

NAME

CURLOPT_PROTOCOLS_STR – allowed protocols

SYNOPSIS

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

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

DESCRIPTION

Pass a pointer to a string that holds a comma-separated list of case insensitive protocol names (URL schemes) to allow in the transfer. This option allows applications to use 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 accepts all protocols it was built with support for. See also *CURLOPT_REDIR_PROTOCOLS_STR(3)*.

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

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.

curl_version_info(3) can be used to get a list of all supported protocols in the current libcurl. *CURLINFO_SCHEME(3)* is the recommended way to figure out the protocol used in a previous transfer.

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

DEFAULT

All protocols built-in

PROTOCOLS

This functionality affects all supported protocols

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 HTTP, TFTP and SFTP */
        curl_easy_setopt(curl, CURLOPT_PROTOCOLS_STR, "http,tftp,sftp");

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

HISTORY

RTMP and its related protocol schemes are not supported since curl 8.20.0

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_REDIR_PROTOCOLS_STR(3), CURLOPT_URL(3), curl_version_info(3)