

---

Subject: Socket in a sock

Posted by [Paco](#) on Mon, 18 Aug 2008 22:09:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I am a new in Upp and trying to adapt current running application VBdevelopped tu U++. The programm is a simple telnet client application. I can create a client socket, initialize connection an read the server answer. Until now everything is correct but now the client socket has to be continuously listening to the server and everytime data arrives it has to be extracted to update a string and displayed in a DocEdit widget. My question is : How can I implement such an application ? Has the socket an event that is fired the same moment data arrives ?? . Do I need to create a thread to continuously read the socket ? maybe a timer with a callback?

I read all forum topics where a socket is involved and got no solution Any example code out there ? The application is a RadioAmateur programm and the socket is listening to server continuously, the server send blocks of data called "spots".

Any help is very appreciated. Thank you very much.....

---

---

Subject: Re: Socket in a sock

Posted by [captainc](#) on Tue, 19 Aug 2008 12:30:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Usually when listening for incoming socket connections, you have a socket server that listens and hands off connections to other individual sockets (threads).

U++ has ServerSocket for this.

This is partial code I used in an application to receive data:

```
if( !ServerSocket(_accept_socket, _port, false, 1000, false) ){
// throw exception...
}
//do this in a loop:
if( _accept_socket.IsOpen() ){ // server read from data socket
    dword ip_addr;
    if( (!_accept_socket.IsError()) && _accept_socket.Accept(_data_socket, &ip_addr) ){
        _content_buf.Clear();
        while( _data_socket.IsOpen() && !_data_socket.IsEof() && !_data_socket.IsError())
            {_content_buf.Cat(_data_socket.Read());}
    }
    // do something with content buf...
}
```

So you can see the \_accept\_socket gets the connection and hands it off to \_data\_socket with the Accept() method. Then \_data\_socket has the connection.

Servers use this method with "Thread Pools" of socket connections to allow for a certain number of sockets to exist (usually 1 per thread).

Also do a search online for "BSD Sockets" or "Berkeley Sockets" to get a better understanding of the underlying technologies.

---

---

Subject: Re: Socket in a sock

Posted by [Paco](#) on Tue, 19 Aug 2008 18:22:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Thank's a lot captainc about your quick answer, I have read every post coming from you in this topic.

I am not sure if I need only one socket or at least 2 are needed as only few characters have to be sent to the server.

My application is not a server, is a client that needs to connect to the server, wait for a login, then send 5 characters and from now continuously wait listening while the server sends packets of characters, as they arrive the application has to detect the arrival, extract these characters from the listening socket buffer and update a string that serves to update a DocEdit widget.

All this work has to be made without interrupting the other application works, like logging contacts to a database, moving a rotor antenna via com port, etc. not a continuous work, maybe a TimerCallback is the better option to read the buffer socket every 500 milliseconds

If you can use telnet, create a connection via TCP/IP with a server "dx.ea7urc.org" port 41112, as the server asks for login type "ea7cdu" (my radioamateur call) and then watch the telnet window (and suppose it is a DocEdit widget) for a few minutes, you will see what I mean. The application has to reproduce the telnet behavior.

If you can spend a few minutes you will teach me about to reproduce this behavior and I will thank you very much.

Hope to read you soon... Thank's again...

---

---

Subject: Re: Socket in a sock

Posted by [captainc](#) on Tue, 19 Aug 2008 20:13:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Quote:All this work has to be made without interrupting the other application works, like logging contacts to a database, moving a rotor antenna via com port, etc. not a continuous work, maybe a TimerCallback is the better option to read the buffer socket every 500 milliseconds

You will need to thread the outgoing connection. What I would do is create a class that would handle outgoing connections (make the connection and read from the buffer) and use the U++ Thread to run a class method. Have that thread run a loop checking for new information on the socket. When it receives information, it could update the Gui. Meanwhile, it would not block the program from doing other things, and the thread would terminate after the function exits (on socket close). You have to be careful with the architecture in this scenario with threading and race conditions though. For example, what if you want to close the program while it is listening for connections (thread running)? How do you end the thread and close the socket? You probably will need to use shared resources and locking in this situation, In this case I would look into making a thread-safe class method that would close the socket and have the function that is running in the

thread finish/exit if the socket is closed.

Also, remember that BSD socket connections close when both sides close the socket. The close message that is sent is acknowledged by the other side to officially close the connection.

Sorry I cannot check out the telnet site right now, but I might be able to get to it in a bit.

---