

NAME

`ares_query` – Initiate a single-question DNS query

SYNOPSIS

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

```
typedef void (*ares_callback_dnsrec)(void *arg, ares_status_t status,
                                     size_t timeouts,
                                     const ares_dns_record_t *dnsrec);
```

```
ares_status_t ares_query_dnsrec(ares_channel_t *channel,
                                const char *name,
                                ares_dns_class_t dnsclass,
                                ares_dns_rec_type_t type,
                                ares_callback_dnsrec callback,
                                void *arg,
                                unsigned short *qid);
```

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

```
void ares_query(ares_channel_t *channel, const char *name,
               int dnsclass, int type,
               ares_callback callback, void *arg);
```

DESCRIPTION

The **`ares_query_dnsrec(3)`** and **`ares_query(3)`** functions initiate a single-question DNS query on the name service channel identified by *channel*. The parameter *name* gives the query name as a NUL-terminated C string of period-separated labels optionally ending with a period; periods and backslashes within a label must be escaped with a backslash. The parameters *dnsclass* and *type* give the class and type of the query.

`ares_query_dnsrec(3)` uses the `ares_dns_class_t` and `ares_dns_rec_type_t` defined types. However, **`ares_query(3)`** uses the values defined in `<arpa/nameser.h>`.

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

If this is called from a thread other than which the main program event loop is running, care needs to be taken to ensure any file descriptor lists are updated immediately within the eventloop. 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 argument *arg* is copied from the **`ares_query_dnsrec(3)`** or **`ares_query(3)`** 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 successfully.

ARES_ENODATA The query completed but contains no answers.

ARES_EFORMERR The query completed but the server claims that the query was malformed.

ARES_ESERVFAIL The query completed but the server claims to have experienced a failure. (This code can only occur if the **ARES_FLAG_NOCHECKRESP** flag was specified at channel initialization time; otherwise, such responses are ignored at the **ares_send_dnsrec(3)** level.)

ARES_ENOTFOUND

The query completed but the queried-for domain name was not found.

ARES_ENOTIMP

The query completed but the server does not implement the operation requested by the query. (This code can only occur if the **ARES_FLAG_NOCHECKRESP** flag was specified at channel initialization time; otherwise, such responses are ignored at the **ares_send_dnsrec(3)** level.)

ARES_EREFUSED

The query completed but the server refused the query. (This code can only occur if the **ARES_FLAG_NOCHECKRESP** flag was specified at channel initialization time; otherwise, such responses are ignored at the **ares_send_dnsrec(3)** level.)

ARES_EBADNAME

The query name *name* could not be encoded as a domain name, either because it contained a zero-length label or because it contained a label of more than 63 characters.

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.

ARES_ENOSERVER

The query will not be completed because no DNS servers were configured on the channel.

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

If the query completed (even if there was something wrong with it, as indicated by some of the above error codes), the callback argument *dnsrec* or *abuf* will be non-NULL, otherwise they will be NULL.

AVAILABILITY

ares_query_dnsrec(3) was introduced in c-ares 1.28.0.

SEE ALSO

ares_process(3), **ares_dns_record(3)**