

NAME

libssh2_session_callback_set - set a callback function

SYNOPSIS

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

```
void *
```

```
libssh2_session_callback_set(LIBSSH2_SESSION *session,  
                             int cbtype, void *callback);
```

DESCRIPTION

Sets a custom callback handler for a previously initialized session object. Callbacks are triggered by the receipt of special packets at the Transport layer. To disable a callback, set it to NULL.

session - Session instance as returned by **libssh2_session_init_ex(3)**

cbtype - Callback type. One of the types listed in Callback Types.

callback - Pointer to custom callback function. The prototype for this function must match the associated callback declaration macro.

CALLBACK TYPES

LIBSSH2_CALLBACK_IGNORE

Called when a SSH_MSG_IGNORE message is received

LIBSSH2_CALLBACK_DEBUG

Called when a SSH_MSG_DEBUG message is received

LIBSSH2_CALLBACK_DISCONNECT

Called when a SSH_MSG_DISCONNECT message is received

LIBSSH2_CALLBACK_MACERROR

Called when a mismatched MAC has been detected in the transport layer. If the function returns 0, the packet will be accepted nonetheless.

LIBSSH2_CALLBACK_X11

Called when an X11 connection has been accepted

LIBSSH2_CALLBACK_SEND

Called when libssh2 wants to send data on the connection. Can be set to a custom function to

handle I/O your own way.

The prototype of the callback:

```
ssize_t sendcb(libssh2_socket_t sockfd, const void *buffer,  
               size_t length, int flags, void **abstract);
```

sockfd is the socket to write to, **buffer** points to the data to send, **length** is the size of the data, **flags** is the flags that would have been used to a *send()* call and **abstract** is a pointer to the abstract pointer set in the *libssh2_session_init_ex(3)* call.

The callback returns the number of bytes sent, or -1 for error. The special return code **-EAGAIN** can be returned to signal that the send was aborted to prevent getting blocked and it needs to be called again.

LIBSSH2_CALLBACK_RECV

Called when libssh2 wants to read data from the connection. Can be set to a custom function to handle I/O your own way.

The prototype of the callback:

```
ssize_t recvcb(libssh2_socket_t sockfd, void *buffer,  
               size_t length, int flags, void **abstract);
```

sockfd is the socket to read from, **buffer** where to store received data into, **length** is the size of the buffer, **flags** is the flags that would have been used to a *recv()* call and **abstract** is a pointer to the abstract pointer set in the *libssh2_session_init_ex(3)* call.

The callback returns the number of bytes read, or -1 for error. The special return code **-EAGAIN** can be returned to signal that the read was aborted to prevent getting blocked and it needs to be called again.

LIBSSH2_CALLBACK_AUTHAGENT

Called during authentication process to allow the client to connect to the ssh-agent and perform any setup, such as configuring the agent or adding keys.

The prototype of the callback:

```
void authagent(LIBSSH2_SESSION* session, LIBSSH2_CHANNEL *channel,  
               void **abstract);
```

LIBSSH2_CALLBACK_AUTHAGENT_IDENTITIES

Not called by libssh2. The client is responsible for calling this method when a SSH2_AGENTC_REQUEST_IDENTITIES message has been received.

The prototype of the callback:

```
void identities(LIBSSH2_SESSION* session, void *buffer,  
               const char *agent_path,  
               void **abstract)
```

buffer must be filled in by the callback. Different clients may implement this differently. For example, one client may pass in an unsigned char ** for this parameter, while another may pass in a pointer to a struct.

Regardless of the type of buffer used, the client will need to send back a list of identities in the following format.

uint32 buffer length uint32 number of entries entries

Where each entry in the entries list is of the format:

string data cstring comment

agent_path The path to a running ssh-agent on the client machine, from which identities can be listed.

LIBSSH2_CALLBACK_AUTHAGENT_SIGN

Not called by libssh2. The client is responsible for calling this method when a SSH2_AGENTC_SIGN_REQUEST message has been received.

The prototype of the callback:

```
void sign(LIBSSH2_SESSION* session,  
          unsigned char *blob, unsigned int blen,  
          const unsigned char *data, unsigned int dlen,  
          unsigned char **sig, unsigned int *sig_len,  
          const char *agent_path,  
          void **abstract);
```

When interfacing with an ssh-agent installed on the client system, this method can call

libssh2_agent_sign(3) to perform signing.

RETURN VALUE

Pointer to previous callback handler. Returns NULL if no prior callback handler was set or the callback type was unknown.

SEE ALSO

libssh2_session_init_ex(3) **libssh2_agent_sign(3)**