

---

# NSNetServiceBrowser Class Reference

Networking, Internet, & Web: Services & Discovery



2009-04-18



Apple Inc.  
© 2009 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, Bonjour, 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

---

## **NSNetServiceBrowser Class Reference 5**

---

Overview	5
Tasks	6
Creating Network Service Browsers	6
Configuring Network Service Browsers	6
Using Network Service Browsers	6
Managing Run Loops	6
Instance Methods	7
delegate	7
init	7
removeFromRunLoop:forMode:	7
scheduleInRunLoop:forMode:	8
searchForBrowsableDomains	8
searchForRegistrationDomains	9
searchForServicesOfType:inDomain:	9
setDelegate:	10
stop	10

---

## **Appendix A      Deprecated NSNetServiceBrowser Methods 13**

---

Deprecated in Mac OS X v10.4	13
searchForAllDomains	13

---

## **Document Revision History 15**

---

---

## **Index 17**

---



# NSNetServiceBrowser Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.2 and later.
<b>Declared in</b>	NSNetServices.h
<b>Companion guides</b>	Bonjour Overview NSNetServices and CFNetServices Programming Guide
<b>Related sample code</b>	CocoaEcho GridCalendar PictureSharingBrowser

## Overview

The `NSNetServiceBrowser` class defines an interface for finding published services on a network using multicast DNS. An instance of `NSNetServiceBrowser` is known as a **network service browser**.

Services can range from standard services, such as HTTP and FTP, to custom services defined by other applications. You can use a network service browser in your code to obtain the list of accessible domains and then to obtain an `NSNetService` object for each discovered service. Each network service browser performs one search at a time, so if you want to perform multiple simultaneous searches, use multiple network service browsers.

A network service browser performs all searches asynchronously using the current run loop to execute the search in the background. Results from a search are returned through the associated delegate object, which your client application must provide. Searching proceeds in the background until the object receives a [stop](#) (page 10) message.

To use an `NSNetServiceBrowser` object to search for services, allocate it, initialize it, and assign a delegate. (If you wish, you can also use the [scheduleInRunLoop:forMode:](#) (page 8) and [removeFromRunLoop:forMode:](#) (page 7) methods to execute searches on a run loop other than the current one.) Once your object is ready, you begin by gathering the list of accessible domains using either the [searchForRegistrationDomains](#) (page 9) or [searchForBrowsableDomains](#) (page 8) methods. From the list of returned domains, you can pick one and use the [searchForServicesOfType:inDomain:](#) (page 9) method to search for services in that domain.

The `NSNetServiceBrowser` class provides two ways to search for domains. In most cases, your client should use the `searchForRegistrationDomains` (page 9) method to search only for local domains to which the host machine has registration authority. This is the preferred method for accessing domains as it guarantees that the host machine can connect to services in the returned domains. Access to domains outside this list may be more limited.

## Tasks

### Creating Network Service Browsers

- `init` (page 7)  
Initializes an allocated `NSNetServiceBrowser` (page 5) object.

### Configuring Network Service Browsers

- `delegate` (page 7)  
Returns the receiver's delegate.
- `setDelegate:` (page 10)  
Sets the receiver's delegate.

### Using Network Service Browsers

- `searchForBrowsableDomains` (page 8)  
Initiates a search for domains visible to the host. This method returns immediately.
- `searchForRegistrationDomains` (page 9)  
Initiates a search for domains in which the host may register services.
- `searchForServicesOfType:inDomain:` (page 9)  
Starts a search for services of a particular type within a specific domain.
- `stop` (page 10)  
Halts a currently running search or resolution.
- `searchForAllDomains` (page 13) **Deprecated in Mac OS X v10.4**  
Initiates a search for all domains that are visible to the host. (**Deprecated**. This method has been deprecated. Use `searchForBrowsableDomains` (page 8) or `searchForRegistrationDomains` (page 9) instead.)

### Managing Run Loops

- `scheduleInRunLoop:forMode:` (page 8)  
Adds the receiver to the specified run loop.
- `removeFromRunLoop:forMode:` (page 7)  
Removes the receiver from the specified run loop.

## Instance Methods

### delegate

Returns the receiver's delegate.

```
- (id < NSNetServiceBrowserDelegate >)delegate
```

#### Return Value

Delegate for the receiver.

#### Availability

Available in Mac OS X v10.2 and later.

#### See Also

- [setDelegate:](#) (page 10)

#### Declared In

NSNetServices.h

### init

Initializes an allocated [NSNetServiceBrowser](#) (page 5) object.

```
- (id)init
```

#### Return Value

Initialized [NSNetServiceBrowser](#) (page 5) object.

#### Availability

Available in Mac OS X v10.2 and later.

#### Declared In

NSNetServices.h

### removeFromRunLoop:forMode:

Removes the receiver from the specified run loop.

```
- (void)removeFromRunLoop:(NSRunLoop *)runLoop forMode:(NSString *)runLoopMode
```

#### Parameters

*runLoop*

Run loop from which to remove the receiver.

*runLoopMode*

Run loop mode in which to perform this operation, such as `NSDefaultRunLoopMode`. See the [Run Loop Modes](#) section of the `NSRunLoop` class for other run loop mode values.

**Discussion**

You can use this method in conjunction with [scheduleInRunLoop:forMode:](#) (page 8) to transfer the receiver to a run loop other than the default one. Although it is possible to remove an `NSNetService` object completely from any run loop and then attempt actions on it, you must not do it.

**Availability**

Available in Mac OS X v10.2 and later.

**See Also**

- [scheduleInRunLoop:forMode:](#) (page 8)

**Declared In**

`NSNetServices.h`

**scheduleInRunLoop:forMode:**

Adds the receiver to the specified run loop.

```
- (void)scheduleInRunLoop:(NSRunLoop *)runLoop forMode:(NSString *)runLoopMode
```

**Parameters**

*runLoop*

Run loop from which to remove the receiver.

*runLoopMode*

Run loop mode in which to perform this operation, such as `NSDefaultRunLoopMode`. See the [RunLoop Modes](#) section of the `NSRunLoop` class for other run loop mode values.

**Discussion**

You can use this method in conjunction with [removeFromRunLoop:forMode:](#) (page 7) to transfer the receiver to a run loop other than the default one. You should not attempt to run the receiver on multiple run loops.

**Availability**

Available in Mac OS X v10.2 and later.

**See Also**

- [removeFromRunLoop:forMode:](#) (page 7)

**Declared In**

`NSNetServices.h`

**searchForBrowsableDomains**

Initiates a search for domains visible to the host. This method returns immediately.

```
- (void)searchForBrowsableDomains
```

**Discussion**

The delegate receives a `netServiceBrowser:didFindDomain:moreComing:` message for each domain discovered.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

- [searchForRegistrationDomains](#) (page 9)

**Declared In**

NSNetServices.h

**searchForRegistrationDomains**

Initiates a search for domains in which the host may register services.

```
- (void)searchForRegistrationDomains
```

**Discussion**

This method returns immediately, sending a `netServiceBrowserWillSearch:` message to the delegate if the network was ready to initiate the search. The delegate receives a subsequent `netServiceBrowser:didFindDomain:moreComing:` message for each domain discovered.

Most network service browser clients do not have to use this method—it is sufficient to publish a service with the empty string, which registers it in any available registration domains automatically.

**Availability**

Available in Mac OS X v10.2 and later.

**See Also**

- [searchForBrowsableDomains](#) (page 8)
- [searchForServicesOfType:inDomain:](#) (page 9)
- `netServiceBrowser:didFindDomain:moreComing:` (`NSNetServerBrowserDelegate`)
- `netServiceBrowserWillSearch:` (`NSNetServerBrowserDelegate`)

**Declared In**

NSNetServices.h

**searchForServicesOfType:inDomain:**

Starts a search for services of a particular type within a specific domain.

```
- (void)searchForServicesOfType:(NSString *)serviceType inDomain:(NSString *)domainName
```

**Parameters**

*serviceType*

Type of the service to search for.

*domainName*

Domain name in which to perform the search.

**Discussion**

This method returns immediately, sending a `netServiceBrowserWillSearch:` message to the delegate if the network was ready to initiate the search. The delegate receives subsequent `netServiceBrowser:didFindDomain:moreComing:` messages for each service discovered.

The *serviceType* argument must contain both the service type and transport layer information. To ensure that the mDNS responder searches for services, rather than hosts, make sure to prefix both the service name and transport layer name with an underscore character (“\_”). For example, to search for an HTTP service on TCP, you would use the type string “\_http.\_tcp.”. Note that the period character at the end is required.

The *domainName* argument can be an explicit domain name, the generic local domain @"local." (note trailing period, which indicates an absolute name), or the empty string (@""), which indicates the default registration domains. Usually, you pass in an empty string. Note that it is acceptable to use an empty string for the *domainName* argument when publishing or browsing a service, but do not rely on this for resolution.

### Availability

Available in Mac OS X v10.2 and later.

### See Also

- `netServiceBrowser:didFindDomain:moreComing:` (NSNetServiceBrowserDelegate)
- `netServiceBrowserWillSearch:` (NSNetServiceBrowserDelegate)

### Related Sample Code

GridCalendar

### Declared In

NSNetServices.h

## setDelegate:

Sets the receiver’s delegate.

```
- (void)setDelegate:(id < NSNetServiceBrowserDelegate >)delegate
```

### Parameters

*delegate*

Object to serve as the receiver’s delegate. Must not be `nil`. The delegate must conform to the `NSNetServiceBrowserDelegate Protocol` protocol.

### Discussion

The delegate is not retained. The receiver calls the methods of your delegate to receive information about discovered domains and services.

### Availability

Available in Mac OS X v10.2 and later.

### See Also

- [delegate](#) (page 7)

### Related Sample Code

GridCalendar

### Declared In

NSNetServices.h

## stop

Halts a currently running search or resolution.

- (void)stop

**Discussion**

This method sends a `netServiceBrowserDidStopSearch:` message to the delegate and causes the browser to discard any pending search results.

**Availability**

Available in Mac OS X v10.2 and later.

**See Also**

- `netServiceBrowserDidStopSearch:` (`NSNetServiceBrowserDelegate`)

**Related Sample Code**

GridCalendar

**Declared In**

`NSNetServices.h`



# Deprecated NSNetServiceBrowser Methods

---

A method identified as deprecated has been superseded and may become unsupported in the future.

## Deprecated in Mac OS X v10.4

### **searchForAllDomains**

Initiates a search for all domains that are visible to the host. (Deprecated in Mac OS X v10.4. This method has been deprecated. Use [searchForBrowsableDomains](#) (page 8) or [searchForRegistrationDomains](#) (page 9) instead.)

- (void)searchForAllDomains

#### **Discussion**

This method returns immediately, sending a `netServiceBrowserWillSearch:` message to the delegate if the network was ready to initiate the search. The delegate receives a subsequent `netServiceBrowser:didFindDomain:moreComing:` message for each domain discovered.

This method may find domains in which the localhost does not have registration authority.

#### **Availability**

Available in Mac OS X v10.2 and later.

Deprecated in Mac OS X v10.4.

#### **See Also**

- [searchForRegistrationDomains](#) (page 9)
- `netServiceBrowser:didFindDomain:moreComing:` (`NSNetServiceBrowserDelegate`)

#### **Declared In**

`NSNetServices.h`



# Document Revision History

---

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

Date	Notes
2009-04-18	Updated for Mac OS X v10.6. Delegate methods moved to NSNetServiceBrowserDelegate Protocol Reference.
2009-04-08	Miscellaneous edits.
2007-04-02	Made editorial improvements.
2006-11-07	Added memory-management details to the netServiceBrowser:didFindService:moreComing: delegate method.
2006-05-23	Added "NSNetServices and CFNetServices Programming Guide" as a companion document.
	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## D

---

delegate [instance method 7](#)

## I

---

init [instance method 7](#)

## R

---

removeFromRunLoop:forMode: [instance method 7](#)

## S

---

scheduleInRunLoop:forMode: [instance method 8](#)  
searchForAllDomains [instance method 13](#)  
searchForBrowsableDomains [instance method 8](#)  
searchForRegistrationDomains [instance method 9](#)  
searchForServicesOfType:inDomain: [instance method 9](#)  
setDelegate: [instance method 10](#)  
stop [instance method 10](#)