

init

Freeing an NIDomain object free

Connecting to or disconnecting from a domain

getMasterServer
getCurrentServer
getTag
getServerIPAddress
getDomainHandle
findDirectory:withProperty:

Checking the error status lastError

Assigning a delegate setDelegate:

disconnectFromCurrent

Terminates the connection to a domain but retains the NIDomain object. Resets all instance variables.

(ni_entrylist *)findDirectory:(const char *)parentDirectory
withProperty:(const char *)property

Returns a list containing the values associated with the indicated property in the named NetInfo directory. The caller should free this list when it's no longer needed. This method returns NULL if it couldn't read the property. You can find the reason for the failure with lastError.

free

Deallocates the NIDomain object. Returns nil.

(const char *)getCurrentServer

Returns the host name of the current server of the domain, or NULL if the object isn't currently connected. If the host name couldn't be resolved.

Returns the fully qualified pathname of the current domain, or NULL if the path couldn't be resolved. If the path couldn't be resolved, invoking `lastError` might help you find out the cause.

`(const char *)getMasterServer`

Returns the host name of the master server of the current domain, or NULL if the object isn't currently connected to a master server or the host name couldn't be resolved.

`(const struct sockaddr_in *)getServerIPAddress`

Returns the socket address of the current server of the current domain. If an error occurs, NULL is returned. If the object is connected, invoking `lastError` should return the reason for the failure.

`(const char *)getTag`

Returns the tag of the current domain, or NULL if there's no current connection or if it couldn't be resolved. If the object is connected, invoking `lastError` should return the reason for the failure.

`init`

Initializes a newly allocated `NIDomain` instance. The new instance isn't connected. Returns self.

`(ni_status)lastError`

Returns the status code returned by the most recent `NetInfo` call. This value can be translated to an English error message by the `ni_error()` function, which is described in the `netinfo(3)` UNIX manual page.

`(ni_status)setConnection:(const char *)domain`

Establishes a connection to the named domain. Returns a value indicating status, corresponding to the values in the header file `netinfo/ni_prot.h`. This value can be translated to an English error message by the `ni_error()` function, which is described in the `netinfo(3)` UNIX manual page.

`(ni_status)setConnection:(const char *)domain
readTimeout:(int)rtime
writeTimeout:(int)wtime
canAbort:(BOOL)abortFlag
mustWrite:(BOOL)writeFlag`

Establishes a connection to the named domain with arguments corresponding to the `ni_fancyopen` function in the UNIX manual page for `netinfo`. Values for `rtime` and `wtime` indicate the timeout, in seconds.

Sets the NIDomain object's delegate to anObject. Returns self.

(ni_status)setTaggedConnection:(const char *)tag to:(char *)hostName

Establishes a connection to a domain by host name and tag rather than domain name. Returns a value corresponding to the constants defined in the header file netinfo/ni_prot.h. This value can be translated to an English error message by the ni_error() function, which is described in the netinfo(3) UNIX manual page.

(ni_status)setTaggedConnection:(const char *)tag
to:(char *)hostName
readTimeout:(int)rtime
writeTimeout:(int)wtime
canAbort:(BOOL)abortFlag

Establishes a connection to a domain by host name and tag with arguments. Same as setConnection:writeTimeout:canAbort:mustWrite:, except that it doesn't include the writeFlag argument. Since the connection is made to a specific server, the writeFlag argument is irrelevant. Returns a value indicating status, corresponding to the constants defined in the header file netinfo/ni_prot.h. This value can be translated to an English error message by the ni_error() function, which is described in the netinfo(3) UNIX manual page.

domain:sender willCloseBecause:(int)reason

Indicates that the connection to the current domain will terminate as a result of the disconnectFromDomain: value of reason is always 0, indicating that the program requested closing.