

NAME

curl_easy_cleanup - End a libcurl easy handle

SYNOPSIS

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

```
void curl_easy_cleanup(CURL *handle);
```

DESCRIPTION

This function must be the last function to call for an easy session. It is the opposite of the *curl_easy_init(3)* function and must be called with the same *handle* as input that a *curl_easy_init(3)* call returned.

This might close all connections this handle has used and possibly has kept open until now - unless it was attached to a multi handle while doing the transfers. Don't call this function if you intend to transfer more files, re-using handles is a key to good performance with libcurl.

Occasionally you may get your progress callback or header callback called from within *curl_easy_cleanup(3)* (if previously set for the handle using *curl_easy_setopt(3)*). Like if libcurl decides to shut down the connection and the protocol is of a kind that requires a command/response sequence before disconnect. Examples of such protocols are FTP, POP3 and IMAP.

Any use of the **handle** after this function has been called and have returned, is illegal. *curl_easy_cleanup(3)* kills the handle and all memory associated with it!

For libcurl versions before 7.17.0, after you've called this function, you can safely remove all the strings you've previously told libcurl to use, as it won't use them anymore now.

RETURN VALUE

None

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),