
NSHost

Inherits From:	NSObject
Conforms To:	NSObject (NSObject)
Declared In:	Foundation/NSHost.h

Class Description

An NSHost object holds network name and address information for a host. You use this class to get the current host's name and address and to look up other hosts by name or by address.

To create an NSHost, use the **currentHost**, **hostWithAddress:**, or **hostWithName:** class methods (don't use **alloc** and **init**). These methods use available network administration services (such as NetInfo or the Domain Name Service) to discover all names and addresses for the host requested. They don't attempt to contact the host itself, however. This avoids untimely delays due to a host being unavailable, but it may result in incomplete information about the host.

An NSHost instance contains all of the network addresses and names discovered for a given host by the network administration services. Each NSHost typically contains one unique address, but it may have more than one name. If an NSHost has more than one name, the additional names are variations on the same name, typically the basic host name plus the fully qualified domain name. For example, with a host name "sales" in the domain "anycorp.com", an NSHost can hold both the names "sales" and "sales.anycorp.com".

The NSHost class maintains a cache of previously created instances so that requests for an existing NSHost return that object instead of creating a new one. Use the **setHostCacheEnabled:** method to turn the cache off, forcing lookup of hosts as they're requested. You can also use the **flushHostCache** method to clear the cache of its entries so that subsequent requests look up the host information and create new instances.

Method Types

Creating an NSHost	+ currentHost + hostWithAddress: + hostWithName:
Getting host names and addresses	– address – addresses – name – names

Comparing instances	– isEqualToHost:
Controlling the cache	+ setHostCacheEnabled:
	+ isHostCacheEnabled
	+ flushHostCache

Class Methods

NS+ currentHost

+ (NSHost *)**currentHost**

Returns an NSHost representing the host that the process is running on.

See also: + **hostWithAddress:**, + **hostWithName:**

NS+ flushHostCache

+ (void)**flushHostCache**

Releases the cache of existing NSHost instances so that subsequent requests for NSHost objects create new instances. NSHost instances that were retained before this method was invoked remain valid.

See also: + **isHostCacheEnabled**, + **setHostCacheEnabled:**

NS+ hostWithAddress:

+ (NSHost *)**hostWithAddress:**(NSString *)*address*

Returns the NSHost that has the Internet address *address*. If caching is turned on and the cache already contains an NSHost with *address*, returns that object. Otherwise, this method creates a new instance and returns it. *address* should be an NSString containing an Internet address in decimal notation, such as “127.0.0.1”.

See also: + **hostWithName:**, + **setHostCacheEnabled:**

NS+ hostWithName:

+ (NSHost *)**hostWithName:**(NSString *)*name*

Returns the NSHost named *name*. If caching is turned on and the cache already contains an NSHost with *name*, returns that object. Otherwise, this method creates a new instance and returns it. *name* can be either a simple host name, such as “sales”, or a fully qualified domain name, such as “sales.anycorp.com”.

See also: + **hostWithAddress:**, + **setHostCacheEnabled:**

NS+ **isHostCacheEnabled**

+ (BOOL)**isHostCacheEnabled**

Returns YES if the NSHost class checks the cache before creating new instances, NO if it always creates new instances.

See also: + **setHostCacheEnabled:**, + **flushHostCache**

NS+ **setHostCacheEnabled:**

+ (void)**setHostCacheEnabled:(BOOL)***flag*

Sets whether the NSHost class caches instances as it creates them to avoid creating duplicate instances. If *flag* is YES, the NSHost class checks the cache before creating new instances. If *flag* is NO, it always creates new instances. Caching is turned on by default.

This method doesn't flush the cache. If you turn caching off and then back on, new requests for hosts use what was in the cache at the time caching was turned off. However, NSHost instances created while caching is turned off aren't entered into the cache.

See also: + **isHostCacheEnabled**, + **flushHostCache**

Instance Methods

NS+ **address**

– (NSString *)**address**

Returns an arbitrary Internet address of the receiver as a decimal NSString—for example, “192.42.172.1”.

See also: – **addresses**, – **name**

NS+ **addresses**

– (NSArray *)**addresses**

Returns all of the Internet addresses of the receiver as decimal NSStrings—for example, “192.42.172.1”.

See also: – **address**, – **names**

NS+ isEqualToHost:

– (BOOL)**isEqualToHost:**(NSHost *)*aHost*

Returns YES if the receiver represents the same host as *aHost*, NO otherwise. Two NSHost objects represent the same host if they share at least one address.

See also: – **addresses**

NS+ name

– (NSString *)**name**

Returns an arbitrary name for the receiver. This name can be a fully qualified domain name such as “sales.anycorp.com” or a simple host name such as “sales”.

See also: – **address**, – **names**

NS+ names

– (NSArray *)**names**

Returns all of the names for the receiver as NSSStrings. Host names can be fully qualified domain names, such as “sales.anycorp.com”, or simple host names, such as “sales”.

See also: – **addresses**, – **name**