

NAME

CURLOPT_PROXY_KEYPASSWD – passphrase for the proxy private key

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PROXY_KEYPASSWD, char *pwd);
```

DESCRIPTION

This option is for connecting to an HTTPS proxy, not an HTTPS server.

Pass a pointer to a null-terminated string as parameter. It is used as the password required to use the *CURLOPT_PROXY_SSLKEY(3)* private key. You never need a passphrase to load a certificate but you need one to load your private key.

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: OpenSSL, Schannel, 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/foo.bin");
        curl_easy_setopt(curl, CURLOPT_PROXY, "https://proxy.example:443");
        curl_easy_setopt(curl, CURLOPT_PROXY_KEYPASSWD, "superman");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.52.0

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_KEYPASSWD(3), CURLOPT_PROXY_SSLKEY(3), CURLOPT_SSH_PRIVATE_KEYFILE(3), CURLOPT_SSLKEY(3)