

NAME

CURLOPT_FNMATCH_FUNCTION – wildcard match callback

SYNOPSIS

#include <curl/curl.h>

```
int fnmatch_callback(void *ptr,
                    const char *pattern,
                    const char *string);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_FNMATCH_FUNCTION,
                          fnmatch_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

This callback is used for wildcard matching.

Return *CURL_FNMATCHFUNC_MATCH* if pattern matches the string, *CURL_FNMATCHFUNC_NO-MATCH* if not or *CURL_FNMATCHFUNC_FAIL* if an error occurred.

DEFAULT

NULL == an internal function for wildcard matching.

PROTOCOLS

This functionality affects ftp only

EXAMPLE

```
extern int string_match(const char *s1, const char *s2);

struct local_stuff {
    void *custom;
};
static int my_fnmatch(void *clientp,
                    const char *pattern, const char *string)
{
    struct local_stuff *data = clientp;
    printf("my pointer: %p\n", data->custom);
    if(string_match(pattern, string))
        return CURL_FNMATCHFUNC_MATCH;
    else
        return CURL_FNMATCHFUNC_NOMATCH;
}

int main(void)
{
    struct local_stuff local_data;
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "ftp://ftp.example.com/file*");
        curl_easy_setopt(curl, CURLOPT_WILDCARDMATCH, 1L);
        curl_easy_setopt(curl, CURLOPT_FNMATCH_FUNCTION, my_fnmatch);
        curl_easy_setopt(curl, CURLOPT_FNMATCH_DATA, &local_data);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

}

AVAILABILITY

Added in curl 7.21.0

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_DEBUGFUNCTION(3), CURLOPT_FNMATCH_DATA(3), CURLOPT_WILDCARD-MATCH(3)