

**NAME**

CURLOPT\_DEFAULT\_PROTOCOL – default protocol to use if the URL is missing a scheme name

**SYNOPSIS**

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_DEFAULT_PROTOCOL, char *protocol);
```

**DESCRIPTION**

This option tells libcurl to use *protocol* if the URL is missing a scheme name.

Use one of these protocol (scheme) names:

dict, file, ftp, ftps, gopher, http, https, imap, imaps, ldap, ldaps, pop3, pop3s, rtsp, scp, sftp, smb, smbs, smtp, smtps, telnet, tftp

An unknown or unsupported protocol causes error *CURLE\_UNSUPPORTED\_PROTOCOL* when libcurl parses a schemeless URL. Parsing happens when *curl\_easy\_perform(3)* or *curl\_multi\_perform(3)* is called. The protocols supported by libcurl will vary depending on how it was built. Use *curl\_version\_info(3)* if you need a list of protocol names supported by the build of libcurl that you are using.

This option does not change the default proxy protocol (http).

Without this option libcurl would make a guess based on the host, see *CURLOPT\_URL(3)* for details.

**DEFAULT**

NULL (make a guess based on the host)

**PROTOCOLS**

All

**EXAMPLE**

```
curl = curl_easy_init();
if(curl) {
    /* set a URL without a scheme */
    curl_easy_setopt(curl, CURLOPT_URL, "example.com");

    /* set the default protocol (scheme) for schemeless URLs */
    curl_easy_setopt(curl, CURLOPT_DEFAULT_PROTOCOL, "https");

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

**AVAILABILITY**

Added in 7.45.0

**RETURN VALUE**

CURLE\_OK if the option is supported.

CURLE\_OUT\_OF\_MEMORY if there was insufficient heap space.

CURLE\_UNKNOWN\_OPTION if the option is not supported.

**SEE ALSO**

**CURLOPT\_URL(3)**,