
Subject: request: Socket working example
Posted by [forlano](#) on Mon, 09 Nov 2009 19:43:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I'm doing experiments ONLY about socket in order to pass data between two U++ programs. I have found many nice examples on the net of Client/Server application in C++ with Winsock or Linux. They are less than 20 lines of code. But make no sense to use them in my U++ programs when I have a powerful Web class that should work in both OS.

Unfortunaltely the tutor page [http://www.ultimatepp.org/srcdoc\\$Web\\$ConnectionOriented\\$en-us.html](http://www.ultimatepp.org/srcdocWebConnectionOriented$en-us.html) is not enough for a full beginner as me. In fact that snippets are not working examples. For example the variable "m_ipaddr" where come from? Perhaps I should know it .

So the request for the U++ forum is, if possible, to add to those lines a few others to make them two minimal running examples. Just as minimal as possible: the client send something and exit, the server receive and echo that message. After that I think to be able to expand these minimal examples.

Thanks a lot,
Luigi

Subject: Re: request: Socket working example
Posted by [mirek](#) on Sun, 15 Nov 2009 12:34:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

```
using namespace Upp;
```

```
CONSOLE_APP_MAIN
{
    Socket server;
    if(!ServerSocket(server, 3214)) {
        Cout() << "Unable to initialize server socket!\n";
        SetExitCode(1);
        return;
    }
    Cout() << "Waiting for requests..\n";
    for(;;) {
        Socket s;
        if(server.Accept(s)) {
            String w = s.ReadUntil('\n');
            Cout() << "Request: " << w << '\n';
            if(w == "time")
```

```

    s.Write(AsString(GetSysTime()));
else
    s.Write(AsString(3 * atoi(~w)));
s.Write("\n");
}
}
}

```

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

```
using namespace Upp;
```

```

String Request(const String& r)
{
    Socket s;
    if(!ClientSocket(s, CommandLine().GetCount() ? CommandLine()[0] : "127.0.0.1", 3214)) {
        Cout() << "Unable to connect to server!\n";
        SetExitCode(1);
        return Null;
    }
    s.Write(r + '\n');
    return s.ReadUntil('\n');
}

```

```

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

```

(It is now in reference too).

Subject: Re: request: Socket working example
 Posted by [forlano](#) on Sun, 15 Nov 2009 14:33:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Sun, 15 November 2009 13:34

(It is now in reference too).

Thanks a lot!
 I'll test them as soon as possible,

Luigi

Subject: Re: request: Socket working example
Posted by [forlano](#) on Wed, 18 Nov 2009 08:36:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Sun, 15 November 2009 13:34

(It is now in reference too).

Perhaps these examples can substitute the following page:
[http://www.ultimatepp.org/srcdoc\\$Web\\$ConnectionOriented\\$en-us.html](http://www.ultimatepp.org/srcdocWebConnectionOriented$en-us.html)

While this one
[http://www.ultimatepp.org/srcdoc\\$Web\\$SocketExample1\\$en-us.html](http://www.ultimatepp.org/srcdocWebSocketExample1$en-us.html)

can be simple removed.

Luigi

Subject: Re: request: Socket working example
Posted by [Mindtraveller](#) on Wed, 18 Nov 2009 08:50:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Recently I've made a kind of simple example for U++ nonblocking sockets.
Any comments, suggestions and critics are welcome.

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

```
using namespace Upp;
```

```
int PORT_START    = 5000;
int PORT_COUNT    = 100;
int PORT_IO_TIMEOUT = 10000;
```

```
class Outter
{
public:
    Outter &operator<< (const String &s)
    {
        mutex.Enter(); Cout() << s; mutex.Leave();
        return *this;
    }
    Outter &operator<< (int i)
    {
        mutex.Enter(); Cout() << FormatInt(i); mutex.Leave();
        return *this;
    }
};
```

```

}
private:
    Mutex mutex;
};
Outter print;

class SenderThread : public Thread
{
public:
    void Do()
    {
        Vector<Socket> sockets;

        for (int i=0; i<PORT_COUNT; ++i)
        {
            int port = PORT_START+i;
            Socket &socket = sockets.Add();

            if (!ClientSocket(socket, "localhost", port))
            {
                print << "\nError creating client socket @" << port;
                sockets.Pop();
                break;
            }

            socket.Write(Format("%04d", port));
        }

        print << "\nAll data in ports sent\n";

        Vector<Socket *> read;
        Vector<Socket *> write;
        for (int i=0; i<sockets.GetCount(); ++i)
            write.Add(&sockets[i]);

        int socketsFinished = 0;
        while (socketsFinished < sockets.GetCount())
        {
            if (!Socket::Wait(read, write, PORT_IO_TIMEOUT))
                Sleep(200);
            else
            {
                int curSocketsFinished = 0;

                for (int i=0; i<sockets.GetCount(); ++i)
                    if (sockets[i].IsError())
                    {
                        print << Format("-(%04d) ", PORT_START+i);
                    }
            }
        }
    }
};

```

```

        ++curSocketsFinished;
    }
    else
    if (sockets[i].PeekWrite(PORT_IO_TIMEOUT))
    {
        sockets[i].Clear();
        print << Format("(+(%04d) ", PORT_START+i);
        ++curSocketsFinished;
    }

    socketsFinished += curSocketsFinished;
}
}
};

```

```

class ReceiverThread : public Thread
{
public:
    void Do()
    {
        Vector<Socket> sockets;
        Vector<Socket> newSockets;

        for (int i=0; i<PORT_COUNT; ++i)
        {
            int port = PORT_START+i;
            Socket &socket = sockets.Add();

            if (!ServerSocket(socket, port))
            {
                print << "\nError creating server port @" << port;
                sockets.Pop();
                break;
            }

            Socket &newSocket = newSockets.Add();
            dword newAddr;
            socket.Accept(newSocket, &newAddr);
        }

        print << "\nListening with all ports\n";

        Vector<Socket *> read;
        Vector<Socket *> write;
        for (int i=0; i<newSockets.GetCount(); ++i)
            read.Add(&newSockets[i]);
    }
}

```

```

int socketsFinished = 0;
while (socketsFinished < sockets.GetCount())
{
    if (!Socket::Wait(read, write, PORT_IO_TIMEOUT))
        Sleep(200);
    else
    {
        int curSocketsFinished = 0;
        for (int i=0; i<newSockets.GetCount(); ++i)
        {
            String inData = newSockets[i].PeekCount(4, PORT_IO_TIMEOUT);
            if (!inData.IsEmpty())
            {
                newSockets[i].Read(4,PORT_IO_TIMEOUT);
                print << inData << " ";
                ++curSocketsFinished;
            }
        }

        socketsFinished += curSocketsFinished;
    }
}

};

CONSOLE_APP_MAIN
{
    SenderThread sender;
    ReceiverThread receiver;

    receiver.Run(callback(&receiver, &ReceiverThread::Do)); print << "\nReceiver started";
    Sleep(1000);
    sender .Run(callback(&sender, &SenderThread ::Do)); print << "\nSender started";

    sender.Wait(); print << "\nSender finished";
    receiver.Wait(); print << "\nReceiver finished";
}

```
