

NAME

CURLOPT_RANGE – byte range to request

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_RANGE, char *range);
```

DESCRIPTION

Pass a char pointer as parameter, which should contain the specified range you want to retrieve. It should be in the format "X-Y", where either X or Y may be left out and X and Y are byte indexes.

HTTP transfers also support several intervals, separated with commas as in "X-Y,N-M". Using this kind of multiple intervals causes the HTTP server to send the response document in pieces (using standard MIME separation techniques) as a multiple part response which libcurl returns as-is. It contains meta information in addition to the requested bytes. Parsing or otherwise transforming this response is the responsibility of the caller.

Unfortunately, the HTTP standard (RFC 7233 section 3.1) allows servers to ignore range requests so even when you set *CURLOPT_RANGE(3)* for a request, you may end up getting the full response sent back.

For RTSP, the formatting of a range should follow RFC 2326 Section 12.29. For RTSP, byte ranges are **not** permitted. Instead, ranges should be given in **npt**, **utc**, or **smpte** formats.

For HTTP PUT uploads this option should not be used, since it may conflict with other options.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

The application does not have to keep the string around after setting this option.

DEFAULT

NULL

PROTOCOLS

This functionality affects file, ftp, http, rtsp and sftp

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        /* get the first 200 bytes */
        curl_easy_setopt(curl, CURLOPT_RANGE, "0-199");

        /* Perform the request */
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

HISTORY

FILE since 7.18.0, RTSP since 7.20.0

AVAILABILITY

Added in curl 7.1

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_LOW_SPEED_LIMIT(3), CURLOPT_MAXFILESIZE_LARGE(3), CUR-
LOPT_MAX_RECV_SPEED_LARGE(3), CURLOPT_RESUME_FROM(3)