

NAME

CURLOPT_REDIR_PROTOCOLS_STR – protocols allowed to redirect to

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_REDIR_PROTOCOLS_STR,
                           char *spec);
```

DESCRIPTION

Pass a pointer to a string that holds a comma-separated list of case insensitive protocol names (URL schemes). That list limits what protocols libcurl may use in a transfer that it follows to in a redirect when *CURLOPT_FOLLOWLOCATION(3)* is enabled. This option allows applications to limit specific transfers to only be allowed to use a subset of protocols in redirections.

Protocols denied by *CURLOPT_PROTOCOLS_STR(3)* are not overridden by this option.

By default libcurl allows HTTP, HTTPS, FTP and FTPS on redirects (since 7.65.2).

These are the available protocols:

DICT, FILE, FTP, FTPS, GOPHER, GOPHERS, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, MQTT,
 MQTTS, POP3, POP3S, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET, TFTP, WS, WSS

You can set "ALL" as a short-cut to enable all protocols. Note that by setting all, you may enable protocols that were not supported the day you write this but are introduced in a future libcurl version.

If trying to set a non-existing protocol or if no matching protocol at all is set, it returns error.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to restore to internal default.

The application does not have to keep the string around after setting this option.

DEFAULT

HTTP, HTTPS, FTP and FTPS

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(int argc, char **argv)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        /* pass in the URL from an external source */
        curl_easy_setopt(curl, CURLOPT_URL, argv[1]);

        /* only allow redirects to HTTP and HTTPS URLs */
        curl_easy_setopt(curl, CURLOPT_REDIR_PROTOCOLS_STR, "http,https");

        /* Perform the request */
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.85.0

RETURN VALUE

Returns `CURLE_UNKNOWN_OPTION` if the option is not implemented, `CURLE_UNSUPPORTED_PROTOCOL` if a listed protocol is not supported or disabled, `CURLE_BAD_FUNCTION_ARGUMENT` if no protocol is listed else `CURLE_OK`.

SEE ALSO

`CURLINFO_SCHEME(3)`, `CURLOPT_DEFAULT_PROTOCOL(3)`, `CURLOPT_PROTOCOLS(3)`, `CURLOPT_PROTOCOLS_STR(3)`, `CURLOPT_REDIR_PROTOCOLS(3)`