NAME

CURLOPT_CONNECT_TO − Connect to a specific host and port instead of the URL's host and port

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_CONNECT_TO,
                          struct curl_slist *connect_to);
```

DESCRIPTION

Pass a pointer to a linked list of strings with "connect to" information to use for establishing network connections 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.

Each single string should be written using the format HOST:PORT:CONNECT-TO-HOST:CONNECT-TO-PORT where HOST is the host of the request, PORT is the port of the request, CONNECT-TO-HOST is the host name to connect to, and CONNECT-TO-PORT is the port to connect to.

The first string that matches the request's host and port is used.

Dotted numerical IP addresses are supported for HOST and CONNECT-TO-HOST. A numerical IPv6 address must be written within [brackets].

Any of the four values may be empty. When the HOST or PORT is empty, the host or port will always match (the request's host or port is ignored). When CONNECT-TO-HOST or CONNECT-TO-PORT is empty, the "connect to" feature will be disabled for the host or port, and the request's host or port will be used to establish the network connection.

This option is suitable to direct the request at a specific server, e.g. at a specific cluster node in a cluster of servers.

The "connect to" host and port are only used to establish the network connection. They do NOT affect the host and port that are used for TLS/SSL (e.g. SNI, certificate verification) or for the application protocols.

In contrast to *CURLOPT_RESOLVE(3)*, the option *CURLOPT_CONNECT_TO(3)* does not pre-populate the DNS cache and therefore it does not affect future transfers of other easy handles that have been added to the same multi handle.

The "connect to" host and port are ignored if they are equal to the host and the port in the request URL, because connecting to the host and the port in the request URL is the default behavior.

If an HTTP proxy is used for a request having a special "connect to" host or port, and the "connect to" host or port differs from the requests's host and port, the HTTP proxy is automatically switched to tunnel mode for this specific request. This is necessary because it is not possible to connect to a specific host or port in normal (non-tunnel) mode.

When this option is passed to *curl_easy_setopt(3)*, libcurl will not copy the entire list so you **must** keep it around until you no longer use this *handle* for a transfer before you call *curl_slist_free_all(3)* on the list.

DEFAULT

NULL

PROTOCOLS

All

EXAMPLE

```
CURL *curl;
struct curl_slist *connect_to = NULL;
connect_to = curl_slist_append(NULL, "example.com::server1.example.com:");

curl = curl_easy_init();
if(curl) {
  curl_easy_setopt(curl, CURLOPT_CONNECT_TO, connect_to);
  curl_easy_setopt(curl, CURLOPT_URL, "http://example.com");

  curl_easy_perform(curl);

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

curl_slist_free_all(connect_to);
```

AVAILABILITY

Added in 7.49.0

RETURN VALUE

Returns CURLE_OK if the option is supported, and CURLE_UNKNOWN_OPTION if not.

SEE ALSO

CURLOPT_URL(3), **CURLOPT_RESOLVE**(3), **CURLOPT_FOLLOWLOCATION**(3), **CURLOPT_HTTPPROXYTUNNEL**(3),