



第29篇 (网络) 获取本机网络信息

代码地址：<https://github.com/Lornatang/QtStartQuicklyTutorial/tree/main/Network02>

一、使用 `QHostInfo` 获取主机名和IP地址

二、通过 `QNetworkInterface` 类来获取本机的IP地址和网络接口信息

一、使用 `QHostInfo` 获取主机名和IP地址

我们新建Qt Gui应用，项目名为 `myIP`，基类选择 `QWidget`，类名保持 `Widget` 不变。完成后先打开 `myIP.pro` 文件，添加一行代码：`QT += network`，然后保存该文件。下面打开 `widget.h` 文件添加头文件包含：`#include <QtNetwork>`

(1) 获取主机名。

我们在 `widget.cpp` 文件中的构造函数中添加代码：

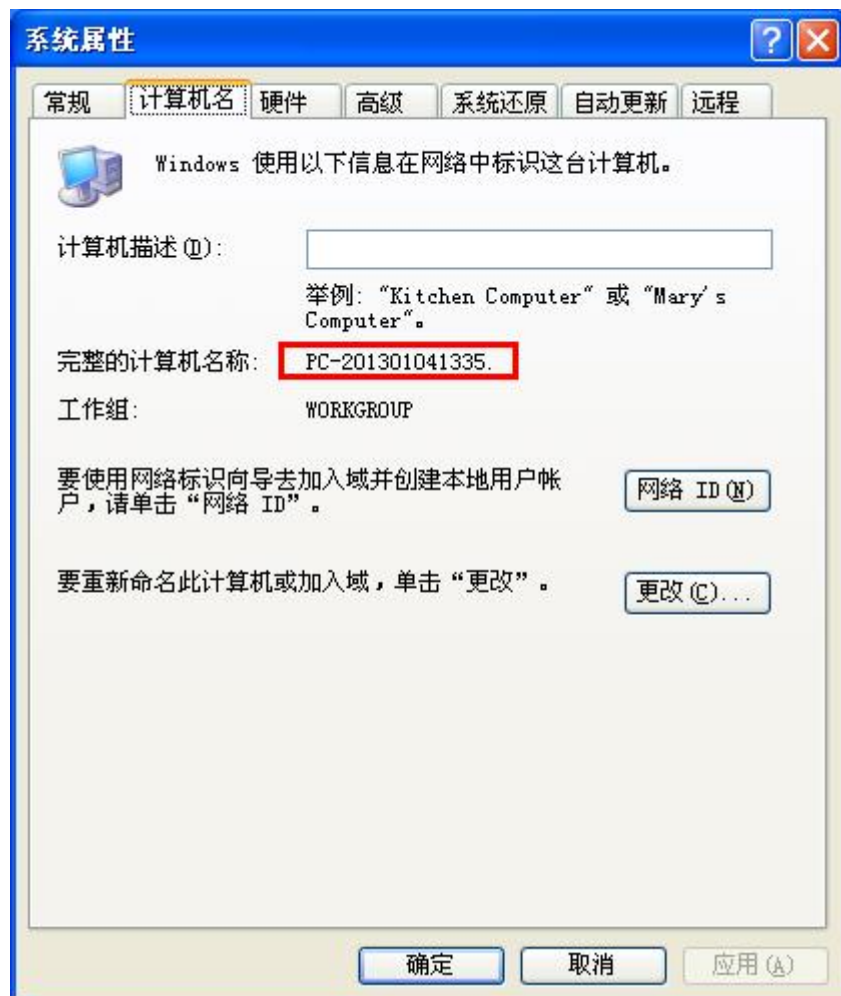
```
QString localHostName = QHostInfo::localHostName();  
QDebug() <<"localHostName: "<<localHostName;
```

这里我们使用了 `QHostInfo` 类的 `localHostName` 类来获取本机的计算机名称。

运行程序，在下面的应用程序输出栏里的信息如下：

```
应用程序输出
myIP
E:\build-myIP-Desktop_Qt_4_8_5-Debug\debug\
localhostName: "PC-201301041335"
```

可以看到，这里获取了计算机名。我们可以在桌面上“我的电脑”图标上点击鼠标右键，然后选择“属性”菜单，查看“计算机名”一项，和这里输出结果是一样的，如下图。



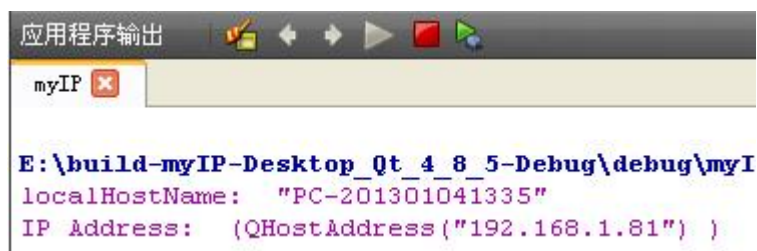
(2) 获取本机的IP地址。

我们继续在构造函数中添加代码：

```
QHostInfo info = QHostInfo::fromName(localhostName);
QDebug() <<"IP Address: "<<info.addresses();
```

调用 `QHostInfo` 类的 `fromName()` 函数，使用上面获得的主机名为参数，来获取本机的信息。然后再利用 `QHostInfo` 类的 `addresses()` 函数，获取本机的所有IP地址信息。运行程

序，输出信息如下：



```
应用程序输出
myIP
E:\build-myIP-Desktop_Qt_4_8_5-Debug\debug\myI
localHostName: "PC-201301041335"
IP Address: (QHostAddress("192.168.1.81") )
```


在我这里只有一条IP地址。但是，在其他系统上，可能出现多条IP地址，其中可能包含了IPv4和IPv6的地址，一般我们需要使用IPv4的地址，所以我们可以只输出IPv4的地址。

我们继续添加代码：

```
foreach(QHostAddress address,info.addresses())
{
    if(address.protocol() == QAbstractSocket::IPv4Protocol)
        qDebug() << address.toString();
}
```

因为IP地址由 `QHostAddress` 类来管理，所以我们可以使用该类来获取一条IP地址，然后使用该类的 `protocol()` 函数来判断其是否为IPv4地址。如果是IPv6地址，可以使用 `QAbstractSocket::IPv6Protocol` 来判断。最后我们将IP地址以 `QString` 类型输出。

我们以后要使用的IP地址都是用这个方法获得的，所以这个一定要掌握。运行效果如下：



```
应用程序输出
myIP
E:\build-myIP-Desktop_Qt_4_8_5-Debug\debug\myI
localHostName: "PC-201301041335"
IP Address: (QHostAddress("192.168.1.81") )
"192.168.1.81"
```

(3) 以主机名获取IP地址。

在上面讲述了用本机的计算机名获取本机的IP地址。其实 `QHostInfo` 类也可以用来获取任意主机名的IP地址，如一个网站的IP地址。在这里我们可以使用 `lookupHost()` 函数。它是基于信号和槽的，一旦查找到了IP地址，就会触发槽函数。具体用法如下。

我们在 `widget.h` 文件中添加一个私有槽函数：

```
private slots:
void lookedUp(const QHostInfo &host);
```

然后在 `widget.cpp` 中的构造函数中先将上面添加的代码全部注释（可以通过选中所有代码，然后按下 `Ctrl+/` 快捷键来注释代码），然后添加以下代码：

```
QHostInfo::lookupHost("www.qter.org",
                      this,SLOT(lookedUp(QHostInfo)));
```

这里我们查询Qter开源社区的IP地址，如果查找到，就会执行我们的 `lookedUp()` 函数。

在 `widget.cpp` 中添加 `lookedUp()` 函数的实现代码：

```
void Widget::lookedUp(const QHostInfo &host)
{
    qDebug() << host.addresses().first().toString();
}
```

这里我们只是简单地输出第一个IP地址。输出信息如下：



其实，我们也可以使用 `lookupHost()` 函数，通过输入IP地址反向查找主机名，只需要将上面代码中的 `www.qter.org` 换成一个IP地址就可以了，如果你有兴趣可以研究一下，不过返回的结果可能不是你想象中的那样。

可以看到 `QHostInfo` 类的作用：通过主机名来查找IP地址，或者通过IP地址来反向查找主机名。

二、通过 `QNetworkInterface` 类来获取本机的IP地址和网络接口信息

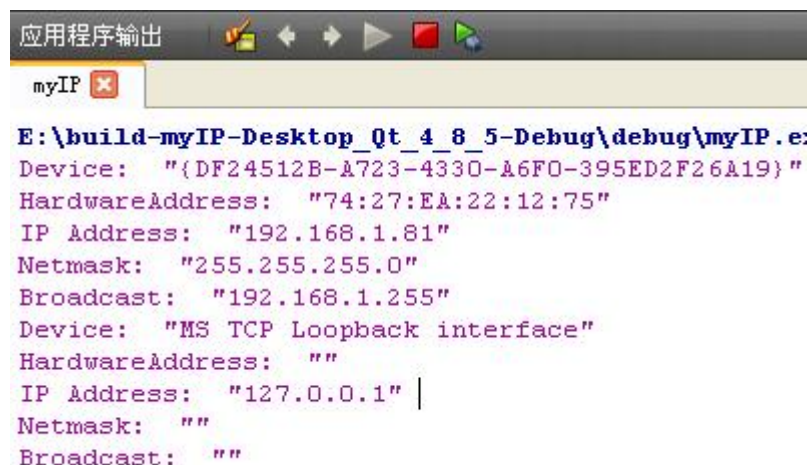
`QNetworkInterface` 类提供了程序所运行时的主机的IP地址和网络接口信息的列表。在每一个网络接口信息中都包含了0个或多个IP地址，而每一个IP地址又包含了和它相关的

子网掩码和广播地址，它们三者被封装在一个 `QNetworkAddressEntry` 对象中。网络接口信息中也提供了硬件地址信息。我们将 `widge.cpp` 构造函数中以前添加的代码注释掉，然后添加以下代码。

```
//获取所有网络接口的列表
QList<QNetworkInterface> list = QNetworkInterface::allInterfaces();
foreach(QNetworkInterface interface, list) //遍历每一个网络接口
{
    qDebug() << "Device: " << interface.name(); //设备名
    //硬件地址
    qDebug() << "HardwareAddress: " << interface.hardwareAddress();

    //获取IP地址条目列表，每个条目中包含一个IP地址，
    //一个子网掩码和一个广播地址
    QList<QNetworkAddressEntry> entryList= interface.addressEntries();
    foreach(QNetworkAddressEntry entry, entryList) //遍历每个IP地址条目
    {
        qDebug() << "IP Address: " << entry.ip().toString(); //IP地址
        qDebug() << "Netmask: " << entry.netmask().toString(); //子网掩码
        qDebug() << "Broadcast: " << entry.broadcast().toString(); //广播地址
    }
}
```

这里我们获取了本机的网络设备的相关信息。运行程序，输出如下：



```
E:\build-myIP-Desktop Qt_4_8_5-Debug\debug\myIP.e:
Device:  "{DF24512B-A723-4330-A6F0-395ED2F26A19}"
HardwareAddress:  "74:27:EA:22:12:75"
IP Address:  "192.168.1.81"
Netmask:  "255.255.255.0"
Broadcast:  "192.168.1.255"
Device:  "MS TCP Loopback interface"
HardwareAddress:  ""
IP Address:  "127.0.0.1" |
Netmask:  ""
Broadcast:  ""
```

其实，如果我们只想利用 `QNetworkInterface` 类来获取IP地址，那么就没必要像上面那样复杂，这个类提供了一个便捷的函数 `allAddresses()` 来获取IP地址，例如：

```
QString address = QNetworkInterface::allAddresses().first().toString();
```