

NAME

CURLOPT_PROTOCOLS – set allowed protocols

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_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 the transfer. This allows you to have a libcurl built to support a wide range of protocols but still limit specific transfers to only be allowed to use a subset of them. By default libcurl will accept all protocols it supports (`CURLPROTO_ALL`). See also `CURLOPT_REDIR_PROTOCOLS(3)`.

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_RTMPPE
CURLPROTO_RTMPFS
CURLPROTO_RTMPPT
CURLPROTO_RTMPTE
CURLPROTO_RTMPFS
CURLPROTO_RTSP
CURLPROTO_SCP
CURLPROTO_SFTP
CURLPROTO_SMTP
CURLPROTO_SMTPS
CURLPROTO_TELNET
CURLPROTO_TFTP

DEFAULT

All protocols built-in

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 HTTP, TFTP and SFTP */
    curl_easy_setopt(curl, CURLOPT_PROTOCOLS,
                     CURLPROTO_HTTP | CURLPROTO_TFTP | CURLPROTO_SFTP);
}
```

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

AVAILABILITY

Added in 7.19.4

RETURN VALUE

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

SEE ALSO

CURLOPT_REDIRECT_PROTOCOLS(3), CURLOPT_URL(3),