

NAME

CURLOPT_RESOLVE – provide custom hostname to IP address resolves

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_RESOLVE,  
                           struct curl_slist *hosts);
```

DESCRIPTION

Pass a pointer to a linked list of strings with hostname resolve information to use for requests with this handle. The linked list should be a fully valid list of **struct curl_slist** structs properly filled in. Use *curl_slist_append(3)* to create the list and *curl_slist_free_all(3)* to clean up an entire list.

libcurl does not copy the list, it needs to be kept around until after the transfer has completed.

Each resolve rule to add should be written using the format

```
[+]HOST:PORT:ADDRESS[,ADDRESS]
```

HOST is the name libcurl wants to resolve, PORT is the port number of the service where libcurl wants to connect to the HOST and ADDRESS is one or more numerical IP addresses. If you specify multiple IP addresses they need to be separated by comma. If libcurl is built to support IPv6, each of the ADDRESS entries can of course be either IPv4 or IPv6 style addressing.

Specify the host as a single asterisk (*) to match all names. This wildcard is resolved last so any resolve with a specific host and port number is given priority.

This option effectively populates the DNS cache with entries for the host+port pair so redirects and everything that operations against the HOST+PORT instead use your provided ADDRESS.

The optional leading plus (+) specifies that the new entry should timeout. Entries added without the leading plus character never times out whereas entries added with +HOST:... times out like ordinary DNS cache entries.

If the DNS cache already has an entry for the given host+port pair, the new entry overrides the former one.

An ADDRESS provided by this option is only used if not restricted by the setting of *CURLOPT_IPRESOLVE(3)* to a different IP version.

To remove names from the DNS cache again, to stop providing these fake resolves, include a string in the linked list that uses the format

```
-HOST:PORT
```

The entry to remove must be prefixed with a dash, and the hostname and port number must exactly match what was added previously.

Provide IPv6 addresses within [brackets].

Using this option multiple times makes the last set list override the previous ones. Set it to NULL to disable its use again.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl;
    struct curl_slist *host = NULL;
    host = curl_slist_append(NULL, "example.com:443:127.0.0.1");
    host = curl_slist_append(host, "example.com:443:[2001:db8::252f:efd6]");

    curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_RESOLVE, host);
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        result = curl_easy_perform(curl);

        /* always cleanup */
        curl_easy_cleanup(curl);
    }

    curl_slist_free_all(host);
}
```

HISTORY

Added in 7.21.3. Removal support added in 7.42.0.

Support for providing the ADDRESS within [brackets] was added in 7.57.0.

Support for providing multiple IP addresses per entry was added in 7.59.0.

Support for adding non-permanent entries by using the "+" prefix was added in 7.75.0.

Support for specifying the host component as an IPv6 address was added in 8.13.0.

AVAILABILITY

Added in curl 7.21.3

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

CURLOPT_CONNECT_TO(3), CURLOPT_DNS_CACHE_TIMEOUT(3), CURLOPT_IPRESOLVE(3)