

# udpclient:C#使用UDP Client编写聊天程序

疯狂代码 <http://www.crazycoder.cn/> j:<http://www.crazycoder.cn/DotNet/Article49067.html>

UDPClient 类使用 UDP 和网络服务通讯UDP 优点是简单易用并且能够同时向多个地址广播消息但由于 UDP 协议是个无连接协议因此发送到远程终结点 UDP 数据文报不定能够到达也不定能够以发送相同顺序到达使用 UDP 应用必须准备处理丢失和顺序有误数据文报 若要使用 UDP 发送数据文报必须知道承载所需服务网络设备网络地址以及该服务用于通讯 UDP 端口号

特殊网络地址用于支持基于 IP 网络上 UDP 广播消息下面探讨内容以 Internet 上使用 IP 版本 4 地址族作为举例

IP 版本 4 地址使用 32 位指定网络地址对于使用 255.255.255.0 网络掩码 C 类地址这些位被分为 4 个 8 位字节当以十进制数表示时这 4 个 8 位字节构成熟悉以点分隔 4 部分表示法如 192.168.100.2 前两个 8 位字节(此举例中为 192.168)构成网络号;第 3 个 8 位字节 (100) 定义子网;最后个 8 位字节 (2) 是主机标识符

将 IP 地址所有位均设置为 1(即 255.255.255.255)可构成有限广播地址将 UDP 数据文报发送到此地址可将消息传递到该广播网络上任何主机由于路由器从不转发发送到此地址消息因此只有已连接网络上主机才可看到这些广播

通过将部分地址所有位全都设置为 1 可以将广播定向到特定网络部分例如若要将广播发送到以 192.168 打头 IP 地址标识网络上所有主机请将地址子网和主机部分全都设置为 1 如 192.168.255.255 若要将广播限制在单个子网则只将主机部分设置全都为 1 如 192.168.100.255

UdpClient 类可向任何网络广播地址广播但它无法侦听发送到网络广播必须使用 Socket 类才能侦听网络广播

当所有接收者都位于单个网络中时或者当许多客户端需要接收广播时广播地址将起作用当接收者为网络小部分时应将消息发送到多路广播组在那里只有加入此组客户端才能接收到消息范围从 224.0.0.2 到 244.255.255.255 IP 地址保留为主机组地址IP 号 224.0.0.0 被保留而 224.0.0.1 分配给所有 IP 主机固定组

下面举例使用 UdpClient 侦听端口8080 上多路广播地址组 10.0.0.1 UDP 数据文报广播它接收消息串并将消息写入控制台

例:在IDE新建个工程项目窗体类命名为FormChat在窗体上放个Control控件ListBox用于接收消息放个TextBoxControl控件用于输入要发送消息放个Button用于发送消息全源代码如下:

```
using;  
using.Drawing;
```

```

using.Collections;
using.ComponentModel;
using.Windows.Forms;
using.Data;
using.Net;
using.Net.Sockets;
using.Threading;
using.Text;
WinUDPChat
{
    ///<summary>
    ///Form1摘要介绍说明
    ///</summary>
    public FormChat : Windows.Forms.Form
    {
        ///<summary>
        ///必需设计器变量
        ///</summary>
        private.ComponentModel.Container components = null;
        private UdpClient m_Client;
        //private UdpClient m_Server;

        private ListenerPort = 8080;
        private SenderPort = 8080;
        private LocalPort;
        private RemotePort;

        private m_szHostName;

        private IPAddress m_GroupAddress_C;
        private IPAddress m_GroupAddress_S;
        private IPHostEntry m_LocalHost;
        private IPEndPoint m_RemoteEP;

        private.Windows.Forms.Button button_sendMSG;
        private.Windows.Forms.TextBox textBox_msg;
    }
}

```

```

private.Windows.Forms.ListBoxlistBox_msg;
privateThreadth;
publicFormChat
{
//
//Windows窗体设计器支持所必需
//
InitializeComponent;
//
//TODO:在InitializeComponent后添加任何构造代码
//
}
///<summary>
///清理所有正在使用资源
///</summary>
protectedoverridevoidDispose(booldisposing)
{
(disposing)
{
(components!=null)
{
components.Dispose;
}
}
base.Dispose(disposing);
}
#regionWindows窗体设计器生成代码
///<summary>
///设计器支持所需思路方法-不要使用代码编辑器修改
///此思路方法内容
///</summary>
privatevoidInitializeComponent
{
this.button_sendMSG=.Windows.Forms.Button;
this.textBox_msg=.Windows.Forms.TextBox;
this.listBox_msg=.Windows.Forms.ListBox;

```

```
this.SuspendLayout;
//
//button_sendMSG
//
this.button_sendMSG.Location=.Drawing.Po(96,234);
this.button_sendMSG.Name="button_sendMSG";
this.button_sendMSG.TabIndex=0;
this.button_sendMSG.Text="发送";
this.button_sendMSG.Click.EventHandler(this.button_sendMSG_Click);
//
//textBox_msg
//
this.textBox_msg.Location=.Drawing.Po(80,197);
this.textBox_msg.Name="textBox_msg";
this.textBox_msg.TabIndex=1;
this.textBox_msg.Text="";
//
//listBox_msg
//
this.listBox_msg.ItemHeight=12;
this.listBox_msg.Location=.Drawing.Po(21,22);
this.listBox_msg.Name="listBox_msg";
this.listBox_msg.Size=.Drawing.Size(248,136);
this.listBox_msg.TabIndex=2;
//
//FormChat
//
this.AutoScaleBaseSize=.Drawing.Size(6,14);
this.ClientSize=.Drawing.Size(292,273);
this.Controls.Add(this.listBox_msg);
this.Controls.Add(this.textBox_msg);
this.Controls.Add(this.button_sendMSG);
this.Name="FormChat";
this.Text="聊天室";
this.Load.EventHandler(this.FormChat_Load);
this.Closed.EventHandler(this.FormChat_Closed);
```

```

this.ResumeLayout(false);
}
#endregion
///<summary>
///应用主入口点
///</summary>
[STAThread]
void Main
{
    Application.Run(FormChat);
}
private void FormChat_Load(object sender, EventArgs e)
{
    LocalPort = SenderPort;
    RemotePort = ListenerPort;
    m_szHostName = Dns.GetHostName();

    m_LocalHost = Dns.GetHostByName(m_szHostName);

    Initialize;

    th = Thread(ThreadStart(Listener));
    th.Start;

}
public void Terminate
{
    m_Client.DropMulticastGroup(m_GroupAddress_C);
}
public void Initialize
{
    //
    //例子化UdpClient
    //
    m_Client = UdpClient(LocalPort);

```

```

//
//创建对方主机终结点
//
m_GroupAddress_S=IPAddress.Parse("10.0.0.1");//要发送到计算机IP
m_RemoteEP=IPEndPo(m_GroupAddress_S,RemotePort);

}
publicvoidListener
{
//
//创建多路广播组对象
//
.Net.IPHostEntrylocalhost=Dns.GetHostByName(Dns.GetHostName);
local_IP=localhost.AddressList[0].;//接收消息本地IP,用于监听
m_GroupAddress_C=IPAddress.Parse(local_IP);

//
//联接组
//
try
{
    m_Client.JoinMulticastGroup(m_GroupAddress_C,100);//不添加到多路广播组也可以
}
catch(Exceptionerr)
{
    throw(err);//无法联接多路广播组
}

//
//侦听器等待数据到来
//并用缓冲区保存它

Thread.Sleep(2000);//确保client2正在接收

```

```
Encoding ASCII=Encoding.ASCII;
```

```
// while(!m_Done)
while(true)
{
    IPEndPointendpo=null;
    //endpo=IPEndPoint(m_GroupAddress_C,LocalPort);//这句代码不要也可以
    Bytedata=m_Client.Receive(refendpo);

    StringstrData=ASCII.GetString(data);
    listBox_msg.Items.Add(strData);
}

}
privatevoidFormChat_Closed(objectsender,.EventArgs)
{
    try
    {
        try
        {
            (th!=null)
            {
                (th.IsAlive)
                {
                    th.Abort;

                }

                th=null;

            }

        }
        catch
        {
            try
```

```

    {
        .Threading.Thread.ReAbort;
    }
    catch
    {}
}

    Terminate;
    m_Client.Close;
}
catch
{
}
}
private void button_sendMSG_Click(object sender, EventArgs e)
{
    Bytebuffer = null;

    Encoding ASCII = Encoding.ASCII;

    s = textBox_msg.Text;

    buffer = Byte[s.Length + 1];
    //
    // 将数据发送给远程对方主机
    //

    len = ASCII.GetBytes(s.ToCharArray, 0, s.Length, buffer, 0);

    ecode = m_Client.Send(buffer, len, m_RemoteEP);

}
}
}

```



2009-1-9 13:23:08

疯狂代码 <http://www.crazycoder.cn/>