NAME

CURLOPT_BUFFERSIZE − receive buffer size

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_BUFFERSIZE, long size);
```

DESCRIPTION

Pass a long specifying your preferred *size* (in bytes) for the receive buffer in libcurl. The main point of this would be that the write callback gets called more often and with smaller chunks. Secondly, for some protocols, there is a benefit of having a larger buffer for performance.

This is treated as a request, not an order. You cannot be guaranteed to actually get the given size.

This buffer size is by default *CURL_MAX_WRITE_SIZE* (16kB). The maximum buffer size allowed to be set is *CURL_MAX_READ_SIZE* (10MB). The minimum buffer size allowed to be set is 1024.

DO NOT set this option on a handle that is currently used for an active transfer as that may lead to unintended consequences.

The maximum size was 512kB until 7.88.0.

Starting in libcurl 8.7.0, there is a single transfer buffer allocated per multi handle. This buffer is used by all easy handles added to a multi handle no matter how many parallel transfers there are. The buffer remains allocated as long as there are active transfers.

DEFAULT

CURL_MAX_WRITE_SIZE (16kB)

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode result;
    curl_easy_setopt(curl, CURLOPT_URL, "sftp://example.com/foo.bin");

    /* ask libcurl to allocate a larger receive buffer */
    curl_easy_setopt(curl, CURLOPT_BUFFERSIZE, 120000L);

    result = curl_easy_perform(curl);

    curl_easy_cleanup(curl);
  }
}
```

AVAILABILITY

Added in curl 7.10

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_MAXFILESIZE(3), **CURLOPT_MAX_RECV_SPEED_LARGE**(3), **CURLOPT_UPLOAD_BUFFERSIZE**(3), **CURLOPT_WRITEFUNCTION**(3)