

NAME

CURLOPT_PRE_PROXY – pre-proxy host to use

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PRE_PROXY, char *preproxy);
```

DESCRIPTION

Set the *preproxy* to use for the upcoming request. The parameter should be a char * to a null-terminated string holding the hostname or dotted numerical IP address. A numerical IPv6 address must be written within [brackets].

To specify port number in this string, append `:[port]` to the end of the host name. The proxy's port number may optionally be specified with the separate option `CURLOPT_PROXYPORT(3)`. If not specified, libcurl defaults to using port 1080 for proxies.

A pre proxy is a SOCKS proxy that curl connects to before it connects to the HTTP(S) proxy specified in the `CURLOPT_PROXY(3)` option. The pre proxy can only be a SOCKS proxy.

The pre proxy string should be prefixed with `[scheme]://` to specify which kind of socks is used. Use `socks4://`, `socks4a://`, `socks5://` or `socks5h://` (the last one to enable socks5 and asking the proxy to do the resolving, also known as `CURLPROXY_SOCKS5_HOSTNAME` type) to request the specific SOCKS version to be used. Otherwise SOCKS4 is used as default.

Setting the pre proxy string to "" (an empty string) explicitly disables the use of a pre proxy.

When you set a hostname to use, do not assume that there is any particular single port number used widely for proxies. Specify it.

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string 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 = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/file.txt");
        curl_easy_setopt(curl, CURLOPT_PRE_PROXY, "socks4://socks-proxy:1080");
        curl_easy_setopt(curl, CURLOPT_PROXY, "http://proxy:80");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.52.0

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_HTTPPROXYTUNNEL(3), CURLOPT_PROXY(3)