NSPasteboard Class Reference

Developer

Contents

NSPasteboard Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in OS X v10.0 and later.
Declared in	NSPasteboard.h
Companion guides	Pasteboard Programming Guide Drag and Drop Programming Topics Services Implementation Guide
Related sample code	ClipboardViewer DemoAssistant iSpend MyMediaPlayList ZipBrowser

Overview

NSPasteboard objects transfer data to and from the pasteboard server. The server is shared by all running applications. 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 applications and service providers listed in each application's Services menu. The drag pasteboard (NSDragPboard) is used to transfer data that is being dragged by the user.

> **Important:** This reference and the *Pasteboard Programming Guide* describe the API and behavior for OS X v10.6 and later. To learn about the behavior on OS X v10.5 and earlier, see "Pasteboard Programming Topics for Cocoa" in the Legacy Reference Library.

On OS X v10.6 and later, 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.

The writing methods available on OS X v10.5 and earlier 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. On OS X v10.6 and later, the writing methods such as setData:forType: (page 26) still provide a convenient means of writing to the first pasteboard item, without having to create the first pasteboard item. For example, instead of

```
[pboard clearContents];
NSPasteboardItem *item = [[[NSPasteboardItem alloc] init] autorelease];
[item setData:data forType:type];
[pboard writeObjects:[NSArray arrayWithObject:item]];
```

you can write:

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

In addition to being more compact, this is also a little more efficient—it avoids the need to create a separate pasteboard item and array.

Tasks

Creating and Releasing a Pasteboard

+ generalPasteboard (page 8)

Returns the general NSPasteboard object.

+ `pasteboardByFilteringData:ofType:` (page 9)
  Creates and returns a new pasteboard with a unique name that supplies the specified data in as many types as possible given the available filter services.

+ `pasteboardByFilteringFile:` (page 10)
  Creates and returns a new pasteboard with a unique name that supplies the specified file data in as many types as possible given the available filter services.

+ `pasteboardByFilteringTypesInPasteboard:` (page 10)
  Creates and returns a new pasteboard with a unique name that supplies the specified pasteboard data in as many types as possible given the available filter services.

+ `pasteboardWithName:` (page 11)
  Returns the pasteboard with the specified name.

+ `pasteboardWithUniqueName` (page 11)
  Creates and returns a new pasteboard with a name that is guaranteed to be unique with respect to other pasteboards on the computer.

– `releaseGlobally` (page 25)
  Releases the receiver's resources in the pasteboard server.

Writing Data

– `clearContents` (page 17)
  Clears the existing contents of the pasteboard

– `writeObjects:` (page 31)
  Writes an array of objects to the receiver.

– `setData:forType:` (page 26)
  Sets the given data as the representation for the specified type for the first item on the receiver.

– `setPropertyList:forType:` (page 26)
  Sets the given property list as the representation for the specified type for the first item on the receiver.

– `setString:forType:` (page 27)
  Sets the given string as the representation for the specified type for the first item on the receiver.

Reading Data

– `readObjectsForClasses:options:` (page 23)
  Reads from the receiver objects that best match the specified array of classes.

– `pasteboardItems` (page 21)
  Returns all the items held by the receiver.

– `indexOfPasteboardItem:` (page 20)
  Returns the index of the specified pasteboard item.

– `dataForType:` (page 17)
  Returns the data for the specified type from the first item in the receiver that contains the type.

– `propertyListForType:` (page 21)
  Returns the property list for the specified type from the first item in the receiver that contains the type.

– `stringForType:` (page 28)
  Returns a concatenation of the strings for the specified type from all the items in the receiver that contain the type.

Validating Contents

– `availableTypeFromArray:` (page 14)
  Scans the specified types for a type that the receiver supports.

– `canReadItemWithDataConformingToTypes:` (page 15)
  Returns a Boolean value that indicates whether the receiver contains any items that conform to the specified UTIs.

– `canReadObjectForClasses:options:` (page 15)
  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.

– `types` (page 29)
  Returns an array of the receiver's supported data types.

+ `typesFilterableTo:` (page 12)
  Returns the data types that can be converted to the specified type using the available filter services.

Getting Information About a Pasteboard

– `name` (page 20)
  Returns the receiver's name.

– `changeCount` (page 16)
  Returns the receiver's change count.

Writing Data (OS X v10.5 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.

– `declareTypes:owner:` (page 18)
Prepares the receiver for a change in its contents by declaring the new types of data it will contain and a new owner.

– `addTypes:owner:` (page 13)
Adds promises for the specified types to the first pasteboard item.

– `writeFileContents:` (page 30)
Writes the contents of the specified file to the pasteboard.

– `writeFileWrapper:` (page 30)
Writes the serialized contents of the specified file wrapper to the pasteboard.

Reading Data (OS X v10.5 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.

– `readFileContentsType:toFile:` (page 22)
Reads data representing a file's contents from the receiver and writes it to the specified file.

– `readFileWrapper` (page 23)
Reads data representing a file's contents from the receiver and returns it as a file wrapper.

Class Methods

generalPasteboard

Returns the general `NSPasteboard` object.

```
+ (NSPasteboard *)generalPasteboard
```

Return Value
The general pasteboard.

Discussion
Invokes `pasteboardWithName:` (page 11) to obtain the pasteboard.

Availability

Available in OS X v10.0 and later.

Related Sample Code

ClipboardViewer

iSpend

MyMediaPlayList

Sketch

Sketch+Accessibility

Declared in

`NSPasteboard.h`

pasteboardByFilteringData:ofType:

Creates and returns a new pasteboard with a unique name that supplies the specified data in as many types as possible given the available filter services.

```
+ (NSPasteboard *)pasteboardByFilteringData:(NSData *)data ofType:(NSString *)type
```

Parameters

`data`

The data to be placed on the pasteboard.

`type`

The type of data in the `data` parameter.

Return Value

The new pasteboard object.

Discussion

The returned pasteboard also declares data of the supplied `type`.

No filter service is invoked until the data is actually requested, so invoking this method is reasonably inexpensive.

Availability

Available in OS X v10.0 and later.

Declared in

`NSPasteboard.h`

pasteboardByFilteringFile:

Creates and returns a new pasteboard with a unique name that supplies the specified file data in as many types as possible given the available filter services.

```
+ (NSPasteboard *)pasteboardByFilteringFile:(NSString *)filename
```

Parameters

`filename`

The filename to put on the pasteboard.

Return Value

The new pasteboard object.

Discussion

No filter service is invoked until the data is actually requested, so invoking this method is reasonably inexpensive.

Availability

Available in OS X v10.0 and later.

Declared in

`NSPasteboard.h`

pasteboardByFilteringTypesInPasteboard:

Creates and returns a new pasteboard with a unique name that supplies the specified pasteboard data in as many types as possible given the available filter services.

```
+ (NSPasteboard *)pasteboardByFilteringTypesInPasteboard:(NSPasteboard *)pasteboard
```

Parameters

`pasteboard`

The original pasteboard object.

Return Value

The new pasteboard object. This method returns the object in the `pasteboard` parameter if the pasteboard was returned by one of the `pasteboardByFiltering...` methods. This prevents a pasteboard from being expanded multiple times.

Discussion

This process can be thought of as expanding the pasteboard, because the new pasteboard generally contains more representations of the data than `pasteboard`.

This method only returns the original types and the types that can be created as a result of a single filter; the pasteboard does not have defined types that are the result of translation by multiple filters.

No filter service is invoked until the data is actually requested, so invoking this method is reasonably inexpensive.

Availability

Available in OS X v10.0 and later.

Declared in

`NSPasteboard.h`

pasteboardWithName:

Returns the pasteboard with the specified name.

```
+ (NSPasteboard *)pasteboardWithName:(NSString *)name
```

Parameters

`name`

The name of the pasteboard. The names of standard pasteboards are given in "Pasteboard Names" (page 32).

Return Value

The pasteboard associated with the given name, or a new `NSPasteboard` object if the application does not yet have a pasteboard object for the specified name.

Discussion

Other names can be assigned to create private pasteboards for other purposes.

Availability

Available in OS X v10.0 and later.

Related Sample Code

ClipboardViewer

ZipBrowser

Declared in

`NSPasteboard.h`

pasteboardWithUniqueName

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

```
+ (NSPasteboard *)pasteboardWithUniqueName
```

Return Value

The new pasteboard object.

Discussion

This method is useful for applications that implement their own interprocess communication using pasteboards.

Availability

Available in OS X v10.0 and later.

Related Sample Code
PhotoSearch

Declared in

`NSPasteboard.h`

typesFilterableTo:

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

```
+ (NSArray *)typesFilterableTo:(NSString *)type
```

Parameters

`type`

The target data type.

Return Value

An array of `NSString` objects containing the types that can be converted to the target data type.

Discussion

The array also contains the original type.

Availability

Available in OS X v10.0 and later.

Declared in

`NSPasteboard.h`

Instance Methods

addTypes:owner:

Adds promises for the specified types to the first pasteboard item.

```
- (NSInteger)addTypes:(NSArray *)newTypes owner:(id)newOwner
```

Parameters

`newTypes`

An array of `NSString` objects, each of which specifies a type of data that can be provided to the pasteboard.

`newOwner`

The object that provides the data for the specified types.

If the data for those types is provided immediately, the owner can be `nil`. If the data for the added types will be provided lazily when requested from the pasteboard, an owner object must be provided that implements the `-pasteboard:provideDataForType:` method of the `NSPasteboardOwner` informal protocol.

Return Value

The new change count, or `0` if there was an error adding the types.

Discussion

This method adds promises for the specified types to the first pasteboard item.

You use this methods to declare additional types of data for the first pasteboard item in the receiver. You can also use it to replace existing types added by a previous `declareTypes:owner:` (page 18) or `addTypes:owner:` (page 13) message.

The `newTypes` parameter specifies the types of data you are promising to the pasteboard. The types should be ordered according to the preference of the source application, with the most preferred type coming first (typically, the richest representation). New types are added to the end of the list containing any existing types, if any.

If you specify a type that has already been declared, this method replaces the owner of that type with the value in `newOwner`. In addition, any data already already written to the pasteboard for that type is removed.

Availability

Available in OS X v10.0 and later.

See Also

– `changeCount` (page 16)

Declared in

`NSPasteboard.h`

availableTypeFromArray:

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

```
- (NSString *)availableTypeFromArray:(NSArray *)types
```

Parameters

`types`

An array of `NSString` objects specifying the pasteboard types your application supports, in preferred order.

Return Value

The first pasteboard type in `types` that is available on the pasteboard, or `nil` if the receiver does not contain any of the types in `types`.

Discussion

You use this method to determine the best representation available on the pasteboard. For example, if your application supports RTFD, RTF, and string data, then you might invoke the method as follows:

```
NSArray *supportedTypes =
    [NSArray arrayWithObjects: NSRTFDPboardType, NSRTFPboardType, NSStringPboardType,
 nil];
NSString *bestType = [[NSPasteboard generalPasteboard]
    availableTypeFromArray:supportedTypes];
```

If the pasteboard contains RTF and string data, then `bestType` would contain `NSRTFPboardType`. If the pasteboard contains none of the types in `supportedTypes`, then `bestType` would be `nil`.

You must send a `types` (page 29) or `availableTypeFromArray:` message before reading any data from an `NSPasteboard` object. If you need to see if a type in the returned array matches a type string you have stored locally, use the `isEqualToString:` method to perform the comparison.

Availability

Available in OS X v10.0 and later.

Related Sample Code

Document Package with iCloud

Sketch

Sketch+Accessibility
TextEdit

Declared in
`NSPasteboard.h`

canReadItemWithDataConformingToTypes:

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

```
- (BOOL)canReadItemWithDataConformingToTypes:(NSArray *)types
```

Parameters

`types`
An array of `NSString` objects containing UTIs.

Return Value

`YES` if the receiver contains any items that conform to the UTIs specified in `types`, otherwise `NO`.

Availability

Available in OS X v10.6 and later.

See Also

– `canReadObjectForClasses:options:` (page 15)
– `readObjectsForClasses:options:` (page 23)

Declared in
`NSPasteboard.h`

canReadObjectForClasses:options:

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.

```
- (BOOL)canReadObjectForClasses:(NSArray *)classArray
options:(NSDictionary *)options
```

Parameters

`classArray`
An array of class objects.

Classes in the array must conform to the `NSPasteboardReading` protocol.

`options`

A dictionary that specifies options to refine the search for pasteboard items, for example to restrict the search to file URLs with particular content types. For valid dictionary keys, see "Pasteboard Reading Options" (page 39).

Return Value

`YES` if the receiver contains any items that can be represented as an instance of a class specified in `classArray`, otherwise `NO`.

Availability

Available in OS X v10.6 and later.

See Also

– `canReadItemWithDataConformingToTypes:` (page 15)

– `readObjectsForClasses:options:` (page 23)

Related Sample Code

MyMediaPlayList

TableViewPlayground

Declared in

`NSPasteboard.h`

changeCount

Returns the receiver's change count.

```
- (NSInteger)changeCount
```

Return Value

The change count of the receiver.

Discussion

The change count starts at zero when a client creates the receiver and becomes the first owner. The change count subsequently increments each time the pasteboard ownership changes.

The change count is also returned from `clearContents` (page 17) and `declareTypes:owner:` (page 18). You can therefore record the change count at the time that you take ownership of the pasteboard and later compare it with the value returned from `changeCount` to determine whether you still have ownership.

Availability

Available in OS X v10.0 and later.

See Also

– `clearContents` (page 17)

– `declareTypes:owner:` (page 18)

Related Sample Code

Sketch

Sketch+Accessibility

Declared in

`NSPasteboard.h`

clearContents

Clears the existing contents of the pasteboard

```
– (NSInteger)clearContents
```

Return Value

The change count of the receiver.

Discussion

Clears the existing contents of the pasteboard, preparing it for new contents. This is the first step in providing data on the pasteboard.

Availability

Available in OS X v10.6 and later.

Related Sample Code

IconCollection

MyMediaPlayList

Declared in

`NSPasteboard.h`

dataForType:

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

```
– (NSData *)dataForType:(NSString *)dataType
```

Parameters

`dataType`

The type of data you want to read from the pasteboard. This value should be one of the types returned by `types` (page 29) or `availableTypeFromArray:` (page 14).

Return Value

A data object containing the data for the specified type from the first item in the receiver that contains the type, or `nil` if the contents of the pasteboard changed since they were last checked.

This method may also return `nil` if the pasteboard server cannot supply the data in time—for example, if the pasteboard's owner is slow in responding to a `pasteboard:provideDataForType:` message and the interprocess communication times out.

Discussion

Errors other than a timeout raise a `NSPasteboardCommunicationException`.

If `nil` is returned, the application should put up a panel informing the user that it was unable to carry out the paste operation.

Special Considerations

For standard text data types such as string, RTF, and RTFD, the text data from each item is returned as one combined result separated by newlines.

Availability

Available in OS X v10.0 and later.

See Also

– `setData:forType:` (page 26)

Related Sample Code

Document Package with iCloud

iSpend

Sketch

Sketch+Accessibility

TextEdit

Declared in

`NSPasteboard.h`

declareTypes:owner:

Prepares the receiver for a change in its contents by declaring the new types of data it will contain and a new owner.

```
- (NSInteger)declareTypes:(NSArray *)newTypes owner:(id)newOwner
```

Parameters

`newTypes`

An array of `NSString` objects that specify the types of data that may be added to the new pasteboard. The types should be ordered according to the preference of the source application, with the most preferred type coming first (typically, the richest representation).

`newOwner`

The object responsible for writing data to the pasteboard, or `nil` if you provide data for all types immediately. If you specify a `newOwner` object, it must support all of the types declared in the `newTypes` parameter and must remain valid for as long as the data is *promised* on the pasteboard.

Return Value

The receiver's new change count.

Discussion

This method is the equivalent of invoking `clearContents` (page 17), implicitly writing the first pasteboard item, and then calling`addTypes:owner:` (page 13) to promise types for the first pasteboard item.

> **OS X v10.5 and earlier:** In OS X v10.5 and earlier, this method is the first step in writing data to the pasteboard and must precede the messages that actually write the data. A `declareTypes:owner:` message essentially changes the contents of the receiver: It invalidates the current contents of the receiver and increments its change count.

Special Considerations

In general, you should not use this method with `writeObjects:` (page 31), since `writeObjects:` will always write additional items to the pasteboard, and will not affect items already on the pasteboard, including the item implicitly created by this method.

Availability

Available in OS X v10.0 and later.

See Also

– `clearContents` (page 17)
– `addTypes:owner:` (page 13)
– `changeCount` (page 16)
`pasteboard:provideDataForType:` (`NSPasteboardOwner` Informal Protocol)

Related Sample Code

SimpleService

SourceView

SpotlightFortunes

With and Without Bindings

ZipBrowser

Declared in

`NSPasteboard.h`

indexOfPasteboardItem:

Returns the index of the specified pasteboard item.

```
- (NSUInteger)indexOfPasteboardItem:(NSPasteboardItem *)pasteboardItem
```

Parameters

`pasteboardItem`

A pasteboard item.

Return Value

The index of the specified pasteboard item. If `pasteboardItem` has not been added to any pasteboard, or is owned by another pasteboard, returns `NSNotFound`.

Discussion

An item's index in the pasteboard is useful for a pasteboard item data provider that has promised data for multiple items, to be able to easily match the pasteboard item to an array of source data from which to derive the promised data.

Availability

Available in OS X v10.6 and later.

Declared in

`NSPasteboard.h`

name

Returns the receiver's name.

```
- (NSString *)name
```

Return Value

The name of the receiver.

Availability

Available in OS X v10.0 and later.

See Also

`+ pasteboardWithName:` (page 11)

Declared in

`NSPasteboard.h`

pasteboardItems

Returns all the items held by the receiver.

```
- (NSArray *)pasteboardItems
```

Return Value

Returns all the items held by the receiver, or `nil` if there is an error retrieving pasteboard items.

Availability

Available in OS X v10.6 and later.

See Also

`– readObjectsForClasses:options:` (page 23)

Declared in

`NSPasteboard.h`

propertyListForType:

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

```
- (id)propertyListForType:(NSString *)dataType
```

Parameters

`dataType`

The pasteboard data type containing the property-list data.

Return Value

The property list for the specified type from the first item in the receiver that contains the type. This object consists of `NSArray`, `NSData`, `NSDictionary`, or `NSString` objects—or any combination thereof.

Discussion

This method invokes the `dataForType:` (page 17) method.

Special Considerations

You must send `types` (page 29) or `availableTypeFromArray:` (page 14) before invoking `propertyListForType:`.

Availability

Available in OS X v10.0 and later.

See Also

– `setPropertyList:forType:` (page 26)

Related Sample Code

DictionaryController

iSpend

Sketch

Sketch+Accessibility

SourceView

Declared in

`NSPasteboard.h`

readFileContentsType:toFile:

Reads data representing a file's contents from the receiver and writes it to the specified file.

```
- (NSString *)readFileContentsType:(NSString *)type toFile:(NSString *)filename
```

Parameters

`type`

The pasteboard data type to read. You should generally specify a value for this parameter. If you specify `nil`, the filename extension (in combination with the `NSCreateFileContentsPboardType` function) is used to determine the type.

`filename`

The file to receive the pasteboard data.

Return Value

The name of the file into which the data was actually written.

Discussion

Data of any file contents type should only be read using this method. If data matching the specified type is not found on the pasteboard, data of type `NSFileContentsPboardType` is requested.

> **OS X v10.6 and later:** In OS X v10.5 and earlier, the file contents pboard type allowed you to synthesize a pboard type for a file's contents based on the file's extension. In OS X v10.5 and later, using the UTI of a file to represent its contents now replaces this functionality.

Special Considerations

You must send an `availableTypeFromArray:` (page 14) or `types` (page 29) message before invoking `readFileContentsType:toFile:`.

Availability

Available in OS X v10.0 and later.

See Also

– `writeFileContents:` (page 30)

Declared in

`NSPasteboard.h`

readFileWrapper

Reads data representing a file's contents from the receiver and returns it as a file wrapper.

```
- (NSFileWrapper *)readFileWrapper
```

Return Value

A file wrapper containing the pasteboard data, or `nil` if the receiver contained no data of type `NSFileContentsPboardType`.

Discussion

In OS X v10.5 and earlier, the file contents pboard type allowed you to synthesize a pboard type for a file's contents based on the file's extension. In OS X v10.5 and later, using the UTI of a file to represent its contents now replaces this functionality.

Availability

Available in OS X v10.0 and later.

Declared in

`NSPasteboard.h`

readObjectsForClasses:options:

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

```
- (NSArray *)readObjectsForClasses:(NSArray *)classArray
options:(NSDictionary *)options
```

Parameters

`classArray`

An array of class objects.

The classes should appear in the array in the order the preferred order of representation. Classes in the array must conform to the `NSPasteboardReading` protocol.

`options`

A dictionary that specifies options to refine the search for pasteboard items, for example to restrict the search to file URLs with particular content types. For valid dictionary keys, see "Pasteboard Reading Options" (page 39).

Return Value

An array containing the best match (if any) for each of the items on the receiver that can be represented by a class specified in `classArray`. Returns `nil` if there is an error in retrieving the requested items from the pasteboard, or if no objects of the specified classes can be created.

Discussion

Classes in `classArray` must implement the `NSPasteboardReading` protocol. Cocoa classes that implement this protocol include `NSImage`, `NSString`, `NSURL`, `NSColor`, `NSAttributedString`, and `NSPasteboardItem`. For every item on the pasteboard, each class in the provided array will be queried for the types it can read using `readableTypesForPasteboard:`. An instance is created of the first class found in the provided array whose readable types match a conforming type contained in that pasteboard item. Any instances that could be created from pasteboard item data is returned to the caller. Additional options, such as restricting the search to file URLs with particular content types, can be specified with an options dictionary.

Only objects of the requested classes are returned. You can always ensure to receive one object per item on the pasteboard by including the NSPasteboardItem class in the array of classes.

Consider the following example: there are five items on the pasteboard, two contain TIFF data, two contain RTF data, one contains a private data type. The following table shows what objects you get back in the returned array for different classes in `classArray`

Classes in `classArray`	Returned objects
`NSImage`	Two `NSImage` objects.
`NSAttributedString`	Two `NSAttributedString` objects.
`NSImage`, `NSAttributedString`	Two `NSImage` objects, and two `NSAttributedString` objects.

Classes in `classArray`	Returned objects
`NSImage`, `NSAttributedString`, `NSPasteboardItem`	Two `NSImage` objects, two `NSAttributedString` objects, and one `NSPasteboardItem` object containing the private data type.

Availability

Available in OS X v10.6 and later.

See Also

– `canReadObjectForClasses:options:` (page 15)
– `canReadItemWithDataConformingToTypes:` (page 15)
– `pasteboardItems` (page 21)

Related Sample Code

LightTable

MultiPhotoFrame

MyMediaPlayList

Declared in

`NSPasteboard.h`

releaseGlobally

Releases the receiver's resources in the pasteboard server.

```
- (void)releaseGlobally
```

Discussion

After this method is invoked, no other application can use the receiver.

Special Considerations

A temporary, privately named pasteboard can be released this way when it is no longer needed, but a standard pasteboard should never be released globally.

Availability

Available in OS X v10.0 and later.

Declared in

`NSPasteboard.h`

setData:forType:

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

```
- (BOOL)setData:(NSData *)data forType:(NSString *)dataType
```

Parameters

`data`

The data to write to the pasteboard.

`dataType`

The type of data in the `data` parameter. The type must have been declared by a previous `declareTypes:owner:` (page 18) message.

Return Value

`YES` if the data was written successfully, otherwise `NO` if ownership of the pasteboard has changed. Any other error raises an `NSPasteboardCommunicationException`.

Availability

Available in OS X v10.0 and later.

See Also

- `dataForType:` (page 17)
- `setPropertyList:forType:` (page 26)
- `setString:forType:` (page 27)

Related Sample Code

CocoaDragAndDrop

DemoAssistant

Sketch

Sketch+Accessibility

Declared in

`NSPasteboard.h`

setPropertyList:forType:

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

```
- (BOOL)setPropertyList:(id)propertyList forType:(NSString *)dataType
```

Parameters

`propertyList`

The property list data to write to the pasteboard.

`dataType`

The type of property-list data in the `propertyList` parameter. The type must have been declared by a previous `declareTypes:owner:` (page 18) message.

Return Value

`YES` if the data was written successfully, otherwise `NO` if ownership of the pasteboard has changed. Any other error raises an `NSPasteboardCommunicationException`.

Discussion

This method invokes `setData:forType:` (page 26) with a serialized property list parameter.

Availability

Available in OS X v10.0 and later.

See Also

– `propertyListForType:` (page 21)
– `setString:forType:` (page 27)

Related Sample Code

ComplexBrowser

iSpend

TextEdit

ZipBrowser

Declared in

`NSPasteboard.h`

setString:forType:

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

```
- (BOOL)setString:(NSString *)string forType:(NSString *)dataType
```

Parameters

`string`

The string to write to the pasteboard.

`dataType`

The type of string data. The type must have been declared by a previous `declareTypes:owner:` (page 18) message.

Return Value

YES if the data was written successfully, otherwise NO if ownership of the pasteboard has changed. Any other error raises an NSPasteboardCommunicationException.

Discussion

This method invokes setData:forType: (page 26) to perform the write.

Availability

Available in OS X v10.0 and later.

See Also

- – stringForType: (page 28)
- – setData:forType: (page 26)
- – setPropertyList:forType: (page 26)

Related Sample Code

DemoAssistant

PhotoSearch

SimpleService

SpotlightFortunes

With and Without Bindings

Declared in

NSPasteboard.h

stringForType:

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

```
- (NSString *)stringForType:(NSString *)dataType
```

Parameters

dataType

The pasteboard data type to read.

Return Value

A concatenation of the strings for the specified type from all the items in the receiver that contain the type, or nil if none of the items contain strings of the specified type.

Discussion

This method invokes `dataForType:` to obtain the string. If the string cannot be obtained, `stringForType:` returns `nil`. See `dataForType:` (page 17) for a description of what will cause `nil` to be returned.

In OS X v10.6 and later, if the receiver contains multiple items that can provide string, RTF, or RTFD data, the text data from each item is returned as a combined result separated by newlines.

Special Considerations

You must send `types` (page 29) or `availableTypeFromArray:` (page 14) before invoking `stringForType:`.

Availability

Available in OS X v10.0 and later.

See Also

– `setString:forType:` (page 27)

Related Sample Code

iSpend

SpotlightFortunes

Declared in

`NSPasteboard.h`

types

Returns an array of the receiver's supported data types.

```
- (NSArray *)types
```

Return Value

An array of `NSString` objects containing the union of the types of data declared for all the pasteboard items on the receiver. The returned types are listed in the order they were declared.

Discussion

You must send a `types` or `availableTypeFromArray:` (page 14) message before reading any data from an `NSPasteboard` object. If you need to see if a type in the returned array matches a type string you have stored locally, use the `isEqualToString:` method to perform the comparison.

Availability

Available in OS X v10.0 and later.

See Also

– `declareTypes:owner:` (page 18)
– `dataForType:` (page 17)

Related Sample Code
iSpend

SimpleService

Declared in
`NSPasteboard.h`

writeFileContents:

Writes the contents of the specified file to the pasteboard.

```
- (BOOL)writeFileContents:(NSString *)filename
```

Parameters
`filename`

The name of the file to write to the pasteboard.

Return Value
`YES` if the data was successfully written, otherwise `NO`.

Discussion
Writes the contents of the file `filename` to the receiver and declares the data to be of type `NSFileContentsPboardType` and also of a type appropriate for the file's extension (as returned by the `NSCreateFileContentsPboardType` function when passed the files extension), if it has one.

Availability
Available in OS X v10.0 and later.

See Also
– `readFileContentsType:toFile:` (page 22)

Declared in
`NSPasteboard.h`

writeFileWrapper:

Writes the serialized contents of the specified file wrapper to the pasteboard.

```
- (BOOL)writeFileWrapper:(NSFileWrapper *)wrapper
```

Parameters

`wrapper`

The file wrapper to write to the pasteboard.

Return Value

`YES` if the data was successfully written, otherwise `NO`.

Discussion

Writes the serialized contents of the file wrapper `wrapper` to the receiver and declares the data to be of type `NSFileContentsPboardType` and also of a type appropriate for the file's extension (as returned by the `NSCreateFileContentsPboardType` function when passed the files extension), if it has one. If `wrapper` does not have a preferred filename, this method raises an exception.

Availability

Available in OS X v10.0 and later.

Declared in

`NSPasteboard.h`

writeObjects:

Writes an array of objects to the receiver.

```
- (BOOL)writeObjects:(NSArray *)objects
```

Parameters

`objects`

An array of objects that implement the `NSPasteboardWriting` protocol (including instances of `NSPasteboardItem`).

Return Value

`YES` if the array was successfully added, otherwise `NO`.

Availability

Available in OS X v10.6 and later.

Related Sample Code

IconCollection

MyMediaPlayList

Declared in

`NSPasteboard.h`

Constants

Pasteboard Names

The `NSPasteboard` class defines the following named pasteboards.

```
NSString *NSGeneralPboard;
NSString *NSFontPboard;
NSString *NSRulerPboard;
NSString *NSFindPboard;
NSString *NSDragPboard;
```

Constants

`NSGeneralPboard`

The pasteboard that's used for ordinary cut, copy, and paste operations.

This pasteboard holds the contents of the last selection that's been cut or copied.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSFontPboard`

The pasteboard that holds font and character information and supports Copy Font and Paste Font commands that may be implemented in a text editor.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSRulerPboard`

The pasteboard that holds information about paragraph formats in support of the Copy Ruler and Paste Ruler commands that may be implemented in a text editor.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSFindPboard`

The pasteboard that holds information about the current state of the active application's find panel.

This information permits users to enter a search string into the find panel, then switch to another application to conduct another search.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSDragPboard`

The pasteboard that stores data to be moved as the result of a drag operation.

For additional information on working with the drag pasteboard, see *Drag and Drop Programming Topics* .

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

Declared in

`AppKit/NSPasteboard.h`

Types for Standard Data (OS X v10.6 and later)

The `NSPasteboard` class uses the following constants to define UTIs for common pasteboard data types.

```
NSString *const NSPasteboardTypeString;
NSString *const NSPasteboardTypePDF;
NSString *const NSPasteboardTypeTIFF;
NSString *const NSPasteboardTypePNG;
NSString *const NSPasteboardTypeRTF;
NSString *const NSPasteboardTypeRTFD;
NSString *const NSPasteboardTypeHTML;
NSString *const NSPasteboardTypeTabularText;
NSString *const NSPasteboardTypeFont;
NSString *const NSPasteboardTypeRuler;
NSString *const NSPasteboardTypeColor;
NSString *const NSPasteboardTypeSound;
NSString *const NSPasteboardTypeMultipleTextSelection;
NSString *const NSPasteboardTypeFindPanelSearchOptions;
```

Constants

`NSPasteboardTypeString`

String data.

> **Important:** Apps that adopt App Sandbox cannot access files identified using the `NSPasteboardTypeString` pasteboard type. Instead, use an `NSURL` object, a bookmark, or a filename pasteboard type.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypePDF`

PDF data.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeTIFF`

Tag Image File Format (TIFF) data.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypePNG`

PNG image data.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeRTF`

Rich Text Format (RTF) data.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeRTFD`

RTFD formatted file contents.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeHTML`

HTML data.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeTabularText`

An `NSString` object containing tab-separated fields of text.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeFont`

Font and character information.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeRuler`

Paragraph formatting information.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeColor`

Color data (an `NSColor` object).

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeSound`

Sound data (an `NSSound` object).

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeMultipleTextSelection`

Multiple text selection.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeFindPanelSearchOptions`

Type for the Find panel metadata property list.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

Types for Standard Data (OS X v10.5 and earlier)

The `NSPasteboard` class uses the following common pasteboard data types.

```
NSString *NSStringPboardType;
NSString *NSFilenamesPboardType;
NSString *NSPostScriptPboardType;
NSString *NSTIFFPboardType;
NSString *NSRTFPboardType;
NSString *NSTabularTextPboardType;
NSString *NSFontPboardType;
NSString *NSRulerPboardType;
NSString *NSFileContentsPboardType;
NSString *NSColorPboardType;
NSString *NSRTFDPboardType;
NSString *NSHTMLPboardType;
NSString *NSPICTPboardType;
NSString *NSURLPboardType;
NSString *NSPDFPboardType;
NSString *NSVCardPboardType;
NSString *NSFilesPromisePboardType;
NSString *NSMultipleTextSelectionPboardType;
```

Constants

`NSColorPboardType`

`NSColor` data.

On OS X v10.6 and later, use `NSPasteboardTypeColor` (page 34) (and you read and write colors directly to and from the pasteboard).

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSFileContentsPboardType`

A representation of a file's contents.

The file contents pboard type allowed you to synthesize a pboard type for a file's contents based on the file's extension.

On OS X v10.6 and later, you should use the UTI of a file to represent its contents instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSFilenamesPboardType`

An array of `NSString` objects designating one or more filenames.

On OS X v10.6 and later, use `writeObjects:` (page 31) to write file URLs to the pasteboard.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSFontPboardType`

Font and character information.

On OS X v10.6 and later, use `NSPasteboardTypeFont` (page 34) instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSHTMLPboardType`

HTML (which an `NSTextView` object can read from, but not write to).

On OS X v10.6 and later, use `NSPasteboardTypeHTML` (page 34) instead.

Available in OS X v10.1 and later.

Declared in `NSPasteboard.h`.

`NSPDFPboardType`

PDF data.

On OS X v10.6 and later, use `NSPasteboardTypePDF` (page 33) instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSPICTPboardType`

QuickDraw picture data.

The PICT format was formally deprecated in OS X v10.4 along with QuickDraw. You should not be explicitly providing or looking for PICT data on the pasteboard.

To aid in this deprecation, if PICT is the only image type on the pasteboard, as is sometimes the case when copying images from 32-bit Carbon applications, a translated image type will be automatically reported and provided by `NSPasteboard`. The translated type is added to the types array ahead of PICT so that the deprecated PICT format is not the preferred format. In addition, when an application provides image data to `NSPasteboard`, the Carbon Pasteboard Manager will automatically make a PICT translation available to 32-bit Carbon applications.

Although `NSPICTPboardType`, and its UTI equivalent `kUTTypePICT`, will appear in a pasteboard's type array retrieved from the existing `NSPasteboard` API, it may cease to be reported in future releases.

Available in OS X v10.0 and later.

Deprecated in OS X v10.6.

Declared in `NSPasteboard.h`.

`NSPostScriptPboardType`

Encapsulated PostScript (EPS) code.

On OS X v10.6 and later, use `@"com.adobe.encapsulated-postscript"` instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSRulerPboardType`

Paragraph formatting information.

On OS X v10.6 and later, use `NSPasteboardTypeRuler` (page 34) instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSRTFPboardType`

Rich Text Format (RTF) data.

On OS X v10.6 and later, use `NSPasteboardTypeRTF` (page 34) instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSRTFDPboardType`

RTFD formatted file contents.

On OS X v10.6 and later, use `NSPasteboardTypeRTFD` (page 34) instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSStringPboardType`

`NSString` data. (Deprecated. In apps that adopt App Sandbox, use an `NSURL` object, a bookmark, or a filename pasteboard type instead. In a nonsandboxed app, you can also use the `NSPasteboardTypeString` (page 33) pasteboard type.)

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSTabularTextPboardType`

An `NSString` object containing tab-separated fields of text.

On OS X v10.6 and later, use `NSPasteboardTypeTabularText` (page 34) instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSTIFFPboardType`

Tag Image File Format (TIFF) data.

On OS X v10.6 and later, use `NSPasteboardTypeTIFF` (page 34) instead.

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSURLPboardType`

`NSURL` data for one file or resource.

On OS X v10.6 and later, use `writeObjects:` (page 31) to write URLs directly to the pasteboard instead.

On OS X v10.5 and earlier: to write an URL to a pasteboard you use `writeToPasteboard:` (NSURL); to get an URL from a pasteboard you use `URLFromPasteboard:` (NSURL).

Available in OS X v10.0 and later.

Declared in `NSPasteboard.h`.

`NSVCardPboardType`

VCard data.

On OS X v10.6 and later, use `(NSString *)kUTTypeVCard` instead.

Available in OS X v10.2 and later.

Declared in `NSPasteboard.h`.

`NSFilesPromisePboardType`

Promised files.

On OS X v10.6 and later, use `(NSString *)kPasteboardTypeFileURLPromise` instead.

For information on promised files, see “Dragging Files” in *Drag and Drop Programming Topics* .

Available in OS X v10.2 and later.

Declared in `NSPasteboard.h`.

`NSInkTextPboardType`

Ink text data.

On OS X v10.6 and later, use `(NSString *)kUTTypeInkText` instead.

For information on ink text objects, see *Using Ink Services in Your Application* .

Available in OS X v10.4 and later.

Declared in `NSPasteboard.h`.

`NSMultipleTextSelectionPboardType`

Multiple text selection.

On OS X v10.6 and later, use `NSPasteboardTypeMultipleTextSelection` (page 35) instead.

Available in OS X v10.5 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardTypeTextFinderOptions`

Type for the Find panel metadata property list..

Available in OS X v10.7 and later.

Declared in `NSPasteboard.h`.

Discussion

See also `NSSoundPboardType` (NSSound).

Version Notes

Pboard types will be deprecated in a future release. On OS X 10.6 and later you should replace any use of pboard types with UTIs, including the constants described in "Types for Standard Data (OS X v10.6 and later)" (page 33).

Pasteboard Reading Options

These options can be used for both the `readObjectsForClasses:options:` (page 23) and `canReadObjectForClasses:options:` (page 15) methods, unless otherwise specified. The currently available options allow for customization of how URLS are read from the pasteboard.

```
NSString *NSPasteboardURLReadingFileURLsOnlyKey;
NSString *NSPasteboardURLReadingContentsConformToTypesKey;
```

Constants

`NSPasteboardURLReadingFileURLsOnlyKey`

Option for reading URLs to restrict the results to file URLs only.

The value for this key is an `NSNumber` object with a boolean value.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

`NSPasteboardURLReadingContentsConformToTypesKey`

Option for reading URLs to restrict the results to URLs with contents that conform to any of the provided UTI types.

If the content type of a URL cannot be determined, it will not be considered to match. The value for this key is an array of UTI type strings.

Available in OS X v10.6 and later.

Declared in `NSPasteboard.h`.

Document Revision History

This table describes the changes to *NSPasteboard Class Reference* .

Date	Notes
2012-05-14	Updated with App Sandbox tweaks.
2011-08-02	Updated for OS X v10.7.
2009-11-17	Moved pasteboard owner methods to NSPasteboardOwner Protocol Reference.
2009-07-20	Corrected deprecation information for pasteboard type constants.
2009-05-16	Describes new API introduced in OS X v10.6.
2008-10-15	Clarified behavior of pasteboardChangedOwner:.
2008-02-08	Enhanced discussions of color and URL pasteboard types.
2007-01-04	Updated to include new API introduced in OS X v10.5.
2006-05-23	Enhanced description of availableTypeFromArray:.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.