

Fixed Tcp Server

```
/*
C# Network Programming
by Richard Blum

Publisher: Sybex
ISBN: 0782141765
*/
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;

public class FixedTcpSrvr
{
    private static int SendData(Socket s, byte[] data)
    {
        int total = 0;
        int size = data.Length;
        int dataleft = size;
        int sent;

        while (total < size) { sent = s.Send(data, total, dataleft,
        SocketFlags.None); total += sent; dataleft -= sent; } return
        total; } private static byte[] ReceiveData(Socket s, int size)
        { int total = 0; int dataleft = size; byte[] data = new
        byte[size]; int recv; while(total < size) { recv =
        s.Receive(data, total, dataleft, 0); if (recv == 0) { data =
        Encoding.ASCII.GetBytes("exit "); break; } total += recv;
        dataleft -= recv; } return data; } public static void Main() {
        byte[] data = new byte[1024]; IPEndPoint ipep = new
        IPEndPoint(IPAddress.Any, 9050); Socket newsock = new
        Socket(AddressFamily.InterNetwork, SocketType.Stream,
        ProtocolType.Tcp); newsock.Bind(ipep); newsock.Listen(10);
        Console.WriteLine("Waiting for a client..."); Socket client =
```

```
newsock.Accept();      IPEndPoint      newclient      =
(IPPEndPoint)client.RemoteEndPoint;
Console.WriteLine("Connected with {0} at port {1}",
newclient.Address, newclient.Port); string welcome = "Welcome
to my test server"; data = Encoding.ASCII.GetBytes(welcome);
int sent = SendData(client, data); for (int i = 0; i < 5; i++)
{      data      =      ReceiveData(client,      9);
Console.WriteLine(Encoding.ASCII.GetString(data));      }
Console.WriteLine("Disconnected from {0}", newclient.Address);
client.Close(); newsock.Close(); } } [/csharp]
```