

**NAME**

CURLOPT\_WILDCARDMATCH – enable directory wildcard transfers

**SYNOPSIS**

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_WILDCARDMATCH, long onoff);
```

**DESCRIPTION**

Set *onoff* to 1 if you want to transfer multiple files according to a file name pattern. The pattern can be specified as part of the *CURLOPT\_URL(3)* option, using an fnmatch-like pattern (Shell Pattern Matching) in the last part of URL (file name).

By default, libcurl uses its internal wildcard matching implementation. You can provide your own matching function by the *CURLOPT\_FNMATCH\_FUNCTION(3)* option.

A brief introduction of its syntax follows:

\* - ASTERISK

ftp://example.com/some/path/\*.txt (for all txt's from the root directory)

? - QUESTION MARK

Question mark matches any (exactly one) character.

ftp://example.com/some/path/photo?.jpeg

[ - BRACKET EXPRESSION

The left bracket opens a bracket expression. The question mark and asterisk have no special meaning in a bracket expression. Each bracket expression ends by the right bracket and matches exactly one character. Some examples follow:

[a-zA-Z0-9] or [f-gF-G] – character interval

[abc] - character enumeration

[^abc] or [!abc] - negation

[[:name:]] class expression. Supported classes are **alnum**, **lower**, **space**, **alpha**, **digit**, **print**, **upper**, **blank**, **graph**, **xdigit**.

[[-!^] - special case – matches only '-', '!', '^'. These characters have no special purpose.

[[]\] - escape syntax. Matches '[', ']' or '\'

Using the rules above, a file name pattern can be constructed:

ftp://example.com/some/path/[a-z[:upper:]]\].jpeg

**PROTOCOLS**

This feature is only supported for FTP download.

**EXAMPLE**

See <http://curl.haxx.se/libcurl/c/ftp-wildcard.html>

**AVAILABILITY**

Added in 7.21.0

**RETURN VALUE**

Returns CURLE\_OK if the option is supported, and CURLE\_UNKNOWN\_OPTION if not.

**SEE ALSO**

**CURLOPT\_FNMATCH\_FUNCTION(3), CURLOPT\_URL(3),**