AppKit / NSPasteboard

Class

NSPasteboard

An object that transfers data to and from the pasteboard server.

macOS

```
class NSPasteboard
```

Mentioned in

▤ Supporting Writing Tools via the pasteboard

Overview

The pasteboard server is shared by all running apps. It contains data that the user has cut or copied, as well as other data that one application wants to transfer to another. `NSPasteboard` objects are an application's sole interface to the server and to all pasteboard operations.

An `NSPasteboard` object is also used to transfer data between apps and service providers listed in each application's Services menu. The drag pasteboard is used to transfer data that is being dragged by the user.

A pasteboard can contain multiple items. You can directly write or read any object that implements the `NSPasteboardWriting` or `NSPasteboardReading` `Protocol` respectively. This allows you to write and read common items such as URLs, colors, images, strings, attributed strings, and sounds without an intermediary object. Your custom classes can also implement these protocols for use with the pasteboard.

Writing methods such as `setData(_:forType:)` provide a convenient means of writing to the first pasteboard item, without having to create the first pasteboard item. You can use code like this, for example:

```
[pboard clearContents];
[pboard setData:data forType:type];
```

The general pasteboard, available by way of the general class method, automatically participates with the Universal Clipboard feature in macOS 10.12 and later and in iOS 10.0 and later. There is no macOS API for interacting with this feature.

Topics

Creating and releasing a pasteboard

class var general: NSPasteboard

The shared pasteboard object to use for general content.

init(byFilteringData: Data, ofType: NSPasteboard.PasteboardType)

Creates a new pasteboard object that supplies the specified data in as many types as possible based on the available filter services.

init(byFilteringFile: String)

Creates a new pasteboard object that supplies the specified file in as many types as possible based on the available filter services.

init(byFilteringTypesInPasteboard: NSPasteboard)

Creates a new pasteboard object that supplies the specified pasteboard data in as many types as possible based on the available filter services.

init(name: NSPasteboard.Name)

Returns the pasteboard with the specified name.

struct Name

Constants that represent the standard pasteboard names.

class func withUniqueName() -> NSPasteboard

Creates and returns a new pasteboard with a name that is guaranteed to be unique with respect to other pasteboards in the system.

func releaseGlobally()

Releases the receiver's resources in the pasteboard server.

Determining pasteboard access

var accessBehavior: NSPasteboard.AccessBehavior

    The current pasteboard access behavior. The user can customize this behavior per-app in System Settings for any app that has triggered a pasteboard access alert in the past.

enum AccessBehavior

    A value indicating pasteboard access behavior.

Writing data

func clearContents() -> Int

    Clears the existing contents of the pasteboard.

func writeObjects([any NSPasteboardWriting]) -> Bool

    Writes an array of objects to the receiver.

func setData(Data?, forType: NSPasteboard.PasteboardType) -> Bool

    Sets the data as the representation for the specified type for the first item on the receiver.

func setPropertyList(Any, forType: NSPasteboard.PasteboardType) -> Bool

    Sets the given property list as the representation for the specified type for the first item on the receiver.

func setString(String, forType: NSPasteboard.PasteboardType) -> Bool

    Sets the given string as the representation for the specified type for the first item on the receiver.

struct PasteboardType

    The supported pasteboard types.

Reading data

func readObjects(forClasses: [AnyClass], options: [NSPasteboard.Reading OptionKey : Any]?) -> [Any]?

    Reads from the receiver objects that best match the specified array of classes.

struct ReadingOptionKey

    Options for reading pasteboard data.

`struct ReadingOptions`

Options that specify how to interpret data on the pasteboard when initializing pasteboard data.

`var pasteboardItems: [NSPasteboardItem]?`

An array that contains all the items held by the pasteboard.

`func index(of: NSPasteboardItem) -> Int`

Returns the index of the specified pasteboard item.

`func data(forType: NSPasteboard.PasteboardType) -> Data?`

Returns the data for the specified type from the first item in the receiver that contains the type.

`func propertyList(forType: NSPasteboard.PasteboardType) -> Any?`

Returns the property list for the specified type from the first item in the receiver that contains the type.

`func string(forType: NSPasteboard.PasteboardType) -> String?`

Returns a concatenation of the strings for the specified type from all the items in the receiver that contain the type.

Validating contents

`func availableType(from: [NSPasteboard.PasteboardType]) -> NSPasteboard.PasteboardType?`

Scans the specified types for a type that the receiver supports.

`func canReadItem(withDataConformingToTypes: [String]) -> Bool`

Returns a Boolean value that indicates whether the receiver contains any items that conform to the specified UTIs.

`func canReadObject(forClasses: [AnyClass], options: [NSPasteboard.ReadingOptionKey : Any]?) -> Bool`

Returns a Boolean value that indicates whether the receiver contains any items that can be represented as an instance of any class in a given array.

`var types: [NSPasteboard.PasteboardType]?`

An array of the receiver's supported data types.

`class func types(filterableTo: NSPasteboard.PasteboardType) -> [NSPasteboard.PasteboardType]`

Returns the data types that can be converted to the specified type using the available filter services.

Detecting patterns and metadata in pasteboard items

`func detectedPatterns(for: Set<PartialKeyPath<NSPasteboard.Detected Values>>) async throws -> Set<PartialKeyPath<NSPasteboard.Detected Values>>`

Determines whether the first pasteboard item matches the specified patterns, without notifying the person using the app.

`func detectedValues(for: Set<PartialKeyPath<NSPasteboard.DetectedValues >>) async throws -> NSPasteboard.DetectedValues`

Determines whether the first pasteboard item matches the specified patterns, reading the contents if it finds a match.

`struct DetectedValues`

A type that contains common types of data that the data detection system matches for a pasteboard.

⁝☰ Pasteboard detection patterns

Common types of data that the data detection system matches for a pasteboard.

`func detectedMetadata(for: Set<PartialKeyPath<NSPasteboard.Detected Metadata>>) async throws -> NSPasteboard.DetectedMetadata`

Determines available metadata from the specified metadata types for the first pasteboard item, without notifying the person using the app.

`struct DetectedMetadata`

An object that contains common types of metadata that the data detection system matches for a pasteboard.

⁝☰ Pasteboard detection metadata types

Common types of metadata that the data detection system matches for a pasteboard.

Preparing the pasteboard for content

`func prepareForNewContents(with: NSPasteboard.ContentsOptions) -> Int`

Prepares the pasteboard to receive new contents, removing the existing pasteboard contents.

```
struct ContentsOptions
```
Options for preparing the pasteboard.

Getting information about a pasteboard

```
var name: NSPasteboard.Name
```
The receiver's name.

```
var changeCount: Int
```
The receiver's change count.

Writing data (macOS 10.5 and earlier)

These methods all operate on what is conceptually the first item on the pasteboard. They accept UTIs and pboard type strings. In a future release they may take only UTIs.

```
func declareTypes([NSPasteboard.PasteboardType], owner: Any?) -> Int
```
Prepares the receiver for a change in its contents by declaring the new types of data it will contain and a new owner.

```
func addTypes([NSPasteboard.PasteboardType], owner: Any?) -> Int
```
Adds promises for the specified types to the first pasteboard item.

```
func writeFileContents(String) -> Bool
```
Writes the contents of the specified file to the pasteboard.

```
func write(FileWrapper) -> Bool
```
Writes the serialized contents of the specified file wrapper to the pasteboard.

Reading data (macOS 10.5 and earlier)

These methods all operate on what is conceptually the first item on the pasteboard. They accept UTIs and pboard type strings. In a future release they may take only UTIs.

```
func readFileContentsType(NSPasteboard.PasteboardType?, toFile: String)
-> String?
```
Reads data representing a file's contents from the receiver and writes it to the specified file.

```
func readFileWrapper() -> FileWrapper?
```
Reads data representing a file's contents from the receiver and returns it as a file wrapper.

Structures

`struct` `WritingOptions`

   Type to specify options for writing to a pasteboard.

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSObjectProtocol`

See Also

Pasteboard

`class` `NSPasteboardItem`

   An item on a pasteboard.

`protocol` `NSPasteboardReading`

   A set of methods that defines the interface for initializing an object from a pasteboard.

`protocol` `NSPasteboardWriting`

   A set of methods that defines the interface for retrieving a representation of an object that can be written to a pasteboard.

protocol **NSPasteboardItemDataProvider**

A set of methods implemented by the data provider of a pasteboard item to provide the data for a particular UTI type.

struct **ContentsOptions**

Options for preparing the pasteboard.

protocol **NSPasteboardTypeOwner**

An object that serves as a data provider for data types that use lazy data fulfillment from a pasteboard request.