

NAME

CURLOPT_READFUNCTION – read callback for data uploads

SYNOPSIS

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

```
size_t read_callback(char *buffer, size_t size, size_t nitems, void *instream);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_READFUNCTION, read_callback);
```

DESCRIPTION

Pass a pointer to your callback function, as the prototype shows above.

This callback function gets called by libcurl as soon as it needs to read data in order to send it to the peer. The data area pointed at by the pointer *buffer* should be filled up with at most *size* multiplied with *nmem* number of bytes by your function.

Your function must then return the actual number of bytes that it stored in that memory area. Returning 0 will signal end-of-file to the library and cause it to stop the current transfer.

If you stop the current transfer by returning 0 "pre-maturely" (i.e before the server expected it, like when you've said you will upload N bytes and you upload less than N bytes), you may experience that the server "hangs" waiting for the rest of the data that won't come.

The read callback may return *CURL_READFUNC_ABORT* to stop the current operation immediately, resulting in a *CURLE_ABORTED_BY_CALLBACK* error code from the transfer.

The callback can return *CURL_READFUNC_PAUSE* to cause reading from this connection to pause. See *curl_easy_pause(3)* for further details.

Bugs: when doing TFTP uploads, you must return the exact amount of data that the callback wants, or it will be considered the final packet by the server end and the transfer will end there.

If you set this callback pointer to NULL, or don't set it at all, the default internal read function will be used. It is doing an *fread()* on the FILE * *userdata* set with *CURLOPT_READDATA(3)*.

DEFAULT

The default internal read callback is *fread()*.

PROTOCOLS

This is used for all protocols when doing uploads.

EXAMPLE

Here's an example setting a read callback for reading that to upload to an FTP site:

```
http://curl.haxx.se/libcurl/c/ftpupload.html
```

AVAILABILITY

CURL_READFUNC_PAUSE return code was added in 7.18.0 and *CURL_READFUNC_ABORT* was added in 7.12.1.

RETURN VALUE

This will return *CURLE_OK*.

SEE ALSO

CURLOPT_READDATA(3), *CURLOPT_WRITEFUNCTION(3)*, *CURLOPT_SEEKFUNCTION(3)*,