Developing NFC Applications

This article explains how to create NFC applications for the latest NFC equipped Symbian phones using Qt or Java technology. The article provides details on the UI considerations in NFC application design and the requirements for NFC applications to be distributed through Nokia Store. A pointer is provided to a project that enables NFC features for Symbian WRT widgets.

Introduction

You can create applications that leverage NFC technology using APIs provided in Qt, Java™ ME, and Symbian C++. Of these options, using Qt is the recommended approach as it's the core development framework for Symbian and is available for MeeGo powered phones as well.

For details on developing apps that use NFC on Windows Phone 8, please see NFC: "Tap and Do" / Reading and writing from tags and NFC: "Tap and Share" / Phone to phone in the article What's new in Windows Phone 8.

Qt based development

You create NFC enabled applications using the Qt Mobility 1.2 Connectivity API. The advantage of this approach is that you are working with the productive Qt cross-platform API, so you can code once for an app that will run on Symbian and MeeGo phones.

To get started you need:

- an NFC enabled phone
- The Qt SDK
- Connectivity API documentation.

In the SDK you will find Qt Mobility 1.2. binaries that can be installed on a Nokia C7 Device with Symbian Anna. The package enables development with the Symbian C++ NFC API also.

Further here are some example apps you might want to use as a starting point:

- Nfc Info (show information about tags and parse NDEF messages, including Smart Posters and vCards)
- Nfc Corkboard (demonstrates NDEF reading and writing)
- Nfc Chat (LLCP peer-to-peer communication).

In addition, you can download a step-by-step walkthrough tutorial that explains how to code the latter two apps plus the basic Qt concepts required for understanding the interaction.

Java ME based development

You can make use of the NFC features in Symbian phones using the Contactless Communication API (JSR-257) and Nokia Extensions to JSR-257. This may be the most convenient approach if you have developed NFC applications for Series 40 phones (such as the Nokia 6131 NFC and Nokia 6212 classic). To get started you will need an NFC enabled phone plus:

- NetBeans or Eclipse IDE
- Symbian^3 SDK 1.0
- NFC plug-in for the Nokia Symbian^3 SDK 1.0
- Developer's guide

You can use the NFC phone SDKs for the Series 40 phones also, if you wish. In particular, these SDKs currently offer better features for testing.

Symbian C++ based development

If you're familiar with Symbian C++ you can use the Symbian C++ APIs in a pure Symbian C++ app. To do this, you need an NFC enabled phone plus:

- Carbide.c++
- Symbian^3 SDK 1.0
- NFC plug-in for the Nokia Symbian^3 SDK
- API documentation

Symbian WRT widgets

An NFC library for API Bridge is available from the NFC API Bridge project.

(C) Copyright Nokia 2014. All rights reserved.

Windows Phone 8 development

Proximity API can be used to acess NFC capabilities in the Windows Phone 8.

UI design considerations

As NFC is supported in a small (but set to grow) number of phones, it's important to take this into account when creating your application.

In order to achieve a consistent and favourable user experience, applications should be aware of the presence of an NFC radio in the phone and check if the radio is turned on in the connectivity settings. This section provides a step-by-step guide to handling these requirements, along with relevant code snippets and references.

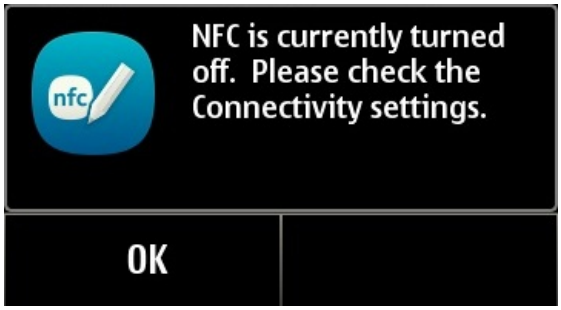
Check whether NFC supported

When an application starts it should check whether NFC is supported in the phones hardware and react appropriately. How the application should react depends on how the application uses NFC. Two design approaches are available:

NFC centric application	NFC enhanced application
This is an application that works on NFC enabled phones only.	This is an application that will works on phones without NFC phones and takes advantage of NFC only if available on a phone.
In this case the application should restrict its installation to NFC capable phones only and/or query whether the NFC functionality is available at runtime. Where NFC is available as a firmware upgrade (such as on the Nokia C7 phone through the Symbian Anna update), the application should ideally alert the user to upgrade the firmware with message "Please update your phone firmware to run this application" and then exit.	In this case the application should query whether NFC is supported in the phone's hardware at install and runtime. If the phone is determines not to support NFC the features of the application that rely on NFC should be disabled and ideally hidden from the user. If the application detects that the phone would offer NFC after a firmware upgrade, the application may alert the user to upgrade the phone's firmware, but continue running.

Check if the NFC radio is turned on or off

On Symbian phones the user has the option to manually turn the NFC radio on or off. This setting affects NFC enabled applications also. To provide a consistent and seamless user experience, applications using NFC should check the status of the NFC radio and alert the user to turn it on to use the application. At start the application queries the status of NFC radio and informs the user if the radio is off with the message "NFC is not active. Please activate NFC from Connectivity settings before using this application".



Implementation

APIs available

(C) Copyright Nokia 2014. All rights reserved.

To determine if a device supports NFC:

- In a Qt or Symbian C++ application use the *IsFeatureSupported* method using the Feature Discovery API of the *TFeatureSet* class and test for *KFeatureIdNfc* (ID = 117).
- In a Java application query the *javax.microedition.contactless.version* system property value for the supported JSR-257 version. If the API is supported on the phone, the property returns 1.0.
- In a Windows Phone 8 application, we can detect if the NFC chipset is present or not using the *ProximityDevice* class. To access this object you must use the static function *ProximityDevice.GetDefault()*. If this method returns null, your device does not support NFC.

Examples

The following examples show how to implement the UX recommendations:

- NFC Launcher (Qt/Symbian) -This example uses a separate launcher application that checks the presence of NFC on the phone and based on that either launches the main application, or displays a notification box alerting the user that NFC is not present.Translations of the notification strings are included.
- NfcSettings application(Qt/Symbian) - This project uses the NFC feature and NFC radio logic from the "Mobile NFC app" template and applies a very simple QML based UI that shows both the availability of NFC and radio status on the phone.
- NFCinteractor (Java ME) - This project demonstrates how to handle the NFC dependency in a Java Midlet. It queries the *microedition.contactless.version* system property value (JSR-257). If the API is supported on the phone, the property returns 1.0.

Testing

The Qt SDK does provide simulation of NFC tag related events and a simple NDEF editor to customize vitrual tag content.

If you choose to develop with the NFC phone SDKs for the Series 40 phones or make use of the Symbian^3 SDK with the NFC plug-in, these tools provide some emulation of NFC tags and devices suitable for application testing.

An NFC phone is recommended also to complete testing prior to distributing your application.

Application distribution via Store

NFC enabled applications can be distributed as any other application through Nokia Store. This section describes the requirements related to NFC feature handling and guidelines that must be implemented by the application to ensure a seamless user experience. These requirements need to be addressed by the publisher before submitting the application to Quality Assurance

Requirements for the Nokia C7 phone with Symbian Anna

- The application **must** handle the cases where the user has not upgraded their Nokia C7 phone's firmware using the approach outlined above
- Only free applications are accepted
- Check whether the NFC radio is turned on or off

Distribution to other Symbian NFC (also MeeGo) capable phones

- Applications may be free or premium
- Check whether the NFC radio is turned on or off

Obtaining an NFC enabled phone

If you’re a Nokia Developer Launchpad or PRO member you can obtain an NFC enabled phone through the Prototype Device Loaning Service or the Discounted Devices Program (link requres you have an active Nokia Developer Launchpad or PRO account/login).

As part of the Discounted Devices Program Nokia is making available an NFC Device Kit to Nokia Developer Launchpad and PRO members. The Kit includes an NFC mobile phone and sample NFC tags. You can purchase it from here: Discounted Devices Program (link requres you have an active Nokia Developer Launchpad or PRO account/login)..

If you’re not a member of Launchpad or PRO, you can find details on how to join Launchpad here.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.