



## NSProtocolChecker

Inherits From:	NSObject
Declared In:	Foundation/NSProtocolChecker.h

### Class Description

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 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 delegate, and consequently appears to be the delegate itself. However, when it receives a message not in its protocol, it raises an `NSInvalidArgumentException` exception to indicate that the message isn't allowed, whether or not the delegate 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 that were forwarded by the checker, but direct references to the object (bypassing the checker) could be passed around by other objects.

### Method Types

Creating a checker	+ protocolCheckerWithTarget:protocol: – initWithTarget:protocol:
Reimplemented NSObject methods	– forwardInvocation: – methodDescriptionForSelector:

## Class Methods

### **protocolCheckerWithTarget:protocol:**

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

Allocates and initializes an NSProtocolChecker instance that will forward any messages in the *aProtocol* protocol to *anObject*, its delegate. Thus, the checker can be vended in lieu of *anObject* to restrict the messages that can be sent to *anObject*. Returns the new instance.

## Instance Methods

### **forwardInvocation:**

– (void)**forwardInvocation:**(NSInvocation \*)*anInvocation*

Forwards any message to the delegate if the method is declared in the checker’s protocol; otherwise raises an NSInvalidArgumentException.

### **initWithTarget:protocol:**

– (id)**initWithTarget:**(NSObject \*)*anObject* **protocol:**(Protocol \*)*aProtocol*

Initializes a newly allocated NSProtocolChecker instance that will forward any messages in the *aProtocol* protocol to *anObject*, its delegate. 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*. Returns the new instance.

### **methodDescriptionForSelector:**

– (struct objc\_method\_description \*) **methodDescriptionForSelector:**(SEL)*aSelector*

Returns an Objective C description for a method in the checker’s protocol, or NULL if *aSelector* isn’t declared as an instance method in the protocol.