

NAME

CURLOPT_CHUNK_END_FUNCTION – callback after a transfer with FTP wildcard match

SYNOPSIS

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

```
long chunk_end_callback(void *ptr);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_CHUNK_END_FUNCTION,
                           chunk_end_callback);
```

DESCRIPTION

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

This function gets called by libcurl as soon as a part of the stream has been transferred (or skipped).

Return *CURL_CHUNK_END_FUNC_OK* if everything is fine or **CURL_CHUNK_END_FUNC_FAIL** to tell the lib to stop if some error occurred.

DEFAULT

NULL

PROTOCOLS

This functionality affects ftp only

EXAMPLE

```
#include <stdio.h>

struct callback_data {
    FILE *output;
};

static long file_is_downloaded(void *ptr)
{
    struct callback_data *data = ptr;
    if(data->output) {
        fclose(data->output);
        data->output = 0x0;
    }
    return CURL_CHUNK_END_FUNC_OK;
}

int main()
{
    /* data for callback */
    struct callback_data callback_info;

    CURL *curl = curl_easy_init();

    curl_easy_setopt(curl, CURLOPT_CHUNK_END_FUNCTION, file_is_downloaded);
    curl_easy_setopt(curl, CURLOPT_CHUNK_DATA, &callback_info);
}
```

AVAILABILITY

Added in curl 7.21.0

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_CHUNK_BGN_FUNCTION(3), **CURLOPT_WILDCARDMATCH**(3)