

Forum:[Hack Forums](#)

Thread: [\[C#\] TCP Modules](#)

Local Celeb - September 18, 2010, 12:20 pm

These are just some modules I wrote up, basic, but pretty full proof.

If there are any errors I'm sorry. I spent quite some time working the bugs out, although it may not seem like I have, I was learning as I was going.

Anyways, here's a demonstration video on what they do etc.

And here is the code :

If you use it, please say thanks or something ?

TCPClient.cs :

PHP Code

```
<?php
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net.Sockets;

namespace TCP_Modules
{
    class TCPClient
    {
        #region Declarations
        static TcpClient Client = new TcpClient();
        static NetworkStream CStream;
        static ASCIIEncoding CEncode = new ASCIIEncoding();
        #endregion

        public static Boolean Connect(String IPAddress, int Port)
        {
            try
            {
                Client.Connect(IPAddress, Port);
                if (Client.Connected != false)
                {
                    CStream = Client.GetStream();
                    return true;
                }
            }
            else
            {
                return false;
            }
        }
    }
}
```

```

    }
}
catch (Exception Ex)
{
    return false;
}
}
public static String GetData()
{
    if (CStream.DataAvailable)
    {
        Byte[] A = new Byte[1024];
        int b = 0;
        try
        {
            StringBuilder StringBuild = new StringBuilder();
            b = CStream.Read(A, 0, A.Length);
            StringBuild.AppendFormat("{0}", Encoding.ASCII.GetString(A, 0, b));
            CStream.Flush();
            return StringBuild.ToString();
        }
        catch (Exception Ex)
        {
            return null;
        }
    }
    else
    {
        return null;
    }
}
public static Boolean Disconnect()
{
    try
    {
        Client.Close();
        if (Client.Connected != true)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception Ex)
    {
        return false;
    }
}
public static void SendData(String Data)
{
    Byte[] Sending = CEncode.GetBytes(Data);

```

```

        CStream.Write(Sending, 0, Sending.Length);
    }
}
}

```

TCPServer.cs :

PHP Code

```
<?php
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net.Sockets;
using System.Net;
using System.IO;

```

```
namespace TCP_Modules
```

```

{
    class TCPServer
    {
        static int Port = 2143;
        static TcpListener Server;
        static TcpClient AServer;
        static NetworkStream ServerStream;
        static ASCIIEncoding SEncode = new ASCIIEncoding();

        public static Boolean ListenAccept()
        {
            if (Server.Pending() == true)
            {
                AServer = Server.AcceptTcpClient();
                ServerStream = AServer.GetStream();
                return true;
            }
            return false;
        }

        public static void Start()
        {
            Server = new TcpListener(IPAddress.Any, Port);
            Server.Start();
        }

        public static void SetPort(int PortNum)
        {
            Port = PortNum;
        }

        public static String GetData()
        {
            if (ServerStream.DataAvailable)
            {
                Byte[] A = new Byte[1024];
                int b = 0;
            }
        }
    }
}

```

```

    try
    {
        StringBuilder StringBuild = new StringBuilder();
        b = ServerStream.Read(A, 0, A.Length);
        StringBuild.AppendFormat("{0}", Encoding.ASCII.GetString(A, 0, b));
        ServerStream.Flush();
        return StringBuild.ToString();
    }
    catch (Exception Ex)
    {
        return null;
    }
}
else
{
    return null;
}
}
public static Boolean IsConnected()
{
    if (AServer.Connected == true)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public static void SendData(String Data)
{
    Byte[] Sending = SEncode.GetBytes(Data);
    ServerStream.Write(Sending, 0, Sending.Length);
}
}
}
}

```

You can remove the namespaces if you want, it makes no difference.

The Weeknd. - September 19, 2010, 3:57 am

Nice snippet mate.

Although I won't need this as I already know how to use it I'm glad to thank you for sharing!

Ah, maybe if you provide a VB.net conversion you'll have more visibility.

Local Celeb - October 6, 2010, 1:38 pm

Bumping some old topics of mine :)

Rue - October 6, 2010, 1:41 pm

Why static classes?

What if i want multiple servers or multiple clients?

Local Celeb - October 6, 2010, 1:43 pm

(Oct 6, 2010 01:41 PM)Rue Wrote: [\[CITE\]](#) Why static classes?

What if i want multiple servers or multiple clients?
>.>

I think you can change that on your own.

That's just the way I made them.

Rue - *October 6, 2010, 1:45 pm*

(Oct 6, 2010 01:43 PM)Asydix Wrote: [\[CITE\]](#) >.>

I think you can change that on your own.

That's just the way I made them.
Just saying...

Static classes are not appropriate for this sort of thing.