

[SwiftUI](#) / [ModifiedContent](#)

## Structure

# ModifiedContent

A value with a modifier applied to it.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 13.0+ | visionOS 1.0+ | watchOS 6.0+

```
@frozen
struct ModifiedContent<Content, Modifier>
```

## Topics

### Creating a modified content view

`init(content: Content, modifier: Modifier)`

A structure that defines the content and modifier needed to produce a new view or view modifier.

`var content: Content`

The content that the modifier transforms into a new view or new view modifier.

`var modifier: Modifier`

The view modifier.

### Instance Methods

~~`func accessibility(activationPoint:)`~~

Specifies the point where activations occur in the view.

Deprecated

```
func accessibility(addTraits: AccessibilityTraits) -> ModifiedContent<Content, Modifier>
```

Adds the given traits to the view.

Deprecated

```
func accessibility(hidden: Bool) -> ModifiedContent<Content, Modifier>
```

Specifies whether to hide this view from system accessibility features.

Deprecated

```
func accessibility(hint: Text) -> ModifiedContent<Content, Modifier>
```

Communicates to the user what happens after performing the view's action.

Deprecated

```
func accessibility(identifier: String) -> ModifiedContent<Content, Modifier>
```

Uses the specified string to identify the view.

Deprecated

```
func accessibility(inputLabels: [Text]) -> ModifiedContent<Content, Modifier>
```

Sets alternate input labels with which users identify a view.

Deprecated

```
func accessibility(label: Text) -> ModifiedContent<Content, Modifier>
```

Adds a label to the view that describes its contents.

Deprecated

```
func accessibility(removeTraits: AccessibilityTraits) -> ModifiedContent<Content, Modifier>
```

Removes the given traits from this view.

Deprecated

```
func accessibility(selectionIdentifier: AnyHashable) -> ModifiedContent<Content, Modifier>
```

Sets a selection identifier for this view's accessibility element.

Deprecated

```
func accessibility(sortPriority: Double) -> ModifiedContent<Content, Modifier>
```

Sets the sort priority order for this view's accessibility element, relative to other elements at the same level.

Deprecated

```
func accessibility(value: Text) -> ModifiedContent<Content, Modifier>
```

Adds a textual description of the value that the view contains.

Deprecated

```
func accessibilityAction(AccessibilityActionKind, () -> Void) -> ModifiedContent<Content, Modifier>
```

Adds an accessibility action to the view. Actions allow assistive technologies, such as the VoiceOver, to interact with the view by invoking the action.

```
func accessibilityAction<I>(AccessibilityActionKind, intent: I) -> ModifiedContent<Content, Modifier>
```

Adds an accessibility action representing actionKind to the view. Actions allow assistive technologies, such as the VoiceOver, to interact with the view by invoking the action. When the action is performed, the intent will be invoked.

```
func accessibilityAction(named:_:)
```

Adds an accessibility action to the view. Actions allow assistive technologies, such as the VoiceOver, to interact with the view by invoking the action.

```
func accessibilityAction(named:intent:)
```

Adds an accessibility action labeled name to the view. Actions allow assistive technologies, such as the VoiceOver, to interact with the view by invoking the action. When the action is performed, the intent will be invoked.

```
func accessibilityActivationPoint(_:)
```

The activation point for an element is the location assistive technologies use to initiate gestures.

```
func accessibilityActivationPoint(_:isEnabled:)
```

The activation point for an element is the location assistive technologies use to initiate gestures.

```
func accessibilityAddTraits(AccessibilityTraits) -> ModifiedContent<Content, Modifier>
```

Adds the given traits to the view.

```
func accessibilityAdjustableAction((AccessibilityAdjustmentDirection) -> Void) -> ModifiedContent<Content, Modifier>
```

Adds an accessibility adjustable action to the view. Actions allow assistive technologies, such as the VoiceOver, to interact with the view by invoking the action.

```
func accessibilityCustomContent(_:_:importance:)
```

Add additional accessibility information to the view.

```
func accessibilityDirectTouch(Bool, options: AccessibilityDirectTouchOptions) -> ModifiedContent<Content, Modifier>
```

Explicitly set whether this accessibility element is a direct touch area. Direct touch areas passthrough touch events to the app rather than being handled through an assistive technology, such as VoiceOver. The modifier accepts an optional AccessibilityDirectTouchOptions option set to customize the functionality of the direct touch area.

```
func accessibilityDragPoint(_:description:)
```

The point an assistive technology should use to begin a drag interaction.

```
func accessibilityDragPoint(_:description:isEnabled:)
```

The point an assistive technology should use to begin a drag interaction.

```
func accessibilityDropPoint(_:description:)
```

The point an assistive technology should use to end a drag interaction.

```
func accessibilityDropPoint(_:description:isEnabled:)
```

The point an assistive technology should use to end a drag interaction.

```
func accessibilityHeading(AccessibilityHeadingLevel) -> ModifiedContent<Content, Modifier>
```

Set the level of this heading.

```
func accessibilityHidden(Bool) -> ModifiedContent<Content, Modifier>
```

Specifies whether to hide this view from system accessibility features.

```
func accessibilityHidden(Bool, isEnabled: Bool) -> ModifiedContent<Content, Modifier>
```

Specifies whether to hide this view from system accessibility features.

```
func accessibilityHint(_:)
```

Communicates to the user what happens after performing the view's action.

```
func accessibilityHint(_:isEnabled:)
```

Communicates to the user what happens after performing the view's action.

```
func accessibilityIdentifier(String) -> ModifiedContent<Content, Modifier>
```

Uses the string you specify to identify the view.

```
func accessibilityIdentifier(String, isEnabled: Bool) -> ModifiedContent<Content, Modifier>
```

Uses the string you specify to identify the view.

```
func accessibilityInputLabels(_:)
```

Sets alternate input labels with which users identify a view.

```
func accessibilityInputLabels(_:isEnabled:)
```

Sets alternate input labels with which users identify a view.

```
func accessibilityLabel(_:)
```

Adds a label to the view that describes its contents.

```
func accessibilityLabel(_:isEnabled:)
```

Adds a label to the view that describes its contents.

```
func accessibilityRemoveTraits(AccessibilityTraits) -> ModifiedContent<Content, Modifier>
```

Removes the given traits from this view.

```
func accessibilityRespondsToUserInteraction(Bool) -> ModifiedContent<Content, Modifier>
```

Explicitly set whether this Accessibility element responds to user interaction and would thus be interacted with by technologies such as Switch Control, Voice Control or Full Keyboard Access.

```
func accessibilityRespondsToUserInteraction(Bool, isEnabled: Bool) -> ModifiedContent<Content, Modifier>
```

Explicitly set whether this Accessibility element responds to user interaction and would thus be interacted with by technologies such as Switch Control, Voice Control or Full Keyboard Access.

```
func accessibilityScrollAction((Edge) -> Void) -> ModifiedContent<Content, Modifier>
```

Adds an accessibility scroll action to the view. Actions allow assistive technologies, such as the VoiceOver, to interact with the view by invoking the action.

```
func accessibilityScrollStatus(_:isEnabled:)
```

Changes the announcement provided by accessibility technologies when a user scrolls a scroll view within this view.

```
func accessibilitySortPriority(Double) -> ModifiedContent<Content, Modifier>
```

Sets the sort priority order for this view's accessibility element, relative to other elements at the same level.

```
func accessibilityTextContentType(AccessibilityTextContentType) -> ModifiedContent<Content, Modifier>
```

Sets an accessibility text content type.

```
func accessibilityValue(_:)
```

Adds a textual description of the value that the view contains.

```
func accessibilityValue(_:isEnabled:)
```

Adds a textual description of the value that the view contains.

```
func accessibilityZoomAction((AccessibilityZoomGestureAction) -> Void) -> ModifiedContent<Content, Modifier>
```

Adds an accessibility zoom action to the view. Actions allow assistive technologies, such as VoiceOver, to interact with the view by invoking the action.

---

## Relationships

### Conforms To

Animatable

Conforms when Content conforms to Animatable and Modifier conforms to Animatable.

Chart3DContent

Copyable

CustomHoverEffect

Conforms when Content conforms to CustomHoverEffect and Modifier conforms to CustomHoverEffect.

DynamicMapContent

DynamicTableRowContent

Conforms when Content conforms to DynamicTableRowContent and Modifier conforms to \_TableRowContentModifier.

DynamicViewContent

Conforms when Content conforms to DynamicViewContent and Modifier conforms to ViewModifier.

Equatable

HoverEffectContent

Conforms when Content conforms to HoverEffectContent and Modifier conforms to HoverEffectContent.

## MapContent

### Scene

Conforms when Content conforms to Scene and Modifier conforms to \_SceneModifier.

## Sendable

### SendableMetatype

### TableRowContent

Conforms when Content conforms to TableRowContent and Modifier conforms to \_TableRowContentModifier.

## View

Conforms when Content conforms to View and Modifier conforms to ViewModifier.

## ViewModifier

Conforms when Content conforms to ViewModifier and Modifier conforms to ViewModifier.

## VisualEffect

Conforms when Content conforms to VisualEffect and Modifier conforms to VisualEffect.

---

# See Also

## Modifying a view

### Configuring views

Adjust the characteristics of a view by applying view modifiers.

### Reducing view modifier maintenance

Bundle view modifiers that you regularly reuse into a custom view modifier.

`func modifier<T>(T) -> ModifiedContent<Self, T>`

Applies a modifier to a view and returns a new view.

`protocol ViewModifier`

A modifier that you apply to a view or another view modifier, producing a different version of the original value.

`struct EmptyModifier`

An empty, or identity, modifier, used during development to switch modifiers at compile time.

`protocol EnvironmentalModifier`

A modifier that must resolve to a concrete modifier in an environment before use.

```
struct ManipulableModifier  
struct ManipulableResponderModifier  
struct ManipulableTransformBindingModifier  
struct ManipulationGeometryModifier  
struct ManipulationGestureModifier  
struct ManipulationUsingGestureStateModifier
```

```
enum Manipulable
```

A namespace for various manipulable related types.