Documentation Language: Swift

SwiftUI / Entry()

Macro

Entry()

Creates an environment values, transaction, container values, or focused values entry.

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

```
@attached(accessor) @attached(peer, names: prefixed(__Key_))
macro Entry()
```

Mentioned in

Building and customizing the menu bar with SwiftUI

Environment Values

Create `EnvironmentValues` entries by extending the `EnvironmentValues` structure with new properties and attaching the @Entry macro to the variable declarations:

```
extension EnvironmentValues {
    @Entry var myCustomValue: String = "Default value"
    @Entry var anotherCustomValue = true
}
```

Transaction Values

Create `Transaction` entries by extending the `Transaction` structure with new properties and attaching the @Entry macro to the variable declarations:

```
extension Transaction {
    @Entry var myCustomValue: String = "Default value"
}
```

Container Values

Create `ContainerValues` entries by extending the `ContainerValues` structure with new properties and attaching the @Entry macro to the variable declarations:

```
extension ContainerValues {
    @Entry var myCustomValue: String = "Default value"
}
```

Focused Values

Since the default value for `FocusedValues` is always `nil`, `FocusedValues` entries cannot specify a different default value and must have an Optional type.

Create `FocusedValues` entries by extending the `FocusedValues` structure with new properties and attaching the @Entry macro to the variable declarations:

```
extension FocusedValues {
    @Entry var myCustomValue: String?
}
```

See Also

Moving an animation to another view

`func withTransaction<Result>(Transaction, () throws -> Result) rethrows -> Result`

Executes a closure with the specified transaction and returns the result.

`func withTransaction<R, V>(WritableKeyPath<Transaction, V>, V, () throws -> R) rethrows -> R`

Executes a closure with the specified transaction key path and value and returns the result.

```
func transaction((inout Transaction) -> Void) -> some View
```

Applies the given transaction mutation function to all animations used within the view.

```
func transaction(value: some Equatable, (inout Transaction) -> Void) ->
some View
```

Applies the given transaction mutation function to all animations used within the view.

```
func transaction<V>((inout Transaction) -> Void, body: (Placeholder
ContentView<Self>) -> V) -> some View
```

Applies the given transaction mutation function to all animations used within the body closure.

```
struct Transaction
```

The context of the current state-processing update.

```
protocol TransactionKey
```

A key for accessing values in a transaction.