

NAME

CURLOPT_UPKEEP_INTERVAL_MS – connection upkeep interval

SYNOPSIS

#include <curl/curl.h>

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_UPKEEP_INTERVAL_MS,
                          long upkeep_interval_ms);
```

DESCRIPTION

Some protocols have "connection upkeep" mechanisms. These mechanisms usually send some traffic on existing connections in order to keep them alive; this can prevent connections from being closed due to overzealous firewalls, for example.

The user needs to explicitly call *curl_easy_upkeep(3)* in order to perform the upkeep work.

Currently the only protocol with a connection upkeep mechanism is HTTP/2: when the connection upkeep interval is exceeded and *curl_easy_upkeep(3)* is called, an HTTP/2 PING frame is sent on the connection.

DEFAULT

CURL_UPKEEP_INTERVAL_DEFAULT (currently defined as 60000L, which is 60 seconds)

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        /* Make a connection to an HTTP/2 server. */
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        /* Set the interval to 30000ms / 30s */
        curl_easy_setopt(curl, CURLOPT_UPKEEP_INTERVAL_MS, 30000L);

        result = curl_easy_perform(curl);

        /* Perform more work here. */

        /* While the connection is being held open, curl_easy_upkeep() can be
           called. If curl_easy_upkeep() is called and the time since the last
           upkeep exceeds the interval, then an HTTP/2 PING is sent. */
        curl_easy_upkeep(curl);

        /* Perform more work here. */

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

AVAILABILITY

Added in curl 7.62.0

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_TCP_KEEPALIVE(3)