

**NAME**

`ares_getnameinfo` – Address-to-nodename translation in protocol-independent manner

**SYNOPSIS**

```
#include <ares.h>
```

```
typedef void (*ares_nameinfo_callback)(void *arg, int status,
int timeouts, char *node, char *service)
```

```
void ares_getnameinfo(ares_channel channel, const struct sockaddr *sa,
ares_socklen_t salen, int flags, ares_nameinfo_callback callback,
void *arg)
```

**DESCRIPTION**

The **`ares_getnameinfo`** function is defined for protocol-independent address translation. The function is a combination of *`ares_gethostbyaddr(3)`* and *`getservbyport(3)`*. The function will translate the address either by executing a host query on the name service channel identified by *channel* or it will attempt to resolve it locally if possible. The parameters *sa* and *len* give the address as a *sockaddr* structure, and *flags* gives the options that the function will use. Valid flags are listed below:

**ARES\_NI\_NOFQDN**

Only the nodename portion of the FQDN is returned for local hosts.

**ARES\_NI\_NUMERICHOST**

The numeric form of the hostname is returned rather than the name.

**ARES\_NI\_NAMEREQD**

An error is returned if the hostname cannot be found in the DNS.

**ARES\_NI\_NUMERICSERV**

The numeric form of the service is returned rather than the name.

**ARES\_NI\_TCP** The service name is to be looked up for the TCP protocol.

**ARES\_NI\_UDP** The service name is to be looked up for the UDP protocol.

**ARES\_NI\_SCTP** The service name is to be looked up for the SCTP protocol.

**ARES\_NI\_DCCP** The service name is to be looked up for the DCCP protocol.

**ARES\_NI\_NUMERICSCOPE**

The numeric form of the scope ID is returned rather than the name.

**ARES\_NI\_LOOKUPHOST**

A hostname lookup is being requested.

**ARES\_NI\_LOOKUPSERVICE**

A service name lookup is being requested.

When the query is complete or has failed, the *ares* library will invoke *callback*. Completion or failure of the query may happen immediately, or may happen during a later call to *`ares_process(3)`*, *`ares_destroy(3)`* or *`ares_cancel(3)`*.

The callback argument *arg* is copied from the **`ares_getnameinfo`** argument *arg*. The callback argument *status* indicates whether the query succeeded and, if not, how it failed. It may have any of the following values:

**ARES\_SUCCESS** The host lookup completed successfully.

**ARES\_ENOTIMP** The *ares* library does not know how to look up addresses of type *family*.

**ARES\_ENOTFOUND**

The address *addr* was not found.

**ARES\_ENOMEM** Memory was exhausted.

**ARES\_EDESTRUCTION**

The name service channel *channel* is being destroyed; the query will not be completed.

**ARES\_EBADFLAGS**

The *flags* parameter contains an illegal value.

The callback argument *timeouts* reports how many times a query timed out during the execution of the given request.

On successful completion of the query, the callback argument *node* contains a string representing the host-name (assuming **ARES\_NI\_LOOKUPHOST** was specified). Additionally, *service* contains a string representing the service name (assuming **ARES\_NI\_LOOKUPSERVICE** was specified). If the query did not complete successfully, or one of the values was not requested, *node* or *service* will be **NULL**.

**SEE ALSO**

**ares\_process(3)**, **ares\_getaddrinfo(3)**

**AUTHOR**

Dominick Meglio

Copyright 2005 by Dominick Meglio.