

NAME

CURLOPT_HSTSWRITEFUNCTION – write callback for HSTS hosts

SYNOPSIS

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

struct curl_hstsentry {
    char *name;
    size_t namelen;
    unsigned int includeSubDomains:1;
    char expire[18]; /* YYYYMMDD HH:MM:SS [null-terminated] */
};

struct curl_index {
    size_t index; /* the provided entry's "index" or count */
    size_t total; /* total number of entries to save */
};

CURLSTSCode hstswrite(CURL *easy, struct curl_hstsentry *sts,
                     struct curl_index *count, void *clientp);

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_HSTSWRITEFUNCTION, hstswrite);
```

DESCRIPTION

Pass a pointer to your callback function, as the prototype shows above.

This callback function gets called by libcurl repeatedly to allow the application to store the in-memory HSTS cache when libcurl is about to discard it.

Set the *clientp* argument with the *CURLOPT_HSTS_WRITEDATA(3)* option or it is NULL. When the callback is invoked, the *sts* pointer points to a populated struct: Read the hostname to 'name' (it is *namelen* bytes long and null terminated. The *includeSubDomains* field is non-zero if the entry matches subdomains. The *expire* string is a date stamp null-terminated string using the syntax YYYYMMDD HH:MM:SS.

The callback should return *CURLSTS_OK* if it succeeded and is prepared to be called again (for another host) or *CURLSTS_DONE* if there is nothing more to do. It can also return *CURLSTS_FAIL* to signal error.

This option does not enable HSTS, you need to use *CURLOPT_HSTS_CTRL(3)* to do that.

DEFAULT

NULL – no callback.

PROTOCOLS

This functionality affects http only

EXAMPLE

```
struct priv {
    void *custom;
};

static CURLSTSCode hswr_cb(CURL *easy, struct curl_hstsentry *sts,
                          struct curl_index *count, void *clientp)
{
    /* save the passed in HSTS data somewhere */
    return CURLSTS_OK;
}
```

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        struct priv my_stuff;
        CURLcode result;

        /* set HSTS read callback */
        curl_easy_setopt(curl, CURLOPT_HSTSWRITEFUNCTION, hswr_cb);

        /* pass in suitable argument to the callback */
        curl_easy_setopt(curl, CURLOPT_HSTSWRITEDATA, &my_stuff);

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

AVAILABILITY

Added in curl 7.74.0

RETURN VALUE

This returns CURLE_OK.

SEE ALSO

CURLOPT_HSTS(3), CURLOPT_HSTSWRITEDATA(3), CURLOPT_HSTSWRITEFUNCTION(3), CURLOPT_HSTS_CTRL(3)