

NAME

`ares_send` – Initiate a DNS query

SYNOPSIS

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

```
typedef void (*ares_callback)(void *arg, int status,
int timeouts, unsigned char *abuf, int alen)
```

```
void ares_send(ares_channel channel, const unsigned char *qbuf,
int qlen, ares_callback callback, void *arg)
```

DESCRIPTION

The `ares_send` function initiates a DNS query on the name service channel identified by *channel*. The parameters *qbuf* and *qlen* give the DNS query, which should already have been formatted according to the DNS protocol. 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)` or `ares_destroy(3)`.

The callback argument *arg* is copied from the `ares_send` 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 query completed.

ARES_EBADQUERY

The query buffer was poorly formed (was not long enough for a DNS header or was too long for TCP transmission).

ARES_ETIMEOUT No name servers responded within the timeout period.

ARES_ECONNREFUSED

No name servers could be contacted.

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.

If the query completed, the callback argument *abuf* points to a result buffer of length *alen*. If the query did not complete, *abuf* will be NULL and *alen* will be 0.

Unless the flag **ARES_FLAG_NOCHECKRESP** was set at channel initialization time, `ares_send` will normally ignore responses whose questions do not match the questions in *qbuf*, as well as responses with reply codes of **SERVFAIL**, **NOTIMP**, and **REFUSED**. Unlike other query functions in the ares library, however, `ares_send` does not inspect the header of the reply packet to determine the error status, so a callback status of **ARES_SUCCESS** does not reflect as much about the response as for other query functions.

SEE ALSO

`ares_process(3)`

AUTHOR

Greg Hudson, MIT Information Systems

Copyright 1998 by the Massachusetts Institute of Technology.