

NAME

CURLOPT_SSH_PRIVATE_KEYFILE – private key file for SSH auth

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSH_PRIVATE_KEYFILE,
                           char *filename);
```

DESCRIPTION

Pass a char pointer pointing to a *filename* for your private key. If not used, libcurl defaults to **\$HOME/.ssh/id_rsa** or **\$HOME/.ssh/id_dsa** if the HOME environment variable is set, and in the current directory if HOME is not set.

If the file is password-protected, set the password with *CURLOPT_KEYPASSWD(3)*.

The SSH library derives the public key from this private key when possible. If the SSH library cannot derive the public key from the private one and no public one is provided with *CURLOPT_SSH_PUBLIC_KEYFILE(3)*, the transfer fails.

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

DEFAULT

As explained above

PROTOCOLS

This functionality affects scp and sftp

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "sftp://example.com/file");
        curl_easy_setopt(curl, CURLOPT_SSH_PRIVATE_KEYFILE,
                          "/home/clarkkent/.ssh/id_rsa");
        curl_easy_setopt(curl, CURLOPT_KEYPASSWD, "password");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.16.1

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_AUTH_TYPES(3), **CURLOPT_SSH_PUBLIC_KEYFILE(3)**