Get Storage information using Qt

Overview

This code example shows how to access storage information from the device. For example to get name of available drives, available space on each drive, total space on each drive and type of drive of the device. Code example use `QSystemStorageInfo` API from `QtMobility` APIs.

This snippet requires `ReadDeviceData` capabilities. **Self-signing** is not possible, a Developer certificate is needed to sign application for testing.

Prerequisite

- Install Qt for Symbian:Installing Qt on Symbian
- Install QtMobility APIs:Setting up environment for Qt Mobility API

Headers required

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

.pro file

Modify your **.pro** file as follows.

```
QT += core \
    gui \
    network
INCLUDEPATH += d:/qtmobility/src/systeminfo
INCLUDEPATH += D:/QtMobility/src/global

CONFIG += mobility
MOBILITY = systeminfo
symbian:TARGET.CAPABILITY = ReadDeviceData
```

Header file

```
/* GetStorageInfo.h */
#ifndef GETSTORAGEINFO_H
#define GETSTORAGEINFO_H

#include <QtGui/QMainWindow>
#include <qsysteminfo.h>

QTM_USE_NAMESPACE;

class QListWidget;
class QScrollArea;

QTM_USE_NAMESPACE;

class GetStorageInfo : public QMainWindow
{
    Q_OBJECT

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

private:
 void resizeEvent (QResizeEvent* event);
 void setWidgetGeometry(QSize* screenSize);
    void getStorageInformation();

private:
    QSystemStorageInfo* sysInfo;
 QListWidget     *listWidget;
 QScrollArea     *scrollArea;
};

#endif // GETSTORAGEINFO_H
```

Source file

```
/* GetStorageInfo.cpp */
#include "GetStorageInfo.h"
#include <QListWidget>
#include <QScrollArea>
```


(C) Copyright Nokia 2013. All rights reserved.

```
#include <QResizeEvent>

GetStorageInfo::GetStorageInfo(QWidget *parent)
    : QMainWindow(parent)
{
 setWindowTitle("Get storage information");

 /*The sysInfo becomes a child of the GetDisplayInfo, and will be destroyed when the GetDisplayInfo is deleted */
 sysInfo = new QSystemStorageInfo(this);

 /* To show items in list view */
 listWidget = new QListWidget();
 listWidget->setObjectName(QString::fromUtf8("listWidget"));

 /* To make UI scrollable */
 scrollArea = new QScrollArea();
 setWidgetGeometry(&(size()));

 /* scrollArea takes ownership of the listWidget, so no need to delete it */
 scrollArea->setWidget(listWidget);
 scrollArea->setAlignment(Qt::AlignLeft);
 scrollArea->setWidgetResizable(true);

 /* QMainWindow takes ownership of the scrollArea, so no need to delete it */
 setCentralWidget(scrollArea);
 getStorageInformation();
}

GetStorageInfo::~GetStorageInfo()
{

}

void GetStorageInfo::resizeEvent (QResizeEvent* event)
{
 QSize widgetSize = event->size();
 setWidgetGeometry(&widgetSize);
 QMainWindow::resizeEvent(event);
}

void GetStorageInfo::setWidgetGeometry(QSize* screenSize)
{
 resize(*screenSize);
 QRect rect(0,0,screenSize->width(), screenSize->height());
 listWidget->setGeometry(rect);
}

void GetStorageInfo::getStorageInformation()
{
 QString strInfo;
        //* qlonglong defined in 4.2 same as qint64

 qlonglong   diskSpace;

 /*get logical drives */
 QStringList driveList = sysInfo->logicalDrives();

 strInfo.append("Drives information: ") ;
 listWidget->addItem(strInfo);

 foreach ( QString str, driveList )
 {
  strInfo.clear();
  strInfo.append("   Drive name: " + str);
  listWidget->addItem(strInfo);

  /* get available space on drive. */
  diskSpace = sysInfo->availableDiskSpace(str);

  strInfo.clear();

  if(diskSpace > 1000000)
  {
   diskSpace /= 1000000;
   strInfo.setNum(diskSpace);
   strInfo.prepend("   Available space: ");
   strInfo.append("MB");

  }
  else if(diskSpace > 1000)
  {
   diskSpace /= 1000;
   strInfo.setNum(diskSpace);
   strInfo.prepend("   Available space: ");
   strInfo.append("KB");

  }
  else
  {
   strInfo.setNum(diskSpace);
   strInfo.prepend("   Available space: ");
   strInfo.append("Bytes");
  }

  listWidget->addItem(strInfo);

  /* get total space on drive */
  diskSpace = sysInfo->totalDiskSpace(str);

  strInfo.clear();

  if(diskSpace > 1000000)
  {
   diskSpace /= 1000000;
   strInfo.setNum(diskSpace);
   strInfo.prepend("   Total space: ");
   strInfo.append("MB");

  }
  else if(diskSpace > 1000)
  {
   diskSpace /= 1000;
   strInfo.setNum(diskSpace);
   strInfo.prepend("   Total space: ");
   strInfo.append("KB");

  }
  else
  {
   strInfo.setNum(diskSpace);
```

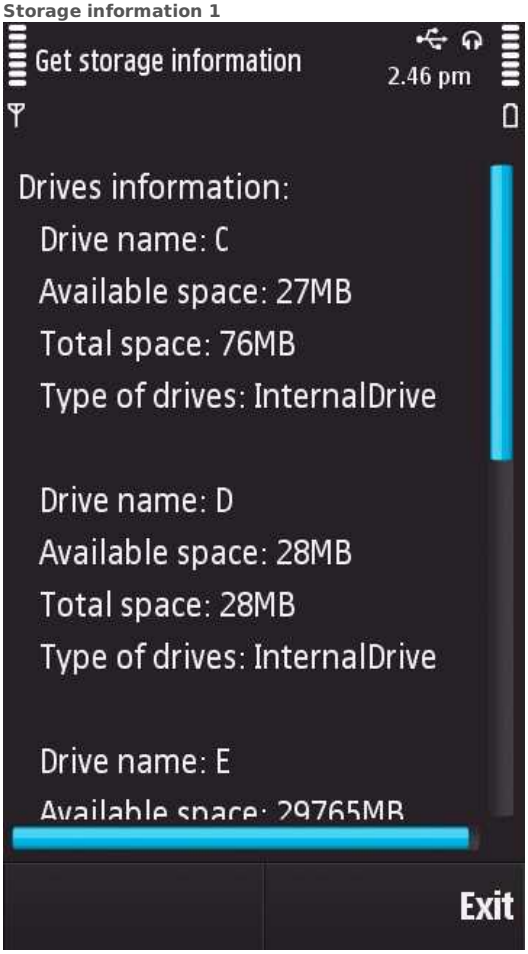

(C) Copyright Nokia 2013. All rights reserved.

```
    strInfo.prepend("   Total space: ");
    strInfo.append("Bytes");
   }

   listWidget->addItem(strInfo);

   strInfo.clear();
   strInfo.append("   Type of drives: ");

   /* get type of drive */
   switch(sysInfo->typeForDrive(str))
   {
   case QSystemStorageInfo::InternalDrive:
    {
    strInfo.append("InternalDrive");
    break;
    }
   case QSystemStorageInfo::RemovableDrive:
    {
    strInfo.append("RemovableDrive");
    break;
    }
   case QSystemStorageInfo::RemoteDrive:
    {
    strInfo.append("RemoteDrive");
    break;
    }
   case QSystemStorageInfo::CdromDrive:
    {
    strInfo.append("CdromDrive");
    break;
    }
   case QSystemStorageInfo::NoDrive:
   default:
    {
    strInfo.append("NoDrive available");
    break;
    }
   }

   listWidget->addItem(strInfo);

   strInfo.clear();
   strInfo.append("  ");
   listWidget->addItem(strInfo);
  }
 }
```

Postconditions

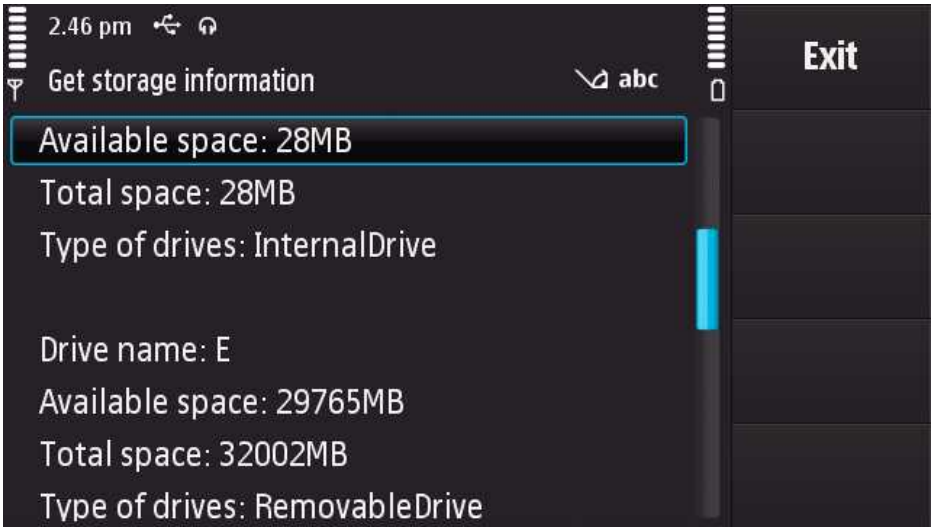
The code snippet is expected to read storage information from device and show it on screen.

Storage information 1



Storage information 2

(C) Copyright Nokia 2013. All rights reserved.



Download Code Example

- Working Code Example is available to download from here.

Related articles

- Get Device information using Qt for Symbian
- Get Display information using Qt for Symbian
- Get Network information using Qt for Symbian
- Get System information using Qt for Symbian

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.