

NAME

`ares_send` – Initiate a 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_send_dnsrec(ares_channel_t *channel,
                               const ares_dns_record_t *dnsrec,
                               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_send(ares_channel_t *channel, const unsigned char *qbuf,
              int qlen, ares_callback callback, void *arg);
```

DESCRIPTION

The `ares_send_dnsrec(3)` function initiates a DNS query formatted using the `ares_dns_record_t *` data structure created via `ares_dns_record_create(3)` in the `dnsrec` parameter. The supplied callback in the `callback` parameter also returns the response using a `ares_dns_record_t *` data structure.

The `ares_send(3)` function similarly initiates a DNS query, but instead uses raw binary buffers with fully formatted DNS messages passed in the request via the `qbuf` and `qlen` parameters. The supplied callback in the `callback` parameter also returns the raw binary DNS response in the `abuf` and `alen` parameters. This method should be considered deprecated in favor of `ares_send_dnsrec(3)`.

Both functions take an initialized ares channel identified by `channel`.

The `ares_send_dnsrec(3)` also can be supplied an optional output parameter of `qid` to populate the query id as it was placed on the wire.

The `ares_send_dnsrec(3)` function returns an `ares_status_t` response code. This may be useful to know that the query was enqueued properly. The response code does not reflect the result of the query, just the result of the enqueueing of the query.

Completion or failure of the query may happen immediately (even before the function returning), or may happen later as network events are processed.

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_send_dnsrec(3)` or `ares_send(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.

The callback argument `arg` is copied from the `ares_send_dnsrec(3)` or `ares_send(3)` `arg` parameter.

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.

ARES_ENOSERVER

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

ARES_EBADQUERY

Misformatted DNS query.

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 *dnsrec* for *ares_send_dnsrec(3)* or *abuf* and *alen* for *ares_send(3)* will be non-NULL.

Unless the flag **ARES_FLAG_NOCHECKRESP** was set at channel initialization time, *ares_send_dnsrec(3)* and *ares_send(3)* will normally ignore responses whose questions do not match the supplied questions, as well as responses with reply codes of **SERVFAIL**, **NOTIMP**, and **REFUSED**. Unlike other query functions in the ares library, however, *ares_send_dnsrec(3)* and *ares_send(3)* do 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.

AVAILABILITY

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

SEE ALSO

ares_dns_record_create(3), **ares_process(3)**, **ares_search(3)**, **ares_dns_record(3)**