

NAME

CURLOPT_SEEKFUNCTION – user callback for seeking in input stream

SYNOPSIS

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

/* These are the return codes for the seek callbacks */
#define CURL_SEEKFUNC_OK    0
#define CURL_SEEKFUNC_FAIL  1 /* fail the entire transfer */
#define CURL_SEEKFUNC_CANTSEEK 2 /* tell libcurl seeking cannot be done, so
                                libcurl might try other means instead */

int seek_callback(void *clientp, curl_off_t offset, int origin);

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SEEKFUNCTION, seek_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

This function gets called by libcurl to seek to a certain position in the input stream and can be used to fast forward a file in a resumed upload (instead of reading all uploaded bytes with the normal read function/callback). It is also called to rewind a stream when data has already been sent to the server and needs to be sent again. This may happen when doing an HTTP PUT or POST with a multi-pass authentication method, or when an existing HTTP connection is reused too late and the server closes the connection. The function shall work like `fseek(3)` or `lseek(3)` and it gets `SEEK_SET`, `SEEK_CUR` or `SEEK_END` as argument for *origin*, although libcurl currently only passes `SEEK_SET`.

clientp is the pointer you set with `CURLOPT_SEEKDATA(3)`.

The callback function must return `CURL_SEEKFUNC_OK` on success, `CURL_SEEKFUNC_FAIL` to cause the upload operation to fail or `CURL_SEEKFUNC_CANTSEEK` to indicate that while the seek failed, libcurl is free to work around the problem if possible. The latter can sometimes be done by instead reading from the input or similar.

If you forward the input arguments directly to `fseek(3)` or `lseek(3)`, note that the data type for *offset* is not the same as defined for `curl_off_t` on many systems.

DEFAULT

NULL

PROTOCOLS

This functionality affects ftp, http and sftp

EXAMPLE

```
#include <unistd.h> /* for lseek */

struct data {
    int our_fd;
};

static int seek_cb(void *clientp, curl_off_t offset, int origin)
{
    struct data *d = (struct data *)clientp;
    lseek(d->our_fd, offset, origin);
    return CURL_SEEKFUNC_OK;
}

int main(void)
```

```
{
    struct data seek_data;
    CURL *curl = curl_easy_init();
    if(curl) {
        curl_easy_setopt(curl, CURLOPT_SEEKFUNCTION, seek_cb);
        curl_easy_setopt(curl, CURLOPT_SEEKDATA, &seek_data);
    }
}
```

AVAILABILITY

Added in curl 7.18.0

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_DEBUGFUNCTION(3), CURLOPT_IOCTLFUNCTION(3), CURLOPT_SEEK-
DATA(3), CURLOPT_STDERR(3)