

**NAME**

CURLOPT\_PROGRESSFUNCTION – callback to progress meter function

**SYNOPSIS**

#include &lt;curl/curl.h&gt;

```
int progress_callback(void *clientp,
                      double dltotal,
                      double dlnow,
                      double ultotal,
                      double ulnow);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PROGRESSFUNCTION, progress_callback);
```

**DESCRIPTION**

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

We encourage users to use the newer *CURLOPT\_XFERINFOFUNCTION(3)* instead, if you can.

This function gets called by libcurl instead of its internal equivalent with a frequent interval. While data is being transferred it will be called very frequently, and during slow periods like when nothing is being transferred it can slow down to about one call per second.

*clientp* is the pointer set with *CURLOPT\_PROGRESSDATA(3)*, it is not used by libcurl but is only passed along from the application to the callback.

The callback gets told how much data libcurl will transfer and has transferred, in number of bytes. *dltotal* is the total number of bytes libcurl expects to download in this transfer. *dlnow* is the number of bytes downloaded so far. *ultotal* is the total number of bytes libcurl expects to upload in this transfer. *ulnow* is the number of bytes uploaded so far.

Unknown/unused argument values passed to the callback will be set to zero (like if you only download data, the upload size will remain 0). Many times the callback will be called one or more times first, before it knows the data sizes so a program must be made to handle that.

Returning a non-zero value from this callback will cause libcurl to abort the transfer and return *CURLE\_ABORTED\_BY\_CALLBACK*.

If you transfer data with the multi interface, this function will not be called during periods of idleness unless you call the appropriate libcurl function that performs transfers.

*CURLOPT\_NOPROGRESS(3)* must be set to 0 to make this function actually get called.

**DEFAULT**

By default, libcurl has an internal progress meter. That's rarely wanted by users.

**PROTOCOLS**

All

**EXAMPLE**

<http://curl.haxx.se/libcurl/c/progressfunc.html>

**AVAILABILITY**

Always

**RETURN VALUE**

Returns *CURLE\_OK*.

CURLOPT\_PROGRESSFUNCTION(3)      curl\_easy\_setopt options      CURLOPT\_PROGRESSFUNCTION(3)

**SEE ALSO**

**CURLOPT\_VERBOSE(3), CURLOPT\_NOPROGRESS(3),**