

[AppKit](#) / [NSWritingToolsCoordinator](#) / `NSWritingToolsCoordinator.AnimationParameters`

Class

NSWritingToolsCoordinator.AnimationParameters

An object you use to configure additional tasks or animations to run alongside the Writing Tools animations.

macOS 15.2+

```
class AnimationParameters
```

Overview

When Writing Tools replaces text in one of your context objects, it provides an `NSWritingToolsCoordinator.AnimationParameters` object for you to use to configure any additional animations. During a Writing Tools session, you hide the text under evaluation and provide a targeted preview of your content. Writing Tools animations changes to that preview, but you might need to provide additional animations for other parts of your view's content. For example, you might need to animate any layout changes caused by the insertion or removal of text in other parts of your view. Use this object to configure those animations.

You don't create an `NSWritingToolsCoordinator.AnimationParameters` object directly. Instead, the system creates one and passes it to the `NSWritingToolsCoordinator/writingToolsCoordinator(_:replaceRange:inContext:proposedText:reason:animationParameters:completion:)` method of your [`NSWritingToolsCoordinator.Delegate`](#) object. Use that object to specify the blocks to run during and after the system animations.

Topics

Getting the animation values

`var duration: CGFloat`

The number of seconds it takes the system animations to run.

`var delay: CGFloat`

The number of seconds the system waits before starting its animations.

Creating custom animations

`var progressHandler: ((Float) -> Void)?`

A custom block that runs at the same time as the system animations.

`var completionHandler: (() -> Void)?`

A custom block to run when the system animations finish.

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`

`CustomDebugStringConvertible`

`CustomStringConvertible`

`Equatable`

`Hashable`

`NSObjectProtocol`

`Sendable`

`SendableMetatype`

See Also

Writing Tools for custom views

Supporting Writing Tools via the pasteboard

Adopt a simplified version of the Writing Tools experience in a custom view using the pasteboard and macOS services.

Adding Writing Tools support to a custom AppKit view

Integrate Writing Tools support, including support for inline replacement animations, to your custom text views on macOS.

`class NSWritingToolsCoordinator`

An object that manages interactions between Writing Tools and your custom text view.

`protocol Delegate`

An interface that you use to manage interactions between Writing Tools and your custom text view.

`class Context`

A data object that you use to share your custom view's text with Writing Tools.

Enhancing your custom text engine with Writing Tools

Add Writing Tools support to your custom text engine to enhance the text editing experience.