AppKit / NSIsControllerMarker(_:)

Function

NSIsControllerMarker(_:)

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

macOS

```
func NSIsControllerMarker(_ object: Any?) -> Bool
```

Parameters

`object`
Specify the object you want to check. This parameter can be `nil`.

Return Value

`true` if the object is one of the designated controller markers or `false` if it is not.

Discussion

This function helps you to create bindings between user interface elements and controller objects. The Application Kit predefines several special marker objects used as values for indicating selection state; currently these are `NSMultipleValuesMarker`, `NSNoSelectionMarker`, and `NSNotApplicableMarker`. These markers are typed as `id` and only exist for the purpose of indicating a state; they are never archived and cannot be used as object values in controls. You use this function to test whether a given object value is a marker, in which case it is not directly assignable to the object that is bound. This check is important, especially since additional markers may be added in the future.

See the `NSKeyValueBinding.h` header file for further details.

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 NSBindingOption`

`struct NSBindingInfoKey`

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.