

NAME

CURLOPT_FTP_CREATE_MISSING_DIRS – create missing directories for FTP and SFTP

SYNOPSIS

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

```
#define CURLFTP_CREATE_DIR_NONE 0L
#define CURLFTP_CREATE_DIR 1L
#define CURLFTP_CREATE_DIR_RETRY 2L
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_FTP_CREATE_MISSING_DIRS,
                           long create);
```

DESCRIPTION

Pass a long telling libcurl to *create* the dir. If the value is *CURLFTP_CREATE_DIR* (1), libcurl may create any remote directory that it fails to "move" into.

For FTP requests, that means a CWD command fails. CWD being the command that changes working directory.

For SFTP requests, libcurl may create the remote directory if it cannot obtain a handle to the target-location. The creation fails if a file of the same name as the directory to create already exists or lack of permissions prevents creation.

Setting *create* to *CURLFTP_CREATE_DIR_RETRY* (2), tells libcurl to retry the CWD command again if the subsequent **MKD** command fails. This is especially useful if you are doing many simultaneous connections against the same server and they all have this option enabled, as then CWD may first fail but then another connection does **MKD** before this connection and thus **MKD** fails but trying CWD works.

DEFAULT

CURLFTP_CREATE_DIR_NONE (0)

PROTOCOLS

This functionality affects ftp only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL,
                        "ftp://example.com/non-existing/new.txt");
        curl_easy_setopt(curl, CURLOPT_FTP_CREATE_MISSING_DIRS,
                        CURLFTP_CREATE_DIR_RETRY);

        result = curl_easy_perform(curl);

        curl_easy_cleanup(curl);
    }
}
```

HISTORY

CURLFTP_CREATE_* enums became *long* types in 8.16.0, prior to this version a *long* cast was necessary when passed to *curl_easy_setopt*(3).

AVAILABILITY

Added in curl 7.10.7

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_FTP_FILEMETHOD(3), **CURLOPT_FTP_USE_EPSV**(3)