

[SwiftUI](#) / [View](#) / `textSelection(_:)`

Instance Method

textSelection(_:)

Controls whether people can select text within this view.

iOS 15.0+ | iPadOS 15.0+ | Mac Catalyst 15.0+ | macOS 12.0+ | visionOS 1.0+

```
nonisolated
func textSelection<S>(_ selectability: S) -> some View where S : Text
Selectability
```

Discussion

People sometimes need to copy useful information from [Text](#) views — including error messages, serial numbers, or IP addresses — so they can then paste the text into another context. Enable text selection to let people select text in a platform-appropriate way.

You can apply this method to an individual text view, or to a container to make each contained text view selectable. In the following example, the person using the app can select text that shows the date of an event or the name or email of any of the event participants:

```
var body: some View {
    VStack {
        Text("Event Invite")
            .font(.title)
        Text(invite.date.formatted(date: .long, time: .shortened))
            .textSelection(.enabled)

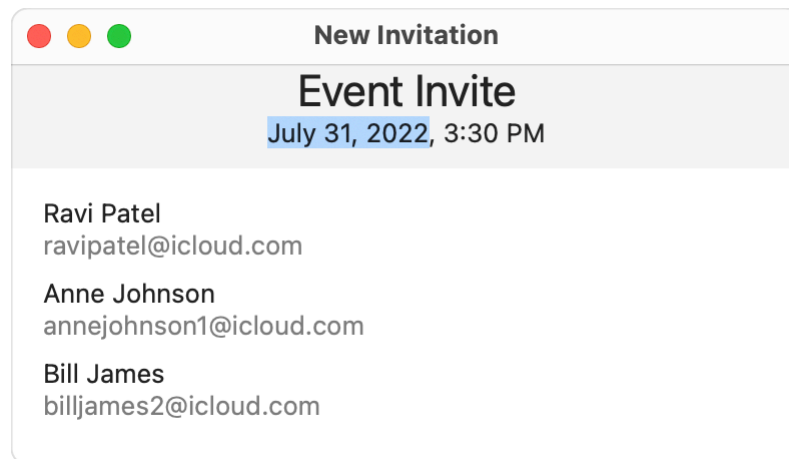
        List(invite.recipients) { recipient in
            VStack (alignment: .leading) {
                Text(recipient.name)
                Text(recipient.email)
            }
        }
    }
}
```

```

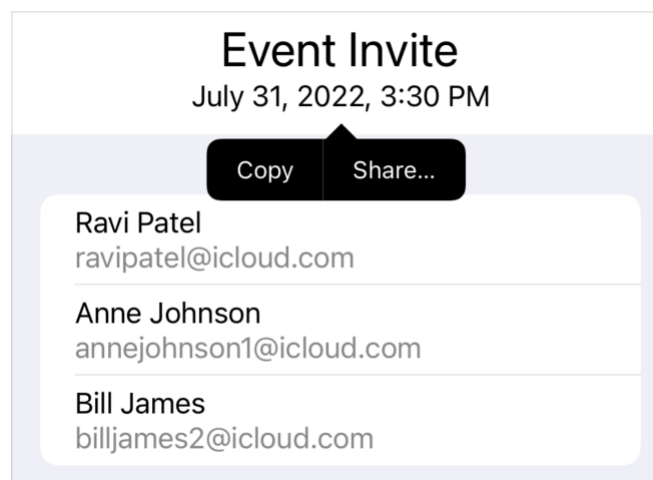
        .foregroundColor(.secondary)
    }
}
.textSelection(.enabled)
}
.navigationTitle("New Invitation")
}

```

On macOS, people use the mouse or trackpad to select a range of text, which they can quickly copy by choosing Edit > Copy, or with the standard keyboard shortcut.



On iOS, the person using the app touches and holds on a selectable Text view, which brings up a system menu with menu items appropriate for the current context. These menu items operate on the entire contents of the Text view; the person can't select a range of text like they can on macOS.



Note

Button views don't support text selection.

See Also

Selecting text

`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).

`struct AttributedTextSelection`

Represents a selection of attributed text.