Real Time TCP/IP using C#

By Jibin Pan.

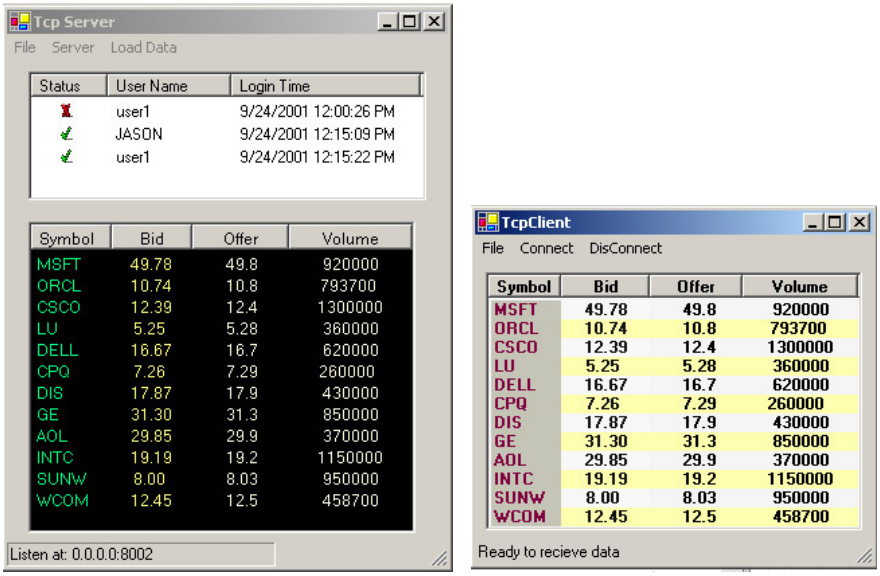
This sample shows the communication techniques between a client and a server application using a Socket class on each side.

C#
Windows, .NET (.NET 1.0)
Win32, VS (VS.NET2002)
Dev
Posted : 5 Oct 2001
Updated : 13 Jan 2002
Views : 144,860

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.

```
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.

Collapse

```
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.

Collapse

```
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.

Collapse

```
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.

Collapse

```
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.

Jibin Pan

Click here to view Jibin Pan's online profile.

Other popular Internet / Network articles:

- An Asynchronous Socket Server and Client
  An asynchronous socket server and client with encryption and compression.
- A Simple .NET TCP Socket Component
  Reusable C# code for client-server programming in .NET
- Asynchronous socket communication
  An article on using sockets to communicate in a non-blocking manner. The sample works through building a simple chat client and server.
- A POP3 Client in C# .NET
  A POP3 client in C# .NET for reading and processing emails (including attachments).

[Top]

Note: You must **Sign in** to post to this message board.

FAQ Message score threshold 3.0 Search comments Set Options

View Message View Per page 25

Msgs 1 to 25 of 35 (Total: 35) (Refresh) First Prev Next

Subject	Author	Date
MaskedTextBox Method missing	Kasie	5:40 26 Mar '06
tcp client	vbytesdc	1:55 3 Jun '05
thread	vbytesdc	19:07 2 Jun '05
Info needed,,, Client IP	Md Saleem Navalur	6:58 29 Mar '05

Updated Sample to get ...	**Christian Uhlig**	**4:21 8 Apr '04**
Re: Updated Sample to get ...	Chuck Duncan	15:43 7 Aug '06
VC++ 6.0 client	**mduarte**	**0:52 18 Jan '04**
Let us appreciate	**fp (Not Far Pointer; Fact Pandit)**	**12:10 10 Dec '03**
Re: Let us appreciate	Christian Uhlig	4:01 8 Apr '04
absolutelly NOT a real-time appl	**Anonymous**	**12:14 13 Oct '03**
Re: absolutelly NOT a real-time appl	3ddA	12:39 13 Oct '03
Thank you for this example	**robert135**	**1:00 8 Oct '03**
Re: Thank you for this example	Chris A.R.	23:22 21 Mar '04
Re: Thank you for this example	Christian Uhlig	4:01 8 Apr '04
I can't find the "MaskedTextBox"	**brook**	**23:34 17 Aug '03**
Re: I can't find the "MaskedTextBox"	cristiansje	10:14 22 Jan '04
Problems in ReadSocket() method	**eyasso**	**9:52 19 May '03**
hi□It is a bug	**jhlcss**	**1:13 7 Mar '03**
Re: hi□It is a bug	Anonymous	11:56 10 Mar '03
Hi, I have read your thread, and....a prb	**steve_cluj**	**1:57 21 Jun '02**
Re: Hi, I have read your thread, and....a prb	AK	19:56 23 Dec '02
Terrible!	**Ian Griffiths**	**16:32 14 Jan '02**
Re: Terrible!	Jibin Pan	9:02 22 Jan '02
Re: Terrible!	Ian Griffiths	11:26 22 Jan '02
Re: Terrible!	Jibin Pan	15:43 22 Jan '02

Last Visit: 11:22 Wednesday 6th June, 2007 First Prev Next

General comment News / Info Question Answer Joke / Game Admin message

Updated: 13 Jan 2002