Working with QSystemDeviceInfo - System Information API - Part 1

ID		**Creation date**	8-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): QSystemDeviceInfo, QString, QStringList, QButton, QLabel

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 1 of the ***QSystemDeviceInfo API*** which is a part of the ***System Information API*** from ***Qt Mobility project***.

What is covered in QSystemDeviceInfo API Part 1

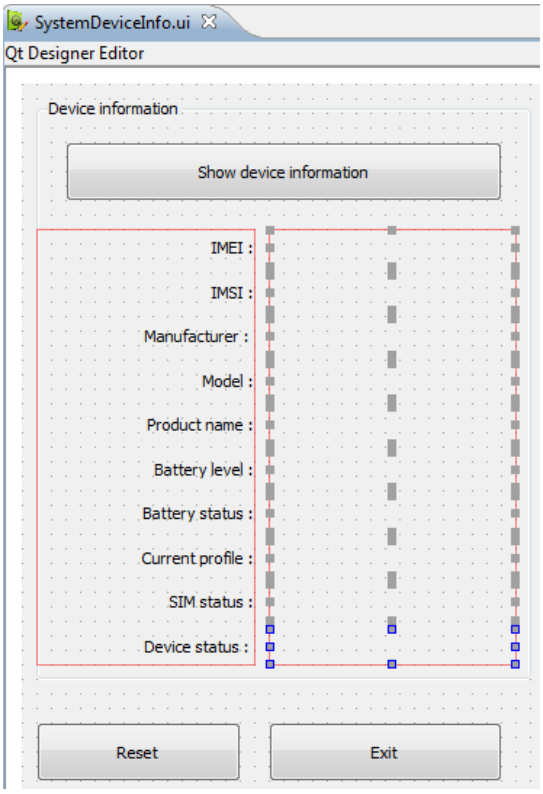
- It is a part of the System Information API.
- This article covers information about
  - IMEI of the device
  - IMSI information
  - Manufacturer information
  - Model information
  - Product information
  - Battery level
  - Battery status
  - Current profile
  - SIM card status
  - Device status

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 QSystemDeviceInfo API

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

```
using namespace QtMobility;
```

Header file

```
#ifndef SYSTEMDEVICEINFO_H
#define SYSTEMDEVICEINFO_H

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

using namespace QtMobility;
class SystemDeviceInfo : public QWidget
{
    Q_OBJECT

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

public slots:
    void listDeviceInformation();
    void resetAllFields();
private:
    Ui::SystemDeviceInfo ui;
    QSystemDeviceInfo* deviceInfo;
};

#endif // SYSTEMDEVICEINFO_H
```

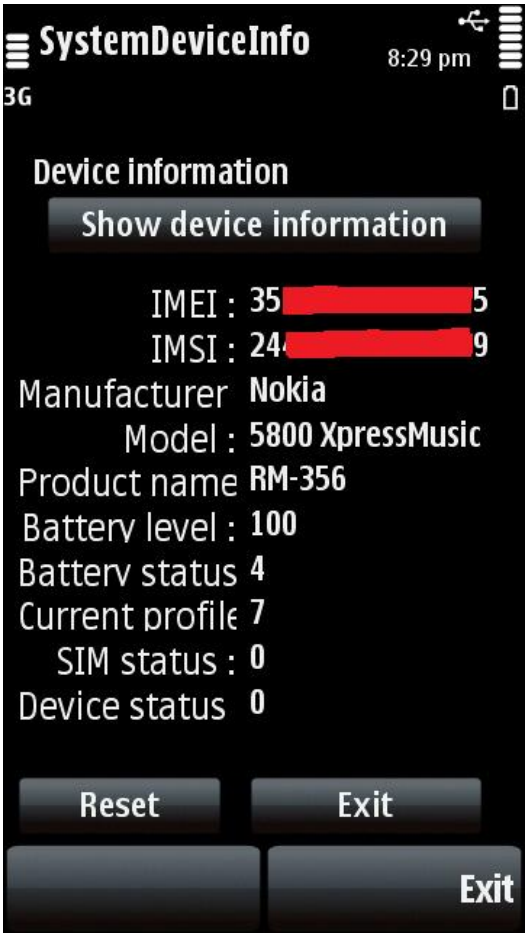

(C) Copyright Nokia 2013. All rights reserved.

Source file

```
#include "SystemDeviceInfo.h"

SystemDeviceInfo::SystemDeviceInfo(QWidget *parent)
    : QWidget(parent), deviceInfo(NULL)
{
 ui.setupUi(this);
 deviceInfo = new QSystemDeviceInfo;
 QObject::connect(ui.listDeviceInfoButton,
     SIGNAL(clicked()),
     this,
     SLOT(listDeviceInformation()));
 QObject::connect(ui.resetButton,
     SIGNAL(clicked()),
     this,
     SLOT(resetAllFields()));
}

SystemDeviceInfo::~SystemDeviceInfo()
{
 delete deviceInfo;
}

void SystemDeviceInfo::listDeviceInformation()
{
 if(deviceInfo)
 {
  ui.imeiLabel->setText(deviceInfo->imei());
  ui.imsiLabel->setText(deviceInfo->imsi());
  ui.manufacturerLabel->setText(deviceInfo->manufacturer());
  ui.modelLabel->setText(deviceInfo->model());
  ui.productLabel->setText(deviceInfo->productName());
  ui.batteryLabel->setText(QString::number(deviceInfo->batteryLevel()));
  ui.batteryStatusLabel->setText(QString::number(
    (QSystemDeviceInfo::BatteryStatus)deviceInfo->batteryStatus()));
  ui.profileLabel->setText(QString::number(
    (QSystemDeviceInfo::Profile)deviceInfo->currentProfile()));
  ui.simLabel->setText(QString::number(
    (QSystemDeviceInfo::SimStatus)deviceInfo->simStatus()));
  ui.deviceLabel->setText(QString::number((int)deviceInfo->isDeviceLocked()));
 }
}

void SystemDeviceInfo::resetAllFields()
{
 if(deviceInfo)
 {
  ui.imeiLabel->clear();
  ui.imsiLabel->clear();
  ui.manufacturerLabel->clear();
  ui.modelLabel->clear();
  ui.productLabel->clear();
  ui.batteryLabel->clear();
  ui.batteryStatusLabel->clear();
  ui.profileLabel->clear();
  ui.simLabel->clear();
  ui.deviceLabel->clear();
 }
}
```

Output

(C) Copyright Nokia 2013. All rights reserved.



Useful functions

- QString imei ();
- QString imsi ();
- QString manufacturer ();
- QString model ();
- QString productName ();
- int batteryLevel () const;
- QSystemDeviceInfo::BatteryStatus batteryStatus ();
- QSystemDeviceInfo::Profile currentProfile ();
- bool isDeviceLocked ();
- QSystemDeviceInfo::SimStatus simStatus ();

Keywords

Headers

- #include <qsysteminfo.h>

Classes

- QSystemDeviceInfo
- QString
- QStringList
- QButton
- QLabel

Example Application

- A working example application is available to download from here: QSystemDeviceInfo part1.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 QSystemStorageInfo - System Information API
- Working with QSystemDisplayInfo - System Information API
- Working with QSystemNetworkInfo - System Information API - Part 1
- Working with QSystemNetworkInfo - System Information API - Part 2

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.