public class asyncNetworkConnector

{

public Socket socket = null;

public IPAddress selfIPaddress = null;

public IPAddress targetIPaddress = null;

public IPEndPoint selfEndPoint = null;

public IPEndPoint targetEndPoint = null;

int selfPortNumber;

int targetPortNumber;

string localPCName;

public bool connected = false;

public bool readyToSend = false;

private ManualResetEvent connectDone = null;

private ManualResetEvent sendDone = null;

private ManualResetEvent receiveDone = null;

const int timeout = 5000;

public asyncNetworkConnector(IPAddress inIPAddress,int selfPortNum,int targetPortNum)

{

this.socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

this.targetIPaddress = inIPAddress;

this.selfPortNumber = selfPortNum;

this.targetPortNumber = targetPortNum;

this.finishSocketGroundWork();

}

public asyncNetworkConnector(string connectTo, int selfPortNum, int targetPortNum)

{

this.socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

IPHostEntry targetInfo = Dns.GetHostEntry(connectTo);

this.targetIPaddress = targetInfo.AddressList[0];

this.selfPortNumber = selfPortNum;

this.targetPortNumber = targetPortNum;

this.finishSocketGroundWork();

}

private void finishSocketGroundWork()

{

this.localPCName = System.Environment.MachineName;

IPHostEntry selfInfo = Dns.GetHostEntry(this.localPCName);

this.selfIPaddress = selfInfo.AddressList[0];

this.connectDone = new ManualResetEvent(false);

this.sendDone = new ManualResetEvent(false);

this.receiveDone = new ManualResetEvent(false);

this.selfEndPoint = new IPEndPoint(this.selfIPaddress, this.selfPortNumber);

this.targetEndPoint = new IPEndPoint(this.targetIPaddress, this.targetPortNumber);

this.socket.Bind(this.selfEndPoint);

}

private void connectCallBack(IAsyncResult ar)

{

try

{

asyncNetworkConnector connector = (asyncNetworkConnector)ar.AsyncState;

connector.socket.EndConnect(ar);

Console.WriteLine("Connection established");

}

catch(Exception e)

{

Console.WriteLine("Error in connectCallBack: " + e.ToString());

}

}

private void recieveCallBack(IAsyncResult ar)

{

}

private void sendCallBack(IAsyncResult ar)

{

try

{

asyncNetworkConnector connector = (asyncNetworkConnector)ar.AsyncState;

int sentBytes = connector.socket.EndSend(ar);

Console.WriteLine("Number of Bytes send = " + sentBytes.ToString());

}

catch(Exception e)

{

Console.WriteLine("Error in sendCallBack: " + e.ToString());

}

}

public void connect()

{

try

{

Console.WriteLine("Before begin connect");

this.socket.BeginConnect(this.targetEndPoint, new AsyncCallback(connectCallBack), this);

Console.WriteLine("After begin connect");

}

catch(Exception e)

{

Console.WriteLine("Error in beginConnect: " + e.ToString());

}

}

public void receive()

{

}

public void send(byte[] data)

{

try

{

this.socket.BeginSend(data, 0, data.Length, 0, new AsyncCallback(sendCallBack), this);

}

catch(Exception e)

{

Console.WriteLine("Error in send: " + e.ToString());

}

}

};