# 实 验 一 获得主机 IP 和网卡信息

## 1.1 实验目的

获取主机网络信息是网络程序的准备工作，.NET 平台提供的网络通信相关类可以方便获取主机 IP 网卡等信息，IP 地址是 TCP 或 UDP 程序的核心数据，MAC 地址不仅用于低层协议，还由于其固定性特殊性还常作为主机的标识信息。

## 1.2 实验内容

软件项目采用 VS2008 工具的 C# 语言。

### 1.2.1 IP 地址转换类

System.Net 命名空间提供网络编程多个类与接口，其中 IPAddress 类提供 IP 地址转换、处理等功能，IPEndPoint 类包括主机和端口信息，IPHostEntry 与 Dns 类用于域名解析可以对本地或远程主机的域名和 IP 地址进行转换。示例程序演示 IPAddress 类、Dns 类、IPHostEntry 类和 IPEndPoint 类的使用方法。参考界面如图1-1。


图 1-1 程序运行界面一

获取远程主机信息示例程序代码如下：

```
private void buttonRemoteIP_Click(object sender, EventArgs e)
```

```
{

    this.listBoxRemoteInfo.Items.Clear();
    IPHostEntry remoteHost = Dns.GetHostEntry(this.textBoxRmoteIP.Text);
    IPAddress[] remoteIP = remoteHost.AddressList;
    IPEndPoint iep;
    foreach (IPAddress ip in remoteIP)
    {
        iep = new IPEndPoint(ip, 80);
        listBoxRemoteInfo.Items.Add(iep);
    }
}
```

获取本机 IP 信息的示例代码如下：

```
private void buttonLocalIP_Click(object sender, EventArgs e)
{
    listBoxLocalInfo.Items.Clear();
    string name = Dns.GetHostName();
    listBoxLocalInfo.Items.Add(" 本机主机名:" + name);
    IPHostEntry me = Dns.GetHostEntry(name);
    listBoxLocalInfo.Items.Add(" 本机所有 IP 地址：");
    foreach (IPAddress ip in me.AddressList)
    {
        listBoxLocalInfo.Items.Add(ip);
    }
    IPAddress localip = IPAddress.Parse("127.0.0.1");
    IPEndPoint iep = new IPEndPoint(localip, 80);
    listBoxLocalInfo.Items.Add("IP 端点: " + iep.ToString());
    listBoxLocalInfo.Items.Add("IP 端口: " + iep.Port);
    listBoxLocalInfo.Items.Add("IP 地址: " + iep.Address);
    listBoxLocalInfo.Items.Add("IP 地址族: " + iep.AddressFamily);
    listBoxLocalInfo.Items.Add(" 可分配端口最大值: " + IPEndPoint.MaxPort);
    listBoxLocalInfo.Items.Add(" 可分配端口最小值: " + IPEndPoint.MinPort);
}
```

### 1.2.2 网卡信息

网卡是计算机用来连接网络的硬件设备，下面代码获取本机网络适配器个数、描述信息、名称、类型、速度、MAC 地址以及 DNS 服务器信息。程序运行效果如图1-2。


图 1-2 程序运行界面二

示例代码如下：

```
private void ShowAdapterInfo()
{
    NetworkInterface[] adapters = NetworkInterface.GetAllNetworkInterfaces();
    listBoxAdpterInfo.Items.Add(" 适配器个数：" + adapters.Length);
    int index = 0;
    foreach (NetworkInterface adapter in adapters)
    {
        index++;
        //显示网络适配器描述信息、名称、型号、速度、MAC 地址
        listBoxAdpterInfo.Items.Add("————————--第" + index + " 个适配器信息 ————————-");
        listBoxAdpterInfo.Items.Add(" 描述信息：{0}" + adapter.Description);
        listBoxAdpterInfo.Items.Add(" 名称：{0}" + adapter.Name);
        listBoxAdpterInfo.Items.Add(" 类型：{0}" + adapter.NetworkInterfaceType);
        listBoxAdpterInfo.Items.Add(" 速度：{0}" + adapter.Speed);
        listBoxAdpterInfo.Items.Add("MAC 地址：{0}" + adapter.GetPhysicalAddress());
        //获取 IPInterfaceProperties 实例
        IPInterfaceProperties adapterProperties = adapter.GetIPProperties();
        //获取并显示 DNS 服务器 IP 地址信息
```

```
        IPAddressCollection dnsServers = adapterProperties.DnsAddresses;
        if (dnsServers.Count > 0)
        {
          foreach (IPAddress dns in dnsServers)
          {
            listBoxAdpterInfo.Items.Add("DNS 服务器 IP 地址：{0} " + dns + "\n");
          }
        }
        else
        {
          listBoxAdpterInfo.Items.Add("DNS 服务器 IP 地址：{0} " + "\n");
        }
      }
    }
```

### 1.2.3 Ping 类使用

在 System.Net.NetworkInformation 命名空间下提供了 Ping 类、PingOptions 类和 PingReply 类，可用来实现类似 Ping 的功能。由于 WindowsXp 操作系统默认情况下关闭了 Ping 功能，打开方式步骤为控制面版 ->windows 防火墙 -> 高级 -> 设置 ->ICMP，选中允许传入的回显请求，即可用来测试 Ping 程序了，程序运行界面如图1-3。

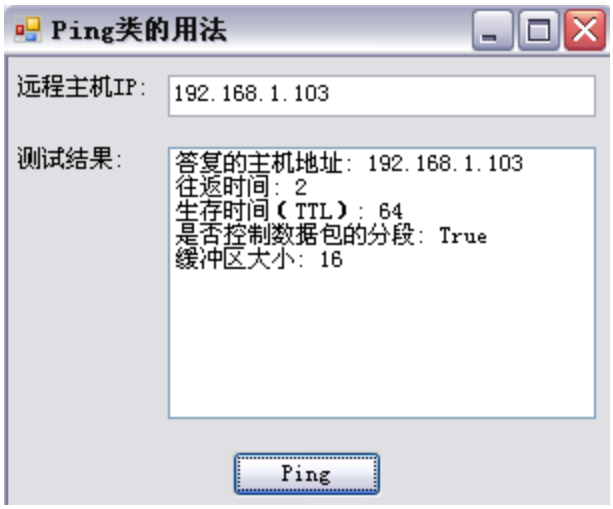


图 1-3 程序运行界面三

主要代码如下：

```
  private void buttonPing_Click(object sender, EventArgs e)
  {
    this.listBoxResult.Items.Clear();
    //远程服务器 IP
    string ipString = this.textBoxRemoteIp.Text.ToString().Trim();
```

```
    //构造 Ping 实例
    Ping pingSender = new Ping();
    //Ping 选项设置
    PingOptions options = new PingOptions();
    options.DontFragment = true;
    //测试数据
    string data = "test data abcabc";
    byte[] buffer = Encoding.ASCII.GetBytes(data);
    //设置超时时间
    int timeout = 120;
    //调用同步 send 方法发送数据，将返回结果保存至 PingReply 实例
    PingReply reply = pingSender.Send(ipString, timeout, buffer, options);
    if (reply.Status == IPStatus.Success)
    {
      listBoxResult.Items.Add(" 答复的主机地址: " + reply.Address.ToString());
      listBoxResult.Items.Add(" 往返时间: " + reply.RoundtripTime);
      listBoxResult.Items.Add(" 生存时间（TTL）: " + reply.Options.Ttl);
      listBoxResult.Items.Add(" 是否控制数据包的分段: " + reply.Options.DontFragment);
      listBoxResult.Items.Add(" 缓冲区大小: " + reply.Buffer.Length);
    }
    else
    {
      listBoxResult.Items.Add(reply.Status.ToString());
    }
  }
```

## 1.3 实验作业

完成本实验代码任务。