

[SwiftUI](#) / [AttributedTextSelection](#)

Structure

AttributedTextSelection

Represents a selection of attributed text.

iOS 26.0+ | iPadOS 26.0+ | Mac Catalyst 26.0+ | macOS 26.0+ | tvOS 26.0+ | visionOS 26.0+ | watchOS 26.0+

```
struct AttributedTextSelection
```

Overview

A selection is either an insertion point (e.g. a cursor in the text), or spans over a range of characters. While that range is always visually contiguous, it may not be logically contiguous in the text storage. Specifically, a single selection value cannot represent multiple cursors.

This is frequently used to represent selection of text in a `TextEditor`. The following example shows a text editor that leverages text selection to offer live suggestions based on the current selection.

```
struct SuggestionTextEditor: View {
    @State var text: AttributedString = ""
    @State var selection = AttributedTextSelection()

    var body: some View {
        VStack {
            TextEditor(text: $text, selection: $selection)
            // A helper view that offers live suggestions based on selection.
            SuggestionsView(substrings: getSubstrings(
                text: text, indices: selection.indices(in: text))
        )
    }
}
```

```
private func getSubstrings(
    text: String, indices: AttributedTextSelection.Indices
) -> [Substring] {
    // Resolve substrings representing the current selection...
}

struct SuggestionsView: View { ... }
```

You can also use the `textSelectionAffinity(_:)` modifier to specify a selection affinity on the given hierarchy:

```
struct SuggestionTextEditor: View {
    @State var text: AttributedString = ""
    @State var selection = AttributedTextSelection()

    var body: some View {
        VStack {
            TextEditor(text: $text, selection: $selection)
            // A helper view that offers live suggestions based on selection.
            SuggestionsView(substrings: getSubstrings(
                text: text, indices: selection.indices(in: text))
        )
        .textSelectionAffinity(.upstream)
    }

    private func getSubstrings(
        text: String, indices: AttributedTextSelection.Indices
    ) -> [Substring] {
        // Resolve substrings representing the current selection...
    }
}

struct SuggestionsView: View { ... }
```

See Also

[TextSelectionAffinity](#), [TextEditor](#)

Topics

Structures

```
struct Attributes
```

A sequence of all attribute values a selection has in a certain text.

Initializers

```
init()
```

Initialize the default selection for a new text editor.

```
init(insertionPoint: AttributedString.Index, typingAttributes: AttributeContainer?)
```

Initialize a selection to a single insertion point.

```
init(range: Range<AttributedString.Index>)
```

Initialize a selection to a character range.

```
init(ranges: RangeSet<AttributedString.Index>)
```

Initialize a selection to character ranges.

Instance Methods

```
func affinity(in: AttributedString) -> TextSelectionAffinity
```

Return the selection affinity of the selection.

```
func attributes(in: AttributedString) -> AttributedTextSelection.Attributes<AttributedString>
```

Obtain a lazy sequence of all attribute values the selection has in a given text.

```
func indices(in: AttributedString) -> AttributedTextSelection.Indices
```

The current text selection indices.

```
func typingAttributes(in: AttributedString) -> AttributeContainer
```

Returns the typing attributes for a corresponding text.

Enumerations

```
enum Indices
```

The indices of the current selection.

Relationships

Conforms To

Equatable, Sendable, SendableMetatype

See Also

Selecting text

```
func textSelection<S>(S) -> some View
```

Controls whether people can select text within this view.

```
protocol TextSelectability
```

A type that describes the ability to select text.

```
struct TextSelection
```

Represents a selection of text.

```
func textSelectionAffinity(TextSelectionAffinity) -> some View
```

Sets the direction of a selection or cursor relative to a text character.

```
var textSelectionAffinity: TextSelectionAffinity
```

A representation of the direction or association of a selection or cursor relative to a text character. This concept becomes much more prominent when dealing with bidirectional text (text that contains both LTR and RTL scripts, like English and Arabic combined).

```
enum TextSelectionAffinity
```

A representation of the direction or association of a selection or cursor relative to a text character. This concept becomes much more prominent when dealing with bidirectional text (text that contains both LTR and RTL scripts, like English and Arabic combined).