NAME

libssh2_channel_write_ex - write data to a channel stream blocking

SYNOPSIS

```
#include <libssh2.h>

ssize_t
libssh2_channel_write_ex(LIBSSH2_CHANNEL *channel,
                         int stream_id, char *buf,
                         size_t buflen);
```

DESCRIPTION

Write data to a channel stream. All channel streams have one standard I/O substream (stream_id == 0), and may have up to 2^32 extended data streams as identified by the selected *stream_id*. The SSH2 protocol currently defines a stream ID of 1 to be the stderr substream.

channel - active channel stream to write to.

stream_id - substream ID number (e.g. 0 or SSH_EXTENDED_DATA_STDERR)

buf - pointer to buffer to write

buflen - size of the data to write

libssh2_channel_write(3) and *libssh2_channel_write_stderr(3)* are convenience macros for this function.

libssh2_channel_write_ex(3) will use as much as possible of the buffer and put it into a single SSH protocol packet. This means that to get maximum performance when sending larger files, you should try to always pass in at least 32K of data to this function.

RETURN VALUE

Actual number of bytes written or negative on failure. LIBSSH2_ERROR_EAGAIN when it would otherwise block. While LIBSSH2_ERROR_EAGAIN is a negative number, it is not really a failure per se.

ERRORS

LIBSSH2_ERROR_ALLOC - An internal memory allocation call failed.

LIBSSH2_ERROR_SOCKET_SEND - Unable to send data on socket.

LIBSSH2_ERROR_CHANNEL_CLOSED - The channel has been closed.

LIBSSH2_ERROR_CHANNEL_EOF_SENT - The channel has been requested to be

LIBSSH2_ERROR_BAD_USE - This can be returned if you ignored a previous return for LIBSSH2_ERROR_EAGAIN and rather than sending the original buffer with the original size, you sent a new buffer with a different size.

closed.

SEE ALSO

libssh2_channel_open_ex(3) libssh2_channel_read_ex(3)