NAME

curl_easy_upkeep − keep existing connections alive

SYNOPSIS

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

CURLcode curl_easy_upkeep(CURL *handle);
```

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.

For HTTP/2 we have an upkeep mechanism: when the connection upkeep interval is exceeded and *curl_easy_upkeep(3)* is called, an HTTP/2 PING frame is sent on the connection.

For MQTT the upkeep interval defines when to send ping requests to prevent the server from disconnecting.

This function must be explicitly called in order to perform the upkeep work. The connection upkeep interval is set with *CURLOPT_UPKEEP_INTERVAL_MS(3)*.

If you call this function on an easy handle that uses a shared connection cache then upkeep is performed on the connections in that cache, even if those connections were never used by the easy handle. (Added in 8.10.0)

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
 CURL *curl = curl_easy_init();
 if(curl) {
  /* 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);

  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

This function returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non−zero means an error occurred, see *libcurl−errors(3)*. If *CURLOPT_ERRORBUFFER(3)* was set with *curl_easy_setopt(3)* there can be an error message stored in the error buffer when non−zero is returned.

SEE ALSO

CURLOPT_TCP_KEEPALIVE(3), **CURLOPT_TCP_KEEPIDLE**(3)