

NAME

CURLINFO_OS_ERRNO – errno number from last connect failure

SYNOPSIS

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

```
CURLcode curl_easy_getinfo(CURL *handle, CURLINFO_OS_ERRNO, long *errnop);
```

DESCRIPTION

Pass a pointer to a long to receive the errno variable from a connect failure. Note that the value is only set on failure, it is not reset upon a successful operation. The number is OS and system specific.

libcurl network-related errors that may have a saved errno are: CURLE_COULDNT_CONNECT, CURLE_FAILED_INIT, CURLE_INTERFACE_FAILED, CURLE_OPERATION_TIMEDOUT, CURLE_RECV_ERROR, CURLE_SEND_ERROR.

Since 8.8.0 libcurl clears the easy handle's saved errno before performing the transfer. Prior versions did not clear the saved errno, which means if a saved errno is retrieved it could be from a previous transfer on the same handle.

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");
        result = curl_easy_perform(curl);
        if(result != CURLE_OK) {
            long error;
            result = curl_easy_getinfo(curl, CURLINFO_OS_ERRNO, &error);
            if(!result && error) {
                printf("Errno: %ld\n", error);
            }
        }
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.12.2

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

curl_easy_getinfo(3), *curl_easy_setopt(3)*