# Introduction to NFC

NOKIA Developer

Document created on 8 July 2011

Version 1.1

# Table of contents

# Change history

| | | |
|---|---|---|
| 19 April 2011 | 1.0 | Initial document release |
| 8 July 2011 | 1.1 | Added information on NFC Standards and tags supported in Symbian and MeeGo 1.2 Harmattan platforms. |

# 1. Introduction

This document introduces Near Field Communication (NFC) technology in general and the specific features that are supported on Nokia phones.

The document is intended for developers who are new to NFC and are looking to develop applications for Nokia phones using the Qt/Symbian/Java™ APIs.

# 2. What is NFC?

NFC stands for Near Field Communication. It is a short-range radio technology that enables communication between devices that either touch or are momentarily held close together.

- NFC is an open-platform technology that is being standardised in the NFC Forum.
- NFC is based on and extends on RFID. It operates on 13.56 MHz frequency.
- The NFC communication range is up to 10 cm. However, for the best user experience with Nokia phones it is recommended that the devices touch each other.
- The NFC standard supports different data transmission rates such as 106 kBps, 212 kBps, and 424 kBps.

## 2.1 Tag and reader

NFC-based communication between two devices is possible when one device acts as a reader/writer and the other as a tag.

### 2.1.1 Tag

The tag is a simple, thin device containing an antenna and a small amount of memory. It is a passive device, powered by a magnetic field. Depending on the tag type, the memory can be read only, rewritable, or writable once.

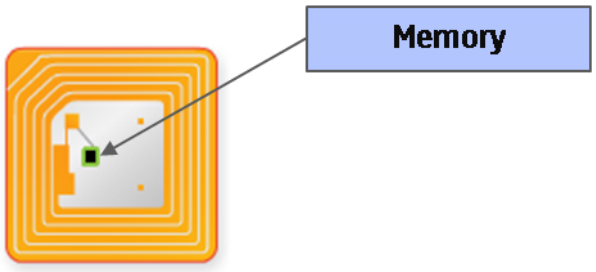


Figure 1: Tag

### 2.1.2 Reader

The reader is an active device that generates radio signals to communicate with the tags. The reader powers the passive device when the two are engaged in the passive mode of communication.


Figure 2: Reader

# 3. NFC modes

## 3.1 Communication modes

NFC devices support two communication modes.

### 3.1.1 Active

In this mode, the target and the initiator devices have power supplies and can communicate with one another by alternate signal transmission.

### 3.1.2 Passive

In this mode, the initiator device generates radio signals and the target device is powered by this electromagnetic field. The target device responds to the initiator by modulating the existing electromagnetic field.

## 3.2 Operating modes

NFC devices can operate in three different modes based on the ISO/IEC 18092, NFC IP-1, and ISO/IEC 14443 contactless smart card standards.

### 3.2.1 Read/write

In this mode, the NFC-enabled phone can read or write data to any of the supported tag types in a standard NFC data format.

### 3.2.2 Peer to peer

In this mode, two NFC-enabled devices can exchange data. For example, you can share Bluetooth or Wi-Fi link setup parameters to initiate a Bluetooth or Wi-Fi link. You can also exchange data such as virtual business cards or digital photos. Peer-to-peer mode is standardised on the ISO/IEC 18092 standard.

**Note:** The Symbian implementation for NFC currently supports initiation of a Bluetooth link, while Wi-Fi is not supported yet. Also, the Java implementation for NFC does not support this mode of operation.

### 3.2.3 Card emulation

An NFC-enabled phone acts as a reader when in contact with tags. In this mode, the phone can act as a tag or contactless card for existing readers.

**Note:** The Symbian and Java implementations for NFC do not currently support this mode of operation.


Figure 3: Card emulation

# 4. NFC use cases

This section discusses the use cases that NFC offers and describes some scenarios.

## 4.1 Service initiation

In the service initiation scenario, an NFC-enabled Nokia phone tapped against an NFC tag can receive information such as a URL, a phone number, or text, based upon which an action can be defined.

A smart poster serves as an example of this, where NFC tags have been placed on the backside of the poster under each hot spot. Each tag contains a piece of data in the NFC data format known as Smart Poster. When an NFC-enabled Nokia phone touches a tag (or the hot spot) in the poster, it reads the information in the tag. Depending on the type of information, the Nokia phone may start a video stream, open a web browser, or ask the user if he or she wants to place a telephone call.


Figure 4: Smart poster

## 4.2 Sharing

NFC-enabled devices can share data through a peer-to-peer communication mode.

For example, two NFC-enabled Nokia phones can share business card information. When a user chooses to share contact information, the phone sends a radio signal to the other phone in the NFC radio range and initiates peer-to-peer communication. The information is shared in the NFC data format recognised by both devices.

**Note:** Sharing is not done over NFC. Rather, NFC is used to initiate Bluetooth service and sharing is done via Bluetooth.

## 4.3 Connecting devices

NFC helps connect devices easily without the need for any configuration or setup. For example, a user can pair an NFC-enabled Nokia phone with a Bluetooth headset by touching the headset with the phone. The Bluetooth headset contains an embedded NFC tag that contains all the information required for Bluetooth pairing, stored in the NFC data format. The NFC-enabled Nokia phone acts as a reader; it reads Bluetooth information from the tag (that is, from the headset) and performs the Bluetooth pairing.

## 4.4 Ticketing

An NFC-enabled phone can be used in a ticketing application as a travel ticket, replacing a physical travel ticket. To present the ticket, the user must touch the ticket reader with the phone. Users can refill tickets by paying and touching the ticketing machine. The NFC-enabled phone acts as a contactless card to the ticket reader and so operates in card emulation mode.

To use an NFC-enabled phone as a travel ticket, a secure environment known as secure element (SE) must be implemented on the phone. The secure environment runs the ticketing application, which may, for example, keep track of remaining travel time or prepaid journeys.

**Note:** The ticketing use case is not yet supported in Nokia phones.

## 4.5 Payment

An NFC-enabled phone can be used in a payment application like a credit card to make payments. To pay, the user must touch the payment terminal with the phone. The NFC-enabled phone acts as a contactless card (chip-based credit card) to the payment terminal (reader) and so operates in card emulation mode.

The NFC-enabled phone must implement the secure element, which runs payment applications similar to the applications on normal chip-based credit cards.

**Note:** The payment use case is not yet supported in Nokia phones.

# 5. NFC tags

NFC tags can be used in applications such as posters, where small amounts of data can be stored and transferred to active NFC devices. The stored data on the NFC tag may contain any form of data such as a URL, a phone number, or calendar information.

To ensure interoperability between different NFC tag providers and NFC device manufacturers, the NFC Forum has defined four tag types.

For information on tags that are supported in Symbian and MeeGo 1.2 Harmattan platforms, refer to Section 8.5, 'NFC standards and tags supported'.

## 5.1 NFC Forum tag types

The NFC Forum is a consortium that was formed to advance the use of NFC technology by developing specifications, ensuring interoperability among devices and services, and educating the market about NFC technology.

For operation specifications of the NFC Forum tag types, see the technical specifications at the NFC Forum website (http://www.nfc-forum.org/specs/spec_list/). The operation specifications for the tags provide the technical information required for implementing the reader/writer and the associated control functionality of the NFC device to interact with the tags.

### 5.1.1 Type 1 Tag

Type 1 Tags are cost effective and ideal for many NFC applications.

- Based on ISO-14443A standard
- Read and rewrite capable; also users can configure the tag to be read-only
- 96 bytes of memory, expandable up to 2 kB
- Communication speed 106 kbits/s
- No data collision protection

– Compatible products available in the market — Innovision Topaz, Broadcom BCM20203

For information about how the NFC Forum device should operate in the NFC Forum Type 1 Tag platform and how the NFC Forum Device detects, reads, and writes NDEF (see Section 7.1, 'NDEF') data into the Type 1 Tag platform, see the Type 1 Tag Operation Specification at the NFC Forum website: http://www.nfc-forum.org/specs/spec_list/.

### 5.1.2 Type 2 Tag

Type 2 Tags are similar to Type 1 Tags and are derived from the NXP/Philips MIFARE Ultralight tag.

– Based on ISO-14443A standard
– Read and rewrite capable; also users can configure the tag to be read-only
– 96 bytes of memory, expandable up to 2 kB
– Communication speed 106 kbits/s
– Anticollision support
– Compatible products available in the market — NXP MIFARE Ultralight

### 5.1.3 Type 3 Tag

Type 3 Tags are derived from the nonsecure parts of Sony FeliCa tags. These tags are costlier than Types 1 and 2.

– Based on the Japanese Industrial Standard (JIS) X 6319-4
– Preconfigured at manufacture to be either read and rewritable, or read-only
– Variable memory, up to 1 MB per service
– Supports two communication speeds: 212 or 424 kbits/s
– Anticollision support
– Compatible products available in the market — Sony FeliCa

### 5.1.4 Type 4 Tag

Type 4 Tags are similar to Type 1 Tags and are derived from the NXP DESFire tag.

- Based on ISO-14443A standard
- Preconfigured at manufacture to be either read and rewritable, or read-only
- Variable memory, up to 32 kB per service
- Supports three different communication speeds: 106, 212, or 424 kbits/s
- Anticollision support
- Compatible products available in the market — NXP DESFire, SmartMX-JCOP

## 5.2 NXP-specific tag type

This is a proprietary tag type defined by NXP Semiconductors.

### 5.2.1 Type MIFARE Classic Tag

- Based on ISO-14443A standard
- Read and rewrite capable; also users can configure the tag to be read-only
- Variable memory 192/768/3584 bytes
- Communication speed 106 kbits/s
- Anticollision support
- Compatible products available in the market — NXP MIFARE Classic 1K, MIFARE Classic 4K, and Classic Mini

For more information on the MIFARE Classic Tag, see the NXP website: http://www.nxp.com/.

# 6. NFC-related specifications

For information on standards that are supported in Symbian and MeeGo 1.2 Harmattan platforms, see Section 8.5, 'NFC standards and tags supported'.

## 6.1 ISO 14443

ISO 14443 is a well-known international standard originally developed for contactless chip card communication over a 13.56 MHz radio.

ISO 14443 defines a protocol stack from the radio layer up to a command protocol.

There are two versions of the radio layer ISO 14443-2, with different modulation and bit-encoding methods. These versions are known as the –A and –B versions of the ISO 14443. Similarly, ISO 14443 specifies two versions of the packet framing and low-level protocol part (ISO 14443-3). The topmost layer of the ISO protocol stack defines a command interface (ISO 14443-4) for transferring information.


Figure 5: ISO 14443 protocol stack

## 6.2 NFCIP-1

Peer-to-peer communication between two NFC devices is made possible by mechanisms defined in the Near Field Communication — Interface and Protocol Specification, NFCIP-1. This key NFC specification is also known as ISO 18092 and ECMA-340.

The protocol stack in NFCIP-1 is based on ISO 14443. The main difference is a new command protocol, which replaces the topmost part of the stack.

NFCIP-1 includes two communication modes that allow an NFC device to communicate with other NFC devices in a peer-to-peer manner, as well as with NFCIP-1 based NFC tags.


Figure 6: ISO 14443 protocol stack

## 6.3 MIFARE

MIFARE refers to an NFC tag type developed by NXP Semiconductors. MIFARE tags are widely used as memory cards in transportation applications. ISO 14443 defines a protocol stack from the radio layer up to a command protocol.

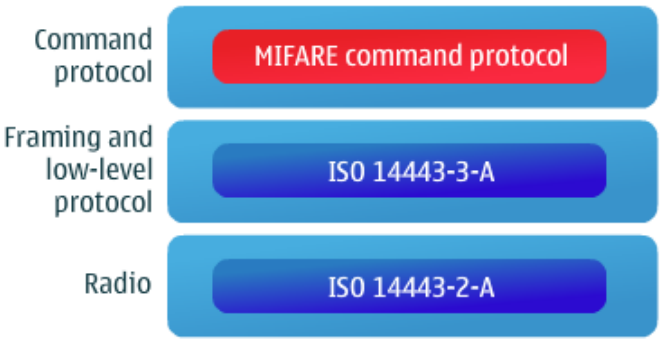


Figure 7: Protocol stack

## 6.4 FeliCa

FeliCa is a proprietary NFC tag technology developed by Sony, and it is widely used in proprietary payment and transportation applications in the Asian markets. FeliCa tags have also been integrated with select mobile phone models in the Mobile FeliCa system. FeliCa tags are standardised as a Japanese industry standard. The tags are based on the passive mode of ISO 18092, with added authentication and encryption capabilities.

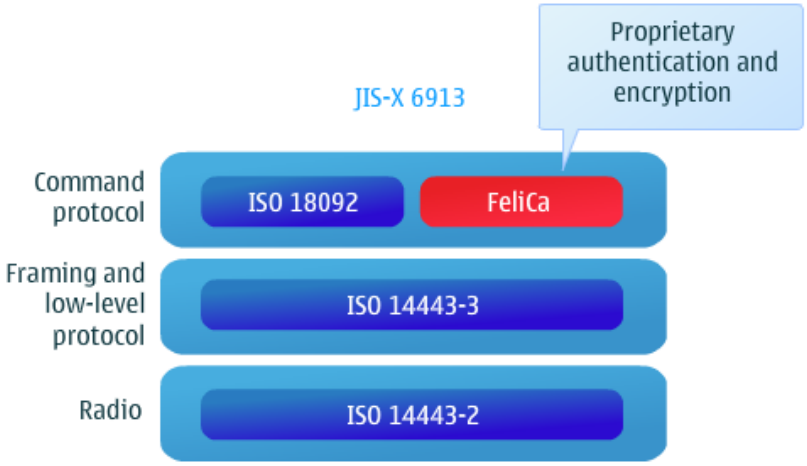


Figure 8: Protocol stack

# 7. NFC Forum standards

## 7.1 NDEF

To enable interoperability when transferring data to and from tags and between NFC devices, the NFC Forum has specified a common data format known as the NFC Data Exchange Format (NDEF).

NDEF is a lightweight and compact binary format that can carry URLs, vCards, and NFC-specific data types.

NDEF allows NFC functionality to easily use any supported tag type to transfer data, because NDEF hides all the tag type-specific details from the application.

NDEF is exchanged in messages that consist of a sequence of records. Each record carries a payload. The payload contents can be of type URL, MIME media, or an NFC-specific data type. For NFC-specific data types, the payload contents must be defined in an NFC Record Type Definition (RTD) file.

The type of data in the record and the size of the record are indicated in a header attached to the payload.

The header includes a type field for identifying the type of payload. The payload length indicates the number of octets in the payload. The optional payload identifier allows user applications to identify the payload carried within an NDEF record.

The format of the TYPE field value is indicated using the Type Name Format (TNF) field. For information on supported types and corresponding TNF values, see Section 3.2.6 of the NDEF technical specification on the NFC Forum website (http://www.nfc-forum.org/specs/spec_list/).

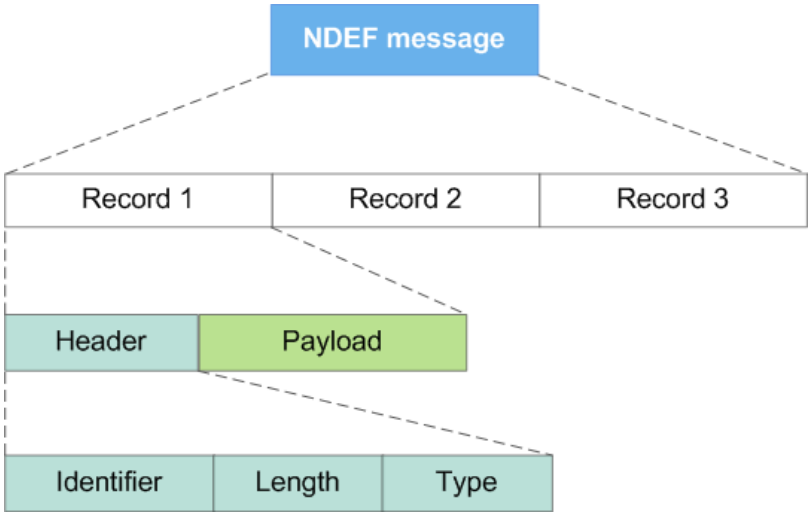


Figure 9: NDEF message

## 7.2 RTD

The NFC Forum has specified several optimised record types that can be carried in NDEF records. Each NFC Forum record type is specified in a Record Type Definition (RTD) document. NFC defines the following RTDs:

- NFC Text RTD
- NFC URI RTD
- NFC Smart Poster RTD
- NFC Generic Control RTD
- NFC Signature RTD

The simplest is a Text record type, which can carry a Unicode string. A text record can be included in an NDEF message as a descriptive text for another record. The URI record type can be used to store a URI such as a web address, an email, or a phone number in an optimised binary form.

The Smart Poster RTD defines how to put URLs, SMSs, or phone numbers on an NFC Forum tag and how to transport them between devices.

## 7.3 LLCP

To enhance the peer-to-peer mode of operation, the NFC Forum has specified the link-level protocol known as the Logical Link Control Protocol (LLCP). In the NFC peer-to-peer protocol stack, LLCP provides additional communication capabilities on top of the NFCIP-1/ ISO 18092.

LLCP introduces a two-way, link-level connection, allowing both peers to send and receive data using the following methods of data exchange:

- **Connection-oriented transfer**, where data exchanges are acknowledged;
- **Connectionless transfer**, where data exchanges are unacknowledged.

# 8. NFC in Nokia phones

Qt, Symbian, and Java™ technologies provide a choice of APIs for NFC support.

## 8.1 Qt APIs

The Qt NFC API is included as a cross-platform API in the Qt Mobility project. This API is part of the connectivity API and integrates into the Qt SDK.

**Note:** The Qt NFC API is currently available only at a Technology Preview level of maturity.

For information on Qt Mobility APIs for NFC, go to http://doc.qt.nokia.com/qtmobility-1.2.0-tp1/connectivity-api.html.

For a source package of the Qt Mobility 1.2 APIs at a Technology Preview level of maturity, go to http://labs.qt.nokia.com/2010/12/24/qt-mobility-1-2-technology-preview/.

### 8.1.1 Use cases

The following use cases are supported by Qt NFC APIs:

- Interacting with NFC Forum tags and NFC Forum devices
- Target detection and loss
- Registering NDEF message handlers
- Reading and writing NDEF messages to NFC Forum tags
- Sending tag-specific commands
- Client and server LLCP sockets for peer-to-peer communication

### 8.1.2 Classes

| API | Description |
|---|---|
| QLlcpServer | NFC LLCP socket-based server |
| QLlcpSocket | NFC LLCP socket |
| QNdefFilter | Filter for matching NDEF messages |
| QNdefMessage | NFC NDEF message |
| QNdefNfcTextRecord | NFC RTD-Text |
| QNdefNfcUriRecord | NFC RTD-URI |
| QNdefRecord | NFC NDEF record |
| QNearFieldManager | Access to notifications for NFC events |
| QNearFieldTagType1 | Interface for communicating with NFC Tag Type 1 tag |
| QNearFieldTagType2 | Interface for communicating with NFC Tag Type 2 tag |
| QNearFieldTagType3 | Interface for communicating with NFC Tag Type 3 tag |
| QNearFieldTagType4 | Interface for communicating with NFC Tag Type 4 tag |
| QNearFieldTarget | Interface for communicating with a target device |

### 8.1.3 Examples

– NFC Annotated URL: Displays URL along with caption text and an image from an NFC tag containing a specially formatted NDEF message.

– NDEF Editor: Creates NDEF messages using the NDEF records supported by the Connectivity API and saves them to supported NFC Forum tags.

## 8.2 Symbian APIs

In Symbian S^3, the following native APIs are provided for NFC:

| API | Description |
|---|---|
| NFC Discovery API | Provides methods to subscribe and discover tags and NDEF messages |
| NFC Tag Extension API | Provides implementation and access (read/write) methods for different types of NFC targets |
| NDEF API | Provides methods for handling NDEF messages and accessing NDEF messages on tags |
| NFC Peer-to-Peer API | Provides LLCP API for peer-to-peer communication between NFC devices |
| NFC Content Handler API | Provides ECOM plug-in interface for subscribing and receiving NDEF messages |

Table 1: Native APIs in Symbian S^3

### 8.2.1 Use cases

The following use cases are supported by Symbian APIs:

- Creating content handler plug-ins
- Setting up a Bluetooth connection
- Sharing a file or data
- Reading NDEF messages
- Discovering NFC tags
- Exchanging data with NFC Forum Type 4 Tags
- Reading NFC Forum Type 1, 2, 3 Tags
- Writing to NFC Forum Type 1, 2, 3 Tags
- Transferring and receiving "Hello World!" ASCII text using the LLCP stack (connectionless)
- Transferring and receiving "Hello World!" ASCII text using the LLCP stack (connection-oriented)

**Note:** The Symbian implementation for NFC does not support the Card emulation mode of operation.

### 8.2.2 Examples

– Sharing using the NFC stack and AIW framework APIs: Demonstrates the usage of Tag Discovery and NDEF Discovery APIs and various read\write operations that can be performed with the tags.

– Tag discovery and NDEF discovery: Demonstrates how two devices get paired using the NFC channel and share a file using the secondary bearer channel (for example, Bluetooth). Using the AIW framework, this example shows how to share a vCard or vCal from one device to another.

See the Nokia Symbian^3 Developer's Library at Nokia Developer for documentation about Symbian API use cases and to download the examples:

http://library.developer.nokia.com/topic/GUID-E35887BB-7E58-438C-AA27-97B2CDE7E069/GUID-7A72B008-901E-454D-AD93-F99555CA904A.html#GUID-7A72B008-901E-454D-AD93-F99555CA904A.

## 8.3 Java APIs

The JSR 257 Contactless Communication API 1.0 is supported on Symbian devices with NFC capabilities.

The Contactless Communication API allows MIDlets to use the NFC capabilities of a mobile device to communicate with contactless targets. Specifically, MIDlets can read and write small amounts of data to the tags (integrated circuits) in the contactless targets.

### 8.3.1 Use cases

The following use cases are supported by the Contactless Communication API:

– Detecting contactless targets

– Exchanging data with detected contactless targets

**Note:** The Java implementation for NFC does not support the Card emulation mode or Peer to peer mode of operation.

### 8.3.2 Examples

– UIDReader: Shows how to create a MIDlet that reads data from contactless targets.

– URLWriter: Shows how to create a MIDlet that writes data to contactless targets.

See the Java Developer's Library at Nokia Developer for documentation about Java API use cases and to download the examples:

http://library.developer.nokia.com/index.jsp?topic=/Java_Developers_Library/GUID-9248DD7F-23BE-40A6-A8DC-823783443F10.html.

## 8.4 Nokia phones with NFC support

For information about Nokia phones that provide support for NFC, go to: http://www.developer.nokia.com/Devices/Device_specifications/?filter1=all&filter2=nfc.

## 8.5 NFC standards and tags supported

The following NFC standards and tags are supported in both Symbian and MeeGo 1.2 Harmattan platforms:

- NFC A technology
- NFC B technology
- NFC F technology (JIS 6319-A)
- ISO DEP protocol (ISO 14443)
- NFC DEP protocol (NFCIP1 - ISO 18092)

The following NFC Forum tag types are supported:

- **NFC Forum Type 1**
- **NFC Forum Type 2**
- **NFC Forum Type 3**
- **NFC Forum Type 4**

**Note**: The Java implementation for NFC does not support **NFC Forum Type 4** Tags.

Other proprietary tag types supported:

- **MIFARE Classic**

## 8.6 NFC data handling in Nokia phones

In the Symbian implementation, applications in the background are not allowed to handle Reserved RTDs such as Smart Poster, URI, BT headset, and other common MIME types.

When a tag containing a reserved RTD is detected (see Section 8.6.2, 'Reserved RTDs supported'), the application in the UI foreground can only get a callback. Similarly, sharing (or easy setup) can be initiated only when the application is in the foreground.

However, users can define their own data types and in which case the applications can handle the user-defined data types from both foreground and background.

### 8.6.1 NDEF record handling

The Symbian implementation supports handling of multiple NDEF records. In case of multiple NDEF records, it reads the first records and checks if there is a handler available (either an NFC Discovery API or an NFC Content Handler API client) to handle the data according to record type. If YES, it will stop and give the whole NDEF message to that particular handler. If a handler for the first record is not found, it will move to the second record and repeat the process, and so on. In other words, if for a record any handle is found, then the whole NDEF message is given to that particular handler.

Note: The Symbian implementation supports multiple NDEF records but not multiple NDEF messages.

### 8.6.2 Reserved RTDs supported

The NFC implementation in the Symbian platform provides support for handling the following RTDs:

| **RTD** | **Identifier** |
|---|---|
| Bluetooth 2.0 | urn:nfc:ext:nokia.com:bt |
| Bluetooth 2.1 | application/vnd.bluetooth.ep.oob |
| Handover Request | urn:nfc:wkt:Hr |
| Handover Select | urn:nfc:wkt:Hs |
| Smart Poster | urn:nfc:wkt:Sp |
| URI record | urn:nfc:wkt:U |

Table 2: RTDs supported in Symbian platform

Applications must be in the foreground to handle Reserved RTDs. It is not possible to read a Reserved NDEF from a background process. For more information on NFC data handling, see Section 8.6, 'NFC data handling in Nokia phones'.

## 8.6.3 MIME types supported

The Symbian implementation can read the following file types from the NFC tags:

| Type | Subtype |
|---|---|
| image | All subtypes are supported |
| audio | All subtypes are supported |
| text | All subtypes are supported |
| video | All subtypes are supported |
| application | vnd.Nokia.ringing-tone |
| application | vnd.symbian.install |
| application | java-archive |
| application | x-java-archive |
| application | vnd.wap |

Table 3: MIME types supported in Symbian platform

## 8.6.4 Smart Poster RTD handling

The Smart Poster message can have one or more records. The NFC implementation in the Symbian platform supports the URI record (mandatory record), and the optional records **Action** and **Title**. For information on records supported in the Smart Poster RTD, see the NFC Forum Technical Specifications (http://www.nfc-forum.org/specs/spec_list/#rtds).

The **Action** record specifies the action the device must perform with the content.

| Value | Action |
|---|---|
| 0 | Specifies that the action is to be done immediately |
| 1 | Specifies that the content in URL and SMS cases is to be saved for later use |

Table 4: Action record values

- The content of the **Title** record is used to describe the action specified in the **Action** record.
- The mandatory **URI** record contains a URI Identifier Code for specifying the URI scheme and shortening the actual URI field.

## 8.6.5 Smart Poster RTD/URI stand-alone use cases

The following use cases are supported in the Symbian platform:

### 8.6.5.1 Open URL

| URI identifier codes | Schemes |
|---|---|
| 0x01 | http://www. |
| 0x02 | https://www. |
| 0x03 | http:// |
| 0x04 | https:// |

### 8.6.5.2 Make a phone call

| URI identifier codes | Schemes |
|---|---|
| 0x01 | http://www. |
| 0x05 | tel: |

### 8.6.5.3 Send SMS

| URI identifier codes | Schemes |
|---|---|
| 0x00 | sms: |

### 8.6.5.4 Open a local file or an application

| URI identifier codes | Schemes |
|---|---|
| 0x1D | file:// |

# 8.7 API feature comparisons

For information on standards and tags that are supported in the Symbian and MeeGo 1.2 Harmattan platforms, see Section 8.5, 'NFC standards and tags supported'.

| Core NFC API capabilities | | JSR-257 | JSR-257 Nokia extensions | Symbian Java support | Symbian Native API | Qt Mobility 1.2 API |
|---|---|---|---|---|---|---|
| Register application for launch upon... | Detecting a specified RTD Type in a tag | ✓ | N/A | ✓ | ✓ | ✓ |
| | Connection to a specified LLCP SAP/ Service name | × | × | × | ✓ | ✓ |
| | | | | | | |
| Event notifications | NDEF Tag detected | ✓ | N/A | ✓ | ✓ | ✓ |
| | Specified RTD Type detected | ✓ | N/A | ✓ | ✓ | ✓ |
| | Target detected | ✓ | N/A | ✓ | ✓ | ✓ |
| | | | | | | |
| NDEF reading | Provide NDEF and RTD concepts | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Read NDEF record(s) | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Write NDEF record(s) | ✓ | ✓ | ✓ | ✓ | ✓ |
| | | | | | | |
| LLCP | Open connection | × | × | × | ✓ | ✓ |
| | Send data | × | × | × | ✓ | ✓ |
| | Receive data | × | × | × | ✓ | ✓ |
| | | | | | | |
| Tag Type support | Tag Type 1 specific commands | × | ✓ | ✓ | ✓ | ✓ |
| | Tag Type 2 specific commands | × | ✓ | ✓ | ✓ | ✓ |
| | Tag Type 3 specific commands | × | ✓ | ✓ | ✓ | ✓ |
| | Tag Type 4 specific commands | × | ✓ | × | ✓ | ✓ |
| | MIFARE Std | × | ✓ | ✓ | ✓ | × |

Table 5: API feature comparisons

| Legend | |
|---|---|
| ✓ | Supported |
| × | Not Supported |
| N/A | Not Applicable |

# 9. Terms and abbreviations

This section defines all relevant terms and acronyms used in this document.

| Term or abbreviation | Meaning |
|---|---|
| ECMA | European Computer Manufacturers' Association |
| ISO | International Organization for Standardization |
| LLCP | Logical Link Control Protocol |
| NDEF | NFC Data Exchange Format |
| NFC | Near Field Communications |
| RFID | Radio Frequency Identification |
| RTD | Record Type Definition |
| URL | Uniform Resource Locator |

# 10. References

[1] Technical Specifications at NFC Forum, http://www.nfc-forum.org/specs/

[2] NXP Specific Type Tag details from NXP Semiconductors, http://www.nxp.com/