Your Company Name

Tel: +44 1234 567 9898
Fax: +44 1234 545 9999
email: info@@company.com

Microsoft Visual Studio

C# Project

Source Code Output

Created using

VScodePrint

Macro Variables Substitution Example

General Date	24/01/2017 6:41:20 PM
Long Date	Tuesday, 24 January 2017
Short Date	24/01/2017
Long Time	6:41:20 PM
Short Time	18:41
Today	24/01/2017 6:41:20 PM
Project Name	ServerApp

Prepared by
Joginder S Nahil
on
24/01/2017 6:41:20 PM

WWW.STARPRINTTOOLS.COM

```
1   using System;
2   using System.Collections.Generic;
3   using System.ComponentModel;
4   using System.Data;
5   using System.Drawing;
6   using System.Linq;
7   using System.Net;
8   using System.Net.Sockets;
9   using System.Text;
10  using System.Windows.Forms;
11
12  namespace ClientApp
13  {
14      public partial class Form1 : Form
15      {
16          private string myMessage = "";
17          private TcpClient client = new TcpClient();
18          private IPEndPoint serverEndPoint = new IPEndPoint(IPAddress.Parse(
            "127.0.0.1"), 3000);
19
20          public Form1()
21          {
22              InitializeComponent();
23              client.Connect(serverEndPoint);
24
25          }
26
27          private void RtbClientKeyDown(object sender, KeyEventArgs e)
28          {
29              if (e.KeyData != Keys.Enter || e.KeyData != Keys.Return)
30              {
31                  myMessage += (char)e.KeyValue;
32              }
33              else
34              {
35                  SendMessage(myMessage);
36                  myMessage = "";
37
38              }
39          }
40
41          private void SendMessage(string msg)
42          {
43              NetworkStream clientStream = client.GetStream();
44
45              ASCIIEncoding encoder = new ASCIIEncoding();
46              byte[] buffer = encoder.GetBytes(msg);
47
48              clientStream.Write(buffer, 0, buffer.Length);
49              clientStream.Flush();
50
51              // Receive the TcpServer.response.
52
53              // Buffer to store the response bytes.
54              Byte[] data = new Byte[256];
55
56              // String to store the response ASCII representation.
57              String responseData = String.Empty;
58
59              // Read the first batch of the TcpServer response bytes.
60              Int32 bytes = clientStream.Read(data, 0, data.Length);
61              responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
62
63              rtbClient.AppendText(Environment.NewLine + "From Server: " +
                responseData);
64          }
65
66
67      }
68  }
```

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Windows.Forms;
5
6   namespace ClientApp
7   {
8       static class Program
9       {
10          /// <summary>
11          /// The main entry point for the application.
12          /// </summary>
13          [STAThread]
14          static void Main()
15          {
16              Application.EnableVisualStyles();
17              Application.SetCompatibleTextRenderingDefault(false);
18              Application.Run(new Form1());
19          }
20      }
21  }
```

```
1   using System;
2   using System.Collections.Generic;
3   using System.ComponentModel;
4   using System.Data;
5   using System.Drawing;
6   using System.Linq;
7   using System.Net;
8   using System.Net.Sockets;
9   using System.Text;
10  using System.Threading;
11  using System.Windows.Forms;
12
13  namespace ServerApp
14  {
15      public partial class Form1 : Form
16      {
17          private TcpListener tcpListener;
18          private Thread listenThread;
19          private int connectedClients = 0;
20          private delegate void WriteMessageDelegate(string msg);
21
22          public Form1()
23          {
24              InitializeComponent();
25              Server();
26          }
27
28          private void Server()
29          {
30              this.tcpListener = new TcpListener(IPAddress.Loopback, 3000); // Cha
                nge to IPAddress.Any for internet wide Communication
31              this.listenThread = new Thread(new ThreadStart(ListenForClients));
32              this.listenThread.Start();
33          }
34
35          private void ListenForClients()
36          {
37              this.tcpListener.Start();
38
39              while (true) // Never ends until the Server is closed.
40              {
41                  //blocks until a client has connected to the server
42                  TcpClient client = this.tcpListener.AcceptTcpClient();
43
44                  //create a thread to handle communication
45                  //with connected client
46                  connectedClients++; // Increment the number of clients that have c
                    ommunicated with us.
47                  lblNumberOfConnections.Text = connectedClients.ToString();
48
49                  Thread clientThread = new Thread(new ParameterizedThreadStart(
                    HandleClientComm));
50                  clientThread.Start(client);
51              }
52          }
53
54          private void HandleClientComm(object client)
55          {
56              TcpClient tcpClient = (TcpClient)client;
57              NetworkStream clientStream = tcpClient.GetStream();
58
59              byte[] message = new byte[4096];
60              int bytesRead;
61
62              while (true)
63              {
64                  bytesRead = 0;
65
66                  try
67                  {
68                      //blocks until a client sends a message
    1   2   3   4
```

```
1 2 3 4
69                  bytesRead = clientStream.Read(message, 0, 4096);
70              }
71              catch
72              {
73                  //a socket error has occured
74                  break;
75              }
76
77              if (bytesRead == 0)
78              {
79                  //the client has disconnected from the server
80                  connectedClients--;
81                  lblNumberOfConnections.Text = connectedClients.ToString();
82                  break;
83              }
84
85              //message has successfully been received
86              ASCIIEncoding encoder = new ASCIIEncoding();
87
88              // Convert the Bytes received to a string and display it on the Se
                rver Screen
89              string msg = encoder.GetString(message, 0, bytesRead);
90              WriteMessage(msg);
91
92              // Now Echo the message back
93
94              Echo(msg, encoder, clientStream);
95          }
96
97          tcpClient.Close();
98      }
99
100     private void WriteMessage(string msg)
101     {
102         if (this.rtbServer.InvokeRequired)
103         {
104             WriteMessageDelegate d = new WriteMessageDelegate(WriteMessage);
105             this.rtbServer.Invoke(d, new object[] { msg });
106         }
107         else
108         {
109             this.rtbServer.AppendText(msg + Environment.NewLine);
110         }
111     }
112
113     /// <summary>
114     /// Echo the message back to the sending client
115     /// </summary>
116     /// <param name="msg">
117     /// String: The Message to send back
118     /// </param>
119     /// <param name="encoder">
120     /// Our ASCIIEncoder
121     /// </param>
122     /// <param name="clientStream">
123     /// The Client to communicate to
124     /// </param>
125     private void Echo(string msg, ASCIIEncoding encoder, NetworkStream
        clientStream)
126     {
127         // Now Echo the message back
128         byte[] buffer = encoder.GetBytes(msg);
129
130         clientStream.Write(buffer, 0, buffer.Length);
131         clientStream.Flush();
132     }
133 }
134 }
```

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Windows.Forms;
5
6   namespace ServerApp
7   {
8       static class Program
9       {
10          /// <summary>
11          /// The main entry point for the application.
12          /// </summary>
13          [STAThread]
14          static void Main()
15          {
16              Application.EnableVisualStyles();
17              Application.SetCompatibleTextRenderingDefault(false);
18              Application.Run(new Form1());
19          }
20      }
21  }
```