

NAME

CURLOPT_UNRESTRICTED_AUTH – send credentials to other hosts too

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_UNRESTRICTED_AUTH,
                           long goahead);
```

DESCRIPTION

Set the long *goahead* parameter to 1L to make libcurl continue to send authentication (user+password) credentials or explicitly set cookie headers when following locations, even when the host changes. This option is meaningful only when setting *CURLOPT_FOLLOWLOCATION(3)*.

Further, when this option is not used or set to 0L, libcurl does not send custom nor internally generated *Authentication:* or *Cookie:* headers on requests done to other hosts than the one used for the initial URL. Another host means that one or more of hostname, protocol scheme or port number changed.

By default, libcurl only sends *Authentication:* or explicitly set *Cookie:* headers to the initial host as given in the original URL, to avoid leaking username + password to other sites.

This option should be used with caution: when curl follows redirects it blindly fetches the next URL as instructed by the server. Setting *CURLOPT_UNRESTRICTED_AUTH(3)* to 1L makes curl trust the server and sends possibly sensitive credentials to any host the server points to, possibly again and again as the following hosts can keep redirecting to new hosts.

Due to the way HTTP works, almost any header can be made to contain data a client may not want to pass on to other servers than the initially intended host and for all other headers than the two mentioned above, there is no protection from this happening when libcurl is told to follow redirects.

DEFAULT

0

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");
        curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);
        curl_easy_setopt(curl, CURLOPT_UNRESTRICTED_AUTH, 1L);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.10.4

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

CURLINFO_REDIRECT_COUNT(3), **CURLOPT_FOLLOWLOCATION**(3), **CURLOPT_MAXREDIRS**(3), **CURLOPT_REDIRECT_PROTOCOLS_STR**(3), **CURLOPT_USERPWD**(3)