NAME

CURLOPT_POSTFIELDSIZE_LARGE − size of POST data pointed to

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_POSTFIELDSIZE_LARGE,
                          curl_off_t size);
```

DESCRIPTION

If you want to post static data to the server without having libcurl do a strlen() to measure the data size, this option must be used. When this option is used you can post fully binary data, which otherwise is likely to fail. If this size is set to −1, libcurl uses strlen() to get the size or relies on the *CURLOPT_READFUNCTION(3)* (if used) to signal the end of data.

DEFAULT

-1

PROTOCOLS

This functionality affects http, mqtt and rtsp

EXAMPLE

```
extern char *large_chunk; /* pointer to somewhere */

int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode result;
    const char *data = large_chunk;
    curl_off_t length_of_data = 12345; /* set somehow */

    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

    /* size of the POST data */
    curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, length_of_data);

    curl_easy_setopt(curl, CURLOPT_POSTFIELDS, data);

    result = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
  }
}
```

AVAILABILITY

Added in curl 7.11.1

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_COPYPOSTFIELDS(3), **CURLOPT_POSTFIELDS**(3), **CURLOPT_POSTFIELDSIZE**(3)