
NSDistantObject 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, 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

NSDistantObject Class Reference 5

- Overview 5
- Adopted Protocols 6
- Tasks 6
 - Creating a Local Proxy 6
 - Creating a Remote Proxy 6
 - Getting a Proxy's NSConnection 6
 - Setting a Proxy's Protocol 6
- Class Methods 7
 - proxyWithLocal:connection: 7
 - proxyWithTarget:connection: 7
- Instance Methods 8
 - connectionForProxy 8
 - initWithLocal:connection: 8
 - initWithTarget:connection: 9
 - setProtocolForProxy: 9

Document Revision History 11

Index 13

NSDistantObject Class Reference

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

Overview

`NSDistantObject` is a concrete subclass of `NSProxy` that defines proxies for objects in other applications or threads. When a distant object receives a message, in most cases it forwards the message through its `NSConnection` object to the real object in another application, supplying the return value to the sender of the message if one is received, and propagating any exception back to the invoker of the method that raised it.

`NSDistantObject` adds two useful instance methods to those defined by `NSProxy`: [connectionForProxy](#) (page 8) returns the `NSConnection` object that handles the receiver; [setProtocolForProxy](#): (page 9) establishes the set of methods the real object is known to respond to, saving the network traffic required to determine the argument and return types the first time a particular selector is forwarded to the remote proxy.

There are two kinds of distant object: local proxies and remote proxies. A local proxy is created by an `NSConnection` object the first time an object is sent to another application. It is used by the connection for bookkeeping purposes and should be considered private. The local proxy is transmitted over the network using the `NSCoding` protocol to create the remote proxy, which is the object that the other application uses. `NSDistantObject` defines methods for an `NSConnection` object to create instances, but they're intended only for subclasses to override—you should never invoke them directly. Use the `rootProxyForConnectionWithRegisteredName:host:` method of `NSConnection`, which sets up all the required state for an object-proxy pair.

Important: `NSDistantObject` conforms to the `NSCoding` protocol, but only supports coding by an `NSPortCoder`. `NSDistantObject` and its subclasses do not support archiving.

Adopted Protocols

NSCoding

`encodeWithCoder:`
`initWithCoder:`

Tasks

Creating a Local Proxy

- + [proxyWithLocal:connection:](#) (page 7)
Returns a local proxy for a given object and connection, creating the proxy if necessary.
- [initWithLocal:connection:](#) (page 8)
Initializes an `NSDistantObject` object as a local proxy for a given object.

Creating a Remote Proxy

- + [proxyWithTarget:connection:](#) (page 7)
Returns a remote proxy for a given object and connection, creating the proxy if necessary.
- [initWithTarget:connection:](#) (page 9)
Initializes a newly allocated `NSDistantObject` as a remote proxy for `remoteObject`, which is an `id` in another thread or another application's address space.

Getting a Proxy's NSConnection

- [connectionForProxy](#) (page 8)
Returns the connection used by the receiver.

Setting a Proxy's Protocol

- [setProtocolForProxy:](#) (page 9)
Sets the methods known to be handled by the receiver to those in a given protocol.

Class Methods

proxyWithLocal:connection:

Returns a local proxy for a given object and connection, creating the proxy if necessary.

```
+ (NSDistantObject *)proxyWithLocal:(id)anObject connection:(NSConnection *)aConnection
```

Parameters

anObject

An object in the receiver's address space.

aConnection

The connection for the returned proxy.

Return Value

A local proxy for *anObject* and *aConnection*, creating it if necessary.

Discussion

Other applications connect to the proxy using the `NSConnection` `connectionWithRegisteredName:host:` class method.

Local proxies should be considered private to their `NSConnection` objects. Only an `NSConnection` object should use this method to create them, and your code shouldn't retain or otherwise use local proxies.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [initWithLocal:connection:](#) (page 8)

Declared In

`NSDistantObject.h`

proxyWithTarget:connection:

Returns a remote proxy for a given object and connection, creating the proxy if necessary.

```
+ (NSDistantObject *)proxyWithTarget:(id)remoteObject connection:(NSConnection *)aConnection
```

Parameters

remoteObject

An object in another thread or another application's address space.

aConnection

The connection to set as the `NSConnection` object for the returned proxy—it should have been created using the `NSConnection` `connectionWithRegisteredName:host:` class method.

Return Value

A remote proxy for *remoteObject* and *aConnection*, creating the proxy if necessary

Discussion

A remote proxy cannot be used until its connection's peer has a local proxy representing *remoteObject* in the other application.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [initWithTarget:connection:](#) (page 9)

Declared In

NSDistantObject.h

Instance Methods

connectionForProxy

Returns the connection used by the receiver.

```
- (NSConnection *)connectionForProxy
```

Return Value

The connection used by the receiver.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSDistantObject.h

initWithLocal:connection:

Initializes an NSDistantObject object as a local proxy for a given object.

```
- (id)initWithLocal:(id)anObject connection:(NSConnection *)aConnection
```

Parameters

anObject

An object in the receiver's address space.

aConnection

The connection for the returned proxy.

Return Value

An initialized NSDistantObject object that serves as a local proxy for *anObject*. If a proxy for *anObject* and *aConnection* already exists, the receiver is released and the existing proxy is retained and returned.

Discussion

Other applications connect to the proxy using the NSConnectionconnectionWithRegisteredName:host: class method.

Local proxies should be considered private to their `NSConnection` objects. Only an `NSConnection` object should use this method to create them, and your code shouldn't retain or otherwise use local proxies.

This is the designated initializer for local proxies. It returns an initialized object, which might be different than the original receiver

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [proxyWithLocal:connection:](#) (page 7)

Declared In

`NSDistantObject.h`

initWithTarget:connection:

Initializes a newly allocated `NSDistantObject` as a remote proxy for *remoteObject*, which is an `id` in another thread or another application's address space.

```
- (id)initWithTarget:(id)remoteObject connection:(NSConnection *)aConnection
```

Parameters

remoteObject

An object in another thread or another application's address space.

aConnection

The connection to set as the `NSConnection` object for the returned proxy—it should have been created using the `NSConnectionconnectionWithRegisteredName:host:` class method.

Return Value

An `NSDistantObject` object initialized as a remote proxy for *remoteObject*. If a proxy for *remoteObject* and *aConnection* already exists, the receiver is released and the existing proxy is retained and returned.

Discussion

A remote proxy can't be used until its connection's peer has a local proxy representing *remoteObject* in the other application.

This is the designated initializer for remote proxies. It returns an initialized object, which might be different than the original receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [proxyWithTarget:connection:](#) (page 7)

Declared In

`NSDistantObject.h`

setProtocolForProxy:

Sets the methods known to be handled by the receiver to those in a given protocol.

```
- (void)setProtocolForProxy:(Protocol *)aProtocol
```

Parameters

aProtocol

The protocol for the receiver.

Discussion

Setting a protocol for a remote proxy reduces network traffic needed to determine method argument and return types.

In order to encode a message's arguments for transmission over the network, the types of those arguments must be known in advance. When they're not known, the distributed objects system must send an initial message just to get those types, doubling the network traffic for every new message sent. Setting a protocol alleviates this need for methods defined by the protocol. You can still send messages that aren't declared in *aProtocol*—in this case the initial message is sent to determine the types, and then the real message is sent.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

SimpleThreads

Declared In

NSDistantObject.h

Document Revision History

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

Date	Notes
2007-01-19	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

`connectionForProxy` [instance method 8](#)

I

`initWithLocal:connection:` [instance method 8](#)

`initWithTarget:connection:` [instance method 9](#)

P

`proxyWithLocal:connection:` [class method 7](#)

`proxyWithTarget:connection:` [class method 7](#)

S

`setProtocolForProxy:` [instance method 9](#)