Subject: Re: SSH package for U++
Posted by Oblivion on Wed, 08 Aug 2018 09:22:09 GMT
View Forum Message <> Reply to Message

I undestand the appeal, but I have to say that the result is sort of confusing. E.g. I have implemented

int Get(SFtpObject* obj, void* buffer, int size);

but it is clearly incompatible with this sort of async operations.

Well, I don't plan that method to be async (MT). IMO Async methods (that have SFtp:Asyncxxx prefix) should either use streams or consumer function (Event<const void*, int>), which will have the same effect anyway.

Asynchronous (MT) operations on ssh is somewhat complicated. I had to make some compromises for the sake of simplicity and maintainabilily, but first I have to see tha changes you've made to the code, then I'll write a summary of its design and explain those "odd" choices (it's a complex beast but works fine.)

So I guess for now, I will try to pretend that those complex Cmd / ComplexCmd "nonblocking" operations are not there, p

Those methods are the parts that I am not happy with, either :)
I have a plan and a test code to "fix" that ugliness, but actual refactoring will come later.

All those NULL handles, implicit results etc make me uneasy.

Most of them are, I believe, taken care of and contained. Implicit results are there, but shouldn't pose any real danger (Not that Ic can see of, at least). They can, and hopefully will, be reduced (another TODO for me,connected with Cmd/ComplexCmd pair).

In related news, I have also fixed GetWaitEvents

Did it make any difference? Because IIRC values of those upp enums and the defines of libssh2 are the same.

Quote:
I am also thinking that perhaps SFtp should be (derived from) SshSession. I think it is unlikely that sharing SshSession for several protocols is all that important.

I repectfully disagree. Servers usually limit the number of sessions. (e.g. the servers I work with allows only five sessions from the same user.) OTOH, there is no channel limit. (they are only limited with bandwidth). Also this way we will give the developers flexibility. For example,If I have the ability to use Scp to transfer files while browsing and manipulating file system objects with SFtp, I'll prefer Scp for transfers, since it is relatively faster.

Also I have a plan to override libssh2 raw data read and write methods (it allows it) and use TcpSocket directly, and move Wait() there.

Best regards,
Oblivion.
