

Hello Omari,

Quote:but, the first use of SshExec close the session .

In SSH2 protocol, exec channels are fire-and-forget type channels - they are unique to single execution target.

The protocol does not allow the reuse of exec channels.

That's why you need to create a new SshExec instance for each command instance. (They are relatively cheap).

So what you need to do is:

```
String cout, cerr;  
SshExec ExecMkDir(session);  
SshExec ExecRun(session);  
Scp scp(session);  
  
ExecMkDir("mkdir target/", cout, cerr);  
scp.SaveFile("target/binary", LoadFile(localFile));  
ExecRun("target/binary&", cout, cerr);
```

Yet, as mentioned in SSH package's docs, If you need batch operations, I'd recommend SshShell instead.

OTOH, server does not close the connection on my setups (OpenSSH/sshd). It simply fails to reuse the exec channel.

This may be due to different server implementations, I am not sure.

Now if you are talking about the session, then you have to remember that all SSH objects are scope-bound, including the session. Typical C++ rule.

Still, the output of a brief trace (Ssh::Trace()) would be more helpful.

If you have any other questions regarding SSH package, or run into any other issues, feel free to ask. :)

Best regards,
Oblivion