

NAME

`ares_gethostbyname` – Initiate a host query by name

SYNOPSIS

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

```
typedef void (*ares_host_callback)(void *arg, int status,
                                   int timeouts,
                                   struct hostent *hostent)
```

```
void ares_gethostbyname(ares_channel_t *channel, const char *name,
                       int family, ares_host_callback callback,
                       void *arg)
```

DESCRIPTION

The **ares_gethostbyname** function initiates a host query by name on the name service channel identified by *channel*. The parameter *name* gives the hostname as a NUL-terminated C string, and *family* gives the desired type of address for the resulting host entry. 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_gethostbyname** 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 find addresses of type *family*.
- ARES_EBADNAME** The hostname *name* is composed entirely of numbers and periods, but is not a valid representation of an Internet address.
- ARES_ENODATA** There was no data returned to extract a result from.
- ARES_ENOTFOUND** The name *name* was not found.
- ARES_ENOMEM** Memory was exhausted.
- ARES_ECANCELLED** The query was cancelled.
- ARES_EDESTRUCTION** The name service channel *channel* is being destroyed; the query will not be completed.

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 *hostent* points to a **struct hostent** containing the name of the host returned by the query. The callback need not and should not attempt to free the memory pointed to by *hostent*; the ares library will free it when the callback returns. If the query did not complete successfully, *hostent* will be **NULL**.

When the associated callback is called, it is called with a channel lock so care must be taken to ensure any processing is minimal to prevent DNS channel stalls.

The callback may be triggered from a different thread than the one which called *ares_gethostbyname(3)*.

For integrators running their own event loops and not using **ARES_OPT_EVENT_THREAD**, care needs to be taken to ensure any file descriptor lists are updated immediately within the eventloop when notified.

SEE ALSO

ares_process(3), ares_gethostbyaddr(3)