

NAME

CURLMOPT_TIMERFUNCTION – set callback to receive timeout values

SYNOPSIS

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

```
int timer_callback(CURLM *multi, /* multi handle */  
                  long timeout_ms, /* see above */  
                  void *userp); /* private callback pointer */
```

```
CURLMcode curl_multi_setopt(CURLM *handle, CURLMOPT_TIMERFUNCTION, timer_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

This callback function will be called when the timeout value changes. The **timeout_ms** value is at what latest time the application should call one of the "performing" functions of the multi interface (*curl_multi_socket_action(3)* and *curl_multi_perform(3)*) - to allow libcurl to keep timeouts and retries etc to work. A **timeout_ms** value of -1 means that there is no timeout at all, and 0 means that the timeout is already expired. libcurl attempts to limit calling this only when the fixed future timeout time actually changes.

The **userp** pointer is set with *CURLMOPT_TIMERDATA(3)*.

The timer callback should return 0 on success, and -1 on error. This callback can be used instead of, or in addition to, *curl_multi_timeout(3)*.

DEFAULT

NULL

PROTOCOLS

All

EXAMPLE

TODO

AVAILABILITY

Added in 7.16.0

RETURN VALUE

Returns CURLM_OK if the option is supported, and CURLM_UNKNOWN_OPTION if not.

SEE ALSO

CURLMOPT_TIMERDATA(3), *CURLMOPT_SOCKETFUNCTION(3)*,