
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";  
}
```
