

NAME

`ares_search` – Initiate a DNS query with domain search

SYNOPSIS

```
#include <ares.h>

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

void ares_search(ares_channel channel, const char *name,
int dnsclass, int type, ares_callback callback,
void *arg)
```

DESCRIPTION

The `ares_search` function initiates a series of single-question DNS queries on the name service channel identified by `channel`, using the channel's search domains as well as a host alias file given by the HOSTALIAS environment variable. The parameter `name` gives the alias name or the base of the query name as a NUL-terminated C string of period-separated labels; if it ends with a period, the channel's search domains will not be used. 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 using the values defined in `<arpa/nameser.h>`. When the query sequence is complete or has failed, the ares library will invoke `callback`. Completion or failure of the query sequence 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_search` argument `arg`. The callback argument `status` indicates whether the query sequence ended with a successful query and, if not, how the query sequence failed. It may have any of the following values:

- ARES_SUCCESS** A query completed successfully.
- ARES_ENODATA** No query completed successfully; when the query was tried without a search domain appended, a response was returned with no answers.
- ARES_EFORMERR** A query completed but the server claimed that the query was malformed.
- ARES_ESERVFAIL** No query completed successfully; when the query was tried without a search domain appended, the server claimed 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(3)` level.)
- ARES_ENOTFOUND** No query completed successfully; when the query was tried without a search domain appended, the server reported that the queried-for domain name was not found.
- ARES_ENOTIMP** A 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(3)` level.)
- ARES_EREFOUSED** A query completed but the server refused the query. (This code can only occur returned if the **ARES_FLAG_NOCHECKRESP** flag was specified at channel initialization time; otherwise, such responses are ignored at the `ares_send(3)` level.)
- ARES_TIMEOUT** No name servers responded to a query within the timeout period.
- ARES_ECONNREFUSED** No name servers could be contacted.
- ARES_ENOMEM** Memory was exhausted.
- 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 a query completed successfully, the callback argument *abuf* points to a result buffer of length *alen*. If the query did not complete successfully, *abuf* will usually be NULL and *alen* will usually be 0, but in some cases an unsuccessful query result may be placed in *abuf*.

SEE ALSO

ares_process(3)

AUTHOR

Greg Hudson, MIT Information Systems
Copyright 1998 by the Massachusetts Institute of Technology.