

---

# NSProtocolChecker Class Reference

[Cocoa](#) > [Objective-C Language](#)



2006-05-23



Apple Inc.  
© 2006 Apple Computer, 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

---

## **NSProtocolChecker Class Reference 5**

---

Overview	5
Tasks	5
Creating a Checker	5
Getting Information	6
Class Methods	6
protocolCheckerWithTarget:protocol:	6
Instance Methods	6
initWithTarget:protocol:	6
protocol	7
target	7

---

## **Document Revision History 9**

---

## **Index 11**

---



# NSProtocolChecker Class Reference

---

<b>Inherits from</b>	NSProxy
<b>Conforms to</b>	NSObject (NSProxy)
<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>	NSProtocolChecker.h

## Overview

The `NSProtocolChecker` class defines an object that restricts the messages that can be sent to another object (referred to as the checker's delegate). This fact can be particularly useful when an object with many methods, only a few of which ought to be remotely accessible, is made available using the distributed objects system.

A protocol checker acts as a kind of proxy; when it receives a message that is in its designated protocol, it forwards the message to its target and consequently appears to be the target object itself. However, when it receives a message not in its protocol, it raises an `NSInvalidArgumentException` to indicate that the message isn't allowed, whether or not the target object implements the method.

Typically, an object that is to be distributed (yet must restrict messages) creates an `NSProtocolChecker` for itself and returns the checker rather than returning itself in response to any messages. The object might also register the checker as the root object of an `NSConnection`.

The object should be careful about vending references to `self`—the protocol checker will convert a return value of `self` to indicate the checker rather than the object for any messages forwarded by the checker, but direct references to the object (bypassing the checker) could be passed around by other objects.

## Tasks

### Creating a Checker

+ [protocolCheckerWithTarget:protocol:](#) (page 6)

Allocates and initializes an `NSProtocolChecker` instance that will forward any messages in *aProtocol* to *anObject*, the protocol checker's target.

- [initWithTarget:protocol:](#) (page 6)

Initializes a newly allocated NSProtocolChecker instance that will forward any messages in *aProtocol* to *anObject*, the protocol checker's target.

## Getting Information

- [protocol](#) (page 7)

Returns the protocol object the receiver uses.

- [target](#) (page 7)

Returns the target of the receiver.

## Class Methods

### **protocolCheckerWithTarget:protocol:**

Allocates and initializes an NSProtocolChecker instance that will forward any messages in *aProtocol* to *anObject*, the protocol checker's target.

```
+ (id)protocolCheckerWithTarget:(NSObject *)anObject protocol:(Protocol *)aProtocol
```

#### **Discussion**

Thus, the checker can be vended in lieu of *anObject* to restrict the messages that can be sent to *anObject*. Returns the new instance.

#### **Availability**

Available in Mac OS X v10.0 and later.

#### **Declared In**

NSProtocolChecker.h

## Instance Methods

### **initWithTarget:protocol:**

Initializes a newly allocated NSProtocolChecker instance that will forward any messages in *aProtocol* to *anObject*, the protocol checker's target.

```
- (id)initWithTarget:(NSObject *)anObject protocol:(Protocol *)aProtocol
```

#### **Discussion**

Thus, the checker can be vended in lieu of *anObject* to restrict the messages that can be sent to *anObject*. If *anObject* is allowed to be freed or dereferenced by clients, the `free` method should be included in *aProtocol*.

#### **Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSProtocolChecker.h

**protocol**

Returns the protocol object the receiver uses.

- (Protocol \*)protocol

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSProtocolChecker.h

**target**

Returns the target of the receiver.

- (NSObject \*)target

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSProtocolChecker.h





# Document Revision History

---

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

Date	Notes
2006-05-23	First publication of this content as a separate document.

**REVISION HISTORY**

Document Revision History

# Index

---

## I

---

`initWithTarget:protocol:` **instance method** [6](#)

## P

---

`protocol` **instance method** [7](#)  
`protocolCheckerWithTarget:protocol:` **class**  
**method** [6](#)

## T

---

`target` **instance method** [7](#)