
NS+ NSPort

Inherits From:	NSObject
Conforms To:	NSCoding NSCopying NSObject (NSObject)
Declared In:	Foundation/NSPort.h

Class Description

An NSPort represents a communication channel to or from another NSPort, which typically resides in different thread or task. The distributed objects system uses NSPort objects to send NSPortMessages back and forth. You should implement interapplication communication using distributed objects whenever possible, and use NSPorts only when necessary.

Note: An NSPort is essentially the object form of a Mach port. To use NSPorts effectively you should be familiar with Mach ports, port access rights, and Mach messages.

When an NSPort receives an NSPortMessage, it forwards the message to its delegate in a **handleMachMessage:** or **handlePortMessage:** message. The delegate should implement only one of these methods to process the incoming message in whatever form desired. **handleMachMessage:** provides a message as a raw Mach message beginning with a `msg_header_t` structure. **handlePortMessage:** provides a message as an NSPortMessage object, which is an object-oriented wrapper for a Mach message.

NSPorts intended to receive incoming messages need to be added to an NSRunLoop. See the NSRunLoop class specification for more information.

Adopted Protocols

NSCoding	– encodeWithCoder: – initWithCoder:
NSCopying	– copyWithZone:

Method Types

Creating instances	+ port + portWithMachPort: – init – initWithMachPort:
Getting the Mach port	– machPort
Validation	– invalidate – isValid
Setting the delegate	– setDelegate: – delegate

Class Methods

port

+ (NSPort *)**port**

Creates and returns a new NSPort object capable of both sending and receiving messages.

portWithMachPort:

+ (NSPort *)**portWithMachPort:(int)machPort**

Returns an NSPort object that uses the Mach port *machPort* (which should be of type **port_t** cast to **int**). Creates the NSPort object if necessary. Depending on the access rights for *machPort*, the new NSPort may only be able to send messages.

Instance Methods

delegate

– (id)**delegate**

Returns the NSPort's delegate.

See also: – **setDelegate:**

init

– (id)**init**

Initializes a newly allocated NSPort object to be capable of both sending and receiving messages. Returns **self**.

initWithMachPort:

– (id)**initWithMachPort:(int)machPort**

Initializes a newly allocated NSPort object to use the Mach port *machPort* (which should be of type **port_t** cast to **int**). Depending on the access rights for *machPort*, the new NSPort may only be able to send messages. If an NSPort with *machPort* already exists, deallocates the receiver, then retains and returns the existing NSPort.

This method is the designated initializer for the NSPort class. Returns **self**.

invalidate

– (void)**invalidate**

Marks the NSPort as invalid and posts an NSPortDidBecomeInvalidNotification to the default notification center.

See also: – **isValid**

isValid

– (BOOL)**isValid**

Returns NO if the NSPort is known to be invalid, YES otherwise (an NSPort only notes that it has become invalid when it tries to send or receive a message). An NSPort becomes invalid when its underlying communication resource, which is operating-system dependent, is closed or damaged.

See also: – **invalidate**

machPort

– (int)**machPort**

Returns as an **int** the Mach port used by the NSPort. Cast this value to a **port_t** when using it with Mach system calls.

setDelegate:

– (void)**setDelegate:(id)anObject**

Sets the NSPort’s delegate to *anObject*. Doesn’t retain *anObject*.

See also: – **delegate**

Methods Implemented by the Delegate**handleMachMessage:**

– (void)**handleMachMessage:(void *)machMessage**

Processes *machMessage*, an incoming Mach message cast as a pointer to **void**. The delegate should interpret this data as a pointer to a Mach message beginning with a **msg_header_t** structure and should handle the message appropriately.

The delegate should implement only one of **handleMachMessage:** and **handlePortMessage:**.

handlePortMessage:

– (void)**handlePortMessage:(NSPortMessage *)portMessage**

Processes *portMessage*, an incoming message on the NSPort. See the NSPortMessage class specification for more information.

The delegate should implement only one of **handleMachMessage:** and **handlePortMessage:**.

Notifications**NSPortDidBecomeInvalidNotification**

Posted from the **invalidate** method, which is invoked when the NSPort is deallocated or when it notices that its communication channel has been damaged. The notification contains:

Notification Object The NSPort object that has become invalid.

The NSPort object posting this notification is no longer useful, so all receivers should unregister themselves for any notifications involving the NSPort.