

C# Communication

[http://msdn.microsoft.com/en-us/library/system.net.sockets.tcplistener\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.net.sockets.tcplistener(v=vs.110).aspx)

C# Communication

- Target=> Chat Program
 - Chat Server
 - Chat Client

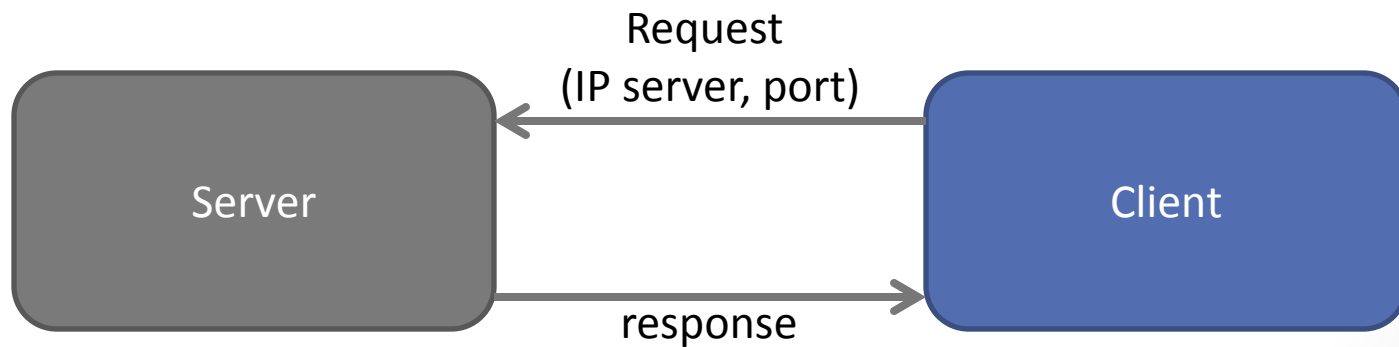
Microsoft .Net framework มี namespace 2 ตัวที่ใช้จัดการกับ Internet protocol ทำให้โปรแกรมของเราสามารถส่งข้อมูลหรือรับข้อมูลผ่าน Internet ได้ นั่นคือ

System.Net และ System.Net.Sockets

C# Socket programming

- Socket คือการเชื่อมต่อการสื่อสารระหว่างจุดสองจุด (เครื่องสองเครื่อง) แบบไปกลับได้ระหว่างโปรแกรมสองโปรแกรม (server กับ client) ภายในเครือข่ายเดียวกัน
 - Server Socket Program เป็นโปรแกรมที่รันบนเครื่องที่มี socket ที่ผูกกับ Port number บนเครื่องและรอ request ที่จะเข้ามาจาก client
 - Client Socket Program จะต้องรู้ว่า IP Address ของเครื่องที่ server socket program ทำงานอยู่ และ port ที่เครื่องนั้นรอฟัง request ด้วย

- เมื่อมีการสร้างการเชื่อมต่อระหว่าง server และ client ก็จะสามารถส่งข้อมูลผ่านทาง socket



ชนิดของการเชื่อมต่อ

- ในการเขียน Socket Programming ใน C# นั้น มี Protocol ในการเชื่อมต่อนั้นมี 2 ชนิด
 - TCP/IP
 - UDP/IP

ต่อไปเราจะมาลองทำเขียน socket programming โดยใช้ TCP/IP

Server Socket Program

- จะใช้ C# เขียนแบบ Console
- Server จะรอ request จาก Client
- เมื่อ Server ได้รับ request จาก client, Server จะส่ง response ไปยัง Client

Client Socket Program

- จะใช้ C# Windows Form
- เมื่อ Client ทำงาน
 - มันจะสร้างการเชื่อมต่อไปยัง Server program
 - ส่ง request ไปยัง Server
 - รับ response จาก Server

Server Socket Program

กำหนดว่าเราจะใช้ port หมายเลข 13000 สำหรับ server socket
เราจะสร้าง instance ของ class TcpListener และสั่งให้ทำงาน

```
// Set the TcpListener on port 13000.
```

```
Int32 port = 13000;
```

```
IPAddress localAddr = IPAddress.Parse("127.0.0.1");
```

```
TcpListener server=null;
```

```
server = new TcpListener(localAddr, port);
```

```
//Start listening for client request
```

```
server.Start();
```


การอ่านเขียนข้อมูล

ขั้นต่อไป เราต้องรองรับ request เรื่อย ๆ
เราจะสร้าง infinite loop สำหรับจัดการ
request

เมื่อ server socket ได้รับ request จาก
client ข้อมูลนั้นเราจะอ่านและเขียนด้วย
NetworkStream

```
using System;
```

```
using System.IO;
```

```
using System.Net;
```

```
using System.Net.Sockets;
```

```
using System.Text;
```

```
class MyTcpListener {  
    public static void Main() {  
        TcpListener server=null;  
  
        try { // Set the TcpListener on port 13000.  
            Int32 port = 13000;  
            IPAddress localAddr = IPAddress.Parse("127.0.0.1");  
            // TcpListener server = new TcpListener(port);  
            server = new TcpListener(localAddr, port);  
            // Start listening for client requests.  
            server.Start();  
            // Buffer for reading data  
            Byte[] bytes = new Byte[256];  
            String data = null;
```

```

// Enter the listening loop.
while(true) { Console.WriteLine("Waiting for a connection... ");

    // Perform a blocking call to accept requests.
    TcpClient client = server.AcceptTcpClient();

    Console.WriteLine("Connected!");

    data = null;

    // Get a stream object for reading and writing
    NetworkStream stream = client.GetStream();

    int i;

    // Loop to receive all the data sent by the client.
    while((i = stream.Read(bytes, 0, bytes.Length))!=0) {

        // Translate data bytes to a ASCII string.
        data = System.Text.Encoding.ASCII.GetString(bytes, 0, i);

        Console.WriteLine("Received: {0}", data);

        // Process the data sent by the client.
        data = data.ToUpper();

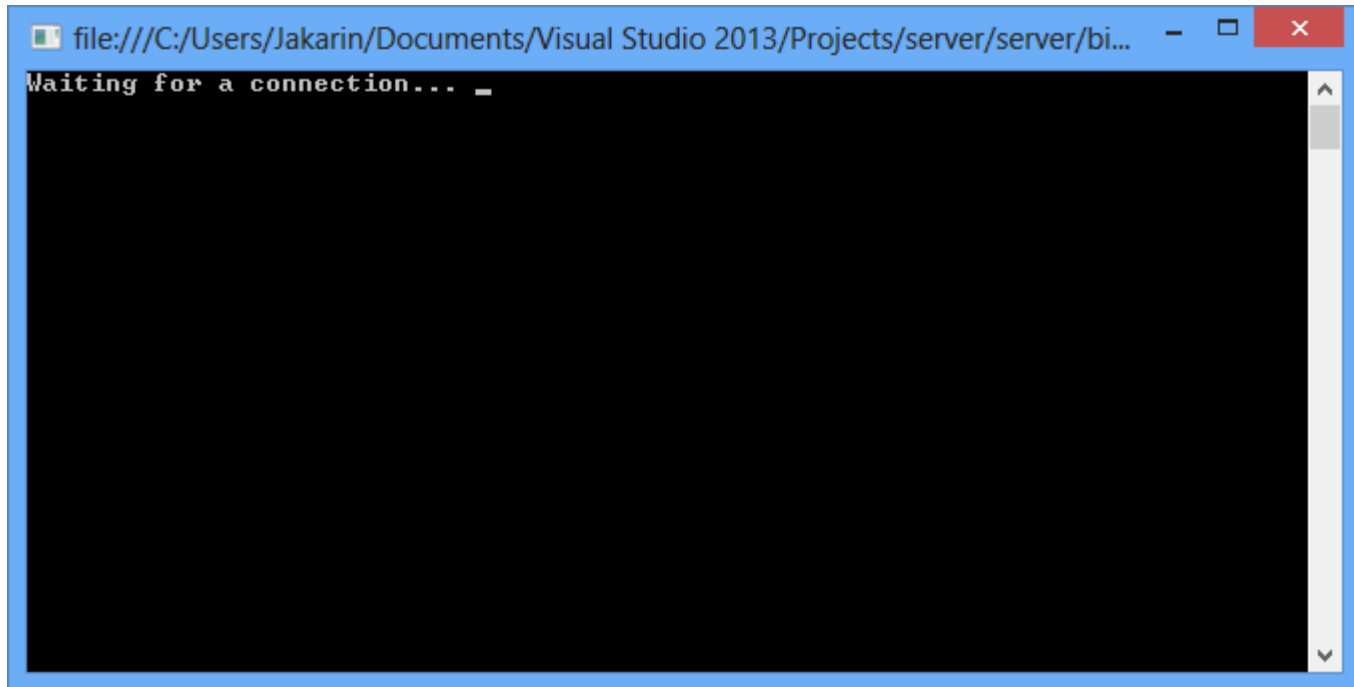
        byte[] msg = System.Text.Encoding.ASCII.GetBytes(data);

        // Send back a response.
        stream.Write(msg, 0, msg.Length);

        Console.WriteLine("Sent: {0}", data);
    }
}

```

```
        // Shutdown and end connection
        client.Close();
    }
} catch(SocketException e) {
    Console.WriteLine("SocketException: {0}", e);
} finally {
    // Stop listening for new clients.
    server.Stop();
}
Console.WriteLine("\nHit enter to continue..."); Console.Read();
}
}
```



A screenshot of a Windows command prompt window. The title bar shows the file path: `file:///C:/Users/Jakarin/Documents/Visual Studio 2013/Projects/server/server/bi...`. The main area of the window is black with white text that reads `Waiting for a connection... _`. A vertical scrollbar is visible on the right side of the window.

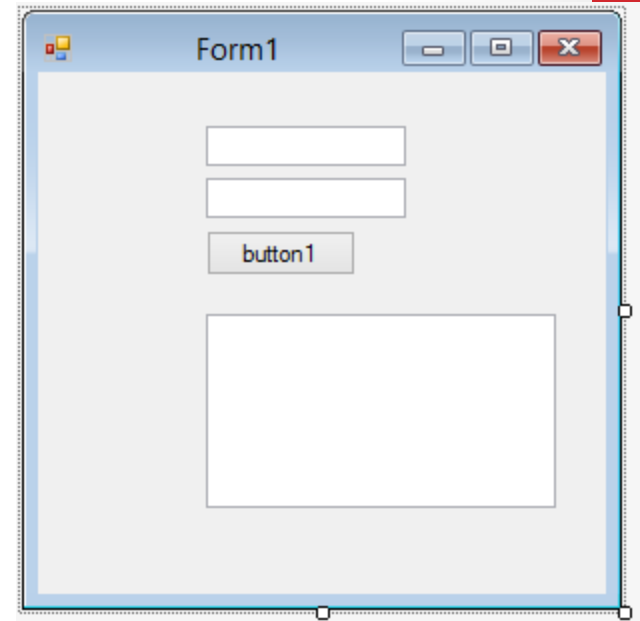
Client Socket Program

- ใช้ window form
- Client จะเชื่อมต่อที่ port 13000
ของ server ที่มี IP 127.0.0.1
(เครื่องตัวเอง)

```
Int32 port = 13000;
```

```
TcpClient client = new TcpClient(server, port);
```

- `using System;`
- `using System.IO;`
- `using System.Net;`
- `using System.Net.Sockets;`
- `using System.Text;`
- `using System.Windows.Forms;`




```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        String server = textBox1.Text;
        String message = textBox2.Text;
        try
        {
            // Create a TcpClient.
            // Note, for this client to work you need to have a TcpServer
            // connected to the same address as specified by the server, port
            // combination.
            Int32 port = 13000;
            TcpClient client = new TcpClient(server, port);
            // Translate the passed message into ASCII and store it as a Byte array.
            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);
```

```
// Get a client stream for reading and writing.
NetworkStream stream = client.GetStream();

// Send the message to the connected TcpServer.
stream.Write(data, 0, data.Length);
Console.WriteLine("Sent: {0}", message);

// Receive the TcpServer.response.

// Buffer to store the response bytes.
data = new Byte[256];

// String to store the response ASCII representation.
String responseData = String.Empty;

// Read the first batch of the TcpServer response bytes.
Int32 bytes = stream.Read(data, 0, data.Length);
responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
textBox3.Text = textBox3.Text + responseData;
Console.WriteLine("Received: {0}", responseData);

// Close everything.
stream.Close();
client.Close();
}
```

```
catch (ArgumentNullException ex)
{
    Console.WriteLine("ArgumentNullException: {0}", ex);
}
catch (SocketException ex)
{
    Console.WriteLine("SocketException: {0}", ex);
}

Console.WriteLine("\n Press Enter to continue...");
Console.Read();
}
}
```

Form1

127.0.0.1

Test

button1

TEST

```
file:///C:/Users/Jakarin/Documents/Visual Studio 2013/Projects/server/server/bi...  
Waiting for a connection... Connected!  
Received: Test  
Sent: TEST  
Waiting for a connection... _
```