

**NAME**

curl\_easy\_perform - Perform a file transfer

**SYNOPSIS**

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

```
CURLcode curl_easy_perform(CURL *handle);
```

**DESCRIPTION**

This function is called after the init and all the curl\_easy\_setopt() calls are made, and will perform the transfer as described in the options. It must be called with the same *handle* as input as the curl\_easy\_init call returned.

libcurl version 7.7 or later (for older versions see below): You can do any amount of calls to curl\_easy\_perform() while using the same handle. If you intend to transfer more than one file, you are even encouraged to do so. libcurl will then attempt to re-use the same connection for the following transfers, thus making the operations faster, less CPU intense and using less network resources. Just note that you will have to use *curl\_easy\_setopt* between the invokes to set options for the following curl\_easy\_perform.

You must never call this function simultaneously from two places using the same handle. Let the function return first before invoking it another time. If you want parallel transfers, you must use several curl handles.

Before libcurl version 7.7: You are only allowed to call this function once using the same handle. If you want to do repeated calls, you must call curl\_easy\_cleanup and curl\_easy\_init again first.

**RETURN VALUE**

0 means everything was ok, non-zero means an error occurred as *<curl/curl.h>* defines. If the CURLOPT\_ERRORBUFFER was set with *curl\_easy\_setopt* there will be a readable error message in the error buffer when non-zero is returned.

**SEE ALSO**

**curl\_easy\_init(3), curl\_easy\_setopt(3),**

**BUGS**

Surely there are some, you tell me!