NSPortMessage

Inherits From:	NSObject
Conforms To:	NSObject (NSObject)
Declared In:	Foundation/NSPortMessage.h

Class Description

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

An NSPortMessage has three major parts: the *send* and *receive ports*, which are NSPorts 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 of NSData and NSPort objects. NSPortMessage's **sendBeforeDate:** 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 can be initialized with a Mach message or with a pair of NSPorts and an NSArray of components. In either case, an NSPortMessage's body can contain only the types in the table below. The types listed under **initWithMachMessage:** are converted to the corresponding object type listed under **initWithSendPort:receivePort:components:**. In the distributed objects system the byte/character arrays are usually encoded NSInvocations that are being forwarded from a proxy to the corresponding real object.

initWithMachMessage:	**initWithSendPort:receivePort:components:**
port_t	NSPort object
MSG_TYPE_BYTE MSG_TYPE_CHAR MSG_TYPE_INTEGER_32	NSData object

An NSPortMessage 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:** and **msgid** methods to access the identifier.

Method Types

Creating instances	– initWithMachMessage: – initWithSendPort:receivePort:components:

Sending the message	– sendBeforeDate:
Getting the components	– components
Getting the ports	– receivePort – sendPort
Accessing the message ID	– setMsgid: – msgid

Instance Methods

components

– (NSArray *)**components**

Returns the data components of the NSPortMessage. See the class description for more information.

initWithMachMessage:

– (id)**initWithMachMessage:**(void *)*buffer*

Initializes a newly allocated NSPortMessage with *buffer*, which is a pointer to a Mach message (beginning with a **msg_header_t** structure) cast as a pointer to **void**. The send and receive ports, message identifier, and components are all extracted from *buffer*. Returns **self**.

See the class description for restrictions on the allowable types in the body of the Mach message.

initWithSendPort:receivePort:components:

– (id)**initWithSendPort:**(NSPort *)*sendPort*
receivePort:(NSPort *)*receivePort*
components:(NSArray *)*components*

Initializes a newly allocated NSPortMessage to send the data in *components* on *sendPort*. Replies to the message will arrive on *receivePort*. An NSPortMessage initialized with this method has a message identifier of 0. This is the designated initializer for the NSPortMessage Class. Returns **self**.

components should contain only NSData and NSPort objects, and the contents of the NSData objects should be in network byte order.

See also: – **setMsgid:**

msgid

– (unsigned int)**msgid**

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

See also: – **setMsgid:**

receivePort

– (NSPort *)**receivePort**

For an outgoing message, returns the NSPort that replies to the NSPortMessage will arrive on. For an incoming message, returns the NSPort that the NSPortMessage did arrive on.

See also: – **sendPort**

sendBeforeDate:

– (BOOL)**sendBeforeDate:**(NSDate *)*aDate*

Attempts to send the message before *aDate*, returning YES if successful, NO otherwise. See the NSPort class specification for information on receiving an NSPortMessage.

sendPort

– (NSPort *)**sendPort**

For an outgoing message, returns the NSPort that the NSPortMessage will send itself through when it receives a **sendBeforeDate:** message. For an incoming message, returns the NSPort that replies to the NSPortMessage should be sent through.

See also: – **receivePort**

setMsgid:

– (void)**setMsgid:**(unsigned int)*msgid*

Sets the identifier for the NSPortMessage to *msgid*. Cooperating applications can use this to define different types of messages, such as connection requests, RPCs, errors, and so on.

See also: – **msgid**