

Copyright ©1994 Christopher J. Kane. Version 1.0.

MiscINETAddress

Inherits From: Object

Declared In: MiscINETAddress.h

Class Description

The MiscINETAddress class encapsulates Internet addresses and common operations on them. Each host connected to the Internet has at least one address assigned to it, which is expressible in a dotted-

decimal format of four small integers (0-255) separated by periods (e.g. "127.45.32.198").

Often, a host will also have been assigned a host name. This class provides the ability to map between addresses and host names. Note that there is no direct correspondence between addresses and host names—a particular address may not map to a host name, multiple addresses may map to the same host name (for instance, if the host has more than one address), and a host name may not correspond to any address (it may be just part of a mail address).

Instances of the class are reusable, in that they may be sent multiple initialization messages. This may allow an application to avoid some memory allocation and deallocation in some cases. Note however that two of the initialization methods (**initWithName:** and **initWith:**) do free the instance on error.

Methods implemented in this class generally return self on success and nil upon failure. When nil is returned, the value of the *errno* global variable can be examined for the cause; values that this variable may take on are described in the header file `<sys/errno.h>` and the UNIX manual page *intro(2)*. If the method is successful, the previous value of *errno* is preserved.

Instance Variables

struct in_addr **address**
char ***hostname**

Internet address

Host name corresponding to address

Method Types

Initializing instances

- init
- initWithName:
- initWithTo:

Getting local host address

+ localAddress

Accessing properties

- address
- hostname
- stringAddress

Class Methods

localAddress
+ **localAddress**

Returns an instance of the class initialized to the address of the host on which the application process is running. Returns nil if the local address cannot be discovered (which should not happen).

Instance Methods

address
- (struct in_addr)**address**

Returns the current Internet address of the instance. The internal fields of the structure are not

guaranteed to be byte-ordered for use outside the structure on a particular architecture.

hostname

- (char *)**hostname**

Returns the host name associated with the instance's current Internet address. If possible, the returned string is standard name format (e.g., "arthur.cmu.com"). If such a name cannot be found, the dotted-decimal format is returned.

init

- **init**

Places the instance in the initialized state (but having no address), and returns self. This method may be applied multiple times to an instance, so that instances may be reused. This method is the designated initializer of MiscINETAddress instances. Subclasses should implement their own version of this method (beginning with [super init]) to initialize its instance variables. To maintain the

multiple-initialization property, the method should check the value of instance variables and free allocated objects and memory as part of the initialization, and extended **init-** methods (**initTo:**, for example) in the class should call [self init] first thing in the method.

initFromName:

- **initFromName:**(const char *)name

Initializes the current address from *name*, which may be a host name in standard or dotted-decimal format. Returns self on success. If *name* cannot be resolved to an Internet address and it is also not in valid dotted-decimal format (or is "255.255.255.255", an invalid address), the instance is freed and nil is returned. The global *errno* may contain the following value: EINVAL (invalid address).

initTo:

- **initTo:**(struct in_addr)addr

Initializes the current address to be *addr*. Returns self on success. If the value of *addr* is -1 (or

"255.255.255.255" in dotted-decimal), the instance is freed and nil is returned. The global *errno* may contain the following value: EINVAL (invalid address).

stringAddress

- (char *)**stringAddress**

Returns the dotted-decimal format of the current address. The returned string resides in static memory, and should be copied if it is to be saved.