Subject: Re: SSH package for U++
Posted by mirek on Thu, 09 Aug 2018 09:54:01 GMT
View Forum Message <> Reply to Message

Quote:
I use it in an app and for a limited number of ssh channels (usually 10-20).
But frankly, that code remains before the the CoWork improvements and the arrival of AsyncWork.
Nowadays in most such cases I use the async methods.

OK, no need to have fully async mode for this....

Quote:
I have a new proposal: What if I get rid of queue mechanism and rewrite the package with only blocking mode and optional async transfer methods(using AsyncWork, and naming them agein SFtp::Asyncxxx)?
It won't take more than a week for me to come up with a working SSH package and the existing public API wont change much (only the NB helpers will be gone).
Besides its SC will be lot cleaner.

Definitely.

Quote:
As to your Get implementation:

I haven't tested this yet, but it shouldn't work in non-blocking mode. (because the execution will be deferred (Get will immediately return) and there is a local variable ("done"))

Sure, as I said that was the point where I decided that fully non-blocking mode is "blocking" this kind of interface.

BTW, digging deeper into the code

```
bool SshChannel::Lock()
{
 if(*lock == 0) {
  LLOG("Channel serialization lock acquired.");
  *lock = ssh->oid;
 }
 return *lock == ssh->oid;
}
```

I think there is a race condition here - two threads can obtain this lock simultaneously. Now I am

not sure whether is this supposed to be MT safe, but if not, why atomic, right?

Mirek
