

NAME

CURLOPT_CAPATH – directory holding CA certificates

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_CAPATH, char *capath);
```

DESCRIPTION

Pass a char pointer to a null-terminated string naming a directory holding multiple CA certificates to verify the peer with. If libcurl is built against OpenSSL, the certificate directory must be prepared using the OpenSSL `c_rehash` utility. This makes sense only when used in combination with the `CURLOPT_SSL_VERIFYPEER(3)` option.

The `CURLOPT_CAPATH(3)` function apparently does not work in Windows due to some limitation in OpenSSL.

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

The default value for this can be figured out with `CURLINFO_CAPATH(3)`.

DEFAULT

A path detected at build time.

PROTOCOLS

This functionality affects all TLS based protocols: HTTPS, FTPS, IMAPS, POP3S, SMTPS etc.

This option works only with the following TLS backends: GnuTLS, OpenSSL, mbedTLS and wolfSSL

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        curl_easy_setopt(curl, CURLOPT_CAPATH, "/etc/cert-dir");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.9.8

RETURN VALUE

CURLE_OK if supported; or an error such as:

CURLE_NOT_BUILT_IN – Not supported by the SSL backend

CURLE_UNKNOWN_OPTION

CURLE_OUT_OF_MEMORY

SEE ALSO

CURLINFO_CAPATH(3), **CURLOPT_CAINFO**(3), **CURLOPT_DEBUGFUNCTION**(3), **CURLOPT_STDERR**(3)