
Subject: Re: SSH package for U++
Posted by [Oblivion](#) on Fri, 04 May 2018 20:08:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello

SSH package is updated.

SSH:

2018-05-04: Ssh::GetWaitEvents() fixed.
SshTunnel::Validate() fixed.

Reference examples added to the package:

SshTunnelExample: Demonstrates the basic SSH tunneling (as tunnel/server) in blocking mode.

Note that the SSH tunnel example requires upp/reference/SocketClient and upp/reference/SocketServer examples.
it acts as a SSH server between the socket client and server. Although the example demonstrates one of the basic tunneling capabilities of the SshTunnel class, very complex SSH tunnels can be built using it.

Code:

```
#include <Core/Core.h>
#include <SSH/SSH.h>

using namespace Upp;

// This example requires upp/reference/SocketServer and upp/reference/SocketClient examples.
// SocketClient: Set the port number to 3215.
//
// |SocketClient (client)|<---> |SshTunnelExample (tunnel/server)| <---> |SocketClient (server)|

bool SocketSendRecv(String& packet)
{
    TcpSocket s;
    if(!s.Connect("127.0.0.1", 3214)) {
        LOG("SocketSend(): " << s.GetErrorDesc());
        return false;
    }
    if(!s.PutAll(packet + '\n'))
        return false;
```

```

packet = s.GetLine();
return !packet.IsEmpty();
}
void StartTunnel(SshSession& session)
{
    SshTunnel listener(session);
    if(!listener.Listen(3215, 5)) {
        LOG("StartTunnel(): " << listener.GetErrorDesc());
        return;
    }
    LOG("SSH tunnel (server mode): Waiting for the requests to be tunneled...");
    for(;;) {
        SshTunnel tunnel(session);
        if(!tunnel.Accept(listener)) {
            LOG("StartTunnel(): " << tunnel.GetErrorDesc());
            return;
        }
        auto data = tunnel.GetLine();
        LOG("Tunneled Request: " << data);
        if(!data.IsEmpty() && SocketSendRecv(data)) {
            LOG("Tunneled Response: " << data);
            tunnel.Put(data + '\n');
        }
    }
}

CONSOLE_APP_MAIN
{
    StdLogSetup(LOG_FILE | LOG_COUT);
    // Ssh::Trace();

    SshSession session;
    if(session.Timeout(30000).Connect("username:password@localhost:22")) {
        StartTunnel(session.Timeout(Null));
        return;
    }
    LOG(session.GetErrorDesc());
}

```

Below is the GIT address where you can always get the latest version.

GIT repo: <https://github.com/ismail-yilmaz/upp-components/tree/master/Core/SSH>
 Examples: <https://github.com/ismail-yilmaz/upp-components/tree/master/Examples>

Best regards,
 Oblivion
