How to read and display contacts from phonebook using Qt Mobility

Overview

This article shows how to read contacts from contact book (phonebook). User might need to read contacts programmatically by one or more reason, this code example shows how to use QtMobility Contacts APIs to read contacts from contact book.

This snippet requires `ReadUserData`, `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 <qtcontacts.h>
```

.pro file

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

```
INCLUDEPATH += D:/QtMobility/src/contacts \
INCLUDEPATH += D:/QtMobility/src/contacts/filters \
INCLUDEPATH += D:/QtMobility/src/contacts/requests \
INCLUDEPATH += D:/QtMobility/src/contacts/details
INCLUDEPATH += D:/QtMobility/src/global

CONFIG += mobility
MOBILITY = contacts

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

Header file

```
#ifndef DISPLAYCONTACTS_H
#define DISPLAYCONTACTS_H

#include <QtGui/QMainWindow>
#include "qtcontacts.h"

QTM_USE_NAMESPACE

class QScrollArea;
class QListWidget;

class DisplayContacts : public QMainWindow
{
    Q_OBJECT

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

private:
    void fetchContcatsFromContactBook();

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

#endif // DISPLAYCONTACTS_H
```

Source file

```
#include "DisplayContacts.h"
#include <QListWidget>
#include <QScrollArea>

DisplayContacts::DisplayContacts(QWidget *parent)
    : QMainWindow(parent)
```

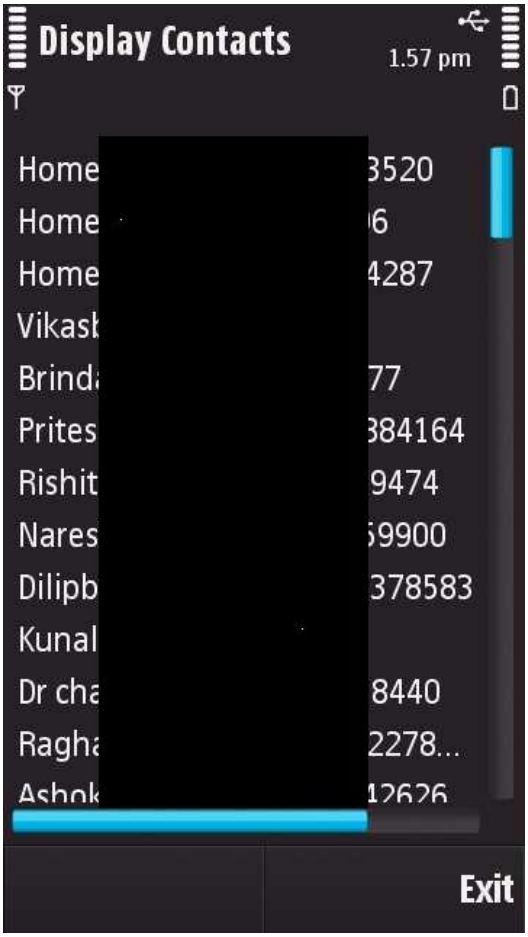

(C) Copyright Nokia 2013. All rights reserved.

```
{
 setWindowTitle("Display Contacts");

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

 QSize screenSize(size());
 resize(screenSize);
 QRect rect(0,0,screenSize.width(), screenSize.height());
 listWidget->setGeometry(rect);

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

 /* 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);

 fetchContcatsFromContactBook();
}

DisplayContacts::~DisplayContacts()
{

}

void DisplayContacts::fetchContcatsFromContactBook()
{
 /* read contact from "symbian" contact manager */
 QContactManager *contactManager = new QContactManager("symbian");

 /* You can also read contact from selected contact manager by selecting
  * appropriate contact manager from available contact managers.

 *QStringList availableManagers = QContactManager::availableManagers();
 *iterate througn available manager and select appropriate one.
 *contactManager = new QContactManager(availableManagers.at(2));
 */

 /* get list of all contacts */
 QList<QContactLocalId> contactIds = contactManager->contactIds();
 QContact currContact;

 /* appens it to list widget to display on screen*/
 foreach (const QContactLocalId& id, contactIds)
 {
        currContact = contactManager->contact(id);
        QString nameAndNumber(currContact.displayLabel() + ": " + currContact.detail(QContactPhoneNumber::DefinitionName).value(QCon
        listWidget->addItem(nameAndNumber);
    }
}
```

Postconditions

The code snippet is expected to read and display contacts from symbian contact manager.

 (C) Copyright Nokia 2013. All rights reserved.



Download Code Example

- Working Code Example is available to download from here.

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.