Subject: Help needed with link errors (serversocket)
Posted by imos on Tue, 11 Jul 2017 21:46:13 GMT
View Forum Message <> Reply to Message

Hi
I am trying to find a dev env to develop a socket oriented deamon (non-blocking) using high level approachs (for sockets and data structures) and I decided to give a try to U++.

I am trying to put to work the ServerSocket example but I have link errors that I can not solve alone. Here the code:

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

using namespace Upp;

Socket accept_socket, data_socket;
int port = 2020;


CONSOLE_APP_MAIN
{
 if(!ServerSocket(accept_socket, port)) // Listen for connections using _accept_socket;
 {
    throw Exc("Couldn't bind socket on the local port.");
 }
 // You can do this in a loop to accept many connections:
 if( accept_socket.IsOpen() )
 {
    dword ip_addr;
    // Hand off successful connection to _data_socket
    if( !accept_socket.IsError() && accept_socket.Accept(data_socket, &ip_addr) )
    {
       //Cout() << "Connection from " << FormatIP(m_ipaddr) << "\n";
       Cout() << "Connection from " << "\n";
       // Read from the socket until it is closed, has an error, or you see an end-of-file marker
       // (EOF optional and application-specific)
       while(data_socket.IsOpen() && !data_socket.IsEof() && !data_socket.IsError())
       {
          Cout() << data_socket.Read();
       }
    }
    Cout() << "\n";
 }
}
```

I also added the Web package to the project but I get these link errors:

...Web/html.cpp (347): error: ambiguous overload for 'operator+' (operand types are 'Upp::HtmlTag' and 'Upp::Htmls')

.../Web/auth.cpp (219): error: ambiguous overload for 'operator+' (operand types are 'Upp::Htmls' and 'Upp::HtmlTag')

Etc.

Thanks a lot

Imos

Subject: Re: Help needed with link errors (serversocket)
Posted by Oblivion on Wed, 12 Jul 2017 09:43:02 GMT
View Forum Message <> Reply to Message

Hello imos, and welcome!

Web package is depreceated. AFAIK, it is kept for historical reasons.
You can use TcpSocket for socket operations.
There is a client/server example in Examples section:
Server: http://www.ultimatepp.org/reference$SocketServer$en-us.html

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



using namespace Upp;



CONSOLE_APP_MAIN

{

    TcpSocket server;

    if(!server.Listen(3214, 5)) {

        Cout() << "Unable to initialize server socket!\n";
        SetExitCode(1);
        return;
    }
    Cout() << "Waiting for requests..\n";
    for(;;) {
        TcpSocket s;
        if(s.Accept(server)) {
            String w = s.GetLine();
            Cout() << "Request: " << w << " from: " << s.GetPeerAddr() << '\n';
            if(w == "time")
                s.Put(AsString(GetSysTime()));
            else
                s.Put(AsString(3 * atoi(~w)));
            s.Put("\n");
        }
    }
}
```

Client: http://www.ultimatepp.org/reference$SocketClient$en-us.html

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

using namespace Upp;

String Request(const String& r)
```

```
{
    TcpSocket s;
    if(!s.Connect(CommandLine().GetCount() ? CommandLine()[0] : "127.0.0.1", 3214)) {
        Cout() << "Unable to connect to server!\n";
        SetExitCode(1);
        return Null;
    }
    s.Put(r + '\n');
    return s.GetLine();
}


// Start reference/SocketServer before starting this program


CONSOLE_APP_MAIN
{
    Cout() << Request("time") << '\n';
    Cout() << Request("33") << '\n';
}
```

Best regards,
Oblivion

Subject: Re: Help needed with link errors (serversocket)
Posted by omari on Wed, 12 Jul 2017 09:49:52 GMT
View Forum Message <> Reply to Message

Hi lmos,

the Web package is depraceted, Socket now is part of Upp Core.

you can find in the reference example an example SocketServer

---

Subject: Re: Help needed with link errors (serversocket)
Posted by imos on Wed, 12 Jul 2017 19:56:06 GMT
View Forum Message <> Reply to Message

Great! It works!

Now I am going to try it using non-blocking approach and single thread if possible... Is it possible to use non-blocking socket using the TcpSocket class?

Thanks a lot

Imos

Subject: Re: Help needed with link errors (serversocket)
Posted by Oblivion on Wed, 12 Jul 2017 21:08:44 GMT
View Forum Message <> Reply to Message

Hello Imos,

Quote:
Now I am going to try it using non-blocking approach and single thread if possible... Is it possible to use non-blocking socket using the TcpSocket class?

Yes it is possible to use non-blocking socket using the TcpSocket class.
TcpSocket class allows blocking, non-blocking, and time-constrained operations.
You have to set Timeout value to 0 to put TcpSocket in a non-blocking mode.
But I suggest you first reading the TcpSocket api docs before you plunge into the world of non-blocking sockets, and then get yourself familiar with U++ core classes & concepts.

Non-blocking socket operations can easily get tricky and complex.
However, there are ways to reduce complexity.
Below you can find a queue model designed exactly for non-blocking socket operations.
It also contains an example code called ClientSockets, which is actually a non-blocking version of SocketClient example with multiple requests.
But it requires some knowledge of U++ callbacks, and C++11 lambdas.

Should you have any further qustions, I may be able to answer them.

Best regards,
Oblivion

File Attachments

1) ClientSockets.zip, downloaded 342 times

Subject: Re: Help needed with link errors (serversocket)
Posted by imos on Thu, 13 Jul 2017 08:50:53 GMT
View Forum Message <> Reply to Message

Thanks Oblivion for your availability for helping...

I am going to try the sample and modify it and find out how Upp handles thousands of sockets handles using just one (or two) thread(s) (which is my main aim)...

Thanks a lot

Imos