

NAME

CURLINFO_LASTSOCKET – last socket used

SYNOPSIS

#include <curl/curl.h>

CURLcode curl_easy_getinfo(CURL *handle, CURLINFO_LASTSOCKET, long *socket);

DESCRIPTIONDeprecated. Use *CURLINFO_ACTIVESOCKET*(3) instead.

Pass a pointer to a long to receive the last socket used by this curl session. If the socket is no longer valid, -1 is returned. When you finish working with the socket, you must call *curl_easy_cleanup*(3) as usual and let libcurl close the socket and cleanup other resources associated with the handle. This is typically used in combination with *CURLOPT_CONNECT_ONLY*(3).

NOTE: this API is deprecated since it is not working on win64 where the SOCKET type is 64 bits large while its 'long' is 32 bits. Use the *CURLINFO_ACTIVESOCKET*(3) instead, if possible.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        long sockfd; /* does not work on win64 */
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        /* Do not do the transfer - only connect to host */
        curl_easy_setopt(curl, CURLOPT_CONNECT_ONLY, 1L);
        result = curl_easy_perform(curl);
        if(result != CURLE_OK) {
            printf("Error: %s\n", curl_easy_strerror(result));
            curl_easy_cleanup(curl);
            return 1;
        }

        /* Extract the socket from the curl handle */
        result = curl_easy_getinfo(curl, CURLINFO_LASTSOCKET, &sockfd);
        if(!result && sockfd != -1) {
            /* operate on sockfd */
        }

        curl_easy_cleanup(curl);
    }
}
```

DEPRECATED

Deprecated since 7.45.0.

AVAILABILITY

Added in curl 7.15.2

RETURN VALUE*curl_easy_getinfo*(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

**CURLINFO_ACTIVESOCKET(3), CURLOPT_CONNECT_ONLY(3), curl_easy_getinfo(3),
curl_easy_setopt(3)**