

NAME

CURLOPT_DNS_CACHE_TIMEOUT – life-time for DNS cache entries

SYNOPSIS

```
#include <curl/curl.h>
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_DNS_CACHE_TIMEOUT, long age);
```

DESCRIPTION

Pass a long, this sets the timeout in seconds. Name resolve results are kept in memory and used for this number of seconds. Set to zero to completely disable caching, or set to -1 to make the cached entries remain forever. By default, libcurl caches this info for 60 seconds.

We recommend users not to tamper with this option unless strictly necessary. If you do, be careful of using large values that can make the cache size grow significantly if many different hostnames are used within that timeout period.

The name resolve functions of various libc implementations do not re-read name server information unless explicitly told so (for example, by calling *res_init(3)*). This may cause libcurl to keep using the older server even if DHCP has updated the server info, and this may look like a DNS cache issue to the casual libcurl-app user.

DNS entries have a "TTL" property but libcurl does not use that. This DNS cache timeout is entirely speculative that a name resolves to the same address for a small amount of time into the future.

libcurl prunes entries from the DNS cache if it exceeds 30,000 entries no matter which timeout value is used. (Added in version 8.1.0)

Since curl 8.16.0, failed name resolves are stored in the DNS cache for half the set timeout period.

DEFAULT

60

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/foo.bin");

        /* only reuse addresses for a short time */
        curl_easy_setopt(curl, CURLOPT_DNS_CACHE_TIMEOUT, 2L);

        result = curl_easy_perform(curl);

        /* in this second request, the cache is not be used if more than
           two seconds have passed since the previous name resolve */
        result = curl_easy_perform(curl);

        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.9.3

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*.

SEE ALSO

CURLMOPT_NETWORK_CHANGED(3), CURLLOPT_CONNECTTIMEOUT_MS(3), CURLLOPT_DNS_SERVERS(3), CURLLOPT_DNS_USE_GLOBAL_CACHE(3), CURLLOPT_MAXAGE_CONN(3), CURLLOPT_RESOLVE(3)