

NAME

CURLOPT_MAXFILESIZE_LARGE – maximum file size allowed to download

SYNOPSIS

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

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

DESCRIPTION

Pass a `curl_off_t` as parameter. This specifies the maximum accepted *size* (in bytes) of a file to download. If the file requested is found larger than this value, the transfer is aborted and *CURLE_FILESIZE_EXCEEDED* is returned. Passing a zero *size* disables this, and passing a negative *size* yields a *CURLE_BAD_FUNCTION_ARGUMENT*.

The file size is not always known prior to the download start, and for such transfers this option has no effect – even if the file transfer eventually ends up being larger than this given limit.

Since 8.4.0, this option also stops ongoing transfers if they reach this threshold.

Since 8.20.0, this option also stops ongoing transfers that would reach this threshold due to automatic decompression using *CURLOPT_ACCEPT_ENCODING*(3).

DEFAULT

0, meaning disabled.

PROTOCOLS

This functionality affects ftp, http and mqtt

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_off_t ridiculous = (curl_off_t)1 << 48;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        /* refuse to download if larger than ridiculous */
        curl_easy_setopt(curl, CURLOPT_MAXFILESIZE_LARGE, ridiculous);
        result = curl_easy_perform(curl);
    }
}
```

AVAILABILITY

Added in curl 7.11.0

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_MAXFILESIZE(3), *CURLOPT_MAX_RECV_SPEED_LARGE*(3)