

[AppKit](#) / NSBindingOption

## Structure

# NSBindingOption

macOS

```
struct NSBindingOption
```

## Discussion

Values that are used as keys in the options dictionary passed to the [`bind\(\_ :to:withKeyPath:options:\)`](#) method.

These keys are also used in the dictionary returned as the [`options`](#) value of [`infoForBinding\(\_ :\)`](#). See the [Cocoa Bindings Reference](#) for more information.

## Topics

### Binding Options

```
static let allowsEditingMultipleValuesSelection: NSBindingOption
```

An `NSNumber` object containing a Boolean value that determines if the binding allows editing when the value represents a multiple selection.

```
static let allowsNullArgument: NSBindingOption
```

An `NSNumber` object containing a Boolean value that determines if the argument bindings allows passing argument values of `nil`.

```
static let alwaysPresentsApplicationModalAlerts: NSBindingOption
```

A number containing a Boolean value that determines if validation and error alert panels displayed as a result of this binding are displayed as application modal alerts.

`static let conditionallySetsEditable: NSBindingOption`

An `NSNumber` object containing a Boolean value that determines if the editable state of the user interface item is automatically configured based on the controller's selection.

`static let conditionallySetsEnabled: NSBindingOption`

An `NSNumber` object containing a Boolean value that determines if the enabled state of the user interface item is automatically configured based on the controller's selection.

`static let conditionallySetsHidden: NSBindingOption`

An `NSNumber` object containing a Boolean value that determines if the hidden state of the user interface item is automatically configured based on the controller's selection.

`static let contentPlacementTag: NSBindingOption`

A number that specifies the tag id of the popup menu item to replace with the content of the array.

`static let continuouslyUpdatesValue: NSBindingOption`

An `NSNumber` object containing a Boolean value that determines whether the value of the binding is updated as edits are made to the user interface item or is updated only when the user interface item resigns as the responder.

`static let createsSortDescriptor: NSBindingOption`

An `NSNumber` object containing a Boolean value that determines if a sort descriptor is created for a table column.

`static let deletesObjectsOnRemove: NSBindingOption`

An `NSNumber` object containing a Boolean value that determines if an object is deleted from the managed context immediately upon being removed from a relationship.

`static let displayName: NSBindingOption`

An `NSString` object containing a human readable string to be displayed for a predicate.

`static let displayPattern: NSBindingOption`

An `NSString` object that specifies a format string used to construct the final value of a string.

`static let handlesContentAsCompoundValue: NSBindingOption`

An `NSNumber` object containing a Boolean value that determines if the content is treated as a compound value.

```
static let insertsNullPlaceholder: NSBindingOption
```

An `NSNumber` object containing a Boolean value that determines if an additional item which represents `nil` is inserted into a matrix or pop-up menu before the items in the content array.

```
static let invokesSeparatelyWithArrayObjects: NSBindingOption
```

An `NSNumber` object containing a Boolean value that determines whether the specified selector is invoked with the array as the argument or is invoked repeatedly with each array item as an argument.

```
static let multipleValuesPlaceholder: NSBindingOption
```

An object that is used as a placeholder when the key path of the bound controller returns the `NSMultipleValuesMarker` marker for a binding.

```
static let noSelectionPlaceholder: NSBindingOption
```

An object that is used as a placeholder when the key path of the bound controller returns the `NSNoSelectionMarker` marker for a binding.

```
static let notApplicablePlaceholder: NSBindingOption
```

An object that is used as a placeholder when the key path of the bound controller returns the `NSNotApplicableMarker` marker for a binding.

```
static let nullPlaceholder: NSBindingOption
```

An object that is used as a placeholder when the key path of the bound controller returns `nil` for a binding.

```
static let predicateFormat: NSBindingOption
```

An `NSString` object containing the predicate pattern string for the predicate bindings. Use `$value` to refer to the value in the search field.

```
static let raisesForNotApplicableKeys: NSBindingOption
```

An `NSNumber` object containing a Boolean value that specifies if an exception is raised when the binding is bound to a key that is not applicable—for example when an object is not key-value coding compliant for a key.

```
static let selectorName: NSBindingOption
```

An `NSString` object that specifies the method selector invoked by the target binding when the user interface item is clicked.

```
static let selectsAllWhenSettingContent: NSBindingOption
```

An `NSNumber` object containing a Boolean value that specifies if all the items in the array controller are selected when the content is set.

```
static let validatesImmediately: NSBindingOption
```

An NSNumber object containing a Boolean value that determines if the contents of the binding are validated immediately.

```
static let valueTransformer: NSBindingOption
```

An NSValueTransformer instance that is applied to the bound value.

```
static let valueTransformerName: NSBindingOption
```

The value for this key is an identifier of a registered NSValueTransformer instance that is applied to the bound value.

## Initializers

```
init(rawValue: String)
```

---

## Relationships

### Conforms To

Equatable

Hashable

RawRepresentable

Sendable

SendableMetatype

---

## See Also

### Key-Value Data

```
class NSDictionaryController
```

A bindings-compatible controller that manages the display and editing of a dictionary of key-value pairs.

```
class NSDictionaryControllerKeyValuePair
```

A set of methods implemented by arranged objects to give access to information about those objects.

```
struct NSBindingName
```

Values that specify a binding for certain methods.

```
struct NSBindingInfoKey
```

```
func NSIsControllerMarker(Any?) -> Bool
```

Tests whether a given object is special marker object used for indicating the state of a selection in relation to a key.

📄 [NSKeyValueBindingCreation](#)

A set of methods that you can use to create and remove bindings between view objects and controllers, or between controllers and model objects.

☰ [Binding dictionary keys](#)

These constants define keys in the binding information dictionary.