.Background = mySolidColorBrush;

## Painting an Area with a Gradient

A gradient brush paints an area with multiple colors that blend into each other along an axis.

The gradient examples in this topic use the default coordinate system for setting start points and end points. The default coordinate system is relative to a bounding box: 0 indicates 0 percent of the bounding box and 1 indicates 100 percent of the bounding box. You can change this coordinate system by setting the [MappingMode](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.gradientbrush.mappingmode) property to the value [Absolute](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brushmappingmode#System_Windows_Media_BrushMappingMode_Absolute).

You may change the orientation and size of the line using the brush's [StartPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.startpoint) and [EndPoint](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.lineargradientbrush.endpoint) properties.

## Radial Gradients

A radial gradient brush's axis is defined by a circle; its colors "radiate" outward from its origin.

The [GradientOrigin](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.radialgradientbrush.gradientorigin) specifies the start point of a radial gradient brush's gradient axis. The gradient axis radiates from the gradient origin to the gradient circle. A brush's gradient circle is defined by its [Center](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.radialgradientbrush.center), [RadiusX](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.radialgradientbrush.radiusx), and [RadiusY](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.radialgradientbrush.radiusy) properties.

[RadiusX](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.radialgradientbrush.radiusx), and [RadiusY](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.radialgradientbrush.radiusy)也是相对值，相对paint area 的长和宽

<RadialGradientBrush

GradientOrigin="0.5,0.5" Center="0.5,0.5"

RadiusX="0.5" RadiusY="0.5">

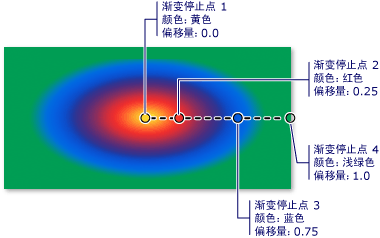
<GradientStop Color="Yellow" Offset="0" />

<GradientStop Color="Red" Offset="0.25" />

<GradientStop Color="Blue" Offset="0.75" />

<GradientStop Color="LimeGreen" Offset="1" />

</RadialGradientBrush>



# Drawing Objects Overview

A [Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawing) object describes visible content, such as a shape, bitmap, video, or a line of text. Different types of drawings describe different types of content. The following is a list of the different types of drawing objects.

* [GeometryDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing) – Draws a shape.
* [ImageDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagedrawing) – Draws an image.
* [GlyphRunDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.glyphrundrawing) – Draws text.
* [VideoDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.videodrawing) – Plays an audio or video file.
* [DrawingGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawinggroup) – Draws other drawings. Use a drawing group to combine other drawings into a single composite drawing.

WPF provides other types of objects that are capable of drawing shapes, bitmaps, text, and media.

when should you use [Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawing) objects? When you can sacrifice framework level features to gain performance benefits or when you need [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) features.

[Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawing) objects lack support for [Layout](https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/advanced/layout), input, and focus, they provide performance benefits

## Draw a Shape

To draw a shape, you use a [GeometryDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing). A geometry drawing's [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing.geometry) property describes the shape to draw, its [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing.brush) property describes how the interior of the shape should be painted, and its [Pen](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing.pen) property describes how its outline should be drawn.

## Draw an Image

To draw an image, you use an [ImageDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagedrawing). An [ImageDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagedrawing) object's [ImageSource](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagedrawing.imagesource) property describes the image to draw, and its [Rect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagedrawing.rect) property defines the region where the image is drawn.

A [DrawingGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawinggroup) also enables you to apply opacity masks, transforms, bitmap effects, and other operations to its contents

## Display a Drawing as an Image

## Play Media (Code Only)

## Draw Text

To draw text, you use a [GlyphRunDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.glyphrundrawing) and a [GlyphRun](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.glyphrun).

GlyphRun theGlyphRun = new GlyphRun(

new GlyphTypeface(new Uri(@"C:\WINDOWS\Fonts\TIMES.TTF")),

0,

false,

13.333333333333334,

new ushort[]{43, 72, 79, 79, 82, 3, 58, 82, 85, 79, 71},

new Point(0, 12.29),

new double[]{

9.62666666666667, 7.41333333333333, 2.96,

2.96, 7.41333333333333, 3.70666666666667,

12.5866666666667, 7.41333333333333,

4.44, 2.96, 7.41333333333333},

null,

null,

null,

null,

null,

null

);

GlyphRunDrawing gDrawing = new GlyphRunDrawing(Brushes.Black, theGlyphRun);

A [GlyphRun](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.glyphrun) is a low-level object intended for use with fixed-format document presentation and print scenarios. A simpler way to draw text to the screen is to use a [Label](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.label) or a [TextBlock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.textblock).

## Composite Drawings

To display a [Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawing) with an [Image](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.image) control, use a [DrawingImage](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawingimage) as the [Image](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.image) control's [Source](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.image.source) and set the [DrawingImage](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawingimage) object's [DrawingImage.Drawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawingimage.drawing) property to the drawing you want to display.

//

// Use a DrawingImage and an Image control

// to display the drawing.

//

DrawingImage geometryImage = new DrawingImage(aGeometryDrawing);

// Freeze the DrawingImage for performance benefits.

geometryImage.Freeze();

Image anImage = new Image();

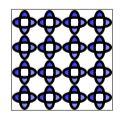
anImage.Source = geometryImage;

anImage.HorizontalAlignment = HorizontalAlignment.Left;

## Paint an Object with a Drawing

A [DrawingBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawingbrush) is a type of brush that paints an area with a drawing object. You can use it to paint just about any graphical object with a drawing

The following examples uses a [DrawingBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawingbrush) to paint the [Fill](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape.fill) of a [Rectangle](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.rectangle) with a pattern created from a [GeometryDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing). This example produces the following output.



DrawingBrush patternBrush = new DrawingBrush(aGeometryDrawing);

patternBrush.Viewport = new Rect(0, 0, 0.25, 0.25);

patternBrush.TileMode = TileMode.Tile;

patternBrush.Freeze();

//

// Create an object to paint.

//

Rectangle paintedRectangle = new Rectangle();

paintedRectangle.Width = 100;

paintedRectangle.Height = 100;

paintedRectangle.Fill = patternBrush;

# Geometry Overview

## What Is a Geometry?

The [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) class and the classes which derive from it enable you to describe the geometry of a 2-D shape. These geometric descriptions have many uses, such defining a shape to paint to the screen or defining hit-test and clip regions. You can even use a geometry to define an animation path.

## Geometries vs. Shapes

For one, the [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) class inherits from the [Freezable](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.freezable) class while the [Shape](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape) class inherits from [FrameworkElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.frameworkelement). Because they are elements, [Shape](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape) objects can render themselves and participate in the layout system, while [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) objects cannot.

Although [Shape](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape) objects are more readily usable than [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) objects, [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) objects are more versatile. While a [Shape](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape) object is used to render 2-D graphics,

### The Path Shape

One [Shape](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape), the [Path](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.path) class, actually uses a [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) to describe its contents. By setting the [Data](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.path.data) property of the [Path](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.path) with a [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) and setting its [Fill](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape.fill) and [Stroke](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.shape.stroke) properties, you can render a [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry).

## Common Properties That Take a Geometry

The following table lists several classes that have properties that take a [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) object.

| **Type** | **Property** |
| --- | --- |
| [DoubleAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimationusingpath) | [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimationusingpath.pathgeometry) |
| [DrawingGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawinggroup) | [ClipGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.drawinggroup.clipgeometry) |
| [GeometryDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing) | [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrydrawing.geometry) |
| [Path](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.path) | [Data](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.shapes.path.data) |
| [UIElement](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement) | [Clip](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.uielement.clip) |

The classes which derive from the [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) class can be roughly grouped into three categories: simple geometries, path geometries, and composite geometries

## Simple Geometry Types

LineGeometry

EllipseGeometry

RectangleGeometry

## Path Geometries

The [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathgeometry) class and its lightweight equivalent, the [StreamGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.streamgeometry) class, provide the means to describe multiple complex figures composed of arcs, curves, and lines

At the heart of a [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathgeometry) is a collection of [PathFigure](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathfigure) objects, so named because each figure describes a discrete shape in the [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathgeometry). Each [PathFigure](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathfigure) is itself comprised of one or more [PathSegment](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathsegment) objects, each of which describes a segment of the figure.

The segments within a [PathFigure](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathfigure) are combined into a single geometric shape with the end point of each segment being the start point of the next segment

Unlike a [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathgeometry), the contents of a [StreamGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.streamgeometry) do not support data binding, animation, or modification.

## Composite Geometries

Composite geometry objects can be created using a [GeometryGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrygroup), a [CombinedGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.combinedgeometry), or by calling the static [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) method [Combine](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry.combine).

The [CombinedGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.combinedgeometry) object and the [Combine](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry.combine) method performs a Boolean operation to combine the area defined by two geometries

The [GeometryGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrygroup) class creates an amalgamation of the [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) objects it contains without combining their area. Any number of [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry) objects can be added to a [GeometryGroup](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrygroup).

## Combined Geometries

the [CombinedGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.combinedgeometry) object and the [Combine](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry.combine) method combine the area defined by the geometries they contain. The [GeometryCombineMode](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrycombinemode) enumeration specifies how the geometries are combined

[Union](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrycombinemode#System_Windows_Media_GeometryCombineMode_Union), [Intersect](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrycombinemode#System_Windows_Media_GeometryCombineMode_Intersect), [Exclude](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrycombinemode#System_Windows_Media_GeometryCombineMode_Exclude), and [Xor](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometrycombinemode#System_Windows_Media_GeometryCombineMode_Xor).

<Path.Data>

<!-- Combines two geometries using the union combine mode. -->

<CombinedGeometry GeometryCombineMode="Union">

<CombinedGeometry.Geometry1>

<EllipseGeometry RadiusX="50" RadiusY="50" Center="75,75" />

</CombinedGeometry.Geometry1>

<CombinedGeometry.Geometry2>

<EllipseGeometry RadiusX="50" RadiusY="50" Center="125,75" />

</CombinedGeometry.Geometry2>

</CombinedGeometry>

</Path.Data>

# Path Markup Syntax

## StreamGeometry and PathFigureCollection Mini-Languages

WPF provides two classes that provide mini-languages for describing geometric paths: [StreamGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.streamgeometry) and [PathFigureCollection](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathfigurecollection).

You use the [StreamGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.streamgeometry) mini-language when setting a property of type [Geometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.geometry)

<Path Stroke="Black" Fill="Gray"

Data="M 10,100 C 10,300 300,-200 300,100" />

You use the [PathFigureCollection](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathfigurecollection) mini-language when setting the [Figures](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathgeometry.figures) property of a [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathgeometry).

<Path Stroke="Black" Fill="Gray">

<Path.Data>

<PathGeometry Figures="M 10,100 C 10,300 300,-200 300,100" />

</Path.Data>

</Path>

## Move Command

Specifies the start point of a new figure.

M startPoint  
  
- or -  
  
m startPoint

## Draw Commands

A draw command can consist of several shape commands.

# Path Animations Overview

A path animation is a type of [AnimationTimeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.animationtimeline) that uses a [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.pathgeometry) as its input. As the path animation progresses, it reads the x, y, and angle information from the path and uses that information to generate its output.

One way to move an object along a path is to use a [MatrixTransform](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.matrixtransform) and a [MatrixAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.matrixanimationusingpath) to transform an object along a complex path.

## Path Animation Types

WPF provides the following path animation classes.

[DoubleAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimationusingpath)

[MatrixAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.matrixanimationusingpath)

[PointAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.pointanimationusingpath)

A [PointAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.pointanimationusingpath) generates [Point](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.point) values from the x- and y-coordinates of its [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.pointanimationusingpath.pathgeometry)

A [DoubleAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimationusingpath) generates [Double](https://docs.microsoft.com/zh-cn/dotnet/api/system.double) values from its [PathGeometry](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimationusingpath.pathgeometry). By setting the [Source](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimationusingpath.source) property, you can specify whether the [DoubleAnimationUsingPath](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.doubleanimationusingpath) uses the x-coordinate, y-coordinate, or angle of the path as its output.

# Imaging Overview

WPF Imaging enables developers to display, transform, and format images.

There are two ways to access the WPF Imaging API, a managed component and an unmanaged component. The unmanaged component provides the following features.

## Displaying Images in WPF

Images can be displayed using an [Image](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.image) control, painted on a visual using an [ImageBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagebrush), or drawn using an [ImageDrawing](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagedrawing).

### Using the Image Control

<Image Width="200">

<Image.Source>

<!-- To save significant application memory, set the DecodePixelWidth or

DecodePixelHeight of the BitmapImage value of the image source to the desired

height and width of the rendered image. If you don't do this, the application will

cache the image as though it were rendered as its normal size rather then just

the size that is displayed. -->

<!-- Note: In order to preserve aspect ratio, only set either DecodePixelWidth

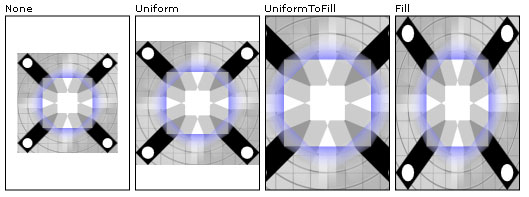
or DecodePixelHeight but not both. -->

<BitmapImage DecodePixelWidth="200"

UriSource="C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Water Lilies.jpg" />

</Image.Source>

The [Stretch](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.controls.image.stretch) property controls how an image is stretched to fill its container.



### Painting with Images

Images can also be displayed in an application by painting with a [Brush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.brush). To paint with images, use an [ImageBrush](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.imagebrush).

<Button.Background>

<ImageBrush ImageSource="sampleImages\blueberries.jpg" />

</Button.Background>