

NAME

CURLOPT_PROXY_ISSUERCERT – proxy issuer SSL certificate filename

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PROXY_ISSUERCERT, char *file);
```

DESCRIPTION

Pass a char pointer to a null-terminated string naming a *file* holding a CA certificate in PEM format. If the option is set, an additional check against the peer certificate is performed to verify the issuer of the HTTPS proxy is indeed the one associated with the certificate provided by the option. This additional check is useful in multi-level PKI where one needs to enforce that the peer certificate is from a specific branch of the tree.

This option makes sense only when used in combination with the *CURLOPT_PROXY_SSL_VERIFYPEER(3)* option. Otherwise, the result of the check is not considered as failure.

A specific error code (CURLE_SSL_ISSUER_ERROR) is defined with the option, which is returned if the setup of the SSL/TLS session has failed due to a mismatch with the issuer of peer certificate (*CURLOPT_PROXY_SSL_VERIFYPEER(3)* has to be set too for the check to fail).

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.

DEFAULT

NULL

PROTOCOLS

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

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

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        /* using an HTTPS proxy */
        curl_easy_setopt(curl, CURLOPT_PROXY, "https://proxy.example:443");
        curl_easy_setopt(curl, CURLOPT_PROXY_ISSUERCERT, "/etc/certs/cacert.pem");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.71.0

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_ISSUERCERT(3), CURLOPT_PROXY_SSL_VERIFYHOST(3), CUR-
LOPT_PROXY_SSL_VERIFYPEER(3), CURLOPT_SSL_VERIFYHOST(3), CUR-
LOPT_SSL_VERIFYPEER(3)