

Forum:[Hack Forums](#)

Thread: [SyncIO || Easy multipurpose networking library. | Now with UDP and Remote Functions |](#)

BahNahNah - December 27, 2016, 6:13 pm

This has been my mini project for the last couple of weeks, and its finally done ?

Download: [GitHub](#)

An example chat program can be found [HERE on the github repository](#) (and it is also included in the source download)

Here is a very basic example:

Server

```
Code var server = new SyncIOServer();
server.SetHandler((SyncIOConnectedClient c, object[] data) => {
    c.Send(data); //Simple echo
});
```

```
server.ListenTCP(12345);
```

```
Console.ReadLine();
```

Client

```
Code var client = new SyncIOClient();
client.SetHandler((SyncIOClient c, object[] data) => {
    Console.WriteLine(string.Join(" ", data));
});
```

```
client.Connect("127.0.0.1", 12345);
```

```
client.WaitForHandshake();
```

```
client.Send("Hello", "World");or non-blocking way of waiting for handshake:
```

```
Code client.OnHandshake += (SyncIOClient c, Guid ID, bool success) => {
    if (success)
        c.Send("Hello", "World", "from", ID);
    else
        Console.WriteLine("Failed handshake.");
};
```

To-DoImplement popper encryption support (Internal structure is done)

Add UDP support (mabe)

Make this thread more presentable than your waifu.

More throughout examples:

Listening on ports

```
Code var port = server.ListenTCP(1000);
```

```
sock.OnDisconnect += (sender, err) => {
    //Called when socket is closed.
```

```
};
```

```
port.Dispose(); //Stop listening on this port.
```

Listening on multiple ports:

```
Code server.ListenTCP(1000);
```

```
server.ListenTCP(1001);
server.ListenTCP(1002);
Looping through open ports:
Code foreach (var s in server) {
    Console.WriteLine(s.EndPoint.Port);
    //s.Dispose() if you want to close all open ports
}
Getting already open port from server without var:
```

```
Code server.ListenTCP(1000);
var sock = server[1000];
```

```
//Equivalent to
var sock = server.ListenTCP(1000);
Custom types
```

To send custom types, the classes must meet two criteriers:
 Impliment IPacket interface
 Marked with Seralizable attribute

For example, a valid class for sending might look like this:

```
Code [Serializable]
class EchoMessage : IPacket {
    public string Message { get; set; }
    public EchoMessage(string msg) { //(Constructor is not nessisary
        Message = msg;
    }
}
```

Now we have to add this class to our client and server so they can send this type. First step is to create a packer that has all of the custom types we want.

An example would be:

```
Code var packer = new Packager(new Type[] {
    typeof(EchoMessage),
});
```

Then create server/client instances with this packer.

(They dont have to be the same instance of the Packager, but both the packager for the client and server must contain all the types you want to send)

```
Code var server = new SyncIOServer(TransportProtocal.IPv4, packer);
var client = new SyncIOClient(TransportProtocal.IPv4, packer);
```

Handlers

There are three types of handlers for both client and server.

1) Generic array handler

```
Code server.SetHandler((SyncIOConnectedClient sender, object[] data) => {

});Code client.SetHandler((SyncIOClient sender, object[] data) => {

});
```

This handler will be called when the sending item is NOT a SINGLE IPacket object.

For example, these will all trigger this event on the server:

```
Code client.Send("Hello World");
```

```
//data[0] = "Hello World"
```

```
client.Send("Hello", "World");  
//data[0] = "Hello"  
//data[1] = "World"
```

```
client.Send("Hello", 1, 2);  
//data[0] = "Hello"  
//data[1] = 1  
//data[2] = 2
```

```
client.Send(new EchoMessage("Echo"), "Message");  
//data[0] = (EchoMessage {Message = "Echo" })  
//data[1] = "Message"
```

```
client.Send(new EchoMessage("Hello"), new EchoMessage("World"));  
//data[0] = (EchoMessage {Message = "Hello" })  
//data[1] = (EchoMessage {Message = "World" })
```

2) Specific packet handler

```
Code server.SetHandler<EchoMessage>((SyncIOConnectedClient sender, EchoMessage packet)  
=> {
```

```
});Code client.SetHandler<EchoMessage>((SyncIOClient sender, EchoMessage packet) => {  
  
});
```

Any valid custom types can be used, but for this example EchoMessage is used.

Each type can only have one handler, but there can be many handlers of different types.

Spoiler ([Click to View](#))

For example

```
Code client.SetHandler<EchoMessage>(MyCallback); //Set the handler for EchoMessage  
client.SetHandler<EchoMessage>(MyNewCallback); //Set the handler for EchoMessage,  
overwriting the previous EchoMessage handler
```

```
client.SetHandler<MyClass1>(MyClass1Handler); //Set the handler for MyClass1  
client.SetHandler<MyClass2>(MyClass2Handler); //Set the handler for MyClass2
```

```
client.SetHandler<MyClass1>(MyClass1Handler2); //Overwrite the MyClass1 handler.
```

This will trigger this event on the server:

```
Code client.Send(new EchoMessage("Echo"));  
//packet = (EchoMessage {Message = "Echo" })
```

2) Generic packet handler

```
Code server.SetHandler<EchoMessage>((SyncIOConnectedClient sender, IPacket packet) => {
```

```
});Code client.SetHandler<EchoMessage>((SyncIOClient sender, IPacket packet) => {  
  
});
```

This handler will handle all valid custom types that do not have a specific packet handler. Generally, this would not be used as all custom types added to the packer should have handlers assigned.

Malum™ - December 27, 2016, 6:18 pm

Holy shit! Perfect timing. I've been using NetworkComms and playing with other libraries and they're always so complicated for simple things. I will be using this, thanks!

CoderEX - December 27, 2016, 6:39 pm

Your code is so clean; it is beautiful! Teach me your ways. LOL.

NinjaLeaks - December 27, 2016, 6:41 pm

Damn thats some sexy ass code lol. Really nice op Ill be using this in the future!

+Haze - December 27, 2016, 6:45 pm

After all this time BahNahNah, I've only just come to realise you're actually a very talented developer. Keep up the work man, honestly, you'll go far.

HFDone - December 27, 2016, 7:09 pm

didnt take alook yet but i hope its not base on your esockt class

BahNahNah - December 27, 2016, 7:13 pm

(Dec 27, 2016 08:09 PM)HFDone Wrote: [\[CITE\]](#) didnt take alook yet but i hope its not base on your esockt class

We dont talk about that class.

HFDone - December 27, 2016, 7:24 pm

whats framework its base on

BahNahNah - December 27, 2016, 7:30 pm

(Dec 27, 2016 08:24 PM)HFDone Wrote: [\[CITE\]](#) whats framework its base on
Its not based on anything.

ubbelol - December 28, 2016, 1:25 am

Looks like a nice simple library. A few typos here and there in the naming of stuff, but that doesn't really matter. :P You might wanna consider adding a timeout for the blocking handshaking at <https://github.com/BahNahNah/SyncIO/blob/master/SyncIO/Client/SyncIOClient.cs#L153> so you don't get a total deadlock if something goes wrong.

BahNahNah - December 28, 2016, 1:52 am

(Dec 28, 2016 02:25 AM)ubbelol Wrote: [\[CITE\]](#) Looks like a nice simple library. A few typos here and there in the naming of stuff, but that doesn't really matter. :P You might wanna consider adding a timeout for the blocking handshaking at <https://github.com/BahNahNah/SyncIO/blob/master/SyncIO/Client/SyncIOClient.cs#L153> so you don't get a total deadlock if something goes wrong.

Diddnt think about that ?

Thanks, updated.

Ninja - December 28, 2016, 4:18 am

First of all, thanks for this awesome release BahNahNah, it is really nice.

Correct me if I am wrong but shouldn't this:

<http://prnt.sc/douhoq>

be "World"?

And can't you make the WaitForHandshake awaitable or something?

BahNahNah - December 28, 2016, 4:41 am

(Dec 28, 2016 05:18 AM)Ninja Wrote: [\[CITE\]](#) First of all, thanks for this awesome release BahNahNah, it is really nice.

Correct me if I am wrong but shouldn't this:

<http://prnt.sc/douhoq>

be "World"?

And can't you make the WaitForHandshake awaitable or something?

Yes it is ;p

I could make it awaitable, but WaitForHandshake is suppose to be blocking. Subscribing to the event is the non-blocking way.

Ninja - December 28, 2016, 4:42 am

(Dec 28, 2016 05:41 AM)BahNahNah Wrote: [\[CITE\]](#) Yes it is ;p

I could make it awaitable, but WaitForHandshake is suppose to be blocking. Subscribing to the event is the non-blocking way.

How about Application.DoEvents? (° ? °)

ubbelol - December 28, 2016, 4:59 am

(Dec 28, 2016 05:42 AM)Ninja Wrote: [\[CITE\]](#) How about Application.DoEvents? (° ? °)

I'm not sure if you're serious or not but that's not how ManualResetEvent works. Also Application.DoEvents is windows forms stuff and not really related.

Ninja - December 28, 2016, 5:03 am

(Dec 28, 2016 05:59 AM)ubbelol Wrote: [\[CITE\]](#) I'm not sure if you're serious or not but that's not how ManualResetEvent works. Also Application.DoEvents is windows forms stuff and not really related.

The "(° ? °)" indicates sarcasm.

HFDone - December 28, 2016, 7:48 am

(Dec 27, 2016 08:30 PM)BahNahNah Wrote: [\[CITE\]](#) Its not based on anything.

once dumb always dumb ?

SIM-1847363 - December 28, 2016, 7:56 am

Really nice job surprised that Ninja didn't bring up his Network/Socks Library. (° ? °) I'm very interested in this project will take a look at it later. Very nice releases, great work keep it up.

Ninja - December 28, 2016, 10:36 am

(Dec 28, 2016 08:56 AM)Kevin. Wrote: [\[CITE\]](#) Really nice job surprised that Ninja didn't bring up his Network/Socks Library. (° ? °) I'm very interested in this project will take a look at it later. Very nice releases, great work keep it up.

About that, link available on my Github. (° ? °)

To be fair it is not on par with this. Mainly because it was designed to only be used by me for HTTP stuff and not general socket applications.

BahNahNah - December 28, 2016, 9:35 pm

Finished UDP support. It has been pushed to the [dev branch](#). I will push it to the master branch when i have done some more testing.

Any feedback is appreciated.

<https://github.com/BahNahNah/SyncIO/releases/tag/1.1.0-dev>

Next step, [Remote variables](#).

Ninja - December 29, 2016, 3:35 am

You could add "implementing proxy protocols" to your to-do list, if you want? It is something that should be in every networking lib imo.

Malum™ - December 29, 2016, 4:09 am

(Dec 28, 2016 10:35 PM)BahNahNah Wrote: [\[CITE\]](#) Finished UDP support. It has been pushed to the [dev branch](#). I will push it to the master branch when i have done some more testing.

Any feedback is appreciated.

<https://github.com/BahNahNah/SyncIO/releases/tag/1.1.0-dev>

Next step, [Remote variables](#).

I'm loving this. Keep updating and the more I'll use it. I'm about to either make a base framework for a RAT or a private (encrypted) chat system with it.

EDIT:

Having fun with it! Thank you.

CoderEX - December 29, 2016, 8:00 am

Out of curiosity, is the logic behind this similar to that of LiteCode?

Malum™ - December 29, 2016, 8:57 am

(Dec 29, 2016 09:00 AM)???? Wrote: [\[CITE\]](#) Out of curiosity, is the logic behind this similar to that of LiteCode?

For the lazy, [this](#) is what he is referring to when he says LiteCode. DragonHunter is god.

kozit - December 29, 2016, 6:14 pm

yay a way to do networking without wanting to kill someone.

USERCRYPT - January 1, 2017, 6:06 am

Thk's for code good job men

BahNahNah - January 20, 2017, 7:16 am

SyncIO v1.1.0

[Download](#)

New features:UDP Support (client -> server)

Remote function calling (client -> server)

Examples of both of these features can be found in their unit testing (linked on the release page above)

Next on the list:Custom connection methods (e.g. Socks5 wich will be default)

Remote variables

Sorry for the lack of updates, been busy with other stuff :/

Barret - January 20, 2017, 7:26 am

(Jan 20, 2017 08:16 AM)BahNahNah Wrote: [\[CITE\]](#) SyncIO v1.1.0

[Download](#)

New features:UDP Support (client -> server)

Remote function calling (client -> server)

Examples of both of these features can be found in their unit testing (linked on the release page above)

Next on the list: Custom connection methods (e.g. Socks5 which will be default)
Remote variables

Sorry for the lack of updates, been busy with other stuff :/
Looks like something i would like to contribute to, im interested :D

gigajew - January 20, 2017, 6:59 pm

Good job! I am a fan of your work. And I enjoy seeing how you improve your code over time

Nick (Name) - February 19, 2017, 10:50 am

Wow! This is actually very readable and organized! Nice work BahNahNah, I'm curious so see how well this kicks off with more updates! I'll probably fork it :)

FDSJKLJFL - October 2, 2017, 12:10 pm

There seems to be a problem with the remote functions. In my setup I have around 10 threads accessing the remote functions and after a while some threads seem to get stuck on the remote function and stall indefinitely. Im using the CallWait method.

Ninja - October 2, 2017, 2:04 pm

(Oct 2, 2017 12:10 PM)tim128 Wrote: [\[CITE\]](#) There seems to be a problem with the remote functions. In my setup I have around 10 threads accessing the remote functions and after a while some threads seem to get stuck on the remote function and stall indefinitely. Im using the CallWait method.
Why don't you make an issue on Github instead of grave digging a thread?

Immortal - October 2, 2017, 2:30 pm

Dude this is fucking amazing great job.