Working with QSystemInfo - System Information API - Part 2

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): QSystemInfo, QString, QStringList, 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 ***QSystemInfo API*** which is a part of the ***System Information API*** from ***Qt Mobility project***.

What is covered in QSystemInfo API Part 2

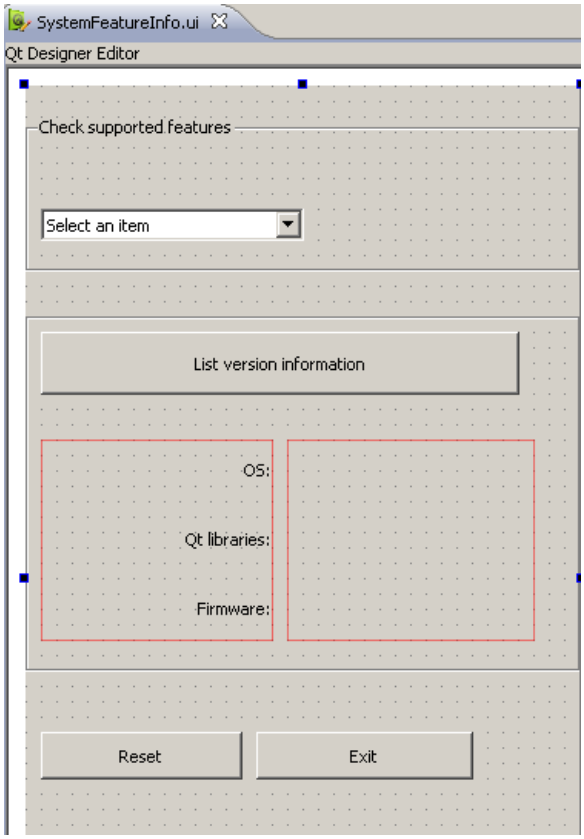
- It is a part of the System Information API.
- This article covers information about
  - Available features of the device
  - Version information of the OS
  - Current firmware information of the device

Capabilities required

- ReadDeviceData

UI design (.ui file)

(C) Copyright Nokia 2014. 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 Qt-specific configuration options in .pro file as shown below

```
QT += network
```

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

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

Implementing QSystemInfo API

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

```
using namespace QtMobility;
```

Header file

```
#ifndef SYSTEMFEATUREINFO_H
#define SYSTEMFEATUREINFO_H
```


(C) Copyright Nokia 2014. All rights reserved.

```
#include <QtGui/QWidget>
#include "ui_SystemFeatureInfo.h"
#include <qsysteminfo.h>

using namespace QtMobility;
class SystemFeatureInfo : public QWidget
{
    Q_OBJECT

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

public slots:
 void checkFeatureAvailability(int aIndex);
 void listFirmwareInformation();
 void resetAllFields();

private:
    Ui::SystemFeatureInfo ui;
    QSystemInfo* sysInfo;
};

#endif // SYSTEMFEATUREINFO_H
```

Source file

```
#include "SystemFeatureInfo.h"

SystemFeatureInfo::SystemFeatureInfo(QWidget *parent)
    : QWidget(parent), sysInfo(NULL)
{
 ui.setupUi(this);
 sysInfo = new QSystemInfo();

 QObject::connect(ui.featureComboBox,
     SIGNAL(currentIndexChanged(int)),
     this,
     SLOT(checkFeatureAvailability(int)));
 QObject::connect(ui.listVersionInfoButton,
     SIGNAL(clicked()),
     this,
     SLOT(listFirmwareInformation()));
 QObject::connect(ui.resetButton,
     SIGNAL(clicked()),
     this,
     SLOT(resetAllFields()));
}

SystemFeatureInfo::~SystemFeatureInfo()
{
 delete sysInfo;
 sysInfo = NULL;
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
void SystemFeatureInfo::checkFeatureAvailability(int aIndex)
{
 ASSERT(aIndex >=0 && aIndex <=13);

 if(! aIndex)
 {
  ui.label->clear();
  return;
 }

 bool val = sysInfo->hasFeatureSupported((QSystemInfo::Feature)(aIndex-1));
 if(val)
 {
  ui.label->setText("Available");
 }
 else
 {
  ui.label->setText("Not available");
 }
}

void SystemFeatureInfo::listFirmwareInformation()
{
 if(sysInfo)
 {
  QString os = sysInfo->version(QSystemInfo::Os);
  ui.OSlabel->setText(sysInfo->version(QSystemInfo::Os));
  ui.qtLabel->setText(sysInfo->version(QSystemInfo::QtCore));
  ui.firmwareLabel->setText(sysInfo->version(QSystemInfo::Firmware));
 }
}
void SystemFeatureInfo::resetAllFields()
{
 ui.featureComboBox->setCurrentIndex(0);
 ui.label->clear();
 ui.OSlabel->clear();
 ui.qtLabel->clear();
 ui.firmwareLabel->clear();
}
```

Output

(C) Copyright Nokia 2014. All rights reserved.

Useful functions

- bool hasFeatureSupported(QSystemInfo::Feature feature);
- QString version(QSystemInfo::Version type, const QString & parameter = QString());

Keywords

Headers

- #include <qsysteminfo.h>

Classes

- QSystemInfo
- QString
- QStringList
- QButton
- QLabel
- QComboBox

Example Application

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

Related articles

- Getting started with Qt Mobility APIs

(C) Copyright Nokia 2014. All rights reserved.

- 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 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
- 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 2014. All rights reserved.