
UIPasteboard Class Reference

Data Management



2010-08-03



Apple Inc.
© 2010 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Apple, the Apple logo, iPhone, and Objective-C are trademarks of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Simultaneously published in the United States and Canada.

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.

Contents

UIPasteboard Class Reference 5

Overview	5
Tasks	6
Getting and Removing Pasteboards	6
Getting and Setting Pasteboard Attributes	6
Determining Types of Single Pasteboard Items	7
Getting and Setting Single Pasteboard Items	7
Determining the Types of Multiple Pasteboard Items	7
Getting and Setting Multiple Pasteboard Items	7
Getting and Setting Pasteboard Items of Standard Data Types	8
Properties	8
changeCount	8
color	9
colors	9
image	9
images	10
items	10
name	10
numberOfItems	11
persistent	11
string	11
strings	12
URL	12
URLs	13
Class Methods	13
generalPasteboard	13
pasteboardWithName:create:	13
pasteboardWithUniqueName	14
removePasteboardWithName:	15
Instance Methods	15
addItem:	15
containsPasteboardTypes:	16
containsPasteboardTypes:itemSet:	16
dataForPasteboardType:	17
dataForPasteboardType:itemSet:	17
itemSetWithPasteboardTypes:	18
pasteboardTypes	18
pasteboardTypesForItemSet:	19
setData:forPasteboardType:	19
setValue:forPasteboardType:	20
valueForPasteboardType:	20

- valuesForPasteboardType:itemSet: 21
- Constants 22
 - Pasteboard Names 22
 - Data Type Extensions 22
 - UserInfo Dictionary Keys 23
- Notifications 24
 - UIPasteboardChangedNotification 24
 - UIPasteboardRemovedNotification 24

Document Revision History 25

UIPasteboard Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 3.0 and later.
Companion guide	Device Features Programming Guide
Declared in	UIPasteboard.h

Overview

The `UIPasteboard` class enables an application to share data within the application or with another application using system-wide or application-specific pasteboards.

Typically, an object in the application writes data to a pasteboard when the user requests a copy or cut operation on a selection in the user interface. Another object in the same or different application then reads that data from the pasteboard and presents it to the user at a new location; this usually happens when the user requests a paste operation.

A pasteboard is a named region of memory where data can be shared. There are two system pasteboards: the General pasteboard ([UIPasteboardNameGeneral](#) (page 22)) and the Find pasteboard ([UIPasteboardNameFind](#) (page 22)). You can use the General pasteboard for copy-paste operations involving any kind of data; the Find pasteboard, which is used in search operations, holds the most recent string value in the search bar. Applications can also create pasteboards for their own use or for use by a family of related applications. Pasteboards must be identified by a unique names. You may also mark an application pasteboard as persistent, so that it continues to exist past the termination of the application and past system reboots. System pasteboards are persistent by default.

When you write an object to a pasteboard, it is stored as a pasteboard item. A pasteboard item is one or more key-value pairs where the key is a string that identifies the representation type of the value. Having multiple representation types per pasteboard item makes it more possible for one application to share data with another application without having to know specific capabilities of that application. For example, the source application could write the same image to the pasteboard in PNG, JPEG, and GIF data formats. If the receiving application can only handle GIF images, it can still obtain the pasteboard data.

A Uniform Type Identifier (UTI) is frequently used for a representation type (sometimes called a pasteboard type). For example, you could use `kUTTypeJPEG` (a constant for `public.jpeg`) as a representation type for JPEG data. However, applications are free to use any string they want for a representation type; however, for application-specific data types, it is recommended that you use reverse-DNS notation to ensure the uniqueness of the type (for example, `com.myCompany.myApp.myType`).

Note: For a discussion of Uniform Type Identifiers and a list of common ones, see *Uniform Type Identifiers Overview*.

UIPasteboard provides methods for reading and writing single pasteboard items at a time as well as multiple pasteboard items. The data written and read can be in two general forms. If the data to be written is a property-list object or can be converted to such an object, use a method such as [setValue:forPasteboardType:](#) (page 20) to write it to the pasteboard. If the data is binary (say, image data) or can't be converted to a property-list type, you would use the [setData:forPasteboardType:](#) (page 19) to write it to the pasteboard. For UIPasteboard, the classes of the property-list objects are NSString, NSArray, NSDictionary, NSDate, NSNumber and NSURL. The class also provides convenience methods for writing and reading strings, images, URLs, and colors to and from single or multiple pasteboard items.

Although UIPasteboard is central to copy-paste operations, several other UIKit classes and protocols are used in these operations as well:

- UINavigationController — Displays a menu with Copy, Cut, Paste, Select, and Select All commands above or below the selection.
- UIResponder — Responders implement the `canPerformAction:withSender:` to enable or disable commands in the above-mentioned menu based on the current context.
- UIResponderStandardEditActions — Responders implement methods declared in this informal protocol to handle the chosen menu commands (for example, `copy:` and `paste:`).

An application that implements copy-paste usually has to handle the management and presentation of selections in its user interface. It must also coordinate the addition and removal of items via paste and cut operations with its data model.

Tasks

Getting and Removing Pasteboards

- + [generalPasteboard](#) (page 13)
Returns the general pasteboard, which is used for general copy-paste operations
- + [pasteboardWithName:create:](#) (page 13)
Returns a pasteboard identified by name, optionally creating it if it doesn't exist.
- + [pasteboardWithUniqueName](#) (page 14)
Returns an application pasteboard identified by a unique system-generated name.
- + [removePasteboardWithName:](#) (page 15)
Invalidates the designated application pasteboard.

Getting and Setting Pasteboard Attributes

- [name](#) (page 10) *property*
The name of the pasteboard. (read-only)

[persistent](#) (page 11) *property*

A Boolean value that indicates whether the pasteboard is persistent.

[changeCount](#) (page 8) *property*

The number of times the pasteboard's contents have changed. (read-only)

Determining Types of Single Pasteboard Items

- [pasteboardTypes](#) (page 18)
Returns the types of the first item on the pasteboard.
- [containsPasteboardTypes:](#) (page 16)
Returns whether the pasteboard holds data of the specified representation type.

Getting and Setting Single Pasteboard Items

- [dataForPasteboardType:](#) (page 17)
Returns the data in the pasteboard for the given representation type.
- [valueForPasteboardType:](#) (page 20)
Returns an object in the pasteboard for the given representation type.
- [setData:forPasteboardType:](#) (page 19)
Puts data in the pasteboard for the specified representation type.
- [setValue:forPasteboardType:](#) (page 20)
Puts a property-list object in the pasteboard for the specified representation type.

Determining the Types of Multiple Pasteboard Items

- [numberOfItems](#) (page 11) *property*
Returns the number of items in the pasteboard (read-only)
- [pasteboardTypesForItemSet:](#) (page 19)
Returns an array of representation types for each specified pasteboard item.
- [itemSetWithPasteboardTypes:](#) (page 18)
Returns an index set identifying pasteboard items having the specified representation types.
- [containsPasteboardTypes:inItemSet:](#) (page 16)
Returns whether the specified pasteboard items contain data of the given representation types.

Getting and Setting Multiple Pasteboard Items

- [items](#) (page 10) *property*
The pasteboard items on the pasteboard.
- [dataForPasteboardType:inItemSet:](#) (page 17)
Returns the data objects in the indicated pasteboard items that have the given representation type.
- [valuesForPasteboardType:inItemSet:](#) (page 21)
Returns the property-list objects in the indicated pasteboard items that have the given representation type.

- `addItem`: (page 15)
Appends pasteboard items to the current contents of the pasteboard.

Getting and Setting Pasteboard Items of Standard Data Types

- `string` (page 11) *property*
The string value of the first pasteboard item.
- `strings` (page 12) *property*
An array of strings in all pasteboard items.
- `image` (page 9) *property*
The image object of the first pasteboard item.
- `images` (page 10) *property*
An array of image objects in all pasteboard items.
- `URL` (page 12) *property*
The URL object of the first pasteboard item.
- `URLs` (page 13) *property*
An array of URL objects in all pasteboard items.
- `color` (page 9) *property*
The color object of the first pasteboard item.
- `colors` (page 9) *property*
An array of color objects in all pasteboard items.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

changeCount

The number of times the pasteboard’s contents have changed. (read-only)

```
@property(readonly, nonatomic) NSInteger changeCount
```

Discussion

Whenever the contents of a pasteboard changes—specifically, when pasteboard items are added, modified, or removed—`UIPasteboard` increments the value of this property. After it increments the change count, `UIPasteboard` posts the notifications named `UIPasteboardChangedNotification` (page 24) (for additions and modifications) and `UIPasteboardRemovedNotification` (page 24) (for removals). These notifications include (in the `userInfo` dictionary) the types of the pasteboard items added or removed. Because `UIPasteboard` waits until the end of the current event loop before incrementing the change count, notifications can be batched. The class also updates the change count when an application reactivates and another application has changed the pasteboard contents. When users restart a device, the change count is reset to zero.

Availability

Available in iOS 3.0 and later.

Declared In

UIPasteboard.h

color

The color object of the first pasteboard item.

```
@property(nonatomic, copy) UIColor *color
```

Discussion

The value stored in this property is a `UIColor` object. The associated array of representation types is [UIPasteboardTypeListColor](#) (page 23), which includes type `. Setting this property replaces all current items in the pasteboard with the new item. If the first item has no value of the indicated type, nil is returned.`

Availability

Available in iOS 3.0 and later.

Declared In

UIPasteboard.h

colors

An array of color objects in all pasteboard items.

```
@property(nonatomic, copy) NSArray *colors
```

Discussion

The value stored in this property is an array of `UIColor` objects. The associated array of representation types is [UIPasteboardTypeListColor](#) (page 23), which includes type `. Setting this property replaces all current items in the pasteboard with the new items. The returned array may have fewer objects than the number of pasteboard items; this happens if a pasteboard item does not have a value of the indicated type.`

Availability

Available in iOS 3.0 and later.

Declared In

UIPasteboard.h

image

The image object of the first pasteboard item.

```
@property(nonatomic, copy) UIImage *image
```

Discussion

The value stored in this property is a `UIImage` object. The associated array of representation types is [UIPasteboardTypeListImage](#) (page 23), which includes types `kUTTypePNG` and `kUTTypeJPEG`. Setting this property replaces all current items in the pasteboard with the new item. If the first item has no value of the indicated type, `nil` is returned.

Availability

Available in iOS 3.0 and later.

Declared In

UIPasteboard.h

images

An array of image objects in all pasteboard items.

```
@property(n nonatomic, copy) NSArray *images
```

Discussion

The value stored in this property is an array of `UIImage` objects. The associated array of representation types is [UIPasteboardTypeListImage](#) (page 23), which includes types `kUTTypePNG` and `kUTTypeJPEG`. Setting this property replaces all current items in the pasteboard with the new items. The returned array may have fewer objects than the number of pasteboard items; this happens if a pasteboard item does not have a value of the indicated type.

Availability

Available in iOS 3.0 and later.

Declared In

UIPasteboard.h

items

The pasteboard items on the pasteboard.

```
@property(n nonatomic, copy) NSArray *items
```

Discussion

The value of the property is an array of dictionaries. Each dictionary represents a pasteboard item, with the key being the representation type and the value the data object or property-list object associated with that type. Setting this property replaces all of the current pasteboard items.

Availability

Available in iOS 3.0 and later.

See Also

- [addItem:](#) (page 15)

Declared In

UIPasteboard.h

name

The name of the pasteboard. (read-only)

```
@property(readonly, nonatomic) NSString *name
```

Discussion

Names of application pasteboard objects should be unique across installed applications. If the object is a system pasteboard, this property returns one of the constants described in ["Pasteboard Names"](#) (page 22).

Availability

Available in iOS 3.0 and later.

See Also

+ [pasteboardWithName:create:](#) (page 13)

+ [pasteboardWithUniqueName](#) (page 14)

Declared In

UIPasteboard.h

numberOfItems

Returns the number of items in the pasteboard (read-only)

```
@property(readonly, nonatomic) NSInteger numberOfItems
```

Availability

Available in iOS 3.0 and later.

Declared In

UIPasteboard.h

persistent

A Boolean value that indicates whether the pasteboard is persistent.

```
@property(getter=isPersistent, nonatomic) BOOL persistent
```

Discussion

When a pasteboard is persistent, it continues to exist past application terminations and across system reboots. Application pasteboards that are not persistent only last until the owning (creating) application quits. System pasteboards are persistent. Application pasteboards by default are not persistent. A persistent application pasteboard is removed when the application that created it is uninstalled.

Availability

Available in iOS 3.0 and later.

See Also

+ [pasteboardWithName:create:](#) (page 13)

+ [pasteboardWithUniqueName](#) (page 14)

Declared In

UIPasteboard.h

string

The string value of the first pasteboard item.

```
@property(n nonatomic, copy) NSString *string
```

Discussion

The value stored in this property is an `NSString` object. The associated array of representation types is [UIPasteboardTypeListString](#) (page 22), which includes type `kUTTypeUTF8PlainText`. Setting this property replaces all current items in the pasteboard with the new item. If the first item has no value of the indicated type, `nil` is returned.

Availability

Available in iOS 3.0 and later.

Declared In

`UIPasteboard.h`

strings

An array of strings in all pasteboard items.

```
@property(n nonatomic, copy) NSArray *strings
```

Discussion

The value stored in this property is an array of `NSString` objects. The associated array of representation types is [UIPasteboardTypeListString](#) (page 22), which includes type `kUTTypeUTF8PlainText`. Setting this property replaces all current items in the pasteboard with the new items. The returned array may have fewer objects than the number of pasteboard items; this happens if a pasteboard item does not have a value of the indicated type.

Availability

Available in iOS 3.0 and later.

Declared In

`UIPasteboard.h`

URL

The URL object of the first pasteboard item.

```
@property(n nonatomic, copy) NSURL *URL
```

Discussion

The value stored in this property is an `NSURL` object. The associated array of representation types is [UIPasteboardTypeListURL](#) (page 23), which includes type `kUTTypeURL`. Setting this property replaces all current items in the pasteboard with the new item. If the first item has no value of the indicated type, `nil` is returned.

Availability

Available in iOS 3.0 and later.

Declared In

`UIPasteboard.h`

URLs

An array of URL objects in all pasteboard items.

```
@property(n nonatomic, copy) NSArray *URLs
```

Discussion

The value stored in this property is an array of `NSURL` objects. The associated array of representation types is [UIPasteboardTypeListURL](#) (page 23), which includes type `kUTTypeURL`. Setting this property replaces all current items in the pasteboard with the new items. The returned array may have fewer objects than the number of pasteboard items; this happens if a pasteboard item does not have a value of the indicated type.

Availability

Available in iOS 3.0 and later.

Declared In

UIPasteboard.h

Class Methods

generalPasteboard

Returns the general pasteboard, which is used for general copy-paste operations

```
+ (UIPasteboard *)generalPasteboard
```

Return Value

A shared system pasteboard object with the name of [UIPasteboardNameGeneral](#) (page 22).

Discussion

You may use the general pasteboard for copying and pasting text, images, URLs, colors, and other data within an application or between applications. The general pasteboard is persistent across device restarts and application uninstalls.

Availability

Available in iOS 3.0 and later.

See Also

- + [pasteboardWithName:create:](#) (page 13)
- + [pasteboardWithUniqueName](#) (page 14)
- + [removePasteboardWithName:](#) (page 15)

Declared In

UIPasteboard.h

pasteboardWithName:create:

Returns a pasteboard identified by name, optionally creating it if it doesn't exist.

```
+ (UIPasteboard *)pasteboardWithName:(NSString *)pasteboardNamecreate:(BOOL)create
```

Parameters*pasteboardName*

A string or string constant that identifies (or should identify) the pasteboard. Specify `nil` if you want `UIPasteboard` to create a pasteboard with a unique name.

create

A Boolean value that indicates whether the pasteboard should be created if it doesn't already exist. Specify `NO` for system pasteboards or if you want an existing application pasteboard.

Return Value

A pasteboard object that can be used for transferring data within and application or between applications.

Discussion

You call this method to obtain the `UIPasteboardNameFind` (page 22) pasteboard and to create custom application pasteboards. (You may also use it to obtain the general pasteboard, but `generalPasteboard` (page 13) exists for that purpose.) If you create a pasteboard for your application, the name should a unique string to prevent possible name collisions with other applications' pasteboards; for this, use of reverse DNS notation (for example, `com.mycompany.myapp.pboard`) is recommended. Application pasteboards returned by this method are not persistent, existing only until the application quits. To make them persistent, set the `persistent` (page 11) property to `YES`.

Availability

Available in iOS 3.0 and later.

See Also

- + [pasteboardWithUniqueName](#) (page 14)
- + [removePasteboardWithName:](#) (page 15)

Declared In

`UIPasteboard.h`

pasteboardWithUniqueName

Returns an application pasteboard identified by a unique system-generated name.

```
+ (UIPasteboard *)pasteboardWithUniqueName
```

Return Value

An application pasteboard object with a unique name.

Discussion

Obtain the value of the `name` (page 10) property to discover the name of the returned pasteboard. Application pasteboards returned by this method are not persistent, existing only until the application quits. To make them persistent, set the `persistent` (page 11) property to `YES`. Calling this method is equivalent to calling `pasteboardWithName:create:` (page 13) with the first parameter set to `nil` and the second set to `YES`.

Availability

Available in iOS 3.0 and later.

See Also

- + [generalPasteboard](#) (page 13)
- + [removePasteboardWithName:](#) (page 15)

Declared In

UIPasteboard.h

removePasteboardWithName:

Invalidates the designated application pasteboard.

```
+ (void)removePasteboardWithName:(NSString *)pasteboardName
```

Parameters*pasteboardName*

The name of the pasteboard to be invalidated.

Discussion

Invalidation of an application pasteboard frees up all resources used by it. Once a pasteboard is invalidated, you cannot use the it; `UIPasteboard` ignores any calls to it. The method has no effect if called with the name of a system pasteboard.

Availability

Available in iOS 3.0 and later.

See Also

+ [pasteboardWithName:create:](#) (page 13)

+ [pasteboardWithUniqueName](#) (page 14)

Declared In

UIPasteboard.h

Instance Methods

addItems:

Appends pasteboard items to the current contents of the pasteboard.

```
- (void)addItems:(NSArray *)items
```

Parameters*items*

An array of dictionaries. Each dictionary represents a pasteboard item, with the key being the representation type and the value the data object or property-list object associated with that type.

Availability

Available in iOS 3.0 and later.

See Also

[@property items](#) (page 10)

Declared In

UIPasteboard.h

containsPasteboardTypes:

Returns whether the pasteboard holds data of the specified representation type.

```
- (BOOL)containsPasteboardTypes:(NSArray *)pasteboardTypes
```

Parameters

pasteboardTypes

An array of strings. Each string should identify a representation of the pasteboard item that the pasteboard reader can handle. These string are frequently UTIs. See the class description for more information about pasteboard item types.

Return Value

YES if the pasteboard item holds data of the indicated representation type, otherwise NO.

Discussion

This method works on the first item in the pasteboard. If there are other items, it ignores them. You can use this method when enabling or disabling the Paste menu command.

Availability

Available in iOS 3.0 and later.

See Also

- [pasteboardTypes](#) (page 18)

Declared In

UIPasteboard.h

containsPasteboardTypes:inItemSet:

Returns whether the specified pasteboard items contain data of the given representation types.

```
- (BOOL)containsPasteboardTypes:(NSArray *)pasteboardTypes inItemSet:(NSIndexSet *)itemSet
```

Parameters

pasteboardTypes

An array of strings, with each string identifying a representation type. Typically you use UTIs as pasteboard types.

itemSet

An index set with each integer value identifying a pasteboard item positionally in the pasteboard. Pass in `nil` to request all pasteboard items.

Return Value

YES if the pasteboard items identified by *itemSet* have data corresponding to the representation types specified by *pasteboardTypes*; otherwise, returns NO.

Availability

Available in iOS 3.0 and later.

See Also

- [@property numberOfItems](#) (page 11)
- [pasteboardTypesForItemSet:](#) (page 19)
- [itemSetWithPasteboardTypes:](#) (page 18)

Declared In

UIPasteboard.h

dataForPasteboardType:

Returns the data in the pasteboard for the given representation type.

```
- (NSData *)dataForPasteboardType:(NSString *)pasteboardType
```

Parameters*pasteboardType*

A string identifying a representation type of a pasteboard item.

Return ValueA data object or `nil` if there is no data in the pasteboard of the given type.**Discussion**

The returned object often holds raw (binary) data, such as image data. This method works on the first item in the pasteboard. If there are other items, it ignores them.

Availability

Available in iOS 3.0 and later.

See Also

- [valueForPasteboardType:](#) (page 20)
- [setData:forPasteboardType:](#) (page 19)

Declared In

UIPasteboard.h

dataForPasteboardType:inItemSet:

Returns the data objects in the indicated pasteboard items that have the given representation type.

```
- (NSArray *)dataForPasteboardType:(NSString *)pasteboardType inItemSet:(NSIndexSet *)itemSet
```

Parameters*pasteboardType*

A string identifying a representation type. Typically this is a UTI.

itemSet

An index set with each integer value identifying a pasteboard item positionally in the pasteboard. Pass in `nil` to request all pasteboard items.

Return Value

An array of `NSData` objects or, if a requested pasteboard item has no data of the the type indicated by *pasteboardType*, a `NSNull` object.

Availability

Available in iOS 3.0 and later.

See Also

- [valuesForPasteboardType:inItemSet:](#) (page 21)

[@property items](#) (page 10)

Declared In

UIPasteboard.h

itemSetWithPasteboardTypes:

Returns an index set identifying pasteboard items having the specified representation types.

```
- (NSIndexSet *)itemSetWithPasteboardTypes:(NSArray *)pasteboardTypes
```

Parameters

pasteboardTypes

An array of strings, with each string identifying a representation type. Typically you use UTIs as pasteboard types.

Return Value

An index set with each integer positionally identifying a pasteboard item that has one of the representation types specified in *pasteboardTypes*.

Discussion

You can pass the index set returned in this method in a call to [dataForPasteboardType:inItemSet:](#) (page 17) or [valuesForPasteboardType:inItemSet:](#) (page 21) to get the data in the indicated pasteboard items.

Availability

Available in iOS 3.0 and later.

See Also

- [@property numberOfItems](#) (page 11)
- [pasteboardTypesForItemSet:](#) (page 19)
- [containsPasteboardTypes:inItemSet:](#) (page 16)

Declared In

UIPasteboard.h

pasteboardTypes

Returns the types of the first item on the pasteboard.

```
- (NSArray *)pasteboardTypes
```

Return Value

An array of strings indicating the representation types of the first item on the pasteboard.

Discussion

A type is frequently, but not necessarily, a UTI (Uniform Type Identifier). It identifies a representation of the data on the pasteboard. For example, a pasteboard item could hold image data under `public.png` and `public.tiff` representations. Applications can define their own types for custom data such as `com.mycompany.myapp.mytype`; however, in this case, only those applications that know of the type could understand the data written to the pasteboard.

With this method, you can determine if the pasteboard holds data of a particular representation type by a line of code such as this:

```
BOOL pngOnPasteboard = [[pasteboard pasteboardTypes]
containsObject:@"public.png"];
```

Availability

Available in iOS 3.0 and later.

See Also

- [containsPasteboardTypes:](#) (page 16)

Declared In

UIPasteboard.h

pasteboardTypesForItemSet:

Returns an array of representation types for each specified pasteboard item.

```
- (NSArray *)pasteboardTypesForItemSet:(NSIndexSet *)itemSet
```

Parameters

itemSet

An index set with each integer value identifying a pasteboard item positionally in the pasteboard. Pass in *nil* to request all pasteboard items.

Return Value

An array of arrays, with each inner array holding the representation types for a particular pasteboard item.

Availability

Available in iOS 3.0 and later.

See Also

- [@property numberOfItems](#) (page 11)
- [itemSetWithPasteboardTypes:](#) (page 18)
- [containsPasteboardTypes:inItemSet:](#) (page 16)

Declared In

UIPasteboard.h

setData:forPasteboardType:

Puts data in the pasteboard for the specified representation type.

```
- (void)setData:(NSData *)data forPasteboardType:(NSString *)pasteboardType
```

Parameters

data

The data object to be written to the pasteboard.

pasteboardType

A string identifying the representation type of the pasteboard item. This is typically a UTI.

Discussion

Use this method to put data on the pasteboard when the data is not a standard property-list object—that is, an object of the `NSString`, `NSArray`, `NSDictionary`, `NSDate`, `NSNumber`, or `NSURL` class. (For property-list objects, use the [setValue:forPasteboardType:](#) (page 20) method.) For example, you could archive a graph of model objects and pass the resulting `NSData` object to a related application via a pasteboard using a custom pasteboard type. This method writes data for the first item in the pasteboard. Calling this method replaces any items currently in the pasteboard.

Availability

Available in iOS 3.0 and later.

See Also

- [dataForPasteboardType:](#) (page 17)

Declared In

UIPasteboard.h

setValue:forPasteboardType:

Puts a property-list object in the pasteboard for the specified representation type.

```
- (void)setValue:(id)value forPasteboardType:(NSString *)pasteboardType
```

Parameters

value

The property-list object to be written to the pasteboard.

pasteboardType

A string identifying the representation type of the pasteboard item. If the type is a UTI, it must be compatible with the class of *value*; otherwise, nothing is written to the pasteboard.

Discussion

Use this method to put an object on the pasteboard that is a standard property-list object—that is an object of the `NSString`, `NSArray`, `NSDictionary`, `NSDate`, `NSNumber`, or `NSURL` class. (For all other data, such as raw binary data, use the [setData:forPasteboardType:](#) (page 19) method.) This method writes the object as the first item in the pasteboard. Calling this method replaces any items currently in the pasteboard.

Availability

Available in iOS 3.0 and later.

See Also

- [valueForPasteboardType:](#) (page 20)

Declared In

UIPasteboard.h

valueForPasteboardType:

Returns an object in the pasteboard for the given representation type.

```
- (id)valueForPasteboardType:(NSString *)pasteboardType
```

Parameters*pasteboardType*

A string identifying a representation type of a pasteboard item.

Return ValueAn object that is an instance of the appropriate class based on *pasteboardType*, a property-list object, or an NSData object containing “raw” data.**Discussion**

This method attempts to return an object that is of a class type appropriate to the representation type, which typically is a UTI. For example, if the representation type is `kUTTypePlainText` (`public.plain-text`), the method returns an `NSString` object. If the method cannot determine the class type from the representation type, it returns the object as a generic property-list object. Property-list objects include `NSString`, `NSArray`, `NSDictionary`, `NSDate`, or `NSNumber` objects, with `NSURL` objects also as a possibility. If the method cannot decode the value as a property-list object, it returns the pasteboard item as an `NSData` object. This method works on the first item in the pasteboard. If there are other items, it ignores them.

Availability

Available in iOS 3.0 and later.

See Also

- [dataForPasteboardType:](#) (page 17)
- [setValue:forPasteboardType:](#) (page 20)

Declared In

UIPasteboard.h

valuesForPasteboardType:inItemSet:

Returns the property-list objects in the indicated pasteboard items that have the given representation type.

```
- (NSArray *)valuesForPasteboardType:(NSString *)pasteboardType inItemSet:(NSIndexSet *)itemSet
```

Parameters*pasteboardType*

A string identifying a representation type. Typically this is a UTI.

*itemSet*An index set with each integer value identifying a pasteboard item positionally in the pasteboard. Pass in `nil` to request all pasteboard items.**Return Value**An array of `NSData` objects or, if a requested pasteboard item has no data of the the type indicated by *pasteboardType*, a `NSNull` object.**Discussion**Returned objects are of one of the following classes: `NSString`, `NSArray`, `NSDictionary`, `NSDate`, `NSNumber`, or `NSURL`.**Availability**

Available in iOS 3.0 and later.

See Also

- [dataForPasteboardType:inItemSet:](#) (page 17)

[@property items](#) (page 10)**Declared In**

UIPasteboard.h

Constants

Pasteboard Names

Names identifying the system pasteboards.

```

UIKIT_EXTERN NSString *const UIPasteboardNameGeneral;
UIKIT_EXTERN NSString *const UIPasteboardNameFind;

```

Constants

UIPasteboardNameGeneral

The name identifying the General pasteboard, which is used for general copy-cut-paste operations.

Available in iOS 3.0 and later.

Declared in UIPasteboard.h.

UIPasteboardNameFind

The name identifying the Find pasteboard, which is used in search operations. In such operations, the most recent search string in the search bar is put in the Find pasteboard.

Available in iOS 3.0 and later.

Declared in UIPasteboard.h.

Discussion

You can access both system pasteboards by calling the class method [pasteboardWithName:create:](#) (page 13), specifying one of these constants as the first argument. You may also access the general pasteboard by calling the [generalPasteboard](#) (page 13) class method. Both system pasteboards are persistent across device restarts, application uninstalls, and restores.

Declared In

UIPasteboard.h

Data Type Extensions

Pasteboard-item representation types for a given object value.

```

UIKIT_EXTERN NSArray *UIPasteboardTypeListString;
UIKIT_EXTERN NSArray *UIPasteboardTypeListURL;
UIKIT_EXTERN NSArray *UIPasteboardTypeListImage;
UIKIT_EXTERN NSArray *UIPasteboardTypeListColor;

```

Constants

UIPasteboardTypeListString

An array of pasteboard-item representation types for strings, including [kUTTypeUTF8PlainText](#).

Related properties are [string](#) (page 11) and [strings](#) (page 12).

Available in iOS 3.0 and later.

Declared in UIPasteboard.h.

UIPasteboardTypeListURL

An array of pasteboard-item representation types for URLs, including `kUTTypeURL`. Related properties are [URL](#) (page 12) and [URLs](#) (page 13).

Available in iOS 3.0 and later.

Declared in `UIPasteboard.h`.

UIPasteboardTypeListImage

An array of pasteboard-item representation types for images, including `kUTTypePNG` and `kUTTypeJPEG`. Related properties are [image](#) (page 9) and [images](#) (page 10).

Available in iOS 3.0 and later.

Declared in `UIPasteboard.h`.

UIPasteboardTypeListColor

An array of pasteboard-item representation types for colors. Related properties are [color](#) (page 9) and [colors](#) (page 9).

Available in iOS 3.0 and later.

Declared in `UIPasteboard.h`.

Declared In

`UIPasteboard.h`

UserInfo Dictionary Keys

You use the following keys to access the representation types of pasteboard items that have been added to or removed from a pasteboard.

```
UIKIT_EXTERN NSString *const UIPasteboardChangedTypesAddedKey;
UIKIT_EXTERN NSString *const UIPasteboardChangedTypesRemovedKey;
```

Constants**UIPasteboardChangedTypesAddedKey**

With the notification named [UIPasteboardChangedNotification](#) (page 24), use this key to access the added representation types. These types are stored as an array in the notification's `userInfo` dictionary.

Available in iOS 3.0 and later.

Declared in `UIPasteboard.h`.

UIPasteboardChangedTypesRemovedKey

With the notification named [UIPasteboardChangedNotification](#) (page 24), use this key to access the removed representation types. These types are stored as an array in the notification's `userInfo` dictionary.

Available in iOS 3.0 and later.

Declared in `UIPasteboard.h`.

Declared In

`UIPasteboard.h`

Notifications

UIPasteboardChangedNotification

Posted by a pasteboard object when its contents change. This happens at the same time the pasteboard's change count ([changeCount](#) (page 8) property) is incremented. Changes include the addition, removal, and modification of pasteboard items. The `userInfo` dictionary may contain the representation types of pasteboard items that have been added to or removed from the pasteboard. See ["UserInfo Dictionary Keys"](#) (page 23) for the keys used to access these representation types. If pasteboard items have been modified but not added or removed, the `userInfo` dictionary is `nil`.

Availability

Available in iOS 3.0 and later.

Declared In

`UIPasteboard.h`

UIPasteboardRemovedNotification

Posted by a pasteboard object just before an application removes it. The removal class method is [`removePasteboardWithName:`](#) (page 15). There is no `userInfo` dictionary.

Availability

Available in iOS 3.0 and later.

Declared In

`UIPasteboard.h`

Document Revision History

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

Date	Notes
2010-08-03	Clarified description of pasteboard persistence and made other corrections.
2009-05-19	Made minor corrections.
2009-03-08	First version of the document that describes the class for transferring data within and between applications.

REVISION HISTORY

Document Revision History