

NAME

CURLOPT_RESUME_FROM – offset to resume transfer from

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_RESUME_FROM, long from);
```

DESCRIPTION

Pass a long as parameter. It contains the offset in number of bytes that you want the transfer to start from. Set this option to 0 to make the transfer start from the beginning (effectively disabling resume). For FTP, set this option to -1 to make the transfer start from the end of the target file (useful to continue an interrupted upload).

When doing uploads with FTP, the resume position is where in the local/source file libcurl should try to resume the upload from and it then appends the source file to the remote target file.

If you need to resume a transfer beyond the 2GB limit, use *CURLOPT_RESUME_FROM_LARGE(3)* instead.

DEFAULT

0, not used

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        long size_of_file = 6789;

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

        /* resume upload at byte index 200 */
        curl_easy_setopt(curl, CURLOPT_RESUME_FROM, 200L);

        /* ask for upload */
        curl_easy_setopt(curl, CURLOPT_UPLOAD, 1L);

        /* set total data amount to expect */
        curl_easy_setopt(curl, CURLOPT_INFILESIZE, size_of_file);

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

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_INFILESIZE*(3), *CURLOPT_RANGE*(3), *CURLOPT_RESUME_FROM_LARGE*(3)**