NAME

CURLOPT_REDIR_PROTOCOLS − set protocols allowed to redirect to

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_REDIR_PROTOCOLS, long bitmask);
```

DESCRIPTION

Pass a long that holds a bitmask of CURLPROTO_* defines. If used, this bitmask limits what protocols libcurl may use in a transfer that it follows to in a redirect when *CURLOPT_FOLLOWLOCATION(3)* is enabled. This allows you to limit specific transfers to only be allowed to use a subset of protocols in redirections.

By default libcurl will allow all protocols on redirect except several disabled for security reasons: Since 7.19.4 FILE and SCP are disabled, and since 7.40.0 SMB and SMBS are also disabled. *CURLPROTO_ALL* enables all protocols on redirect, including those disabled for security.

These are the available protocol defines:
CURLPROTO_DICT
CURLPROTO_FILE
CURLPROTO_FTP
CURLPROTO_FTPS
CURLPROTO_GOPHER
CURLPROTO_HTTP
CURLPROTO_HTTPS
CURLPROTO_IMAP
CURLPROTO_IMAPS
CURLPROTO_LDAP
CURLPROTO_LDAPS
CURLPROTO_POP3
CURLPROTO_POP3S
CURLPROTO_RTMP
CURLPROTO_RTMPE
CURLPROTO_RTMPS
CURLPROTO_RTMPT
CURLPROTO_RTMPTE
CURLPROTO_RTMPTS
CURLPROTO_RTSP
CURLPROTO_SCP
CURLPROTO_SFTP
CURLPROTO_SMB
CURLPROTO_SMBS
CURLPROTO_SMTP
CURLPROTO_SMTPS
CURLPROTO_TELNET
CURLPROTO_TFTP

DEFAULT

All protocols except for FILE, SCP and since 7.40.0 SMB and SMBS.

PROTOCOLS

All

EXAMPLE

```
curl = curl_easy_init();
if(curl) {
  /* 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,
                   CURLPROTO_HTTP | CURLPROTO_HTTPS);

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

AVAILABILITY

Added in 7.19.4, before then it would follow all protocols.

RETURN VALUE

Returns CURLE_OK if the option is supported, and CURLE_UNKNOWN_OPTION if not.

SEE ALSO

CURLOPT_PROTOCOLS(3),