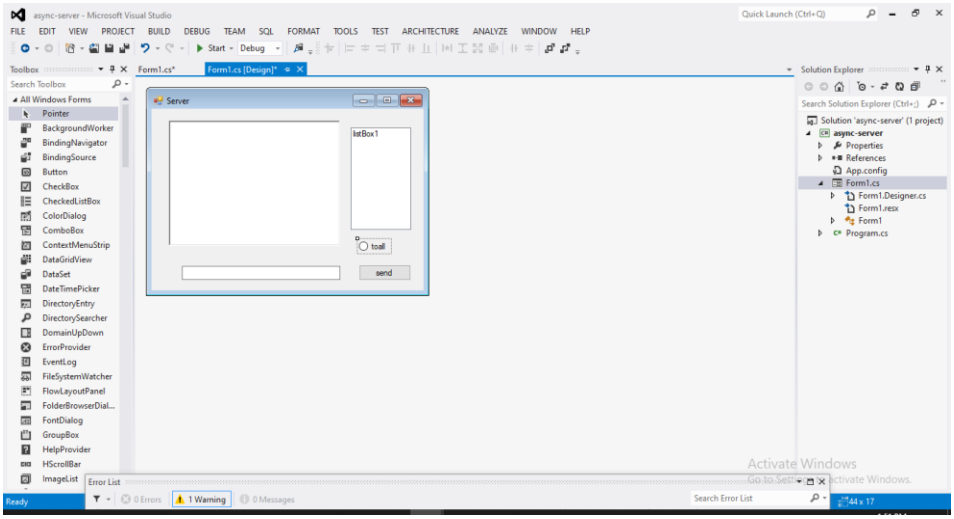
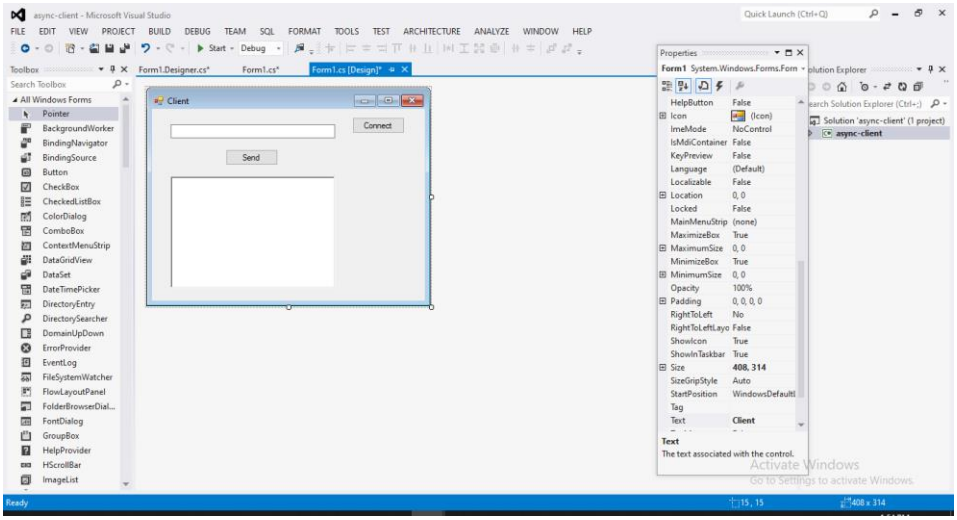
Group Messaging (Task)

Server Form:



Client Form:



Server Code

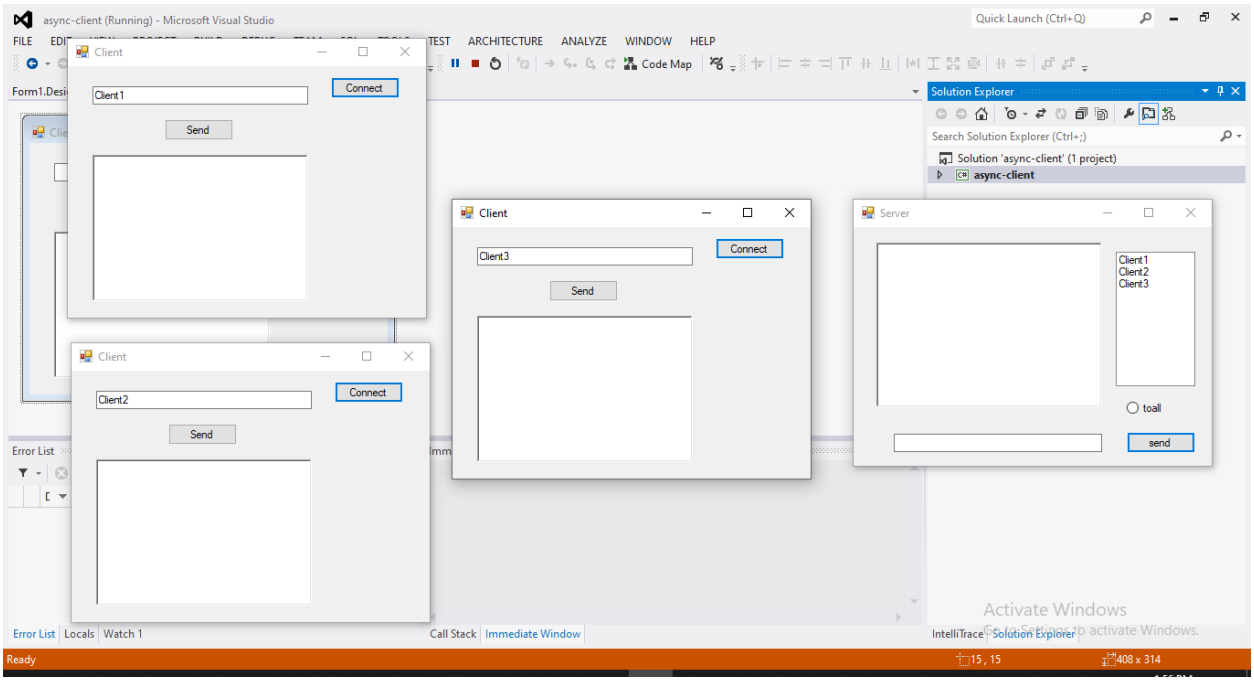
```
using System.Net;
using System.Net.Sockets;
using System.IO;

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

        private void Form1_Load(object sender, EventArgs e)
        {
            CheckForIllegalCrossThreadCalls = false;
            TcpListener listener = new TcpListener(IPAddress.Loopback, 11000);
            listener.Start(10);
            listener.BeginAcceptTcpClient(new AsyncCallback(client_connect),listener);

        }
        Dictionary<string, TcpClient> lstClients = new Dictionary<string, TcpClient>();
        byte[] b = new byte[1024];
        private void client_connect(IAsyncResult ar){

            TcpListener listener = (TcpListener)ar.AsyncState;
            TcpClient client = listener.EndAcceptTcpClient(ar);
            NetworkStream ns = client.GetStream();
            object [] a  = new object[2];
            a[0]=ns;
            a[1]=client;
            ns.BeginRead(b,0,b.Length ,new AsyncCallback(ReadMsg),a);
            listener.BeginAcceptTcpClient(new AsyncCallback(client_connect),listener);

        }
        private void ReadMsg(IAsyncResult ar){

            object [] a = (object[])ar.AsyncState;
            NetworkStream ns = (NetworkStream)a[0];
            TcpClient client = (TcpClient)a[1];
            int count = ns.EndRead(ar);
            string msg=ASCIIEncoding.ASCII.GetString(b,0,count);
            if (msg.Contains("@name@")){

            string name= msg.Replace("@name@","");
            lstClients.Add(name, client);
             listBox1.Items.Add(name);

            }
            else{
            richTextBox1.Text+=msg+Environment.NewLine;

            }
            ns.BeginRead(b,0,b.Length, new AsyncCallback(ReadMsg),a);

        }
```

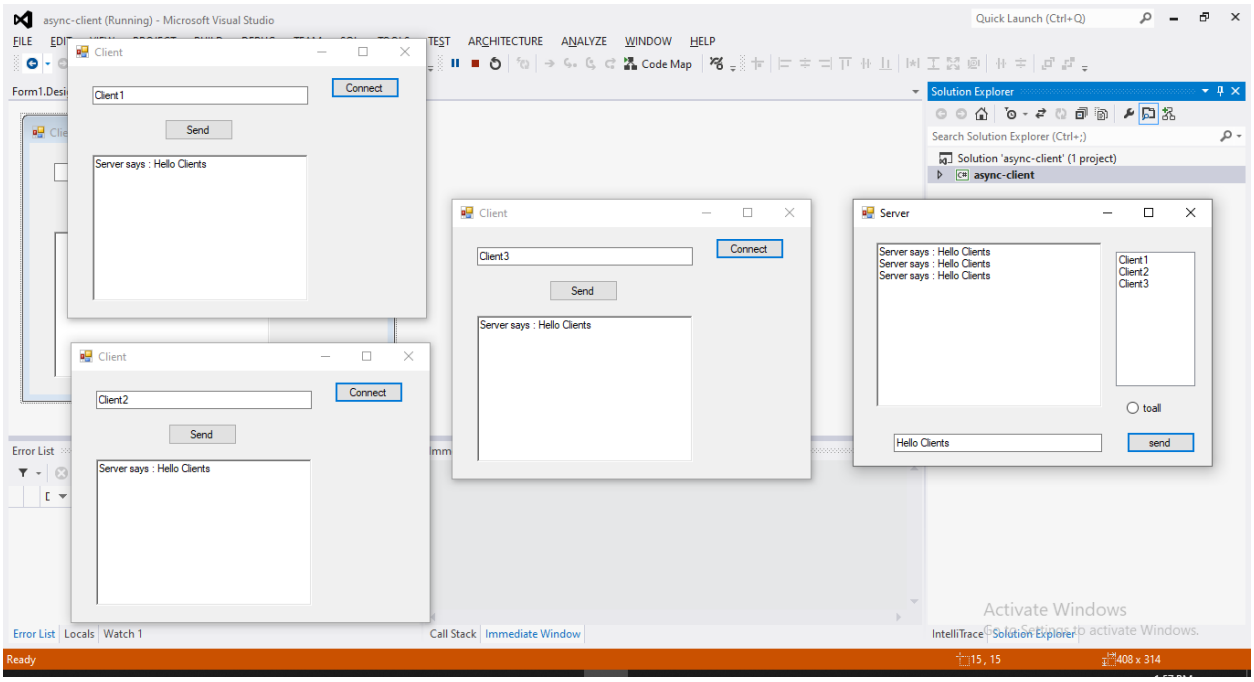
```
        private void button1_Click(object sender, EventArgs e)
        {

            if (radioButton1.Checked == true)
            {
                string items = "";
                foreach (var item in listBox1.Items)
                {
                    TcpClient client = (TcpClient)lstClients[item.ToString()];

                    NetworkStream ns = client.GetStream();
                    StreamWriter sw = new StreamWriter(ns);

                    string texttosend = "Server says : " + textBox1.Text;
                    sw.WriteLine(texttosend);

                    richTextBox1.Text += texttosend + Environment.NewLine;

                    sw.Flush();

                }

            }

            else if (radioButton1.Checked==false)
            {

                TcpClient client =
(TcpClient)lstClients[listBox1.SelectedItem.ToString()];

                NetworkStream ns = client.GetStream();
                StreamWriter sw = new StreamWriter(ns);

                string texttosend = "Server says : " + textBox1.Text;
                sw.WriteLine(texttosend);

                richTextBox1.Text += texttosend + Environment.NewLine;

                sw.Flush();

            }
            radioButton1.Checked = false;
        }
    }
}
```

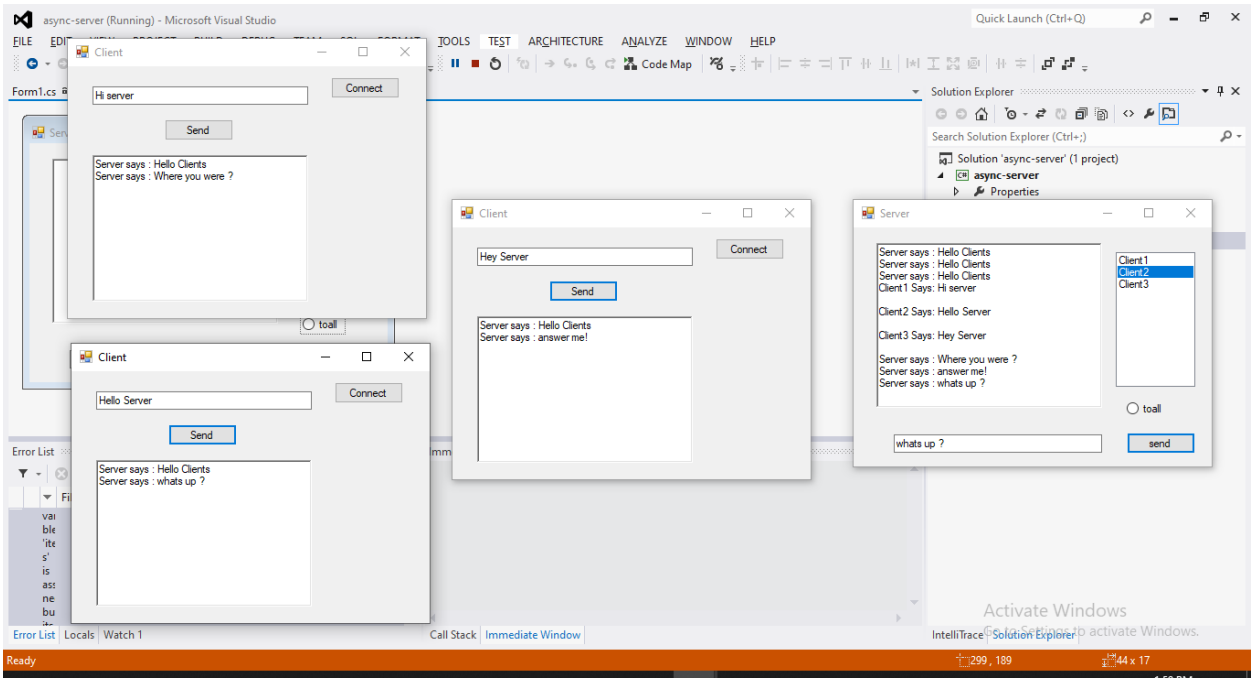
Client Code

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

namespace async_client
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public string name;
        byte[] b = new byte[1024];
        TcpClient client = new TcpClient();
        private void button1_Click(object sender, EventArgs e)
        {
            CheckForIllegalCrossThreadCalls = false;
            client.Connect(IPAddress.Loopback,11000);

            NetworkStream ns = client.GetStream();
            StreamWriter sw = new StreamWriter(ns);
           name = textBox1.Text;
            sw.WriteLine("@name@"+textBox1.Text);
            sw.Flush();
            ns.BeginRead(b,0,b.Length,readmsg,ns);
        }
        private void readmsg(IAsyncResult ar) {

            NetworkStream ns = (NetworkStream)ar.AsyncState;
            int count = ns.EndRead(ar);
            richTextBox1.Text += ASCIIEncoding.ASCII.GetString(b, 0, count);
            ns.BeginRead(b,0,b.Length,readmsg,ns);

        }
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            NetworkStream ns = client.GetStream();
            StreamWriter sw = new StreamWriter(ns);
            sw.WriteLine(name+" Says: "+textBox1.Text);
            sw.Flush();
        }

    }
}
```

Output


async-client (Running) - Microsoft Visual Studio
Client
Client1
Connect
Send
Client2
Client3
Server
Client1
Client2
Client3
toall
send
Solution Explorer
Solution 'async-client' (1 project)
async-client
Activate Windows


async-client (Running) - Microsoft Visual Studio
Client
Client1
Connect
Send
Server says : Hello Clients
Client2
Client3
Server
Hello Clients
toall
send
Solution Explorer
Solution 'async-client' (1 project)
async-client
Activate Windows


async-server (Running) - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE
TOOLS
TEST
ARCHITECTURE
ANALYZE
WINDOW
HELP
Code Map
Client
Hi server
Connect
Send
Server says : Hello Clients
Server says : Where you were ?
toall
Client
Hey Server
Connect
Send
Server says : Hello Clients
Server says : answer me!
Server
Server says : Hello Clients
Server says : Hello Clients
Server says : Hello Clients
Client1 Says: Hi server
Client2 Says: Hello Server
Client3 Says: Hey Server
Server says : Where you were ?
Server says : answer me!
Server says : whats up ?
Client1
Client2
Client3
toall
whats up ?
send
Client
Hello Server
Connect
Send
Server says : Hello Clients
Server says : whats up ?
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'async-server' (1 project)
async-server
Properties
Form1.cs
Error List
Locals
Watch 1
Call Stack
Immediate Window
IntelliTrace
Solution Explorer
Activate Windows
Ready
299 , 189
44 x 17