

[SwiftUI](#) / [TextSelectionAffinity](#)

Enumeration

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).

iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 18.0+ | macOS 15.0+ | tvOS 18.0+ | visionOS 2.0+ | watchOS 11.0+

```
enum TextSelectionAffinity
```

Overview

This type also determines whether, for example, the insertion point appears after the last character on a line or before the first character on the following line in cases where text wraps across line boundaries.

Given the scenario hello | مرحبا, where | represents the cursor & مرحبا represents "hello" in Arabic, the ambiguity arises because:

- If the cursor is associated with the end of the English word, it would be as if you're continuing to type in English (LTR).
- If the cursor is associated with the beginning of the Arabic word, it would also be as if you're continuing to type in Arabic (RTL).

`TextSelectionAffinity` helps resolve this ambiguity by determining the direction or association of the cursor relative to the surrounding text.

You can configure the selection affinity on a given hierarchy by using the `textSelectionAffinity\(_:\)` modifier:

```

struct SuggestionTextEditor: View {
    @State var text: String = ""
    @State var selection: TextSelection? = nil

    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))
        }
        .textSelectionAffinity(.upstream)
    }

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

struct SuggestionsView: View { ... }

```

Topics

Enumeration Cases

case automatic

A selection affinity determined by the framework based on the current context.

case downstream

An downstream selection affinity. In this case, the cursor is associated with the character immediately after it.

case upstream

An upstream selection affinity. In this case, the cursor is associated with the character immediately before it.

Relationships

Conforms To

Equatable
Hashable
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).

```
struct AttributedTextSelection
```

Represents a selection of attributed text.