

Forum:[Hack Forums](#)

Thread: [Asynchronous multi-client/server classes \(make rats, loggers, whatever\)](#)

gigajew - January 28, 2018, 5:31 pm

I've been playing around with Sockets for a while, and after studying Aeonhack's code, I realized that he made some mistakes where he can't receive more than the current buffer size that he had set. So I've created my own (which, it's not perfect - it has a lot of flaws), but it gets the job done. Maybe with some memory leaks, but if you know a thing or two you can fix that easily. With something like this, it would be easy to create your own rat, logger or whatever really.

Edit: going to change some things and update this soon. Hang tight

Example server:

Example output:

```
Code [Server] Server is now listening
[Client] Connected to server
[Server] 127.0.0.1:53060 connected
[Client] Sent 11 bytes to server
[Server] 127.0.0.1:53060 said Hello world
[Server] Sent 19 bytes to 127.0.0.1:53060
[Client] Server said: Hello back to you!!
```

SocketClient source:

Spoiler ([Click to View](#))

SocketServer source:

Spoiler ([Click to View](#))

Ninja - January 28, 2018, 5:53 pm

I like what you did, but what if you receive more than 1024 bytes? Then it will not fit in the buffer. And what about all other socket methods you are hiding here? You would be better off inheriting from Socket and overloading the methods.

Also, one minor detail: the convention for events and delegates dictates that you should use "object sender" instead.

[https://msdn.microsoft.com/en-us/library/dd833085\(v=vs.95\).aspx](https://msdn.microsoft.com/en-us/library/dd833085(v=vs.95).aspx)

gigajew - January 28, 2018, 6:01 pm

It continuously writes to a stream until Socket.Available equals 0. Then it invokes received event, and resets the stream to 0 again for a new transmission. So 1024 is really just an arbitrary value, you can receive more or less.

Ninja - January 28, 2018, 6:48 pm

(Jan 28, 2018 07:01 PM)gigajew Wrote: [\[CITE\]](#) It continuously writes to a stream until Socket.Available equals 0. Then it invokes received event, and resets the stream to 0 again for a new transmission. So 1024 is really just an arbitrary value, you can receive more or less.

Oh, my bad. I didn't read the source fully. Still, it would be better if you would inherit from socket and override the appropriate methods to add events, instead of making a wrapper for it. Or you

would have to add all overloads for the methods you have now so they match the socket methods, else you are limiting functionality.

EDIT: Just remembered I did something similar as well. I was mainly focussed on making it easier to use the methods asynchronously, though I added events as well. You can find it here:
<https://hackforums.net/showthread.php?tid=5735967>

BahNahNah - January 28, 2018, 8:24 pm

You dont need Try/catch on endreceve. Just get the SocketError and check its Success.
[https://msdn.microsoft.com/en-us/library/ms145145\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms145145(v=vs.110).aspx)

gigajew - January 28, 2018, 8:52 pm

(Jan 28, 2018 09:24 PM)BahNahNah Wrote: [\[CITE\]](#) You dont need Try/catch on endreceve. Just get the SocketError and check its Success.

[https://msdn.microsoft.com/en-us/library/ms145145\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms145145(v=vs.110).aspx)

I got a ObjectDisposedException in EndReceive on ServerSocket sometimes

chrizator - January 29, 2018, 11:16 pm

Great share OP, very interesting. If somebody really considers using this for their own project, make sure NOT to create one receive buffer (byte array) for each established connection. Instead, use buffer pooling in order to avoid heap fragmentation due to object pinning. Especially when dealing with hundreds of connections, this becomes essential.

There are some open-source projects:
github.com/tenor/ServerToolkit

Ninja - January 30, 2018, 7:37 am

(Jan 30, 2018 12:16 AM)chrizator Wrote: [\[CITE\]](#) Great share OP, very interesting. If somebody really considers using this for their own project, make sure NOT to create one receive buffer (byte array) for each established connection. Instead, use buffer pooling in order to avoid heap fragmentation due to object pinning. Especially when dealing with hundreds of connections, this becomes essential.

There are some open-source projects:
github.com/tenor/ServerToolkit

I don't know much about buffer pooling and what not, but this is what I found online when I googled it.

Quote:C# sharps memory management is actually quite good, so instead of having a pool of buffers, you could just allocate exactly what you need and stick it into a queue. Once you are done with buffer just let the garbage collector handle it.

Full answer: <https://stackoverflow.com/a/281954>

chrizator - January 30, 2018, 10:10 am

(Jan 30, 2018 08:37 AM)Ninja Wrote: [\[CITE\]](#) I don't know much about buffer pooling and what not, but this is what I found online when I googled it.

Full answer: stackoverflow.com/a/281954

Hm. Seems like opinions differ when talking about buffer pooling.

A while ago I read this post: ahuwanya.net/blog/post/Buffer-Pooling-for-NET-Socket-Operations

Since then I always used buffer pooling for all of my projects. I'm going to cry if all that work turns out to be unnecessary. ?

Looking forward to hear other opinions!

Ninja - January 30, 2018, 10:47 am

(Jan 30, 2018 11:10 AM)chrizator Wrote: [\[CITE\]](#) Hm. Seems like opinions differ when talking about buffer pooling.

A while ago I read this post: ahuwanya.net/blog/post/Buffer-Pooling-for-NET-Socket-Operations

Since then I always used buffer pooling for all of my projects. I'm going to cry if all that work turns out to be unnecessary. ?

Looking forward to hear other opinions!

I actually found a tutorial earlier when I was researching buffer pooling and stumbled upon this:

<https://www.wintellect.com/pooling-buffers-for-better-memory-management/>

Which uses a class called the BufferManager from the Windows Communication Foundation.

gigajew - February 8, 2018, 4:53 pm

(Jan 28, 2018 09:24 PM)BahNahNah Wrote: [\[CITE\]](#) You dont need Try/catch on endreceve. Just get the SocketError and check its Success.

[https://msdn.microsoft.com/en-us/library/ms145145\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms145145(v=vs.110).aspx)

Thanks I didn't know about this overload

Ninja - February 8, 2018, 10:10 pm

(Jan 28, 2018 09:52 PM)gigajew Wrote: [\[CITE\]](#) I got a ObjectDisposedException in EndReceive on ServerSocket sometimes

(Feb 8, 2018 05:53 PM)gigajew Wrote: [\[CITE\]](#) Thanks I didn't know about this overload
2 different answers to the exact same post. you high?

gigajew - February 9, 2018, 11:18 am

(Feb 8, 2018 11:10 PM)Ninja Wrote: [\[CITE\]](#) 2 different answers to the exact same post. you high?
I didn't know what he meant first, and to answer your question; very

michalmonday - July 18, 2018, 3:39 pm

What would be optimal way to implement this code into GUI?

The goal would be to press a button which sends a message to all clients or a single specific client.

Is there some way to still make it event based and have something like:

Code namespace GUI_Server

```
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            clients[0].Send(Encoding.UTF8.GetBytes("Button 1 was pressed, praise the Lord"));
        }
    }
}
```

Edit: I just found this thing:

https://www.codeguru.com/csharp/csharp/cs_network/sockets/article.php/c8781/Asynchronous-Socket-Programming-in-C-Part-II.htm#Client1

It has GUI and it's really easy to accomplish what I intended. Idk how it compares with this one and

what possible flaws it has but it just seems to work properly during my test.

gigajew - July 18, 2018, 7:42 pm

I'll probably be revamping the code soon, keep your ears and eyes open

NinjahZ - April 16, 2019, 2:24 am

Interested in the Development of this, please continue it :) I am using parts of it for a project :) trying to learn TCP/IP Network Libraries

devvariety - August 15, 2019, 4:09 pm

Sorry for the thread grave digging, was testing this and seems whenever I disconnect the client from the remote server it crashes (the server). It works on localhost, but once you use it for production and have the client disconnect it just crashes.

memento - August 15, 2019, 4:16 pm

(Aug 15, 2019 04:09 PM)devvariety Wrote: [\[CITE\]](#) Sorry for the thread grave digging, was testing this and seems whenever I disconnect the client from the remote server it crashes (the server). It works on localhost, but once you use it for production and have the client disconnect it just crashes.

Can you provide some more information? There can be tons of things going wrong, but without a proper question there's nothing to be answered.

devvariety - August 15, 2019, 4:23 pm

Code Unhandled Exception: System.Net.Sockets.SocketException: Transport endpoint is not connected

at System.Net.Sockets.Socket.UpdateStatusAfterSocketErrorAndThrowException(SocketError error, String callerName)

at System.Net.Sockets.Socket.get_RemoteEndPoint()

at rcs.Server.Model.ClientState.get_IPAddress() in D:\rcs\rcs.Server\Model\ClientState.cs:line 13

at rcs.Server.Core.Globals.<>c.<<SetupServer>b__4_2>d.MoveNext() in

D:\rcs\rcs.Server\Core\Globals.cs:line 37

--- End of stack trace from previous location where exception was thrown ---

at System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext, ContextCallback callback, Object state)

--- End of stack trace from previous location where exception was thrown ---

at System.Threading.ThreadPoolWorkQueue.Dispatch()

Exception thrown when the client disconnects.

EDIT: Alright so common sense just slapped me, once the client disconnects it cannot get the IP address since the socket is closed.

The only con with this it seems that it can't send MBs worth of data..

gigajew - August 15, 2019, 5:32 pm

(Aug 15, 2019 04:23 PM)devvariety Wrote: [\[CITE\]](#) Code Unhandled Exception: System.Net.Sockets.SocketException: Transport endpoint is not connected

at System.Net.Sockets.Socket.UpdateStatusAfterSocketErrorAndThrowException(SocketError error, String callerName)

at System.Net.Sockets.Socket.get_RemoteEndPoint()

at rcs.Server.Model.ClientState.get_IPAddress() in D:\rcs\rcs.Server\Model\ClientState.cs:line 13

at rcs.Server.Core.Globals.<>c.<<SetupServer>b__4_2>d.MoveNext() in

D:\rcs\rcs.Server\Core\Globals.cs:line 37

--- End of stack trace from previous location where exception was thrown ---

at System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext, ContextCallback callback, Object state)

--- End of stack trace from previous location where exception was thrown ---

at System.Threading.ThreadPoolWorkQueue.Dispatch()

Exception thrown when the client disconnects.

EDIT: Alright so common sense just slapped me, once the client disconnects it cannot get the IP address since the socket is closed.

The only con with this it seems that it can't send MBs worth of data..

There's a workaround for that too, use streams instead of byte buffers

devvariety - August 15, 2019, 5:39 pm

Makes no sense why it works on localhost, but not on remote server though?

gigajew - August 15, 2019, 6:12 pm

(Aug 15, 2019 05:39 PM)devvariety Wrote: [\[CITE\]](#) Makes no sense why it works on localhost, but not on remote server though?

Have you forwarded the port and changed the listening IP address? May have to change IPAddress.Any to 127.0.0.1 or the lan ip your router assigned to you

devvariety - August 15, 2019, 6:15 pm

(Aug 15, 2019 06:12 PM)gigajew Wrote: [\[CITE\]](#) Have you forwarded the port and changed the listening IP address? May have to change IPAddress.Any to 127.0.0.1 or the lan ip your router assigned to you

I am running the remote socket on a Linux machine. Allowed the port through UFW. I can connect to it just fine from PC, but whenever I send over 1MB of data the client crashes due to the string being too short from the expected string.

I am trying to compress the buffer for now until I have time to use implement NetworkStream, in a rush with this.

For a better view of what I mean.

In short, the socket doesn't fully get the bytes the server sends..

gigajew - August 15, 2019, 8:42 pm

A network stream should be easy to implement, I have a work in progress, more like a scratchpad I'll send it to you in a moment

devvariety - August 16, 2019, 9:51 am

(Aug 15, 2019 08:42 PM)gigajew Wrote: [\[CITE\]](#) A network stream should be easy to implement, I have a work in progress, more like a scratchpad I'll send it to you in a moment

Alright thank you

gigajew - August 17, 2019, 1:03 pm

(Aug 16, 2019 09:51 AM)devvariety Wrote: [\[CITE\]](#) Alright thank you

Server project: https://www.mediafire.com/file/fbjzldju2kqjrvl/server_project.zip/file

Client project: https://www.mediafire.com/file/nmmn1n10nxghdkn/client_project.zip/file

devvariety - August 17, 2019, 1:18 pm

(Aug 17, 2019 01:03 PM)gigajew Wrote: [\[CITE\]](#) Server project: [server_project](#)
Client project: [client_project](#)
!!! Thank you, expect money in your wallet soon!

riu41776 - August 18, 2019, 4:55 am

You'd better off use a standard remote access protocol instead of plain sockets.

gigajew - August 18, 2019, 9:23 am

(Aug 17, 2019 01:18 PM)devvariety Wrote: [\[CITE\]](#) !!! Thank you, expect money in your wallet soon!
Made it a little easier to read code, and created a shared library so there aren't duplicate files
<https://www.mediafire.com/file/joapvak8mk6cu43/RatBase.zip/file>
<https://github.com/gigajew/ratbase>

Hopefully this helps you with whatever you're making