

---

# NSSocketPort Class Reference

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



2008-10-15



Apple Inc.  
© 2008 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, and Mac OS 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

---

## **NSSocketPort Class Reference 5**

---

Overview	5
Tasks	5
Creating Instances	5
Getting Information	6
Instance Methods	6
address	6
init	6
initWithProtocolFamily:socketType:protocol:address:	7
initWithTCPPort:host:	7
initWithProtocolFamily:socketType:protocol:address:	8
initWithProtocolFamily:socketType:protocol:socket:	9
initWithTCPPort:	9
protocol	10
protocolFamily	10
socket	10
socketType	10

---

## **Document Revision History 13**

---

## **Index 15**

---



# NSSocketPort Class Reference

---

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

## Overview

`NSSocketPort` is a subclass of `NSPort` that represents a BSD socket. An `NSSocketPort` object can be used as an endpoint for distributed object connections. Companion classes, `NSMachPort` and `NSMessagePort`, allow for local (on the same machine) communication only. The `NSSocketPort` class allows for both local and remote communication, but may be more expensive than the others for the local case.

**Note:** The `NSSocketPort` class conforms to the `NSCoding` protocol, but only supports coding by an `NSPortCoder`. `NSPort` and its other subclasses do not support archiving.

## Tasks

### Creating Instances

- [init](#) (page 6)  
Initializes the receiver as a local TCP/IP socket of type `SOCK_STREAM`.
- [initWithTCPPort:](#) (page 9)  
Initializes the receiver as a local TCP/IP socket of type `SOCK_STREAM`, listening on a specified port number.
- [initWithProtocolFamily:socketType:protocol:address:](#) (page 8)  
Initializes the receiver as a local socket with the provided arguments.
- [initWithProtocolFamily:socketType:protocol:socket:](#) (page 9)  
Initializes the receiver with a previously created local socket.

- [initWithTCPPort:host:](#) (page 7)  
Initializes the receiver as a TCP/IP socket of type `SOCK_STREAM` that can connect to a remote host on a specified port.
- [initWithProtocolFamily:socketType:protocol:address:](#) (page 7)  
Initializes the receiver as a remote socket with the provided arguments.

## Getting Information

- [address](#) (page 6)  
Returns the receiver's socket address structure.
- [protocol](#) (page 10)  
Returns the protocol that the receiver uses for communication.
- [protocolFamily](#) (page 10)  
Returns the protocol family that the receiver uses for communication.
- [socket](#) (page 10)  
Returns the receiver's native socket identifier on the platform.
- [socketType](#) (page 10)  
Returns the receiver's socket type.

## Instance Methods

### address

Returns the receiver's socket address structure.

```
- (NSData *)address
```

#### Return Value

The receiver's socket address structure stored inside an `NSData` object.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [initWithProtocolFamily:socketType:protocol:address:](#) (page 8)
- [initWithRemoteWithProtocolFamily:socketType:protocol:address:](#) (page 7)

#### Declared In

`NSPort.h`

### init

Initializes the receiver as a local TCP/IP socket of type `SOCK_STREAM`.

```
- (id)init
```

**Return Value**

An initialized local TCP/IP socket port of type `SOCK_STREAM`.

**Discussion**

The port number is selected by the system.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [initWithTCPPort:](#) (page 9)
- [initWithProtocolFamily:socketType:protocol:address:](#) (page 8)

**Declared In**

NSPort.h

**initWithProtocolFamily:socketType:protocol:address:**

Initializes the receiver as a remote socket with the provided arguments.

```
- (id)initWithProtocolFamily:(int)family socketType:(int)type
    protocol:(int)protocol address:(NSData *)address
```

**Parameters**

*family*

The protocol family for the socket port.

*type*

The type of socket.

*protocol*

The specific protocol to use from the the protocol family.

*address*

The family-specific socket address for the receiver copied into an NSData object.

**Discussion**

A connection is not opened to the remote address until data is sent.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [initWithTCPPort:host:](#) (page 7)

**Declared In**

NSPort.h

**initWithTCPPort:host:**

Initializes the receiver as a TCP/IP socket of type `SOCK_STREAM` that can connect to a remote host on a specified port.

```
- (id)initWithTCPPort:(unsigned short)port host:(NSString *)hostName
```

**Parameters***port*

The port to connect to.

*hostName*The host name to connect to. *hostName* may be either a host name or an IPv4-style address.**Return Value**A TCP/IP socket port of type `SOCK_STREAM` that can connect to the remote host *hostName* on port *port*.**Discussion**

A connection is not opened to the remote host until data is sent.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [initWithProtocolFamily:socketType:protocol:address:](#) (page 7)**Declared In**

NSPort.h

**initWithProtocolFamily:socketType:protocol:address:**

Initializes the receiver as a local socket with the provided arguments.

```
- (id)initWithProtocolFamily:(int)family socketType:(int)type protocol:(int)protocol
  address:(NSData *)address
```

**Parameters***family*

The protocol family for the socket port.

*type*

The type of socket.

*protocol*

The specific protocol to use from the the protocol family.

*address*The family-specific socket address for the receiver copied into an `NSData` object.**Return Value**

A local socket port initialized with the provided arguments.

**Discussion**The receiver must be added to a run loop before it can accept connections or receive messages. Incoming messages are passed to the receiver's delegate method `handlePortMessage:`.To create a standard TCP/IP socket, use [initWithTCPPort:](#) (page 9).**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [initWithProtocolFamily:socketType:protocol:address:](#) (page 7)- [initWithProtocolFamily:socketType:protocol:socket:](#) (page 9)



**Declared In**

NSPort.h

**initWithProtocolFamily:socketType:protocol:socket:**

Initializes the receiver with a previously created local socket.

```
- (id)initWithProtocolFamily:(int)family socketType:(int)type protocol:(int)protocol
    socket:(NSSocketNativeHandle)sock
```

**Parameters***family*

The protocol family for the provided socket.

*type*

The type of the provided socket.

*protocol*

The specific protocol the provided socket uses.

*sock*

The previously created socket.

**Return Value**

A local socket port initialized with the provided socket.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [initWithProtocolFamily:socketType:protocol:address:](#) (page 8)

**Declared In**

NSPort.h

**initWithTCPPort:**

Initializes the receiver as a local TCP/IP socket of type `SOCK_STREAM`, listening on a specified port number.

```
- (id)initWithTCPPort:(unsigned short)port
```

**Parameters***port*

The port number for the newly created socket port to listen on. If *port* is 0, the system will assign a port number.

**Return Value**

An initialized local TCP/IP socket of type `SOCK_STREAM`, listening on port *port*.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [init](#) (page 6)

- [initWithProtocolFamily:socketType:protocol:address:](#) (page 8)

**Declared In**

NSPort.h

**protocol**

Returns the protocol that the receiver uses for communication.

- (int)protocol

**Return Value**

The protocol the receiver uses for communication.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSPort.h

**protocolFamily**

Returns the protocol family that the receiver uses for communication.

- (int)protocolFamily

**Return Value**

The protocol family the receiver uses for communication.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSPort.h

**socket**

Returns the receiver's native socket identifier on the platform.

- (NSSocketNativeHandle)socket

**Return Value**

The native socket identifier on the platform. For Mac OS X, this is an integer file descriptor.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSPort.h

**socketType**

Returns the receiver's socket type.

- (int)socketType

**Return Value**

The receiver's socket type.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSPort.h



# Document Revision History

---

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

Date	Notes
2008-10-15	Clarified the types of messaging available for NSSocketPort.
2007-01-30	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

---

## A

---

address [instance method 6](#)

## I

---

init [instance method 6](#)

initRemoteWithProtocolFamily:socketType:protocol:  
address: [instance method 7](#)

initRemoteWithTCPPort:host: [instance method 7](#)

initWithProtocolFamily:socketType:protocol:  
address: [instance method 8](#)

initWithProtocolFamily:socketType:protocol:socket:  
[instance method 9](#)

initWithTCPPort: [instance method 9](#)

## P

---

protocol [instance method 10](#)

protocolFamily [instance method 10](#)

## S

---

socket [instance method 10](#)

socketType [instance method 10](#)