

NAME

CURLOPT_NETRC – enable use of .netrc

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_NETRC, long level);
```

DESCRIPTION

This parameter controls the preference *level* of libcurl between using usernames and passwords from your *~/.netrc* file, relative to usernames and passwords in the URL supplied with *CURLOPT_URL(3)*. If the *NETRC* environment variable is set, that filename is used as the netrc file. (Added in 8.16.0)

On Windows, libcurl primarily checks for *.netrc* in *%HOME%*. If *%HOME%* is not set on Windows, libcurl falls back to *%USERPROFILE%*. If the file does not exist, it falls back to check if there is instead a file named *_netrc* – using an underscore instead of period.

You can also tell libcurl a different filename to use with *CURLOPT_NETRC_FILE(3)*.

libcurl uses a username (and supplied or prompted password) supplied with *CURLOPT_USERPWD(3)* or *CURLOPT_USERNAME(3)* in preference to any of the options controlled by this parameter.

Only machine name, username and password are taken into account (init macros and similar things are not supported).

The netrc file provides credentials for a hostname independent of which protocol and port number that are used.

When providing a username in the URL and a *.netrc* file, libcurl looks for and uses the password for that specific user for the given host if such an entry appears in the file before a "generic" *machine* entry without *login* specified.

libcurl does not verify that the file has the correct properties set (as the standard Unix ftp client does). It should only be readable by user.

level is a long that should be set to one of the values described below.

CURL_NETRC_IGNORED (0)

libcurl ignores the *.netrc* file. This is the default.

CURL_NETRC_OPTIONAL (1)

The use of the *.netrc* file is optional, and information in the URL is to be preferred. The file is scanned for the host and username (to find the password only) or for the host only, to find the first username and password after that *machine*, which ever information is not specified.

CURL_NETRC_REQUIRED (2)

The use of the *.netrc* file is required, and any credential information present in the URL is ignored. The file is scanned for the host and username (to find the password only) or for the host only, to find the first username and password after that *machine*, which ever information is not specified.

FILE FORMAT

The *.netrc* file format is simple: you specify lines with a machine name and follow the login and password that are associated with that machine.

Each field is provided as a sequence of letters that ends with a space or newline. Starting in 7.84.0, libcurl also supports quoted strings. They start and end with double quotes and support the escaped special letters *", n, r, and t*. Quoted strings are the only way a space character can be used in a username or password.

machine <name>

Provides credentials for a host called **name**. libcurl searches the .netrc file for a machine token that matches the hostname specified in the URL. Once a match is made, the subsequent tokens are processed, stopping when the end of file is reached or another "machine" is encountered.

default This is the same as machine name except that default matches any name. There can be only one default token, and it must be after all machine tokens. To provide a default anonymous login for hosts that are not otherwise matched, add a line similar to this in the end:

```
default login anonymous password user@domain
```

login <name>

The username string for the remote machine.

password <secret>

Supply a password. If this token is present, curl supplies the specified string if the remote server requires a password as part of the login process. Note that if this token is present in the .netrc file you really should make sure the file is not readable by anyone besides the user.

macrodef <name>

Define a macro. This feature is not supported by libcurl. In order for the rest of the .netrc to still work fine, libcurl properly skips every definition done with "macrodef" that it finds.

DEFAULT

CURL_NETRC_IGNORED

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "ftp://example.com/");
        curl_easy_setopt(curl, CURLOPT_NETRC, CURL_NETRC_OPTIONAL);
        result = curl_easy_perform(curl);
    }
}
```

HISTORY

CURL_NETRC_* enums became *long* types in 8.13.0, prior to this version a *long* cast was necessary when passed to *curl_easy_setopt(3)*.

AVAILABILITY

Added in curl 7.1

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

CURLOPT_NETRC_FILE(3), **CURLOPT_USERNAME(3)**, **CURLOPT_USERPWD(3)**