

**NAME**

libssh2\_sftp\_symlink\_ex - read or set a symbolic link

**SYNOPSIS**

```
#include <libssh2.h>
```

```
#include <libssh2_sftp.h>
```

```
int
```

```
libssh2_sftp_symlink_ex(LIBSSH2_SFTP *sftp, const char *path,  
                        unsigned int path_len, char *target,  
                        unsigned int target_len, int link_type);
```

**DESCRIPTION**

Create a symlink or read out symlink information from the remote side.

*sftp* - SFTP instance as returned by **libssh2\_sftp\_init(3)**

*path* - Remote filesystem object to create a symlink from or resolve.

*path\_len* - Length of the name of the remote filesystem object to create a symlink from or resolve.

*target* - a pointer to a buffer. The buffer has different uses depending what the *link\_type* argument is set to.

**LIBSSH2\_SFTP\_SYMLINK**: Remote filesystem object to link to.

**LIBSSH2\_SFTP\_READLINK**: Pre-allocated buffer to resolve symlink target into.

**LIBSSH2\_SFTP\_REALPATH**: Pre-allocated buffer to resolve realpath target into.

*target\_len* - Length of the name of the remote filesystem target object.

*link\_type* - One of the three previously mentioned constants which determines the resulting behavior of this function.

These are convenience macros:

**libssh2\_sftp\_symlink(3)** : Create a symbolic link between two filesystem objects.

**libssh2\_sftp\_readlink(3)** : Resolve a symbolic link filesystem object to its next target.

**libssh2\_sftp\_realpath(3)** : Resolve a complex, relative, or symlinked filepath to its effective target.

**RETURN VALUE**

When using **LIBSSH2\_SFTP\_SYMLINK**, this function returns 0 on success or negative on failure.

When using `LIBSSH2_SFTP_READLINK` or `LIBSSH2_SFTP_REALPATH`, it returns the number of bytes it copied to the target buffer (not including the terminating zero) or negative on failure.

It returns `LIBSSH2_ERROR_EAGAIN` when it would otherwise block. While `LIBSSH2_ERROR_EAGAIN` is a negative number, it is not really a failure per se.

From 1.2.8, `LIBSSH2_ERROR_BUFFER_TOO_SMALL` is returned if the given 'target' buffer is too small to fit the requested object name.

## **BUG**

Passing in a too small buffer when receiving data only results in libssh2 1.2.7 or earlier to not copy the entire data amount, and it is not possible for the application to tell when it happens!

## **ERRORS**

*LIBSSH2\_ERROR\_ALLOC* - An internal memory allocation call failed.

*LIBSSH2\_ERROR\_SOCKET\_SEND* - Unable to send data on socket.

*LIBSSH2\_ERROR\_SOCKET\_TIMEOUT* -

*LIBSSH2\_ERROR\_SFTP\_PROTOCOL* - An invalid SFTP protocol response was received on the socket, or an SFTP operation caused an errorcode to be returned by the server.

## **SEE ALSO**

**libssh2\_sftp\_init(3)**