Real Time TCP/IP using C#

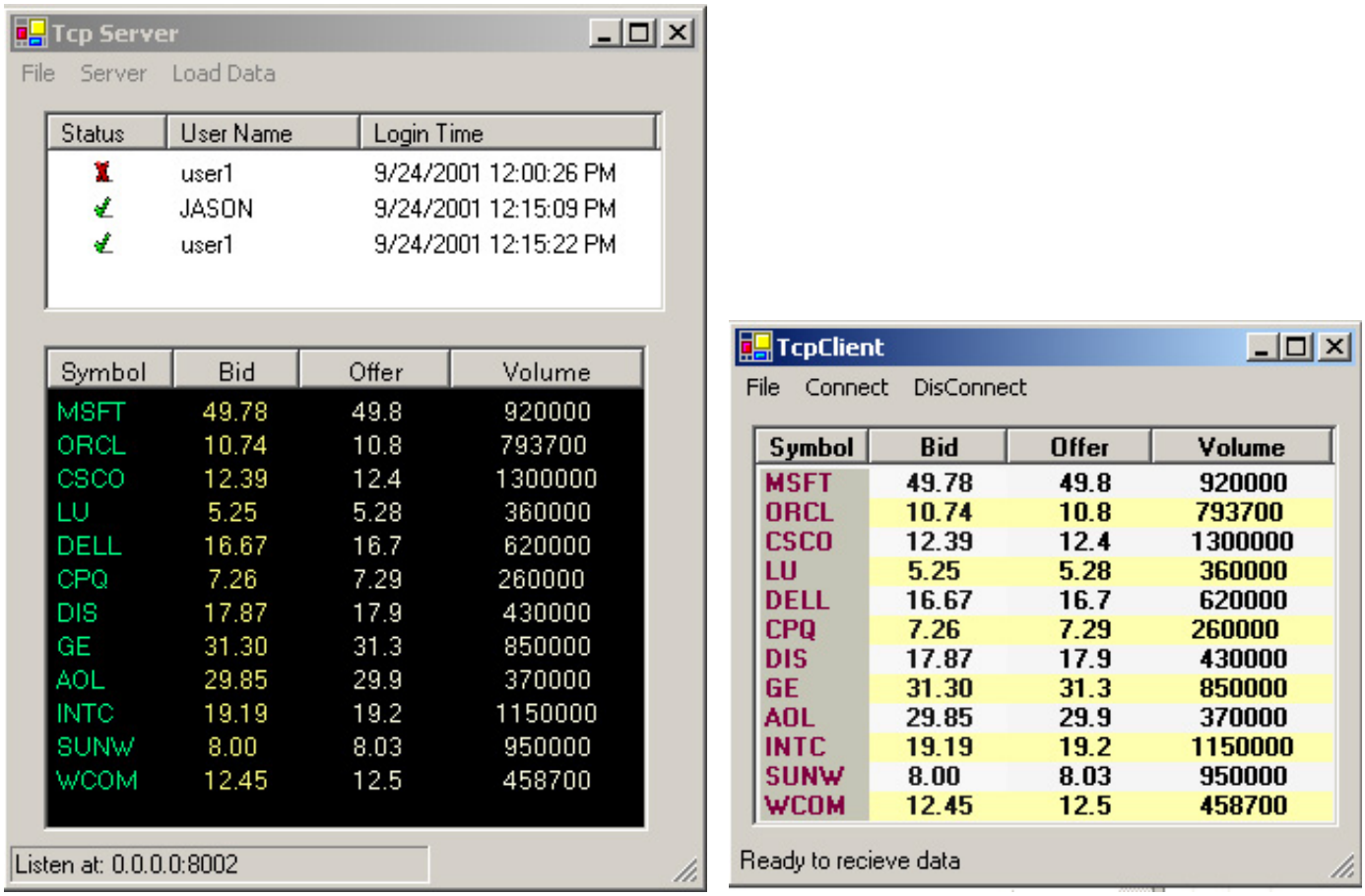
Jibin Pan, 12 Jan 2002

This sample shows the communication techniques between a client and a server application using a Socket class on each side.

Download server files - 44.7 Kb

Introduction

The Real time Application is a sample that shows the communication techniques between a client (TcpClient) and a server (TcpServer) application using Socket class on each side. The project also demonstrates how to using listview control in the real time project.



- TcpServer.exe showing the use of TCP socket communication in a separate thread. Multiple instances of TcpClient can talk to the same instance of TcpServer.
- TcpClient.exe also uses a separate thread to read data from Socket then update the listview control in a form.

The flow of logic

1. TcpServer listens on port 8002 and spawns a thread to waiting clients to connect.

Hide Copy Code

```
Hashtable socketHolder = new Hashtable();
Hashtable threadHolder = new Hashtable();

public Form1()
{
    // Required for Windows Form Designer support
    //
    InitializeComponent();

    tcpLsn = new TcpListener(8002);
    tcpLsn.Start();
    // tcpLsn.LocalEndpoint may have a bug, it only show 0.0.0.0:8002
    stpanel.Text = "Listen at: " + tcpLsn.LocalEndpoint.ToString();
    Thread tcpThd = new Thread(new ThreadStart(WaitingForClient));
    threadHolder.Add(connectId, tcpThd);
    tcpThd.Start() ;

    ...
}
```

2. TcpClient connect to TcpSrv and sends Client information data packet to TcpServer then spawns a thread, which waits to receive data through the Socket.

Hide Shrink ▲ Copy Code

```
private void menuConn_Click(object sender, System.EventArgs e)
{
    ConnectDlg myDlg = new ConnectDlg();
    myDlg.ShowDialog(this);
    if( myDlg.DialogResult==DialogResult.OK)
    {
        s = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
            ProtocolType.Tcp );

        IPAddress hostadd = IPAddress.Parse(myDlg.IpAdd);
        int port=Int32.Parse(myDlg.PortNum);
        IPEndPoint EPhost = new IPEndPoint(hostadd, port);

        Try
        {
            s.Connect(EPhost);

            if (s.Connected)
            {
                Byte[] bBuf;
                string buf;
                buf = String.Format("{0}:{1}", myDlg.UserName,
                    myDlg.PassWord);
                bBuf=ASCII.GetBytes(buf);
                s.Send(bBuf, 0 , bBuf.Length,0);
                t = new Thread(new ThreadStart(StartRecieve));
                t.Start();
                sbar.Text="Ready to recieve data";
            }
        }
        catch (Exception e1)
        {
            MessageBox.Show(e1.ToString());
        }
    }
}
private void StartRecieve()
{
    MethodInvoker miv = new MethodInvoker(this.UpdateListView);
    while (true)
    {
        Byte[] receive = new Byte[38] ;
        Try
        {
            string tmp=null;
            // Receive will block until data coming
            // ret is 0 or Exception happen when Socket connection is
            // broken
            int ret = s.Receive(receive, receive.Length, 0);
            if (ret>0)
            {
                tmp = System.Text.Encoding.ASCII.GetString(receive);
                if(tmp.Length > 0)
                {
                    isu.symbol= Mid(tmp, 0, 4);
                    isu.bid = Mid(tmp, 4, 5);
                    isu.offer = Mid(tmp, 9, 5);
                    isu.volume = Mid(tmp, 16, tmp.Length-16);

                    this.BeginInvoke(miv);
                    Thread.Sleep(300);
```

```
                    // block until finish the
                    // UpdateListview's job JobDone.WaitOne();
                }
            }
        }
        catch (Exception e)
        {
            if( !s.Connected )
            {
                break;
            }
        }
    }
    t.Abort();
}
```

3. TcpServer accepts the connection and saves the socket instance into a Hashtable instance then spawns a thread to handle the socket communication and show the client information in the top listview control.

Hide Shrink Copy Code

```
public void WaitingForClient()
{
      while(true)
      {
            // Accept will block until someone connects
            Socket sckt = tcpLsn.AcceptSocket();
            if (connectId < 10000)
                  Interlocked.Increment(ref connectId);
            Else
                  connectId = 1;
            if (socketHolder.Count < MaxConnected )
            {
                  while (socketHolder.Contains(connectId) )
                  {
                        Interlocked.Increment(ref connectId);
                  }
                  // it is used to keep connected Sockets
                  socketHolder.Add(connectId, sckt);
                  Thread td = new Thread(new ThreadStart(ReadSocket));
                  // it is used to keep the active thread
                  threadHolder.Add(connectId, td);
                  td.Start();
            }
      }
}
// follow function handle the communication from the clients and close the
// socket and the thread when the socket connection is down
public void ReadSocket()
{
      // the connectId is keeping changed with new connection added. it can't
      // be used to keep the real connectId, the local variable realId will
      // keep the value when the thread started.
      long realId = connectId;
      int ind=-1;
      Socket s = (Socket)socketHolder[realId];
      while (true)
      {
            if(s.Connected)
            {
                  Byte[] receive = new Byte[37] ;
                  Try
                  {
                        // Receive will block until data coming
```

```
                        // ret is 0 or Exception happen when Socket connection
                        // is broken
                        int ret=s.Receive(receive,receive.Length,0);
                        if (ret>0)
                        {
                              string tmp = null;
                            tmp=System.Text.Encoding.ASCII.GetString(receive);
                              if(tmp.Length > 0)
                              {
                                    DateTime now1=DateTime.Now;
                                    String strDate;
                                    strDate = now1.ToShortDateString() + " "
                                                + now1.ToLongTimeString();

                                    ListViewItem newItem = new ListViewItem();
                                    string[] strArry=tmp.Split(':');
                                    int code = checkUserInfo(strArry[0]);
                                    if(code==2)
                                    {
                                          userHolder.Add(realId, strArry[0]);
                                          newItem.SubItems.Add(strArry[0]);
                                          newItem.ImageIndex = 0;
                                          newItem.SubItems.Add(strDate);
                                          this.listView2.Items.Add(newItem);
                                    ind=this.listView2.Items.IndexOf(newItem);
                                    }
                                    else if( code==1)

                                          ..............
                              }
                        }
                        else
                        {
                              this.listView2.Items[ind].ImageIndex=1;
                              keepUser=false;
                              break;
                        }
                  }
                  catch (Exception e)
                  {
                        if( !s.Connected )
                        {
                              this.listView2.Items[ind].ImageIndex=1;
                              keepUser=false;
                              break;
                        }
                  }
            }
      }
      CloseTheThread(realId);
}
private void CloseTheThread(long realId)
{
      socketHolder.Remove(realId);
      if(!keepUser) userHolder.Remove(realId);
      Thread thd = (Thread)threadHolder[realId];
      threadHolder.Remove(realId);
      thd.Abort();
}
```

4. Click Load Data Menu to spawns a thread to load the information from a file then sends the information to all the clients that were connected to the TcpServer and update its own listview.

   In both TcpServer and TcpClient, they get the data from a working thread, and then update the Listview

control in the Main thread. Here use the MethodInvoker to work it out.

Hide Shrink Copy Code

```
public void LoadThread()
{
    MethodInvoker mi = new MethodInvoker(this.UpdateListView);
    string tmp = null;
    StreamReader sr = File.OpenText("Issue.txt");
    while((tmp = sr.ReadLine()) !=null )
    {
        if (tmp =="")
            break;
        SendDataToAllClient(tmp);

        isu.symbol= Mid(tmp, 0, 4);
        isu.bid = Mid(tmp, 4, 5);
        isu.offer = Mid(tmp, 9, 5);
        isu.volume = Mid(tmp, 16, tmp.Length-16);

        this.BeginInvoke(mi);
        Thread.Sleep(200);

        JobDone.WaitOne();
    }
    sr.Close();
    fThd.Abort();
}
private void SendDataToAllClient(string str)
{
    foreach (Socket s in socketHolder.Values)
    {
        if(s.Connected)
        {
            Byte[] byteDateLine=ASCII.GetBytes(str.ToCharArray());
            s.Send(byteDateLine, byteDateLine.Length, 0);
        }
    }
}
```

Following function demonstrate how to dynamically set BackColor and Forecolor properties of the Listview in TcpClient.

Hide Shrink Copy Code

```
private void UpdateListView()
{
    int ind=-1;
    for (int i=0; i<this.listView1.Items.Count;i++)
    {
        if (this.listView1.Items[i].Text == isu.symbol.ToString())
        {
            ind=i;
            break;
        }
    }
    if (ind == -1)
    {
        ListViewItem newItem new ListViewItem(isu.symbol.ToString());
        newItem.SubItems.Add(isu.bid);
        newItem.SubItems.Add(isu.offer);
        newItem.SubItems.Add(isu.volume);

        this.listView1.Items.Add(newItem);
```

```
        int i=this.listView1.Items.IndexOf(newItem);
        setRowColor(i, System.Drawing.Color.FromArgb(255, 255, 175));
        setColColorHL(i, 0, System.Drawing.Color.FromArgb(128,0,0));
        setColColorHL(i, 1, System.Drawing.Color.FromArgb(128,0,0));
        this.listView1.Update();
        Thread.Sleep(300);
        setColColor(i, 0, System.Drawing.Color.FromArgb(255, 255,175));
        setColColor(i, 1, System.Drawing.Color.FromArgb(255, 255, 175));
    }
    else
    {
        this.listView1.Items[ind].Text = isu.symbol.ToString();
        this.listView1.Items[ind].SubItems[1].Text = (isu.bid);
        this.listView1.Items[ind].SubItems[2].Text = (isu.offer);
        this.listView1.Items[ind].SubItems[3].Text = (isu.volume);
        setColColorHL(ind, 0, System.Drawing.Color.FromArgb(128,0,0));
        setColColorHL(ind, 1, System.Drawing.Color.FromArgb(128,0,0));
        this.listView1.Update();
        Thread.Sleep(300);
        setColColor(ind, 0, System.Drawing.Color.FromArgb(255,255,175));
        setColColor(ind, 1, System.Drawing.Color.FromArgb(255,255,175));
    }
    JobDone.Set();
}

private void setRowColor(int rowNum, Color colr )
{
    for (int i=0; i<this.listView1.Items[rowNum].SubItems.Count;i++)
        if (rowNum%2 !=0)
            this.listView1.Items[rowNum].SubItems[i].BackColor = colr;
}

private void setColColor(int rowNum, int colNum, Color colr )
{
    if (rowNum%2 !=0)
        this.listView1.Items[rowNum].SubItems[colNum].BackColor=colr;
    else
        this.listView1.Items[rowNum].SubItems[colNum].BackColor =
        System.Drawing.Color.FromArgb(248, 248,248);
    if (colNum==0)
    {
        this.listView1.Items[rowNum].SubItems[colNum].ForeColor =
            System.Drawing.Color.FromArgb(128, 0, 64);
        this.listView1.Items[rowNum].SubItems[colNum].BackColor =
            System.Drawing.Color.FromArgb(197, 197, 182);
    }
    else
        this.listView1.Items[rowNum].SubItems[colNum].ForeColor =
        System.Drawing.Color.FromArgb(20, 20,20);
}

private void setColColorHL(int rowNum, int colNum, Color colr )
{
    this.listView1.Items[rowNum].SubItems[colNum].BackColor = colr;
    this.listView1.Items[rowNum].SubItems[colNum].ForeColor =
        System.Drawing.Color.FromArgb(255,255,255);
}
```

Steps to run the sample:

1. Run TcpServer.exe on machine A.

2. Run TcpClient.exe once or more either on machine A or machine B.
3. On the TcpClient side, Click Menu connect; enter the server machine name where TcpServer is running. Enter user name and password in the edit box. Click Ok.
4. When you see the client in the TcpServer top listview, click Load Data Menu on the TcpServer, and then you will see the real time data in TcpServer and TcpClient.

   Note: Make sure that the Data file, Issue.txt, is in the same directory as TcpSvr.exe.

If you have any comments, I would love to hear about it. You can reach me at Jibin Pan.

Jibin Pan is VC++, C programmer at Interactive Edge Corp. Xtend Communications Corp. MoneyLine Corp in New York City since 1994 and has Master degree at computer science.

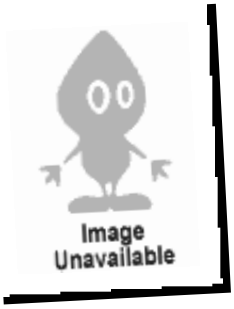
History

13 Jan 2002 - updated source.

License

Share

About the Author

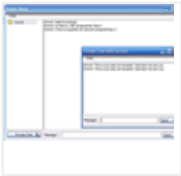


Jibin Pan

United States

No Biography provided

You may also be interested in...



TCP/IP Chat Application Using C#

A Perfect Launch Every Time: 7 Steps to Avoid the Worst Day Of Your Career

The Ultimate TCP/IP Home Page

Using Intel® RealSense™ Technology in combination with the Intel® Edison Development Platform

Generic TCP/IP Client/Server

Understanding the Need for a Single Code Base in Enterprise Mobile App Development

Comments and Discussions

You must Sign In to use this message board.

Search Comments Go

First Prev Next

Control 'listView2' access problem
abasadegh 31-Oct-08 6:51

Great
RRPKannan 17-Oct-07 18:39

Real Intersting dispute !
mekanoo 12-Sep-07 11:37

MaskedTextBox Method missing
Kasie 25-Mar-06 23:40

[Message Deleted]

Member 455866 12-Apr-09 21:40

Re: MaskedTextBox Method missing
rachana singh pawar 2-Jun-09 0:52

Re: MaskedTextBox Method missing
jaredbroad 15-Aug-12 10:42

tcp client
vbytesdc 2-Jun-05 19:55

thread
vbytesdc 2-Jun-05 13:07

Info needed,,, Client IP
Md Saleem Navalur 29-Mar-05 0:58

Updated Sample to get ...
Christian Uhlig 7-Apr-04 22:21

Re: Updated Sample to get ...
Chuck Duncan 7-Aug-06 9:43

VC++ 6.0 client
mduarte 17-Jan-04 18:52

Let us appreciate
fp (Not Far Pointer; Fact Pandit) 10-Dec-03 6:10

Re: Let us appreciate
Christian Uhlig 7-Apr-04 22:01

absolutelly NOT a real-time appl
Anonymous 13-Oct-03 6:14

Re: absolutelly NOT a real-time appl
3ddA 13-Oct-03 6:39

Thank you for this example
robert135 7-Oct-03 19:00

Re: Thank you for this example
Chris A.R. 21-Mar-04 17:22

Re: Thank you for this example
Christian Uhlig 7-Apr-04 22:01

I can't find the "MaskedTextBox"
brook 17-Aug-03 17:34

Re: I can't find the "MaskedTextBox"
cristiansje 22-Jan-04 4:14

Problems in ReadSocket() method
eyasso 19-May-03 3:52

hi，It is a bug
jhlcss 6-Mar-03 19:13

Re: hi，It is a bug
Anonymous 10-Mar-03 5:56

Refresh 1 2 Next »

General News Suggestion Question Bug Answer Joke Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.