

# SSH external for LiveCode

This external uses libssh library (<http://www.libssh.org/>) to provide SSH functions to LiveCode programs.

## Contents of the package

livessh/ — this directory contains copy of libssh with compiled binaries.

binaries/ — this directory contains external builds for Linux (livessh.so) and Windows (livessh.dll).

livessh.c — external source code.

Makefile — build scenario for external.

demos/ — LiveCode demos

## Usage

To use the external, you'll need to load its binary (livessh.so for Linux, livessh.dll for Windows) in Livecode. You also need libssh.dll to be either placed in directory where you run LiveCode or to copy it to WINDOWS/System32 on Windows, on Linux you can install libssh using the package manager.

## Compiling

To compile the external on Windows, you need to install [MinGW32](#), then open a command line prompt in livessh directory and run

**make livessh.dll**

To compile the external on Linux, you need to install gcc, make, libssh with development headers, then run

**make livessh.so**

To compile the external on Mac OS X, you need to install libssh (for example, from [macports](#)), gcc and make, then run

**make livessh**

## Developing

The external exports all the functions from libssh, documentation on which you can see on its site: <http://api.libssh.org/stable/>. Arguments to function calls are translated as follows:

char \* — LiveCode string as is

int, unsigned int, etc. numbers — read from LiveCode string in according format.

Pointers — read from LiveCode string in platform-specific

Exceptions:

ssh\_channel\_read: second argument is name of LiveCode variable, which will be used for output

ssh\_channel\_write: second argument is name of LiveCode variable, which will be used for input

ssh\_channel\_read\_nonblocking: second argument is name of LiveCode variable, which will be used for output

ssh\_scp\_new: 3+ arguments are interpreted as flags combination; possible values are: "write", "read", "recursive"

ssh\_scp\_write: second argument is name of LiveCode variable, which will be used for input

ssh\_scp\_read: second argument is name of LiveCode variable, which will be used for output

sftp\_open: third and fourth arguments are swapped, 4+ arguments are interpreted as flags combination; possible values are "readonly", "writeonly", "readwrite", "create", "excl", "truncate".

sftp\_write: last variable is name of LiveCode variable, which will be used for input

ssh\_bind\_options\_set: second argument is a string. Its possible values:

LiveCode string	Original libssh constant
addr	SSH_BIND_OPTIONS_BINDADDR
log_verbosity	SSH_BIND_OPTIONS_LOG_VERBOSITY
port	SSH_BIND_OPTIONS_BINDPORT_STR
hostkey	SSH_BIND_OPTIONS_HOSTKEY
dsakey	SSH_BIND_OPTIONS_DSAKEY
banner	SSH_BIND_OPTIONS_BANNER

ssh\_message\_auth\_set\_methods: second argument is interpreted as flags combination; possible values are

unknown	SSH_AUTH_METHOD_UNKNOWN
password	SSH_AUTH_METHOD_PASSWORD
publickey	SSH_AUTH_METHOD_PUBLICKEY
hostbased	SSH_AUTH_METHOD_HOSTBASED
interactive	SSH_AUTH_METHOD_INTERACTIVE

ssh\_ssh\_options\_set: second argument is a string. Its possible values:

host	SSH_OPTIONS_HOST
port	SSH_OPTIONS_PORT_STR
bindaddr	SSH_OPTIONS_BINDADDR
user	SSH_OPTIONS_USER
knownhosts	SSH_OPTIONS_KNOWNHOSTS
identity	SSH_OPTIONS_IDENTITY
timeout	SSH_OPTIONS_TIMEOUT
timeout_usec	SSH_OPTIONS_TIMEOUT_USEC
ssh1	SSH_OPTIONS_SSH1
ssh2	SSH_OPTIONS_SSH2
log_verbosity	SSH_OPTIONS_LOG_VERBOSITY
ciphers_c_s	SSH_OPTIONS_CIPHERS_C_S
ciphers_s_c	SSH_OPTIONS_CIPHERS_S_C
compression_c_s	SSH_OPTIONS_COMPRESSION_C_S
compression_s_c	SSH_OPTIONS_COMPRESSION_S_C
proxycommand	SSH_OPTIONS_PROXYCOMMAND

sftp\_read: second argument is a name of LiveCode variable used for output.

sftp\_async\_read: second argument is a name of LiveCode variable used for output.

New function defined:

livessh\_run\_command(ssh\_session, command, outstr)

Runs a given command in shell and stores output in given variable.

Arguments:

ssh\_session: pointer to ssh session

command: string with command to run in shell

outstr: name of variable to use for command output.

Full list of exported functions and commands is available in API.txt

## Demos

Three demos are available in demos/ folder. How to use them: open in LiveCode, input path to SSH external binary in top-most field, press “Load”, input demo-specific data in other fields, press the lowest button. Don’t forget that libssh.dll should be either in current directory or in WINDOWS/System32 on windows for external to work.