

NAME

CURLOPT_ALTSVC_CTRL – control alt-svc behavior

SYNOPSIS

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

```
#define CURLALTSVC_READONLYFILE (1L << 2)
#define CURLALTSVC_H1           (1L << 3)
#define CURLALTSVC_H2           (1L << 4)
#define CURLALTSVC_H3           (1L << 5)
```

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

DESCRIPTION

Populate the long *bitmask* with the correct set of features to instruct libcurl how to handle Alt-Svc for the transfers using this handle.

libcurl only accepts Alt-Svc headers over HTTPS. It also only completes a request to an alternative origin if that origin is properly hosted over HTTPS. These requirements are there to make sure both the source and the destination are legitimate.

Alternative services are only used when setting up new connections. If there exists an existing connection to the host in the connection pool, then that is preferred.

If *CURLOPT_ALTSVC(3)* is set, *CURLOPT_ALTSVC_CTRL(3)* gets a default value corresponding to *CURLALTSVC_H1* | *CURLALTSVC_H2* | *CURLALTSVC_H3* – the HTTP/2 and HTTP/3 bits are only set if libcurl was built with support for those versions.

Setting any bit enables the alt-svc engine.

CURLALTSVC_READONLYFILE

Do not write the alt-svc cache back to the file specified with *CURLOPT_ALTSVC(3)* even if it gets updated. By default a file specified with that option is read and written to as deemed necessary.

CURLALTSVC_H1

Accept alternative services offered over HTTP/1.1.

CURLALTSVC_H2

Accept alternative services offered over HTTP/2. This is only used if libcurl was also built to actually support HTTP/2, otherwise this bit is ignored.

CURLALTSVC_H3

Accept alternative services offered over HTTP/3. This is only used if libcurl was also built to actually support HTTP/3, otherwise this bit is ignored.

DEFAULT

0 – Alt-Svc handling is disabled

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_ALTSVC_CTRL, CURLALTSVC_H1);
        curl_easy_setopt(curl, CURLOPT_ALTSVC, "altsvc-cache.txt");
```

```
    result = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
}
```

HISTORY

CURLALTSVC_* macros became *long* types in 8.16.0, prior to this version a *long* cast was necessary when passed to *curl_easy_setopt(3)*.

AVAILABILITY

Added in curl 7.64.1

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*.

SEE ALSO

CURLOPT_ALTSVC(3), **CURLOPT_CONNECT_TO(3)**, **CURLOPT_RESOLVE(3)**