



PMD2228-AA
DRAFT - PUBLIC

SPRINGCARD NFC SDK FOR PC/SC

Getting Started Guide

DOCUMENT IDENTIFICATION

| | | | |
|-----------------|---|----------------|--------|
| Category | Developer's manual | | |
| Family/Customer | SDK | | |
| Reference | PMD2228 | Version | AA |
| Status | draft | Classification | Public |
| Keywords | H512, NFC'Roll PC/SC, NFC emulation, contactless cards, RFID labels, NFC tags | | |
| Abstract | | | |

| | | | |
|------------|--|--------------|----------|
| File name | V:\Dossiers\notices\SDK\PCSC NFC\[PMD2228-AA] NFC SDK for PCSC Getting Started Guide.odt | | |
| Date saved | 10/10/12 | Date printed | 24/09/12 |

REVISION HISTORY

| Ver. | Date | Author | Valid. by | | Approv. by | Details |
|------|----------|--------|-----------|-------|------------|---------------------------|
| | | | Tech. | Qual. | | |
| AA | 10/10/12 | JDA | | | | First preliminary release |

PRELIMINARY

CONTENTS

| | |
|---|----|
| 1. INTRODUCTION..... | 5 |
| 1.1. ABSTRACT..... | 5 |
| 1.2. SUPPORTED PRODUCTS..... | 5 |
| 1.3. REFERENCE DOCUMENT..... | 5 |
| 1.4. AUDIENCE..... | 5 |
| 1.5. SUPPORT AND UPDATES..... | 6 |
| 1.6. ONE LAST FOREWORD..... | 6 |
| 2. A FIVE-MINUTE GUIDED TOUR TO THE SDK'S SAMPLES..... | 7 |
| 2.1. DOWNLOADING AND INSTALLING THE SDK..... | 7 |
| 2.2. NfcTOOL..... | 7 |
| 2.2.1. Introduction..... | 7 |
| 2.2.2. Characteristics of NfcTool..... | 8 |
| 2.2.3. NfcTool at work..... | 9 |
| 2.2.4. Testing the Tags created by NfcTool..... | 10 |
| 2.2.5. Tracing NfcTool's activity..... | 10 |
| 2.3. NfcTAGEMUL..... | 10 |
| 2.3.1. Characteristics of NfcTagEmul..... | 11 |
| 2.3.2. NfcTagEmul at work..... | 11 |
| 2.3.3. Testing NfcTagEmul..... | 11 |
| 2.3.4. Tracing NfcTagEmul's activity..... | 12 |
| 2.4. NfcBEAM..... | 12 |
| 2.4.1. Introduction..... | 12 |
| 2.4.2. Testing NfcBeam..... | 13 |
| 2.4.3. NfcBeam at work..... | 13 |
| 3. DEVELOPING FOR H512 UNDER .NET..... | 16 |
| 3.1. APIs AND DOCUMENTATION..... | 16 |
| 3.1.1. SpringCard PC/SC API for .NET..... | 16 |
| 3.1.2. SpringCard PC/SC API for .NET – NfcForum extension.. | 16 |
| 3.2. DISCOVERING AND READING A TAG..... | 17 |
| 3.3. MANIPULATING THE NDEF OBJECTS..... | 17 |
| 3.4. (RE)WRITING A TAG..... | 17 |
| 3.5. TAG EMULATION..... | 17 |
| 3.6. IMPLEMENTING SNEP IN INITIATOR MODE..... | 17 |

1. INTRODUCTION

1.1. ABSTRACT

The **H512 series** is a versatile NFC device, supporting the 3 modes of operation:

- **Reader (+Writer) mode**,
- **Card emulation** mode, with in-device support for NFC Forum type 2 and type 4 Tag emulation,
- **Peer-to-peer** mode, either as Initiator or Target.

Despite its innovative NFC features, **H512** remains fully compliant with **PC/SC**, and could be operated through a basic CCID driver using only standard PC/SC function calls (SCardTransmit / SCardControl).

The **SpringCard NFC SDK for PC/SC** is intended to show the potential of the **H512**, and to help the developers design their own applications using the new **NFC** features (card emulation and peer-to-peer). Developers who only want to communicate with contactless smartcards shall use the more classical SpringCard PC/SC SDK instead.

This document is the “getting started guide” associated to the **SpringCard NFC SDK for PC/SC**.

1.2. SUPPORTED PRODUCTS

At the time of writing, this document refers to all SpringCard PC/SC Readers in the **H512** group:

- The **H512S** : OEM module without antenna,
- The **H512-USB** : OEM module with antenna,
- The **NFC'Roll** : desktop reader with LEDs and buzzer.

1.3. REFERENCE DOCUMENT

This **Getting Started Guide** does not pretend to be exhaustive. It is only intended to give an overview of the SDK's content, and a short introduction to the specificities of NFC-related software development.

Document **PMD2176 “H512 Developers' Reference Manual”** is the reference manual for those products.

1.4. AUDIENCE

This manual is written for use by application developers. It assumes that the reader has expert knowledge of computer development and a basic knowledge of PC/SC and NFC.

To get started with PC/SC, please read our [Introduction to PC/SC development and simplified documentation of the API](#), available online at

<http://www.springcard.com/download/find.php?file=pmdz061>

To get started with NFC, please visit NFC Forum (<http://www.nfc-forum.org>) and download their specifications.

1.5. SUPPORT AND UPDATES

Useful related materials (product datasheets, application notes, sample software, HOWTOs and FAQs...) are available at SpringCard's web site:

www.springcard.com

Updated versions of this document and others are posted on this web site as soon as they are available.

For technical support enquiries, please refer to SpringCard support page, on the web at

www.springcard.com/support

1.6. ONE LAST FOREWORD...

All the examples in the SDK are developed in C#, and target Microsoft's .NET Framework.

There's a very good reason to this choice of a **high-level, object oriented language**: the **NFC Forum**, the organisation in charge of developing the application-level standards for NFC, has set up a complex layered architecture, that is very difficult to implement and maintain in a low level language.

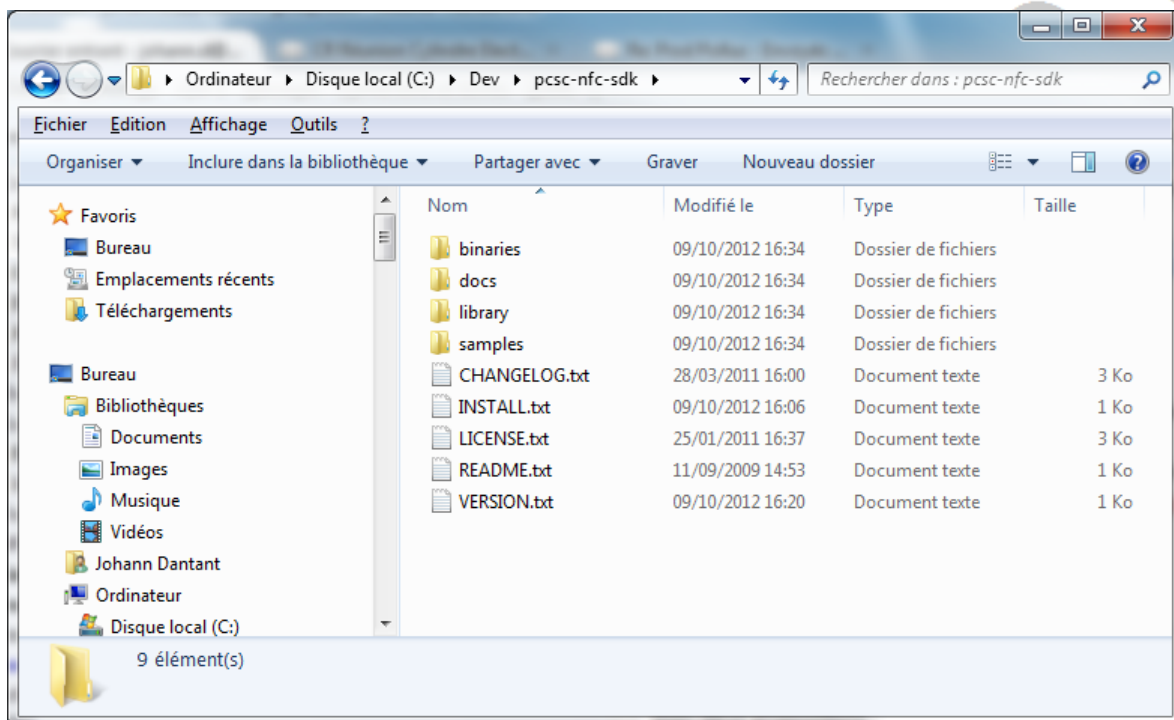
2. A FIVE-MINUTE GUIDED TOUR TO THE SDK'S SAMPLES

2.1. DOWNLOADING AND INSTALLING THE SDK

The SDK is available at

<http://www.springcard.com/download/find.php?file=pcsc-nfc-sdk>

This is a ZIP archive. Just unzip it to a folder of your choice, for instance `c:\dev\pcsc-nfc-sdk`.



The SDK's root directory

Explore the `binaries` folder of the SDK. There are 3 sample applications.

2.2. NfcTOOL

2.2.1. Introduction

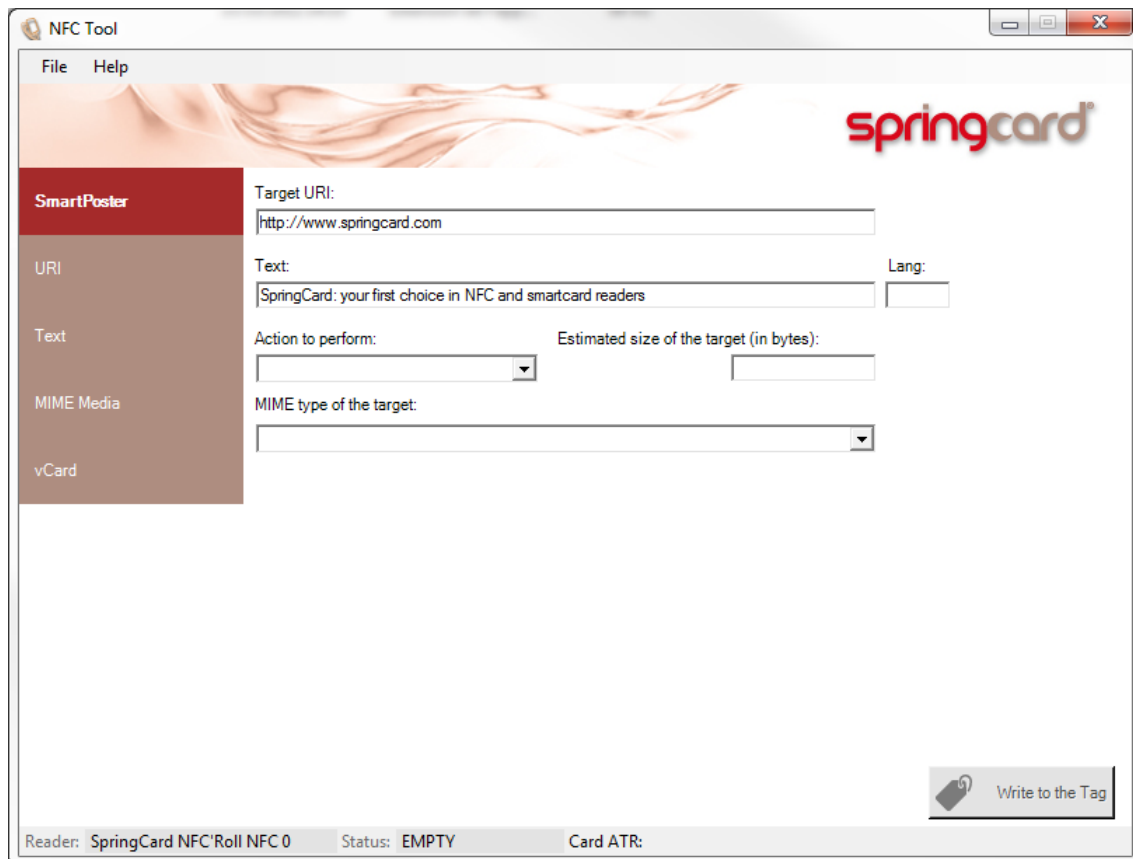
NfcTool is a basic NFC Forum Tag read/write application. "Read/write" means that **NfcTool** is not specific to **H512** and could work with other SpringCard readers as well (**NfcTool** is also included in the classical PC/SC SDK).

The most interesting part of **NfcTool** is in fact its source code, which covers

- encoding and decoding of NFC Forum messages (NDEF),

- reading and writing NFC Forum Tags (type 2 and type 4).

Those 2 key features are basic layers, and the 2 other applications (see below) are built on those layers, so understanding how **NfcTool** works is a good start before going further with either **NfcTagEmul** or **NfcBeam**.



NfcTool's main screen

2.2.2. Characteristics of NfcTool

a. Supported Tags

- NFC Forum type 2 Tag, formatted or blank,
- NFC Forum type 4 Tag, already formatted.

b. Supported NDEF types

- URI (or URL)
- Text

- SmartPoster (URI + Text + information fields)
- Media with a MIME type (text content only)
- vCard (Media with MIME type = vCard)

2.2.3. NfcTool at work

a. Startup

Launch **NfcTool.exe**. The program shows a list of available PC/SC readers. Pick-up your **H512** reader (namely either “**SpringCard H512 NFC 0**” or “**SpringCard NFC'Roll NFC 0**”).

b. Reading a Tag

Put a NFC Forum Tag on the reader. If the Tag is already formatted and has a valid and supported NDEF message, **NfcTool** displays the Tag's content. Otherwise, an error message is raised.

c. Writing a Tag

If the Tag is writable, the “Write to the Tag” button is enabled. If not, select another Tag ;-).

Keep the Tag on the reader while defining the new content: choose the content's type (SmartPoster, URI, Text, MIME Media, or vCard), and then enter the new data.

When finished, click “Write to the Tag”. You may be prompted for confirmation before overwriting an existing content if the Tag is not blank, or before formatting a Tag that has never been formatted yet.

d. Writing and locking a Tag

“Locking” means putting the Tag in read-only mode. To enable this feature, click “File” in the menu, then “Settings”, and check “Enable Locking the Tags”.

A new check-box appears near the “Write to the Tag” button. Check this box before clicking the button if you want to put the Tag in read-only mode after writing. You'll be prompted for confirmation.

Important disclaimer

Once a Tag has been Locked in read-only state, there's no cancel function. The Tag is read-only forever. Don't play with this feature until you're ready to issue your Tags with a permanent content.

2.2.4. Testing the Tags created by NfcTool

All you need is a NFC-enabled smartphone that has complete support for NFC Forum Tags. We recommend a device based on **Android 4.x**, and the **NXP TagInfo** application, to be downloaded from Google Play Store:

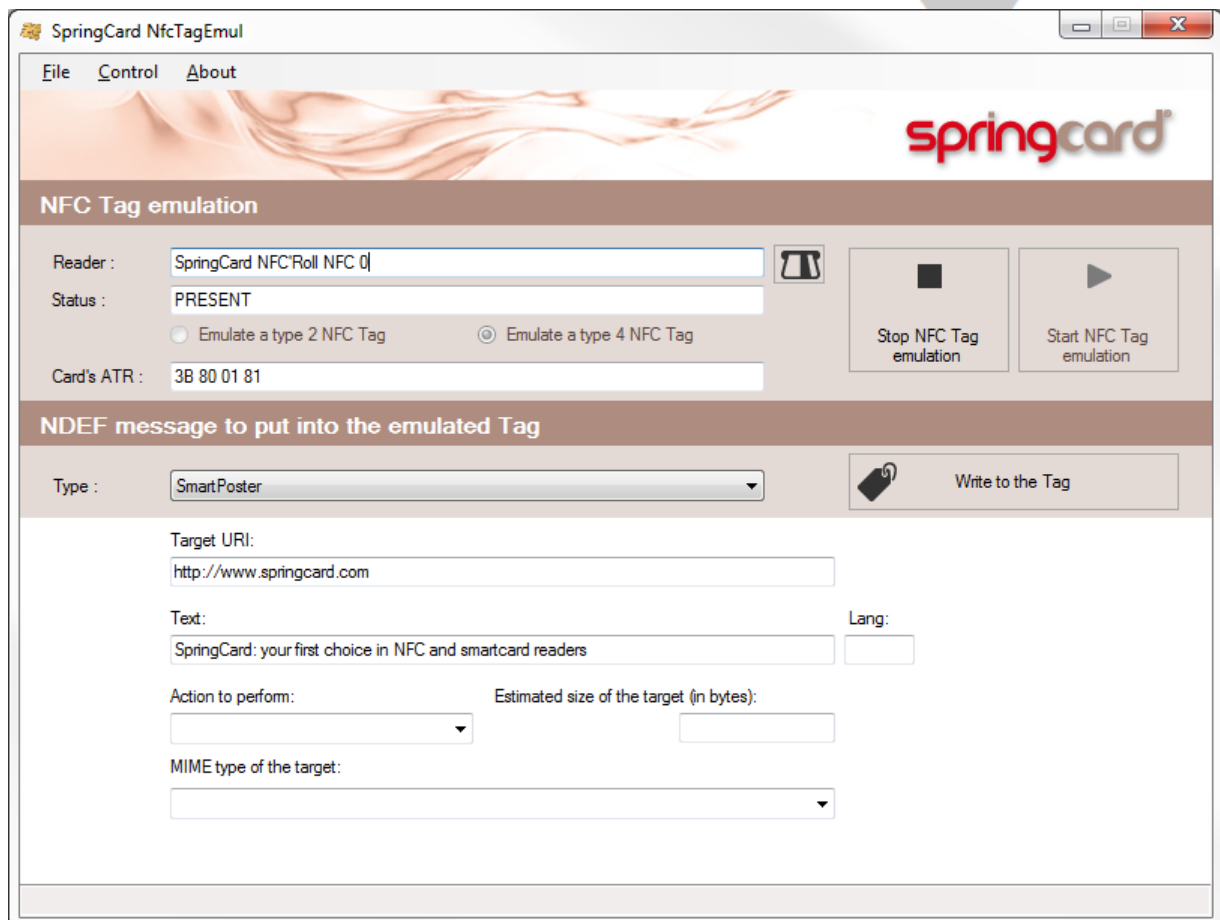
<https://play.google.com/store/apps/details?id=com.nxp.taginfolite>

2.2.5. Tracing NfcTool's activity

NfcTool is primarily a developer's app. If you want to see what's going on between the application and the reader (and therefore with the Tag), just check the "Show trace console" box in the Settings panel.

2.3. NfcTAGEMUL

NfcTagEmul is built on the same roots as **NfcTool**, but instead of Writing data into a Tag, **NfcTagEmul** just turn the **H512** itself into a Tag.



NfcTagEmul's main screen

2.3.1. Characteristics of NfcTagEmul

a. Supported Tags in emulation mode

The Tag emulation is performed by the **H512** itself. Supported types are:

- NFC Forum type 2 Tag, with 1024B of memory,
- NFC Forum type 4 Tag, with 4096B of memory.

b. Supported NDEF types

- URI (or URL)
- Text
- SmartPoster (URI + Text + information fields)
- Media with a MIME type (text content only)
- vCard (Media with MIME type = vCard)

2.3.2. NfcTagEmul at work

Launch **NfcTagEmul.exe**. The program shows a list of available PC/SC readers. Pick-up your **H512** reader (namely either “**SpringCard H512 NFC 0**” or “**SpringCard NFC'Roll NFC 0**”).

a. Entering NFC Tag emulation mode

Select either “Emulate a type 2 NFC Tag” or “Emulate a type 4 NFC Tag” in the top panel. Click “Start NFC Tag emulation”.

If your reader has some LEDs and a buzzer (**NFC'Roll** for instance), the LEDs become pink and the buzzer sounds briefly.

b. Writing a NDEF message into the emulated Tag

Choose the content's type (SmartPoster, URI, Text, MIME Media, or vCard), and then enter the new data in the bottom panel.

When finished, click “Write to the Tag”.

2.3.3. Testing NfcTagEmul

Same as for **NfcTool**: see § 2.2.4. The smartphone makes no difference between an actual NFC Tag and the **H512** running in Tag emulation mode.

If you have more than one **SpringCard** readers, you may also have the **H512** running in Tag emulation mode -under control of **NfcTagEmul** application- and in the same time connect **NfcTool** to the other reader to read the emulated Tag¹.

2.3.4. Tracing NfcTagEmul's activity

NfcTagEmul is primarily a developer's app. If you want to see what's going on between the application and the reader (and therefore with the Tag), just check the "Show trace console" box in the Settings panel.

2.4. NFCBEAM

2.4.1. Introduction

"NFC beaming" is the "end user" name for the exchange of data in NFC peer-to-peer mode.

Technically-speaking, this is a complex layered system, inspired by the OSI network model:

- NFC Forum NFC-DEP, i.e. **ISO 18092** (NFCIP1) is the MAC layer,
- NFC Forum **LLCP** (Logical Link Control Protocol) provides a Connection-oriented Transport and a Connectionless Transport (more or less the same as TCP and UDP on IP),
- NFC Forum **SNEP** (Simple NDEF Exchange Protocol) is a client/server scheme designed for the remote access to application messages (not far from FTP on top of TCP/IP). The **SNEP Server** waits for messages, the **SNEP Client** pushes its message to the remote server.
- NFC Forum **NDEF** is the format of the application messages. They are the same as the messages stored on the NFC Tags (so the link to **NfcTool** and **NfcTagEmul**).

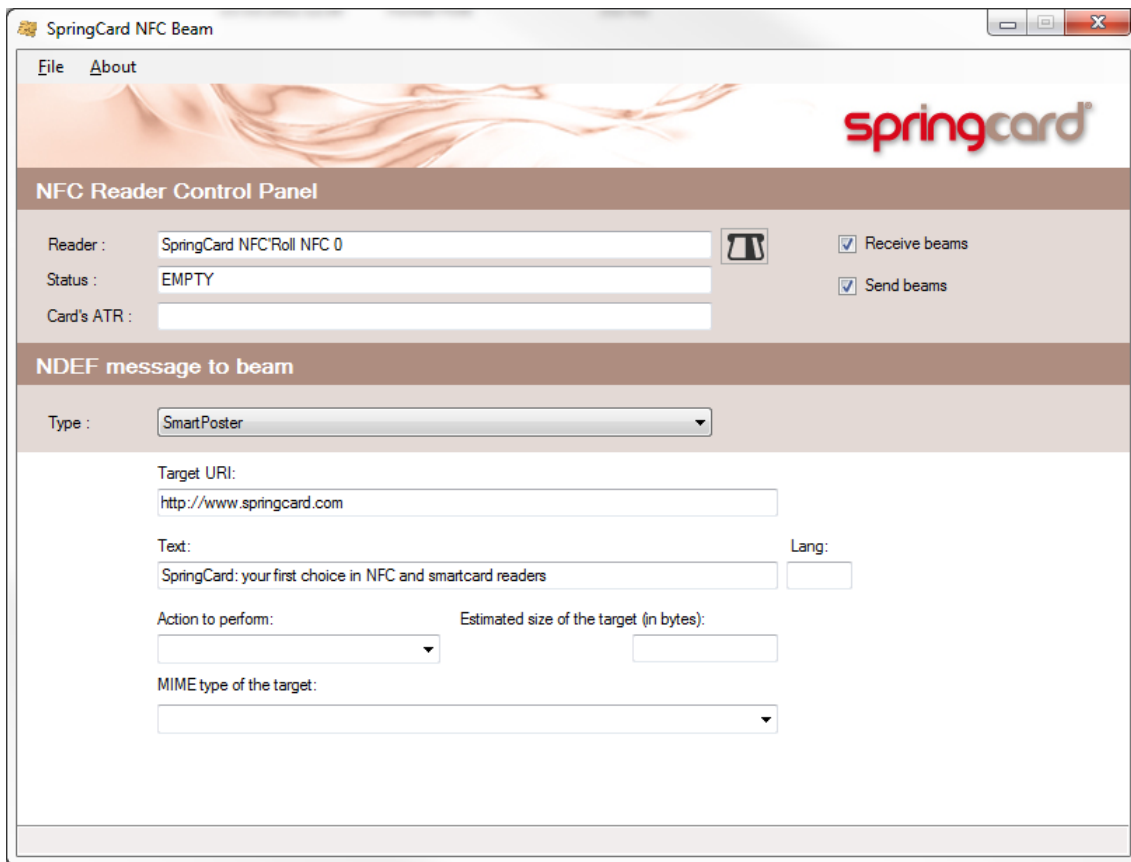
SpringCard NfcBeam implements both a SNEP Client and a SNEP Server, i.e. the application is both able to receive and to send NFC beams from/to a smartphone supporting the SNEP system.

Important disclaimer

NfcBeam has been tested with Android 4.0 and 4.1 targets only. No assumption is made that this application would work with any other NFC-enabled system.

In particular, Android 2.x and 3.x branches have no support for SNEP, and therefore are not able to communicate with NfcBeam.

¹ When accessed through its contactless interface, the emulated Tag behaves as if it were locked in read-only mode. As a consequence, NfcTool will not be able to Write it.



NfcBeam's main screen

2.4.2. Testing NfcBeam

All you need is a NFC-enabled smartphone that has complete support for NFC Forum SNEP. We recommend a device based on **Android 4.x**. In the “System Settings” menu of the Android device, check that “Android Beam” is active.

2.4.3. NfcBeam at work

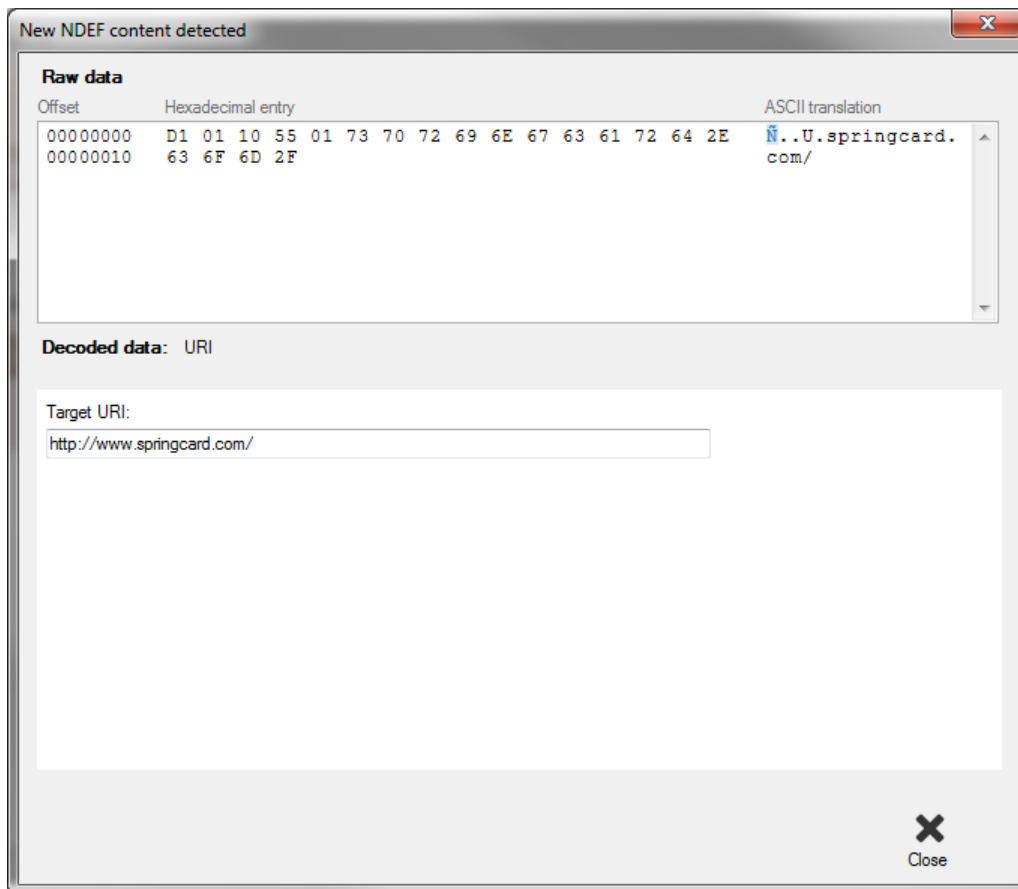
Launch **NfcBeam.exe**. The program shows a list of available PC/SC readers. Pick-up your **H512** reader (namely either “**SpringCard H512 NFC 0**” or “**SpringCard NFC'Roll NFC 0**”).

a. NfcBeam in “Server only” mode

Un-check the box “Send beams”. Check the box “Receive beams”. Doing so, the SNEP Server is running and waiting for beams coming from the SNEP clients.

Activate any SNEP-aware application in your smartphone. For instance, open the web browser and navigate to any URL. Put the smartphone on the reader, and confirm you want to share the content (tap on the screen).

NfcBeam receives the beam (this is actually a NDEF URI Record holding an link to the page you're visiting) and opens a new window to display it:



The “New NDEF content detected screen”, showing an URI Record

Note that the top panel of this screen shows the raw data received by the SNEP Server. This helps debugging a smartphone application sending proprietary messages.

If the message is of a known type (SmartPoster, URI, Text, MIME Media, or vCard), it is displayed “in human-readable form” in the bottom panel.

b. NfcBeam in “Client only” mode

Un-check the box “Receive beams”. Check the box “Send beams”. Doing so, the SNEP Server is halted, and only the SNEP Client is running. As soon as a SNEP Server is discovered, **NfcBeam** will try to push its content.

Before putting your smartphone on the reader, choose the content's type (SmartPoster, URI, Text, MIME Media, or vCard), and enter the new data in the bottom panel.

For instance, select “URI” and enter “<http://www.springcard.com>”.

Put the smartphone on the reader. The smartphone opens its web browser and goes to SpringCard's website!

c. Bi-directional mode

When both "Receive beams" and "Send beams" are checked, **NfcBeam** waits for a NDEF message coming from the smartphone, and sends its own NDEF message in response.

3. DEVELOPING FOR H512 UNDER .NET

3.1. APIs AND DOCUMENTATION

3.1.1. SpringCard PC/SC API for .NET

SpringCard PC/SC API for .NET is a set of classes designed to interface the native PC/SC subsystem (winscard.dll) from .NET programs easily.

It is implemented in a single file: [library/dotnet/SpringCardPCSC.cs](#) .

The documentation is available online at

http://www.springcard.com/support/apidoc/pcsc_dotnet/index.html

For convenience, the SDK archive contains a copy of this documentation under [docs/pcsc_dotnet](#) , but pay attention that the documentation is regularly improved, so the copy may be outdated. Online documentation shall always been taken as reference.

The basic classes are:

- **SCARD** : provides access to winscard.dll's functions through static methods,
- **SCardReader** : the smartcard reader object. Featuring a “Monitor” process to receive notifications of card insertion/removal,
- **SCardReaderList** : allows receiving notifications of card insertion/removal for more than one reader,
- **SCardChannel** : the smartcard channel object, to exchange APDUs with the smartcard in the reader,
- **CardBuffer**, **CAPDU** and **RAPDU** : improved “array of bytes” types, designed to simplify the formatting and the decoding of smartcard-related buffers.

3.1.2. SpringCard PC/SC API for .NET – NfcForum extension

SpringCard PC/SC API for .NET – NfcForum extension is a set of classes designed to interface the native PC/SC subsystem (winscard.dll) from .NET programs easily.

It is implemented in a single file: [library/dotnet/SpringCardPCSC.cs](#) .

The documentation is available online at

http://www.springcard.com/support/apidoc/pcsc_dotnet_nfcforum/index.html

For convenience, the SDK archive contains a copy of this documentation under [docs/pcsc_dotnet_nfcforum](#), but pay attention that the documentation is regularly improved, so the copy may be outdated. Online documentation shall always be taken as reference.

- **NfcTag**, with 2 derivatives: `NfcTagType2` and `NfcTagType4`,
- **Ndef**, and a derivative for every supported type of message or record,
- **Llcp**: minimalistic implementation of LLCP,
- **Snep**: minimalistic implementation of SNEP.

3.2. DISCOVERING AND READING A TAG

To be written.

3.3. MANIPULATING THE NDEF OBJECTS

To be written.

3.4. (RE)WRITING A TAG

To be written.

3.5. TAG EMULATION

To be written.

3.6. IMPLEMENTING SNEP IN INITIATOR MODE

To be written.

DISCLAIMER

This document is provided for informational purposes only and shall not be construed as a commercial offer, a license, an advisory, fiduciary or professional relationship between PRO ACTIVE and you. No information provided in this document shall be considered a substitute for your independent investigation.

The information provided in document may be related to products or services that are not available in your country.

This document is provided "as is" and without warranty of any kind to the extent allowed by the applicable law. While PRO ACTIVE will use reasonable efforts to provide reliable information, we don't warrant that this document is free of inaccuracies, errors and/or omissions, or that its content is appropriate for your particular use or up to date. PRO ACTIVE reserves the right to change the information at any time without notice.

PRO ACTIVE doesn't warrant any results derived from the use of the products described in this document. PRO ACTIVE will not be liable for any indirect, consequential or incidental damages, including but not limited to lost profits or revenues, business interruption, loss of data arising out of or in connection with the use, inability to use or reliance on any product (either hardware or software) described in this document.

These products are not designed for use in life support appliances, devices, or systems where malfunction of these product may result in personal injury. PRO ACTIVE customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify PRO ACTIVE for any damages resulting from such improper use or sale.

COPYRIGHT NOTICE

All information in this document is either public information or is the intellectual property of PRO ACTIVE and/or its suppliers or partners.

You are free to view and print this document for your own use only. Those rights granted to you constitute a license and not a transfer of title : you may not remove this copyright notice nor the proprietary notices contained in this documents, and you are not allowed to publish or reproduce this document, either on the web or by any mean, without written permission of PRO ACTIVE.

Copyright © PRO ACTIVE SAS 2012, all rights reserved.

EDITOR'S INFORMATION

PRO ACTIVE SAS company with a capital of 227 000 €

RCS EVRY B 429 665 482

Parc Gutenberg, 13 voie La Cardon

91120 Palaiseau – FRANCE

CONTACT INFORMATION

For more information and to locate our sales office or distributor in your country or area, please visit

www.springcard.com