

[AppKit](#) / [NSWritingToolsCoordinator](#) / NSWritingToolsCoordinator.Delegate

Protocol

NSWritingToolsCoordinator.Delegate

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

macOS 15.2+

```
protocol Delegate : NSObjectProtocol
```

Mentioned in

 [Adding Writing Tools support to a custom AppKit view](#)

Overview

Adopt the `NSWritingToolsCoordinator.Delegate` protocol in the type you use to manage your custom text view. When you add a `NSWritingToolsCoordinator` object to your view, the coordinator uses this protocol to communicate with that view. The protocol lets Writing Tools fetch your view's text, report suggested changes back to your view, and deliver visual feedback when Writing Tools features are active. Make sure the type that adopts this protocol has access to your view's text storage and can perform relevant tasks on behalf of the view.

Writing Tools expects you to call the provided handler blocks at the end of your delegate methods. It's crucial that you execute these blocks in a timely manner to allow Writing Tools to perform subsequent tasks. For example, Writing Tools waits for you to execute the handlers for animation-related methods before moving on to the next stage of the animations.

Topics

Starting a Writing Tools operation

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requests
ContextsFor: NSWritingToolsCoordinator.ContextScope, completion: ([
NSWritingToolsCoordinator.Context]) -> Void)
```

Asks your delegate to provide the text to evaluate during the Writing Tools operation.

Required

Incorporating Writing Tools suggestions

```
func writingToolsCoordinator(NSWritingToolsCoordinator, replace:
NSRange, in: NSWritingToolsCoordinator.Context, proposedText:
NSAttributedString, reason: NSWritingToolsCoordinator.TextReplacement
Reason, animationParameters: NSWritingToolsCoordinator.Animation
Parameters?, completion: (NSAttributedString?) -> Void)
```

Tells the delegate that there are text changes to incorporate into the view.

Required

```
func writingToolsCoordinator(NSWritingToolsCoordinator, select: [
NSValue], in: NSWritingToolsCoordinator.Context, completion: () -> Void
)
```

Asks the delegate to update your view's current text selection.

Required

Responding to lifecycle changes

```
func writingToolsCoordinator(NSWritingToolsCoordinator, willChangeTo:
NSWritingToolsCoordinator.State, completion: () -> Void)
```

Notifies your delegate of relevant state changes when Writing Tools is running in your view.

Animating inline text changes

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requestsPreview
For: NSWritingToolsCoordinator.TextAnimation, of: NSRange, in:
NSWritingToolsCoordinator.Context, completion: ([NSTextPreview]?) ->
Void)
```

Asks the delegate for a preview image and layout information for the specified text.

Required

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requestsPreviewFor: NSRect, in: NSWritingToolsCoordinator.Context, completion: (NSTextPreview?) -> Void)
```

Asks the delegate for a preview image and layout information for the specified text.

Required

```
func writingToolsCoordinator(NSWritingToolsCoordinator, prepareFor: NSWritingToolsCoordinator.TextAnimation, for: NSRange, in: NSWritingToolsCoordinator.Context, completion: () -> Void)
```

Prepare for animations for the content that Writing Tools is evaluating.

Required

```
func writingToolsCoordinator(NSWritingToolsCoordinator, finish: NSWritingToolsCoordinator.TextAnimation, for: NSRange, in: NSWritingToolsCoordinator.Context, completion: () -> Void)
```

Asks the delegate to clean up any state related to the specified Writing Tools animation.

Required

Displaying proofreading marks

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requestsRangeInContextWithIdentifierFor: NSPoint, completion: (NSRange, UUID) -> Void)
```

Asks the delegate to provide the location of the character at the specified point in your view's coordinate system.

Deprecated

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requestsBoundingBezierPathsFor: NSRange, in: NSWritingToolsCoordinator.Context, completion: ([NSBezierPath]) -> Void)
```

Asks the delegate to provide the bounding paths for the specified text in your view.

Required

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requestsUnderlinePathsFor: NSRange, in: NSWritingToolsCoordinator.Context, completion: ([NSBezierPath]) -> Void)
```

Asks the delegate to provide an underline shape for the specified text during a proofreading session.

Required

Providing animation container views dynamically

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requestsSingle
ContainerSubrangesOf: NSRange, in: NSWritingToolsCoordinator.Context,
completion: ([NSValue]) -> Void)
```

Asks the delegate to divide the specified range of text into the separate containers that render that text.

```
func writingToolsCoordinator(NSWritingToolsCoordinator, requests
DecorationContainerViewFor: NSRange, in: NSWritingToolsCoordinator.
Context, completion: (NSView) -> Void)
```

Asks the delegate to provide a decoration view for the specified range of text.

Relationships

Inherits From

NSObjectProtocol

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.

```
class Context
```

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

```
class AnimationParameters
```

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

Enhancing your custom text engine with Writing Tools

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