NAME

CURLOPT_KEYPASSWD − passphrase to private key

SYNOPSIS

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

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

DESCRIPTION

Pass a pointer to a null−terminated string as parameter. It is used as the password required to use the *CURLOPT_SSLKEY(3)* or *CURLOPT_SSH_PRIVATE_KEYFILE(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_SSLCERT, "client.pem");
  curl_easy_setopt(curl, CURLOPT_SSLKEY, "key.pem");
  curl_easy_setopt(curl, CURLOPT_KEYPASSWD, "superman");
  result = curl_easy_perform(curl);
  curl_easy_cleanup(curl);
 }
}
```

HISTORY

This option was known as CURLOPT_SSLKEYPASSWD up to 7.16.4 and CURLOPT_SSLCERTPASSWD up to 7.9.2.

AVAILABILITY

Added in curl 7.17.0

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_SSH_PRIVATE_KEYFILE(3), **CURLOPT_SSLKEY**(3)