

# Introduction to NFC

19 April 2011

Version 1.0

# Table of contents

<b>1.</b>	<b>Introduction</b>	<b>4</b>
<b>2.</b>	<b>What is NFC</b>	<b>5</b>
2.1	Tag and reader	5
2.1.1	Tag	5
2.1.2	Reader	6
<b>3.</b>	<b>NFC modes</b>	<b>7</b>
3.1	Communication modes	7
3.1.1	Active	7
3.1.2	Passive	7
3.2	Operating modes	7
3.2.1	Read/Write	7
3.2.2	Peer to Peer	7
3.2.3	Card emulation	7
<b>4.</b>	<b>NFC use cases</b>	<b>9</b>
4.1	Service initiation	9
4.2	Sharing	9
4.3	Connecting devices	10
4.4	Ticketing	10
4.5	Payment	10
<b>5.</b>	<b>NFC tags</b>	<b>11</b>
5.1	NFC Forum tag types	11
5.1.1	Tag 1 type	11
5.1.2	Tag 2 type	12
5.1.3	Tag 3 type	12
5.1.4	Tag 4 type	12
5.2	NXP-specific type tag	13
5.2.1	Type MIFARE Classic Tag	13
<b>6.</b>	<b>NFC-related specifications</b>	<b>14</b>
6.1	ISO 14443	14
6.2	NFCIP-1	14
6.3	MIFARE	15
6.4	FeliCa	16
<b>7.</b>	<b>NFC Forum standards</b>	<b>17</b>
7.1	NDEF	17
7.2	RTD	18
7.3	LLCP	18
<b>8.</b>	<b>NFC in Nokia devices</b>	<b>19</b>
8.1	Qt APIs	19
8.1.1	Use cases	19
8.1.2	Classes	19
8.1.3	Examples	20
8.2	Symbian APIs	21
8.2.1	Use cases	21
8.2.2	Examples	22

8.3	Java APIs	22
8.3.1	Use cases	22
8.3.2	Examples	22
8.4	Nokia phones with NFC support	23
8.5	Tags supported	23
8.6	NFC data handling in Nokia devices	23
8.6.1	NDEF record handling	23
8.6.2	Reserved RTDs supported	24
8.6.3	MIME types supported	25
8.6.4	Smart Poster RTD handling	25
8.6.5	Smart Poster RTD / URI stand-alone use cases	26
8.7	API feature comparison	27
<b>9.</b>	<b>Terms and abbreviations</b>	<b>28</b>
<b>10.</b>	<b>References</b>	<b>29</b>

## Change history

19 April 2011	1.0	Initial document release
---------------	-----	--------------------------

# 1. Introduction

**This document introduces the NFC technology in general and the specifics that are supported on Nokia devices.**

**This document is intended for developers who are new to NFC and are planning to develop applications for Nokia devices 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, which is being standardized in the NFC Forum.
- NFC is based on and extends on RFID. It operates on 13.56 MHz frequency.
- NFC communication range is up to 10 cm. However, for best user experience with Nokia phones it is recommended that the devices are in touch with each other.
- NFC standard supports different data transmission rates such as 106kbps, 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 thin simple device containing antenna and small amount of memory. It is a passive device, powered by magnetic field. Depending on the tag type the memory can be read only, re-writable, and writable once.

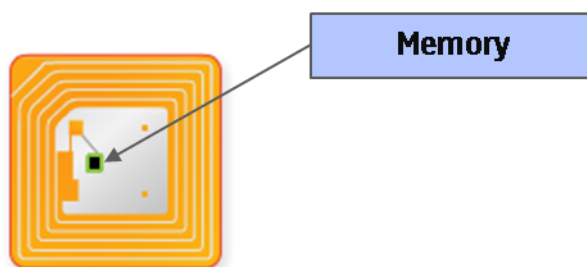


Figure 1: Tag

## 2.1.2 Reader

The reader is an active device, which generates radio signals to communicate with the tags. The reader powers the passive device in case of 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 gets 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 set up 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 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 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 implementation for NFC does not currently support this mode of operation.

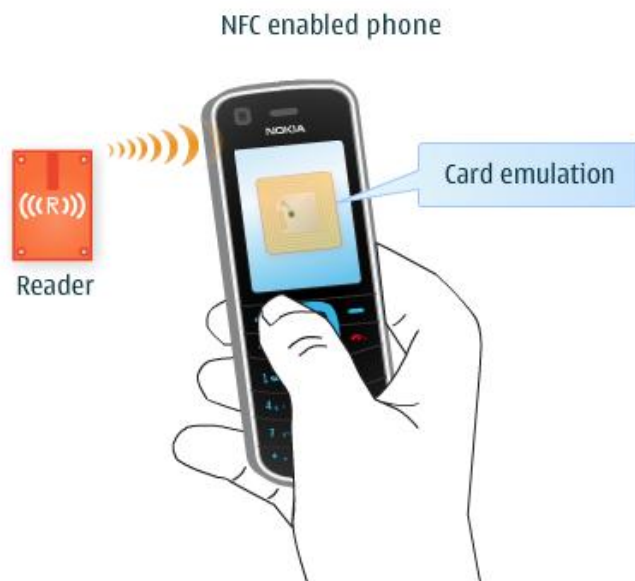


Figure 3: Card emulation



## 4. NFC use cases

This chapter discusses the use cases that NFC offers. Following are some of the scenarios for which NFC can be made use of:

### 4.1 Service initiation

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

Smart poster is an example of service initiation where NFC tags are placed in 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 hot spot) in the poster, information in the tag is read by the phone. Depending on the type of information read, the Nokia phone may start a video stream, open a web browser, or ask the user if he wants to place a telephone call.



Figure 4: Smart poster

### 4.2 Sharing

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

For example, two NFC-enabled Nokia phones can share business card information using NFC. When the user chooses to share the 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:** The sharing is not done over NFC. NFC is used to initiate the Bluetooth service and then the sharing is done over Bluetooth.

## 4.3 Connecting devices

NFC helps in connecting devices easily without the need of any configuration or setup. For example, the user can pair an NFC-enabled Nokia phone with a Bluetooth headset by touching the headset with the phone. The Bluetooth headset contains an NFC tag embedded in it. The tag contains all the information required for Bluetooth pairing stored in NFC data format. The NFC enabled Nokia phone acts as a reader and 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 the travel cards. To present the ticket, user must touch the ticket reader with the phone. The user can refill the 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 the card emulation mode.

To use the 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 the remaining travel time or prepaid journeys.

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

## 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 devices.

## 5. NFC tags

The NFC tags can be used within applications such as posters, where small amounts of data can be stored and transferred to active NFC devices. The data stored on the NFC tag may contain any form of data such as URL, 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.

### 5.1 NFC Forum tag types

The [NFC Forum](#) is a consortium that was formed to advance the use of Near Field Communication technology by developing specifications, ensuring interoperability among devices and services, and educating the market about NFC technology.

For operation specifications of the NFC Forum Type Tags refer to the technical specifications in the NFC Forum website ([http://www.nfc-forum.org/specs/spec\\_list/](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 Tag 1 type

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

- Based on ISO-14443A standard.
- Read and re-write capable, also users can configure the tag to be read-only.
- 96 bytes of memory, expandable up to 2KB.
- Communication speed 106 Kbits/s.
- No data collision protection.
- Compatible products available in the market- Innovision Topaz, Broadcom BCM20203.

For information on how the NFC Forum device should operate in NFC Forum Type 1 tag platform and how the NFC Forum Device detects, reads, and writes NDEF (see Section 7.1) data into the Type 1 tag platform, refer to the Type 1 Tag Operation Specification available at the NFC Forum website ([http://www.nfc-forum.org/specs/spec\\_list/](http://www.nfc-forum.org/specs/spec_list/)).

## 5.1.2 Tag 2 type

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

- Based on ISO-14443A standard.
- Read and re-write capable, also users can configure the tag to be read-only.
- 96 bytes of memory, expandable up to 2KB.
- Communication speed 106 Kbits/s.
- Anti-collision support.
- Compatible products available in the market - NXP MIFARE Ultralight.

## 5.1.3 Tag 3 type

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

- Based on the Japanese Industrial Standard (JIS) X 6319-4.
- Pre-configured at manufacture to be either read and re-writable, or read-only.
- Variable memory, up to 1 MB per service.
- Supports two communication speeds, 212 or 424 Kbits/s.
- Anti-collision support.
- Compatible products available in the market - Sony FeliCa.

## 5.1.4 Tag 4 type

Type 4 tags are similar to type 1 tags and are derived from NXP DesFire tag.

- Based on ISO-14443A standard.
- Pre-configured at manufacture to be either read and re-writable, or read-only.
- Variable memory, up to 32 KB per service.
- Supports three different communication speeds 106 or 212 or 424 Kbits/s.
- Anti-collision support.
- Compatible products available in the market - NXP DESFire, SmartMX-JCOP.

## 5.2 NXP-specific type tag

This is a proprietary tag type defined by NXP Semiconductors.

### 5.2.1 Type MIFARE Classic Tag

- Based on ISO-14443A standard.
- Read and re-write capable, also users can configure the tag to be read-only.
- Variable memory 192/768/3584 Bytes.
- Communication speed 106 Kbits/s.
- Anti-collision support.
- Compatible products available in the market - NXP MIFARE Classic 1k, MIFARE Classic 4K, and Classic Mini.

For more information on MIFARE Classic Tag, refer to the NXP website (<http://www.nxp.com/>).

## 6. NFC-related specifications

### 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

The 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, which 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 widely used in proprietary payment and transportation applications in Asian markets. FeliCa tags have also been integrated with selected 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.

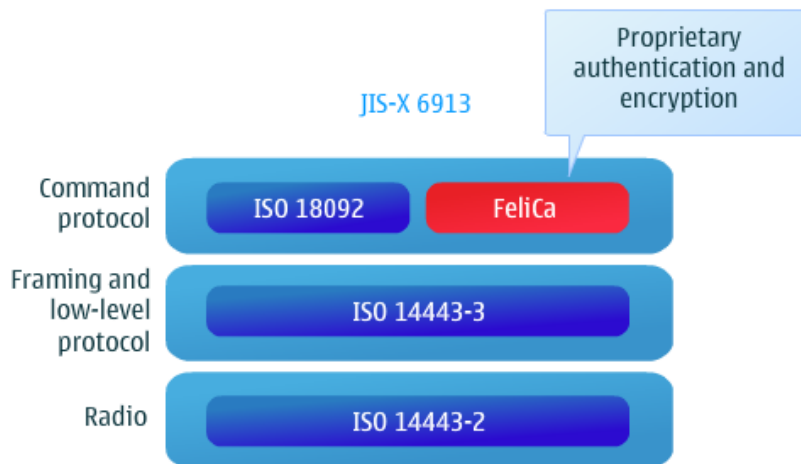


Figure 8: Protocol stack



# 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, which can carry URLs, vCard, and NFC-specific data types.

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

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

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 TNF (Type Name Format) field. For information on supported types and corresponding TNF values, refer to Section 3.2.6 of the NFC Data Exchange Format (NDEF) technical specification in the NFC Forum website ([http://www.nfc-forum.org/specs/spec\\_list/](http://www.nfc-forum.org/specs/spec_list/)).

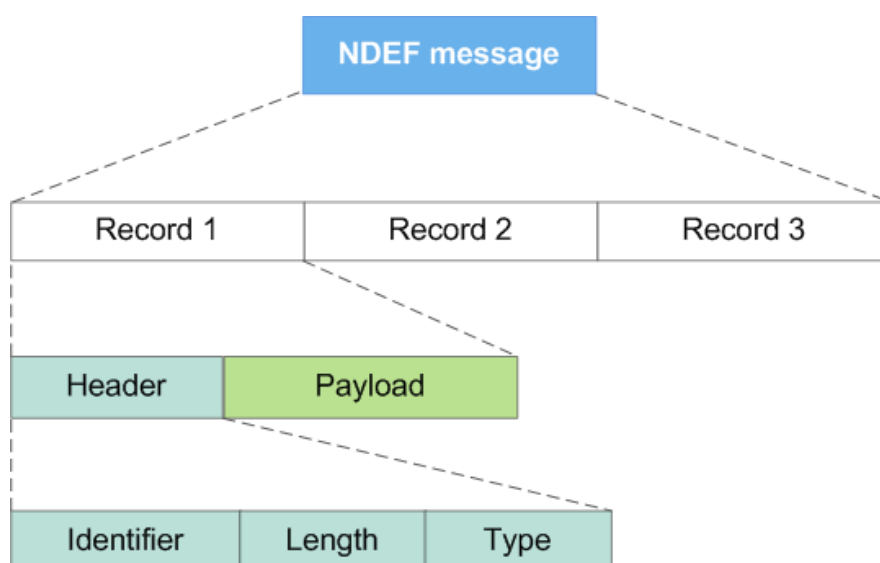


Figure 9: NDEF message

## 7.2 RTD

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

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

The simplest record type is the 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 address, 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 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 the data exchanges are acknowledged.
- **Connectionless transfer**, where the data exchanges are unacknowledged.

## 8. NFC in Nokia devices

The 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, refer to <http://doc.qt.nokia.com/qtmobility-1.2.0-tp1/connectivity-api.html>.

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

#### 8.1.1 Use cases

The following are the use cases 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

API	Description
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 an NFC Tag Type 1 tag
QNearFieldTagType2	Interface for communicating with an NFC Tag Type 2 tag
QNearFieldTagType3	Interface for communicating with an NFC Tag Type 3 tag
QNearFieldTagType4	Interface for communicating with an 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: Create NDEF messages using the NDEF records supported by the Connectivity API and save them to supported NFC Forum tags.

## 8.2 Symbian APIs

In Symbian^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^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 the "Hello World!" ASCII Text Using the LLCP Stack (Connectionless)
- Transferring and Receiving the "Hello World!" ASCII Text Using the LLCP Stack (Connection-Oriented)

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

## 8.2.2 Examples

- Sharing using NFC stack and the AIW framework APIs: Demonstrates the usage of Tag discovery and NDEF discovery APIs and various read\write operations, which 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 other device.

Refer to the Nokia Symbian^3 Developer's Library in Forum Nokia

([http://www.forum.nokia.com/Develop/Other Technologies/Symbian C++/Documentation/](http://www.forum.nokia.com/Develop/Other_Technologies/Symbian_C++/Documentation/)) for the documentation of the Symbian API use cases and for downloading the examples.

## 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 the 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 Card emulation mode and Peer to Peer mode of operation.

### 8.3.2 Examples

- UIDReader: This example shows you how to create a MIDlet that reads data from contactless targets.
- URLWriter: This example shows you how to create a MIDlet that writes data to contactless targets.

Refer to the Java™ Developer's Library in Forum Nokia

(<http://www.forum.nokia.com/Develop/Java/Documentation/>) for the documentation of the Java API use cases and for downloading the examples.

## 8.4 Nokia phones with NFC support

For information on Nokia devices that provide support for NFC refer to [http://www.forum.nokia.com/Devices/Device\\_specifications/?filter1=all&filter2=nfc](http://www.forum.nokia.com/Devices/Device_specifications/?filter1=all&filter2=nfc)

## 8.5 Tags supported

The Symbian platform supports following types of tags currently:

- [NFC Forum Type 1](#)
- [NFC Forum Type 2](#)
- [NFC Forum Type 3](#)
- [NFC Forum Type 4](#)
- [MIFARE Classic](#)

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

## 8.6 NFC data handling in Nokia devices

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

When a tag containing a reserved RTD (see Section 8.6.2) is detected, application in the UI foreground can only get a call-back. 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 (which could either be an NFC Discovery API or NFC Content Handler API client) available to handle the data according to record type. If YES, it will stop and give the whole

NDEF message to that particular handler. If the handler for the first record is not found, it will move to the second record and repeat the process and so on.

So 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 and not multiple NDEF messages.

## 8.6.2 Reserved RTDs supported

The NFC implementation in the Symbian platform provides support for handling the following Record Type Definitions (RTD):

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, refer to Section 8.6.



## 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 the 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 URI record (mandatory record), and the optional records **Action** and **Title**. For information on records supported in Smart Poster RTD, refer to the NFC Forum Technical Specifications ([http://www.nfc-forum.org/specs/spec\\_list/#rtds](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 an 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 is the list of use cases that are supported in the Symbian platform:

### 8.6.5.1 Open URL

URI Identifier Codes	Schemes
0x01	<a href="http://www">http://www.</a>
0x02	https://www.
0x03	http://
0x04	https://

### 8.6.5.2 Make a phone call

URI Identifier Codes	Schemes
0x01	<a href="http://www">http://www.</a>
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 comparison

Core NFC API capabilities		JSR-257	JSR-257 Nokia Extensions	Symbian Java Support	Symbian Native API	Qt Mobility 1.2 API
<b>Register application for launch upon</b>	Detecting a specified RTD Type in a tag	✓	N/A	✓	✓	✓
	Connection to a specified LLCP SAP / Service name	×	×	×	✓	✓
<b>Event notifications</b>	NDEF Tag detected	✓	N/A	✓	✓	✓
	Specified RTD Type detected	✓	N/A	✓	✓	✓
	Target Detected	✓	N/A	✓	✓	✓
<b>NDEF reading</b>	Provide NDEF and RTD concepts	✓	✓	✓	✓	✓
	Read NDEF Record(s)	✓	✓	✓	✓	✓
	Write NDEF Record(s)	✓	✓	✓	✓	✓
<b>LLCP</b>	Open connection	×	×	×	✓	✓
	Send Data	×	×	×	✓	✓
	Receive Data	×	×	×	✓	✓
<b>Tag Type support</b>	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 comparison

Legend	
✓	Supported
×	Not Supported
N/A	Not Applicable

## 9. Terms and abbreviations

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

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

# 10. References

- [1] Technical Specifications in NFC Forum, <http://www.nfc-forum.org/specs/>
- [2] NXP Specific Type Tag details from NXP Semiconductors, <http://www.nxp.com/>

Copyright © 2011 Nokia Corporation. All rights reserved.

Nokia and Forum Nokia are trademarks or registered trademarks of Nokia Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Bluetooth is a registered trademark of Bluetooth SIG, Inc. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

#### Disclaimer

The information in this document is provided 'as is', with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this document at any time, without notice.

#### Licence

A licence is hereby granted to download and print a copy of this document for personal use only. No other licence to any other intellectual property rights is granted herein.