

NAME

CURLOPT_CHUNK_DATA – pointer passed to the FTP chunk callbacks

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_CHUNK_DATA, void *pointer);
```

DESCRIPTION

Pass a *pointer* that is untouched by libcurl and passed as the *ptr* argument to the *CURLOPT_CHUNK_BGN_FUNCTION(3)* and *CURLOPT_CHUNK_END_FUNCTION(3)*.

DEFAULT

NULL

PROTOCOLS

This functionality affects ftp only

EXAMPLE

```
#include <stdio.h>

struct callback_data {
    FILE *output;
};

static long file_is_coming(struct curl_fileinfo *finfo,
                          void *ptr,
                          int remains)
{
    struct callback_data *data = ptr;
    printf("%3d %40s %10luB ", remains, finfo->filename,
          (unsigned long)finfo->size);

    switch(finfo->filetype) {
    case CURLFILETYPE_DIRECTORY:
        printf(" DIR\n");
        break;
    case CURLFILETYPE_FILE:
        printf(" FILE ");
        break;
    default:
        printf(" OTHER\n");
        break;
    }

    if(finfo->filetype == CURLFILETYPE_FILE) {
        /* do not transfer files >= 50B */
        if(finfo->size > 50) {
            printf(" SKIPPED\n");
            return CURL_CHUNK_BGN_FUNC_SKIP;
        }

        data->output = fopen(finfo->filename, "wb");
        if(!data->output) {
            return CURL_CHUNK_BGN_FUNC_FAIL;
        }
    }
}
```

```
    return CURL_CHUNK_BGN_FUNC_OK;
}

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

    CURL *curl = curl_easy_init();

    /* callback is called before download of concrete file started */
    curl_easy_setopt(curl, CURLOPT_CHUNK_BGN_FUNCTION, file_is_coming);
    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)