
NSPortMessage Class Reference

[Cocoa](#) > [Interapplication Communication](#)





Apple Inc.
© 2007 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, Cocoa, Mac, Mac OS, and Objective-C are trademarks of Apple Inc., registered in the United States and other countries.

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

NSPortMessage Class Reference 5

- Overview 5
- Tasks 6
 - Creating Instances 6
 - Sending the Message 6
 - Getting the Components 6
 - Getting the Ports 6
 - Accessing the Message ID 6
- Instance Methods 6
 - components 6
 - initWithSendPort:receivePort:components: 7
 - msgid 7
 - receivePort 8
 - sendBeforeDate: 8
 - sendPort 9
 - setMsgid: 9

Document Revision History 11

Index 13

NSPortMessage Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Distributed Objects Programming Topics
Declared in	NSPortMessage.h

Overview

An `NSPortMessage` defines a low-level, operating system-independent type for inter-application (and inter-thread) messages. Port messages are used primarily by the distributed objects system. You should implement inter-application communication using distributed objects whenever possible and use `NSPortMessage` only when necessary.

An `NSPortMessage` object has three major parts: the send and receive ports, which are `NSPort` object that link the sender of the message to the receiver, and the components, which form the body of the message. The components are held as an `NSArray` object containing `NSData` and `NSPort` objects. `NSPortMessage`'s [sendBeforeDate:](#) (page 8) message sends the components out through the send port; any replies to the message arrive on the receive port. See the `NSPort` class specification for information on handling incoming messages.

An `NSPortMessage` instance can be initialized with a pair of `NSPort` objects and an array of components. A port message's body can contain only `NSPort` objects or `NSData` objects. In the distributed objects system the byte/character arrays are usually encoded `NSInvocation` objects that are being forwarded from a proxy to the corresponding real object.

An `NSPortMessage` object also maintains a message identifier, which can be used to indicate the class of a message, such as an Objective-C method invocation, a connection request, an error, and so on. Use the [setMsgid:](#) (page 9) and [msgid](#) (page 7) methods to access the identifier.

Tasks

Creating Instances

- [initWithSendPort:receivePort:components:](#) (page 7)
Initializes a newly allocated NSPortMessage object to send given data on a given port and to receive replies on another given port.

Sending the Message

- [sendBeforeDate:](#) (page 8)
Attempts to send the message before *aDate*, returning YES if successful or NO if the operation times out.

Getting the Components

- [components](#) (page 6)
Returns the data components of the receiver.

Getting the Ports

- [receivePort](#) (page 8)
For an outgoing message, returns the port on which replies to the receiver will arrive. For an incoming message, returns the port the receiver did arrive on.
- [sendPort](#) (page 9)
For an outgoing message, returns the port the receiver will send itself through. For an incoming message, returns the port replies to the receiver should be sent through.

Accessing the Message ID

- [setMsgid:](#) (page 9)
Sets the identifier for the receiver.
- [msgid](#) (page 7)
Returns the identifier for the receiver.

Instance Methods

components

Returns the data components of the receiver.

- (NSArray *)components

Return Value

The data components of the receiver. See “[Class Description](#)” (page 5) for more information.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSPortMessage.h

initWithSendPort:receivePort:components:

Initializes a newly allocated NSPortMessage object to send given data on a given port and to receiver replies on another given port.

```
- (id)initWithSendPort:(NSPort *)sendPort receivePort:(NSPort *)receivePort
  components:(NSArray *)components
```

Parameters

sendPort

The port on which the message is sent.

receivePort

The port on which replies to the message arrive.

components

The data to send in the message. *components* should contain only NSData and NSPort objects, and the contents of the NSData objects should be in network byte order.

Return Value

An NSPortMessage object initialized to send *components* on *sendPort* and to receiver replies on *receivePort*.

Discussion

An NSPortMessage object initialized with this method has a message identifier of 0.

This is the designated initializer for NSPortMessage.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setMsgid:](#) (page 9)

Declared In

NSPortMessage.h

msgid

Returns the identifier for the receiver.

```
- (uint32_t)msgid
```

Return Value

The identifier for the receiver.

Discussion

Cooperating applications can use this to define different types of messages, such as connection requests, RPCs, errors, and so on.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setMsgid](#): (page 9)

Declared In

NSPortMessage.h

receivePort

For an outgoing message, returns the port on which replies to the receiver will arrive. For an incoming message, returns the port the receiver did arrive on.

```
- (NSPort *)receivePort
```

Return Value

For an outgoing message, the port on which replies to the receiver will arrive. For an incoming message, the port the receiver did arrive on.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [sendPort](#) (page 9)

Declared In

NSPortMessage.h

sendBeforeDate:

Attempts to send the message before *aDate*, returning YES if successful or NO if the operation times out.

```
- (BOOL)sendBeforeDate:(NSDate *)aDate
```

Parameters

aDate

The instant before which the message should be sent.

Return Value

YES if the operation is successful, otherwise NO (for example, if the operation times out).

Discussion

If an error other than a time out occurs, this method could raise an `NSInvalidSendPortException`, `NSInvalidReceivePortException`, or an `NSPortSendException`, depending on the type of send port and the type of error.

If the message cannot be sent immediately, the sending thread blocks until either the message is sent or *date* is reached. Sent messages are queued to minimize blocking, but failure can occur if multiple messages are sent to a port faster than the port's owner can receive them, causing the queue to fill up. Therefore, select a value for *date* that provides enough time for the message to be processed before the next message is sent. See the `NSPort` class specification for information on receiving a port message.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSPortMessage.h`

sendPort

For an outgoing message, returns the port the receiver will send itself through. For an incoming message, returns the port replies to the receiver should be sent through.

- (`NSPort *`)sendPort

Return Value

For an outgoing message, the port the receiver will send itself through when it receives a [sendBeforeDate:](#) (page 8) message. For an incoming message, the port replies to the receiver should be sent through.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [receivePort](#) (page 8)

Declared In

`NSPortMessage.h`

setMsgid:

Sets the identifier for the receiver.

- (void)setMsgid:(`uint32_t`)*msgid*

Parameters

msgid

The identifier for the receiver.

Discussion

Cooperating applications can use this method to define different types of messages, such as connection requests, RPCs, errors, and so on.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [msgid](#) (page 7)

Declared In

NSPortMessage.h

Document Revision History

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

Date	Notes
2007-01-29	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

C

components [instance method 6](#)

I

initWithSendPort:receivePort:components:
[instance method 7](#)

M

msgid [instance method 7](#)

R

receivePort [instance method 8](#)

S

sendBeforeDate: [instance method 8](#)

sendPort [instance method 9](#)

setMsgid: [instance method 9](#)