

**NAME**

curl\_easy\_perform - perform a blocking file transfer

**SYNOPSIS**

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

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

**DESCRIPTION**

Invoke this function after *curl\_easy\_init(3)* and all the *curl\_easy\_setopt(3)* calls are made, and will perform the transfer as described in the options. It must be called with the same **easy\_handle** as input as the *curl\_easy\_init(3)* call returned.

*curl\_easy\_perform(3)* performs the entire request in a blocking manner and returns when done, or if it failed. For non-blocking behavior, see *curl\_multi\_perform(3)*.

You can do any amount of calls to *curl\_easy\_perform(3)* while using the same **easy\_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(3)* 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 **easy\_handle**. Let the function return first before invoking it another time. If you want parallel transfers, you must use several *curl\_easy\_handles*.

While the **easy\_handle** is added to a multi handle, it cannot be used by *curl\_easy\_perform(3)*.

**RETURN VALUE**

CURLE\_OK (0) means everything was ok, non-zero means an error occurred as <curl/curl.h> defines - see *libcurl-errors(3)*. If the **CURLOPT\_ERRORBUFFER(3)** was set with *curl\_easy\_setopt(3)* there will be a readable error message in the error buffer when non-zero is returned.

**EXAMPLE**

```
CURL *curl = curl_easy_init();
if(curl) {
    CURLcode res;
    curl_easy_setopt(curl, CURLOPT_URL, "http://example.com");
    res = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
}
```

**SEE ALSO**

*curl\_easy\_init(3)*, *curl\_easy\_setopt(3)*, *curl\_multi\_add\_handle(3)*, *curl\_multi\_perform(3)*, *libcurl-errors(3)*,