NAME

libssh2_sftp_fstat_ex - get or set attributes on an SFTP file handle

SYNOPSIS

```
#include <libssh2.h>
#include <libssh2_sftp.h>

int
libssh2_sftp_fstat_ex(LIBSSH2_SFTP_HANDLE *handle,
                      LIBSSH2_SFTP_ATTRIBUTES *attrs, int setstat)

#define libssh2_sftp_fstat(handle, attrs) \
  libssh2_sftp_fstat_ex((handle), (attrs), 0)
#define libssh2_sftp_fsetstat(handle, attrs) \
  libssh2_sftp_fstat_ex((handle), (attrs), 1)
```

DESCRIPTION

handle - SFTP File Handle as returned by **libssh2_sftp_open_ex(3)**

attrs - Pointer to an LIBSSH2_SFTP_ATTRIBUTES structure to set file metadata from or into depending on the value of setstat.

setstat - When non-zero, the file's metadata will be updated with the data found in attrs according to the values of attrs->flags and other relevant member attributes.

Get or Set statbuf type data for a given LIBSSH2_SFTP_HANDLE instance.

DATA TYPES

LIBSSH2_SFTP_ATTRIBUTES is a typedefed struct that is defined as below

```
struct _LIBSSH2_SFTP_ATTRIBUTES {

  /* If flags & ATTR_* bit is set, then the value in this
   * struct will be meaningful Otherwise it should be ignored
   */
  unsigned long flags;

  /* size of file, in bytes */
  libssh2_uint64_t filesize;
```

```
    /* numerical representation of the user and group owner of
     * the file
     */
    unsigned long uid, gid;

    /* bitmask of permissions */
    unsigned long permissions;

    /* access time and modified time of file */
    unsigned long atime, mtime;
};
```

You will find a full set of defines and macros to identify flags and permissions on the **libssh2_sftp.h** header file, but some of the most common ones are:

To check for specific user permissions, the set of defines are in the pattern LIBSSH2_SFTP_S_I<action><who> where <action> is R, W or X for read, write and executable and <who> is USR, GRP and OTH for user, group and other. So, you check for a user readable file, use the bit *LIBSSH2_SFTP_S_IRUSR* while you want to see if it is executable for other, you use *LIBSSH2_SFTP_S_IXOTH* and so on.

To check for specific file types, you would previously (before libssh2 1.2.5) use the standard posix S_IS***() macros, but since 1.2.5 libssh2 offers its own set of macros for this functionality:

LIBSSH2_SFTP_S_ISLNK
: Test for a symbolic link

LIBSSH2_SFTP_S_ISREG
: Test for a regular file

LIBSSH2_SFTP_S_ISDIR
: Test for a directory

LIBSSH2_SFTP_S_ISCHR
: Test for a character special file

LIBSSH2_SFTP_S_ISBLK
: Test for a block special file

LIBSSH2_SFTP_S_ISFIFO

Test for a pipe or FIFO special file

LIBSSH2_SFTP_S_ISSOCK
Test for a socket

RETURN VALUE

Return 0 on success 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.

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.

AVAILABILITY

This function has been around since forever, but most of the LIBSSH2_SFTP_S_* defines were introduced in libssh2 0.14 and the LIBSSH2_SFTP_S_IS***() macros were introduced in libssh2 1.2.5.

SEE ALSO

libssh2_sftp_open_ex(3)