Working with QSystemStorageInfo - System Information API

Overview

This article shows how to use `QSystemStorageInfo` API which is a part of the *System Information API* from *Qt Mobility project*.

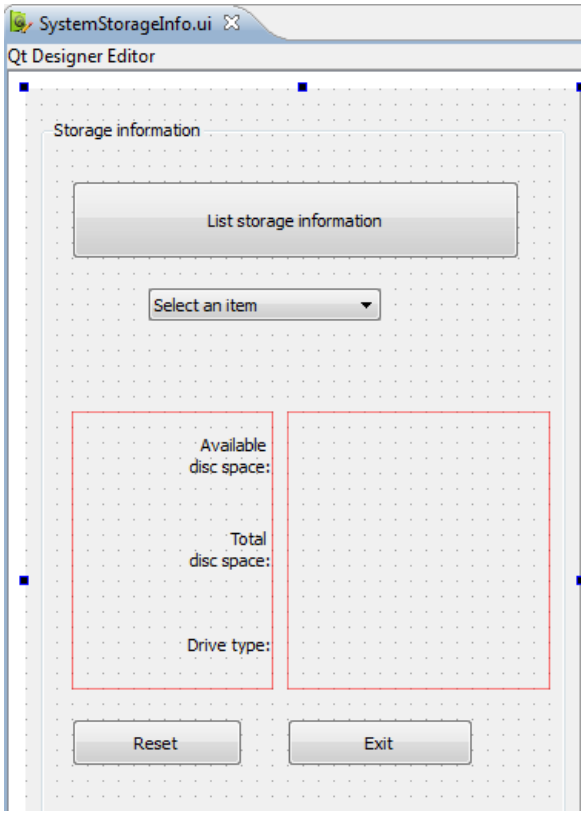
What is covered in QSystemStorageInfo API

- It is a part of the System Information API.
- This article covers information about
  - Available drives of the device
  - Available disc space of the particular drive
  - Total disc space of the particular drive
  - Type of the particular drive

Capabilities required

- None

UI design (.ui file)



Project configuration file (.Pro file)

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

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

Implementing QSystemStorageInfo API

(C) Copyright Nokia 2014. All rights reserved.

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

```
using namespace QtMobility;
```

Header file

```
#ifndef SYSTEMSTORAGEINFO_H
#define SYSTEMSTORAGEINFO_H

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

using namespace QtMobility;
class SystemStorageInfo : public QWidget
{
    Q_OBJECT

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

public slots:
 void listLogicalDrives();
 void listDriveInformation(int aDriveIndex);
 void resetAllFields();

private:
    Ui::SystemStorageInfo ui;
    QSystemStorageInfo* storageInfo;
};

#endif // SYSTEMSTORAGEINFO_H
```

Source file

```
#include "SystemStorageInfo.h"

SystemStorageInfo::SystemStorageInfo(QWidget *parent)
    : QWidget(parent), storageInfo(NULL)
{
 ui.setupUi(this);

 storageInfo = new QSystemStorageInfo();

 QObject::connect(ui.driveComboBox,
     SIGNAL(currentIndexChanged(int)),
     this,
     SLOT(listDriveInformation(int)));
 QObject::connect(ui.storageInfoButton,
     SIGNAL(clicked()),
     this,
```


(C) Copyright Nokia 2014. All rights reserved.

```
     SLOT(listLogicalDrives()));
 QObject::connect(ui.resetButton,
     SIGNAL(clicked()),
     this,
     SLOT(resetAllFields()));
}

SystemStorageInfo::~SystemStorageInfo()
{
 delete storageInfo;
}

void SystemStorageInfo::listLogicalDrives()
{
 if(ui.driveComboBox->count() > 1)
 {
  //this has been called previously and combo box contains data already
  ui.driveComboBox->showPopup();
  return;
 }
 QStringList list = storageInfo->logicalDrives();
 ui.driveComboBox->addItems(list);
 ui.driveComboBox->showPopup();
}

void SystemStorageInfo::listDriveInformation(int aDriveIndex)
{
 if(aDriveIndex == 0)
 {
  resetAllFields();
  return;
 }
 // Set available disc space label
 QString textValue = ui.driveComboBox->itemText(aDriveIndex);
 qlonglong availableSpace = storageInfo->availableDiskSpace(textValue);
 QString str = QString::number(availableSpace);
 ui.availabelDiscSpaceLabel->setText(str);

 // Set total disc space label
 qlonglong totalSpace = storageInfo->totalDiskSpace(textValue);
 str = QString::number(totalSpace);
 ui.totalDiscSpaceLabel->setText(str);

 // Set drive type label
 QString labelValue;
 QSystemStorageInfo::DriveType drvType = storageInfo->typeForDrive(textValue);
 switch(drvType)
 {
  case QSystemStorageInfo::NoDrive:
  {
   labelValue = ("No drive");
   break;
  }
  case QSystemStorageInfo::InternalDrive:
  {
   labelValue = ("Internal Drive");
   break;
```

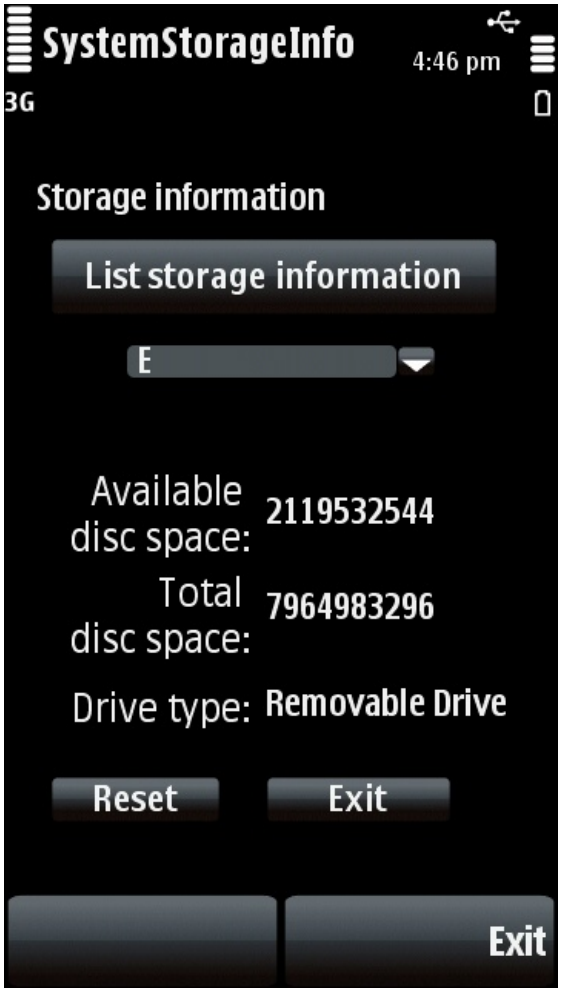

(C) Copyright Nokia 2014. All rights reserved.

```
   }
   case QSystemStorageInfo::RemovableDrive:
   {
    labelValue = ("Removable Drive");
    break;
   }
   case QSystemStorageInfo::RemoteDrive:
   {
    labelValue = ("Remote Drive");
    break;
   }
   case QSystemStorageInfo::CdromDrive:
   {
    labelValue = ("CD ROM Drive");
    break;
   }
   default:
   {
    break;
   }
  }
  ui.driveTypeLabel->setText(labelValue);
 }

 void SystemStorageInfo::resetAllFields()
 {
  ui.driveComboBox->setCurrentIndex(0);
  ui.availabelDiscSpaceLabel->clear();
  ui.totalDiscSpaceLabel->clear();
  ui.driveTypeLabel->clear();
 }
```

Output

(C) Copyright Nokia 2014. All rights reserved.



Useful functions

- QStringList logicalDrives ();
- qlonglong availableDiskSpace (const QString & volumeDrive);
- qlonglong totalDiskSpace (const QString & volumeDrive);
- QSystemStorageInfo::DriveType typeForDrive (const QString & driveVolume);

Headers

- #include <qsysteminfo.h>

Classes

- QSystemStorageInfo
- QString
- QStringList
- QButton
- QLabel
- QComboBox

Example Application

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

Related articles

- Getting started with Qt Mobility APIs

(C) Copyright Nokia 2014. All rights reserved.

- 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 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.