Structure

UICornerConfiguration

A configuration that defines how corner radii are mapped to the corners of a rectangle.

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

```
struct UICornerConfiguration
```

Overview

Create a `UICornerConfiguration` that expresses how you want the corners of your view to appear. Your configuration can apply to corners independently or uniformly, and can form the following types of corners:

- A squared corner
- A rounded corner
- A rounded corner that's concentric relative to the containing view
- Corners that are rounded to form a capsule

Select a method to create a configuration that describes which corners of your view you want to be uniform and which corners you want to be independent, then provide instances of `UICornerRadius` as parameters to indicate which type you want each corner to be.

The system uses squared corners by default, so you don't need to set a configuration to get squared corners.

Configure a rounded corner

To configure a rounded corner with a fixed radius, provide `fixed(_:)` with a value greater than zero for the radius. Since `UICornerRadius` conforms to `ExpressibleByFloatLiteral` and `ExpressibleByIntegerLiteral`, you can also provide a float or integer value for the radius:

```
myView.cornerConfiguration = .corners(radius: 12.0)
```

Configure a concentric rounded corner

To configure a rounded corner that's concentric relative to the containing view, use `containerConcentric(minimum:)`:

```
myView.cornerConfiguration = .corners(radius: .containerConcentric())
```

Set the `minimum` parameter to indicate a minimum radius for the rounded corner.

Configure a corner as a capsule

To configure rounded corners that form a capsule, use `capsule(maximumRadius:)`:

```
myView.cornerConfiguration = .capsule()
```

Set the `maximumRadius` parameter to allow your view to break the capsule paradigm and stretch vertically with an edge if the radius necessary to form a capsule exceeds what you provide.

Topics

Configuring independent corners

```
static func corners(radius: UICornerRadius) -> UICornerConfiguration
```

A configuration that applies the given radius independently to all corners.

```
static func corners(topLeftRadius: UICornerRadius?, topRightRadius: UICornerRadius?, bottomLeftRadius: UICornerRadius?, bottomRightRadius: UICornerRadius?) -> UICornerConfiguration
```

A configuration with independent radii for each corner.

Configuring corners as a capsule

```
static func capsule(maximumRadius: Double?) -> UICornerConfiguration
```

A configuration that rounds the corners into a capsule shape, scaling with the view's size up to the maximum radius you provide.

Configuring uniform corners

```
static func uniformCorners(radius: UICornerRadius) -> UICorner
Configuration
```

A configuration that applies the given radius uniformly to all corners.

```
static func uniformEdges(leftRadius: UICornerRadius, rightRadius:
UICornerRadius) -> UICornerConfiguration
```

A configuration that applies the left radius you provide to the left corners, and the right radius you provide to the right corners.

```
static func uniformEdges(topRadius: UICornerRadius, bottomRadius:
UICornerRadius) -> UICornerConfiguration
```

A configuration that applies the top radius to the top corners, and the bottom radius you provide to the bottom corners.

```
static func uniformBottomRadius(UICornerRadius, topLeftRadius: UICorner
Radius?, topRightRadius: UICornerRadius?) -> UICornerConfiguration
```

A configuration that applies the radius you provide to the bottom corners, with optional independent radii for the top corners.

```
static func uniformLeftRadius(UICornerRadius, topRightRadius: UICorner
Radius?, bottomRightRadius: UICornerRadius?) -> UICornerConfiguration
```

A configuration that applies the left radius to the left corners, with optional independent radii for the right corners.

```
static func uniformRightRadius(UICornerRadius, topLeftRadius: UICorner
Radius?, bottomLeftRadius: UICornerRadius?) -> UICornerConfiguration
```

A configuration that applies the right radius you provide to the right corners, with optional independent radii for the left corners.

```
static func uniformTopRadius(UICornerRadius, bottomLeftRadius: UICorner
Radius?, bottomRightRadius: UICornerRadius?) -> UICornerConfiguration
```

A configuration that applies the top radius you provide to the top corners, with optional independent radii for the bottom corners.

Relationships

Conforms To

`CustomStringConvertible`, `Equatable`, `Hashable`

See Also

Configuring a view's corners

`var cornerConfiguration: UICornerConfiguration`

A configuration that defines the corners of the view.

`struct UICornerRadius`

A type that represents the radius the system uses to round a corner.

`func effectiveRadius(corner: UIRectCorner) -> CGFloat`

Returns the effective radius for the corner you provide, calculated using the view's current corner configuration.