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).

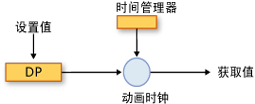
If the [Timeline](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline) contains child timelines, it creates [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) objects for them as well. The resulting [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) 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.



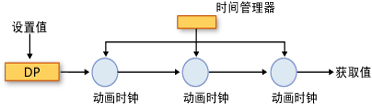
When the time manager ticks, it updates the time of every [Active](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clockstate#System_Windows_Media_Animation_ClockState_Active) [Clock](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.clock) in the application. 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.

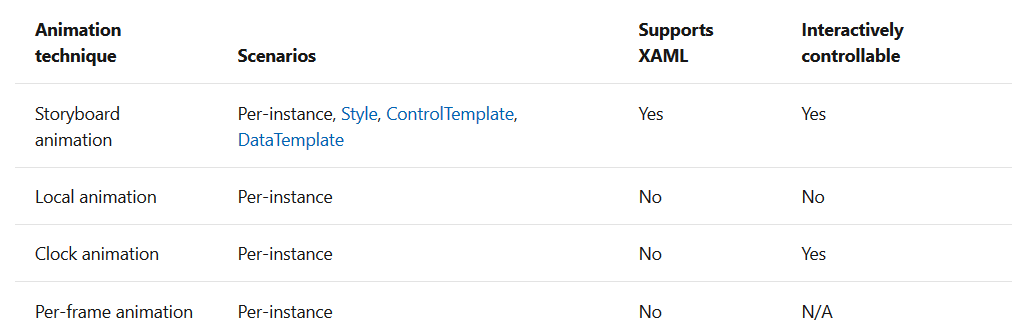
If the animation's [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.duration) is [Automatic](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.duration.automatic) or its [Duration](https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.media.animation.timeline.duration) 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