Working with QSystemNetworkInfo - System Information API - Part 2

ID		**Creation date**	15-March-2010
Platform	S60 3rd Edition, FP1, FP2 S60 5th Edition	**Tested on devices**	Nokia 5800 xPressMusic
Category	Qt	**Subcategory**	Qt Mobility API

Keywords (APIs, classes, methods, functions): QSystemNetworkInfo, QString, QButton, QLabel, QComboBox

Tip:
- Please read this article before moving forward: Archived:Setting up environment for Qt Mobility API
- Setup Carbide.c++ IDE as explained in this article: Working with Carbide.c++ IDE for Qt Mobility APIs

Overview

This is a part 2 of the `QSystemNetworkInfo` API which is a part of the ***System Information API*** from ***Qt Mobility project***. The part 1 of this article can be read from Here

What is covered in `QSystemNetworkInfo` API Part 2

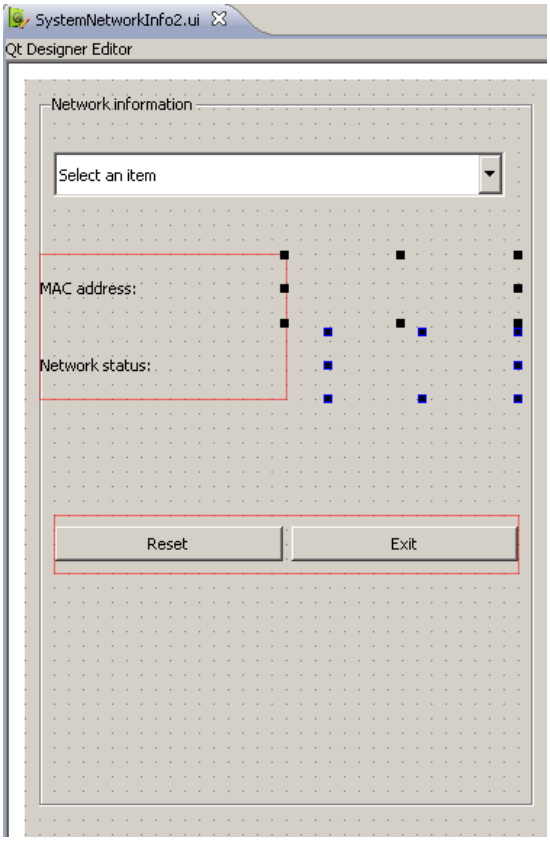
- It is a part of the System Information API.
- This article covers information about
- MAC address of the specified network mode
- Network status of the specified network mode

Capabilities required

- **ReadDeviceData**

UI design (.ui file)

(C) Copyright Nokia 2013. All rights reserved.



Project configuration file (.Pro file)

- Add the Qt Mobility project configuration option in the **.pro** file as shown below

```
CONFIG += mobility
MOBILITY += systeminfo
```

- Add required capability in **.pro** file as shown below:

```
symbian:TARGET.CAPABILITY = ReadDeviceData
```

Implementing QSystemNetworkInfo API

- `QSystemNetworkInfo` API is a part of the `QtMobility` namespace. So declare the `QtMobility` namespace as shown below:

```
using namespace QtMobility;
```

Header file

```
#ifndef SYSTEMNETWORKINFO2_H
#define SYSTEMNETWORKINFO2_H

#include <QtGui/QWidget>
#include "ui_SystemNetworkInfo2.h"
#include <qsysteminfo.h>

using namespace QtMobility;
class SystemNetworkInfo2 : public QWidget
{
    Q_OBJECT

public:
 SystemNetworkInfo2(QWidget *parent = 0);
    ~SystemNetworkInfo2();

public slots:
 void displayNetworkInfo(int aIndex);
 void resetAllFields();

private:
 QString getNetworkStatusString(QSystemNetworkInfo::NetworkStatus aIndex);

private:
```

 (C) Copyright Nokia 2013. All rights reserved.

```
    Ui::SystemNetworkInfo2 ui;
    QSystemNetworkInfo* networkInfo;
};

#endif // SYSTEMNETWORKINFO2_H
```

Source file

```
#include "SystemNetworkInfo2.h"

SystemNetworkInfo2::SystemNetworkInfo2(QWidget *parent)
    : QWidget(parent), networkInfo(NULL)
{
 ui.setupUi(this);
 networkInfo = new QSystemNetworkInfo();
 QObject::connect(ui.comboBox,
   SIGNAL(currentIndexChanged(int)),
   this,
   SLOT(displayNetworkInfo(int)));
 QObject::connect(ui.resetButton,
   SIGNAL(clicked()),
   this,
   SLOT(resetAllFields()));
}

SystemNetworkInfo2::~SystemNetworkInfo2()
{
 delete networkInfo;
 networkInfo = NULL;
}

void SystemNetworkInfo2::displayNetworkInfo(int aIndex)
{
 if(! aIndex)
 {
  resetAllFields();
  return;
 }

 QString macAddress = networkInfo->macAddress((QSystemNetworkInfo::NetworkMode)aIndex);
 if(! macAddress.length())
 {
  ui.macLabel->setText("Not applicable");
 }
 else
 {
  ui.macLabel->setText(macAddress);
 }
 ui.statusLabel->setText(getNetworkStatusString(
     networkInfo->networkStatus(
      (QSystemNetworkInfo::NetworkMode)aIndex)));
}

QString SystemNetworkInfo2::getNetworkStatusString(QSystemNetworkInfo::NetworkStatus aIndex)
{
 switch(aIndex)
 {
  case QSystemNetworkInfo::UndefinedStatus:
  {
   return "Undefined";
  }
  case QSystemNetworkInfo::NoNetworkAvailable:
  {
   return "Unavailable";
  }
  case QSystemNetworkInfo::EmergencyOnly:
  {
   return "Emergency";
  }
  case QSystemNetworkInfo::Searching:
  {
   return "Searching";
  }
  case QSystemNetworkInfo::Busy:
  {
   return "Busy";
  }
  case QSystemNetworkInfo::Connected:
  {
   return "Connected";
  }
  case QSystemNetworkInfo::HomeNetwork:
  {
   return "On home n/w";
  }
  case QSystemNetworkInfo::Denied:
  {
   return "Access denied";
  }
  case QSystemNetworkInfo::Roaming:
  {
   return "On Roaming";
  }
  default:
  {
   return "Error";
  }
 }
}

void SystemNetworkInfo2::resetAllFields()
{
 ui.macLabel->clear();
 ui.statusLabel->clear();
 ui.comboBox->setCurrentIndex(0);
}
```

 (C) Copyright Nokia 2013. All rights reserved.

Output

Useful functions

- `QString macAddress(QSystemNetworkInfo::NetworkMode mode);`
- `QSystemNetworkInfo::NetworkStatus networkStatus(QSystemNetworkInfo::NetworkMode mode);`

Keywords

Headers

```
#include <qsysteminfo.h>
```

Classes

- `QSystemNetworkInfo`
- `QString`
- `QButton`
- `QLabel`
- `QComboBox`

Example Application

- A working example application is available to download from here: QSystemNetworkInfo part2.zip

 (C) Copyright Nokia 2013. All rights reserved.

Related articles

- Getting started with Qt Mobility APIs
- Archived:Setting up environment for Qt Mobility API
- Working with Carbide.c++ IDE for Qt Mobility APIs
- Working with QSystemInfo - System Information API - Part 1
- Working with QSystemInfo - System Information API - Part 2
- Working with QSystemDeviceInfo - System Information API - Part 1
- Working with QSystemStorageInfo - System Information API
- Working with QSystemDisplayInfo - System Information API
- Working with QSystemNetworkInfo - System Information API - Part 1

Reference links

- Qt - cross-platform application and UI framework
- Qt Mobility API
- New Qt APIs Beta - Mobility Project
- SDK help

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.