Type 4 Tag Operation

Technical Specification

NFC Forum™

NFCForum-TS-Type-4-Tag_1.0

2007-03-13

RESTRICTIONS ON USE

Contents

1 Introduction....1
1.1 Objectives....1
1.2 Purpose....1
1.3 Applicable Documents or References....1
1.4 Administration....1
1.5 Special Word Usage....2
1.6 Name and Logo Usage....2
1.7 Intellectual Property....2
1.8 Glossary....2
1.9 Convention and notations....4
1.9.1 Representation of numbers....4
2 Management of the Memory Structure....5
3 RF interface....6
4 Framing / Transmission Handling....7
5 Command Set....8
5.1 Activation of the Transmission Protocol....8
5.2 High Level Command Set....8
5.2.1 Format of C-APDU....8
5.2.2 Format of R-APDU....9
6 NDEF Detection and Access....10
6.1 NDEF Management....10
6.1.1 Version Treating....11
6.1.2 TLV blocks....12
6.2 NDEF Storage....15
6.3 Life Cycle....15
6.3.1 INITIALISED State....16
6.3.2 READ/WRITE State....16
6.3.3 READ-ONLY State....16
6.4 Command Sequence Description....16
6.4.1 NDEF Detection Procedure....16
6.4.2 NDEF Tag Application Select Procedure....17
6.4.3 Capability Container Select Procedure....18
6.4.4 Capability Container Read Procedure....19
6.4.5 NDEF Select Procedure....20
6.4.6 NDEF Read Procedure....21
6.4.7 NDEF Update Procedure....22
6.4.8 State Changes....23

Figures

Figure 1: Length Field Formats....12
Figure 2: Life Cycle with State Changes....23
Figure 3: NDEF Tag Application Example....25

Tables

Table 1: Command Set overview 8
Table 2: Format of C-APDU 9
Table 3: Format of R-APDU 9
Table 4: Data Structure of the Capability Container file 10
Table 5: Handling of the mapping document version numbers 11
Table 6: Defined TLV blocks 12
Table 7: Data Structure of the Proprietary File 14
Table 8: Data Structure of the NDEF file 15
Table 9: NDEF Tag Application Select – C-APDU 17
Table 10: NDEF Tag Application Select – Detailed C-APDU Field Description 18
Table 11: NDEF Tag Application Select – R-APDU 18
Table 12: Capability Container Select – C-APDU 18
Table 13: Capability Container Select – Detailed C-APDU Field Description 19
Table 14: Capability Container Select – R-APDU 19
Table 15: ReadBinary Command – C-APDU 19
Table 16: ReadBinary Command – Detailed C-APDU Field Description 20
Table 17: ReadBinary Command – R-APDU 20
Table 18: NDEF Select – C-APDU 20
Table 19: NDEF Select– Detailed C-APDU Field Description 21
Table 20: NDEF Select – R-APDU 21
Table 21: NDEF Update – C-APDU 22
Table 22: NDEF Update– Detailed C-APDU Field Description 22
Table 23: NDEF Update data structure – R-APDU 22
Table 24: CC File Example 26
Table 25: NDEF File Example 26
Table 26: Revision History 31

1 Introduction

This specification is part of the NFC Forum documentation about tag types that an NFC Forum device needs to support in reader/writer mode.

This specification documents how an NFC Forum Device SHALL operate an NFC Forum Type 4 tag platform. This is not a specification of the NFC Forum Type 4 tag platform itself.

1.1 Objectives

The purpose of this specification is to document the requirements and to specify, with a set of rules and guidelines, the NFC Forum Device operation and management of the Type 4 tag platform.

This specification assumes that the Collision Detection and Device Activation activities have been performed as documented in the Mode Switch specifications [DIGPROT] and [ANINT].

This specification also defines the data mapping and how the NFC Forum Device detects, reads, and writes NDEF data into the Type 4 tag platform in order to achieve and maintain interchangeability and interoperability.

1.2 Purpose

The purpose of this specification is to document the requirements and to specify, with a set of rules and guidelines, the NFC Forum Device operation and management of a Type 4 tag platform.

This specification also defines the data mapping and how the NFC Forum Device detects, reads, and writes NDEF data into the Type 4 tag platform in order to achieve and maintain interchangeability and interoperability.

1.3 Applicable Documents or References

[ANINT]	NFC Analogue Interface Specification.
[DIGPROT]	NFC Digital Protocol Specification.
[ISO 7816-4]	ISO/IEC 7816-4 Identification cards - Integrated circuit cards - Organization, security and commands for interchange.
[NDEF]	“NFC Data Exchange Format (NDEF)” NFC Forum™, May 2006.
[RFC 2119]	S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

1.4 Administration

The NFC Forum Data Exchange Format Specification is an open specification supported by the Near Field Communication Forum, Inc., located at:

401 Edgewater Place, Suite 600
Wakefield, MA, 01880

Tel.: +1 781-876-8955
Fax: +1 781-224-1239

http://www.nfc-forum.org/

The Devices technical working group maintains this specification.

1.5 Special Word Usage

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119.

1.6 Name and Logo Usage

The Near Field Communication Forum’s policy regarding the use of the trademarks *NFC Forum* and the NFC Forum logo is as follows:

- Any company MAY claim compatibility with NFC Forum specifications, whether a member of the NFC Forum or not.
- Permission to use the NFC Forum logos is automatically granted to designated members only as stipulated on the most recent Membership Privileges document, during the period of time for which their membership dues are paid.
- Member’s distributors and sales representatives MAY use the NFC Forum logo in promoting member’s products sold under the name of the member.
- The logo SHALL be printed in black or in color as illustrated on the Logo Page that is available from the NFC Forum at the address above. The aspect ratio of the logo SHALL be maintained, but the size MAY be varied. Nothing MAY be added to or deleted from the logos.
- Since the NFC Forum name is a trademark of the Near Field Communication Forum, the following statement SHALL be included in all published literature and advertising material in which the name or logo appears:

  NFC Forum and the NFC Forum logo are trademarks of the Near Field Communication Forum.

1.7 Intellectual Property

The Type 4 Tag Operation Specification conforms to the Intellectual Property guidelines specified in the NFC Forum's Intellectual Property Right Policy, as approved on November 9, 2004 and outlined in the NFC Forum Rules of Procedures, as approved on December 17, 2004.

1.8 Glossary

APDU

Application Protocol Data Unit

C-APDU

Command APDU

CC

Capability Container

DF

Directory File

EF

Elementary File (file identifier)

Lc

Length command

Le

Length expected

LSB

least significant byte

lsb

least significant bit

MLc

Maximum data Length C-APDU

MLe

Maximum data Length R-APDU

MSB

most significant byte

msb

most significant bit

NDEF

NFC Data Exchange Format

NFC Forum device

A device that supports the following modus operandi: Initiator, Target and Reader/Writer. It may also support Card Emulator. NFC Forum compliant Device. In this document the NFC Forum device is always using the Reader/Writer modus operandi (for more information see [DIGPROT]).

R-APDU

Response APDU

RF

Radio Frequency

RFU

Reserved for future use

Type 4 tag platform

A legacy platform supporting a subset of a Technology (also called Technology Subset). Type 4 tag platform, which uses a particular subset of NFC – Type A technology or NFC- Type B technology, including anticollision (for more information see [DIGPROT]).

1.9 Convention and notations

1.9.1 Representation of numbers

The following conventions and notations apply in this document unless otherwise stated.

Binary numbers are represented by strings of digits 0 and 1 shown with the most significant bit (msb) left and the least significant bit (lsb) right, “b” is added at the end.

Example: 11110101b

Hexadecimal numbers are represented is using the the numbers 0 - 9 and the characters A – F, a “h” is added at the end. The Most Significant Byte (MSB) is shown on the left, the Least Significant Byte (LSB) on the right.

Example: F5h

Decimal numbers are represented as is (without any tailing character).

Example: 245

2 Management of the Memory Structure

The Type 4 tag platform contains at least the NDEF Tag Application. The NDEF Tag Application contains the NDEF messages on a Type 4 tag platform providing a file system composed of at least two EF files (see [ISO 7816-4]): the Capability Container file (CC file, see section 6.1) and the NDEF file (NDEF file, see section 6.2).

Concerning the EF files, the byte with offset value equal to zero is the Most Significant Byte (MSB), and the byte with the highest offset value is the Lost Significant Byte (LSB).

In the definition of this document if not otherwise specified, the bit and byte ordering when defining packets and messages follows the big-endian byte order.

3 RF interface

The RF interface of the NFC Forum device is defined in [ANINT]. The NFC Forum device SHALL comply with the RF interface as defined in the relevant clauses of [ANINT].

4 Framing / Transmission Handling

This chapter describes the framing (also called packet structures), and the transmission handling of the NFC Forum device.

The NFC Forum device SHALL comply with the bit coding, the character coding, the frame coding, the byte coding and as well the commands and responses up to and including the activation sequence of the Type 4 tag as defined in [DIGPROT].

The NFC Forum device SHALL comply with the half duplex protocol (see Type 4 tag in [DIGPROT]) for exchanging the commands and responses defined in chapter 5.

5 Command Set

This chapter describes the command set of the NFC Forum Device.

5.1 Activation of the Transmission Protocol

The activation of the transmission protocol is described in Type 4 tag section of [DIGPROT].

5.2 High Level Command Set

The commands that SHALL be supported at ISO/IEC 7816-4 [ISO 7816-4] by the NFC Forum device are listed in Table 1. The format of the commands and the relative responses of Table 1 are described in section 5.2.1 and 5.2.2. To detect and access the NFC Forum data, the specific settings of the command and response fields are described in chapter 6.

The commands of the NFC Forum device SHALL support short length fields and are compatible with the commands defined in [ISO 7816-4].

The standard [ISO 7816-4] names: command APDU (C-APDU) the commands sent from the NFC Forum device, and responses APDU (R-APDU) the responses to a specific command received by the NFC Forum Device.

NOTE [ISO 7816-4] defines a whole range of commands where only a few are relevant for a Type 4 tag platform implementation.

Table 1: Command Set overview

Command/Response	Description
Select	Selection of applications, or files
ReadBinary	Read data from file
UpdateBinary	Update (erase and write) data to file

NOTE This specification provides means of reading and writing the NDEF file. It does not cover the personalization of the Type 4 tag platform and modifications of access rights. It is assumed that the Type 4 tag platform has already been personalized as expected.

5.2.1 Format of C-APDU

The C-APDU is sent by an NFC Forum device as it is defined in accordance to [ISO 7816-4]. The following part is intended to illustrate the activation of the transmission protocol.

The length fields Lc and Le as well as the data field are optional. Short length field (one byte long) as defined in [ISO 7816-4] SHALL be supported.

Table 2 describes the format of the C-APDU. In chapter 6 the structure of Table 2 is used to describe the different C-APDU.

Table 2: Format of C-APDU

CLA	INS	P1	P2	Lc	Data	Le
Class byte	Instr. Byte	Param. byte 1	Param. byte 2	Lc field	Data bytes (Lc bytes)	Le field

Class Byte: the Class byte SHALL be set 00h (compliant to [ISO 7816-4], no secure messaging).

Instruction Byte: the Instruction byte indicates the command to process.

Parameter Byte 1: the parameter SHALL be set to 00h if no value is specified for instruction use.

Parameter Byte 2: the parameter SHALL be set to 00h if no value is specified for instruction use.

Data Field Length Lc: the data field length Lc is optional. If it is present, it SHALL contain the number of bytes in the data field of the command, and it SHALL be different from zero.

Data Field: the data field is optional.

Expected Response Length Le: the expected response length Le is optional. If it is present, the response R-APDU (see section 5.2.2) contains the number of expected bytes.

5.2.2 Format of R-APDU

The R-APDU is the response to the C-APDU received by NFC Forum device. It is defined in accordance to [ISO 7816-4]. The following part is intended to illustrate the activation of the transmission protocol.

Table 3 describes the format of the C-APDU. In chapter 6 the structure of Table 3 is used to describe the different C-APDU.

Table 3: Format of R-APDU

Response Body	SW1	SW2
Data bytes	Status Word 1	Status Word 2

Response Body: the response body is optional and carries the data of the R-APDU.

Response Status Bytes: the response status bytes SW1 and SW2 are mandatory. The coding of the response status bytes follow the coding as defined in [ISO 7816-4].

6 NDEF Detection and Access

This section describes how the NFC Forum defined data are stored and accessed by the NFC Forum device.

6.1 NDEF Management

To detect and access NFC Forum defined data the NFC Forum device retrieves and uses the Capability Container (CC) file contained inside the NDEF Tag Application. The CC file contains management data, and it is stored inside a read-only EF file, see [ISO 7816-4]. The NFC Forum device SHALL accept NDEF Tag Application having the CC file with file identifier equal to E103h.

The data structure of the CC file is described in Table 4. The CC file SHALL contain from offset 0000h to 0006h the fields: CCLEN, Mapping Version, MLe and MLc. One NDEF File Control TLV SHALL be present at the offset 0007h. Zero, one or more TLV blocks MAY be present from offset 000Fh.

If not specified otherwise, the term NDEF file in the following chapters refers to the NDEF file indicated by the NDEF File Control TLV stored at the offset 0007h in the CC file.

Table 4: Data Structure of the Capability Container file

Offset (bytes)	Size (bytes)	Field	Remarks
0000h	2	CCLEN (bytes)	Indicates the size of this capability container (including this field). Valid CCLEN values are between 000Fh and FFFEh, FFFFh and 0000h-000Eh are reserved for future use.
0002h	1	Mapping Version	Indicates the mapping specification version it is compliant to, see section 6.1.1. The most significant nibble (the 4 most significant bits) SHALL indicate the major version number, and the least significant nibble (the 4 least significant bits) SHALL indicate the minor version number.
0003h	2	MLe (bytes); Maximum R-APDU data size	This value defines the maximum data size that can be read from the Type 4 tag using a single ReadBinary command. The valid values are MLe = 000Fh-FFFFh. The values between 0000h-0000Eh are reserved for future use.
0005h	2	MLc (bytes); Maximum C-APDU data size	This value defines the maximum data size that can be sent to the Type 4 tag using a single UpdateBinary command. The valid range is MLc = 0001h-FFFFh bytes. The value 0000h is RFU.
0007h	8	NDEF File Control TLV	TLV block that contains information to control and manage the NDEF file, see section 0
000Fh	-	TLV Blocks	Zero, one or more TLV blocks MAY be starting from offset Fh

6.1.1 Version Treating

The Mapping Version field in the CC contains the version of the applied mapping document to the NFC Forum Type 4 tag platform. The mapping document version SHALL be indicated with two numbers: major number version (most significant nibble) and minor version number (least significant nibble).

This document specifies the following mapping version: major version number equal to 1h and minor version number equal to 0h i.e. mapping version 1.0.

The handling rules of the different mapping document version numbers applied to the Type 4 tag platform (called T4VNo) and the one implemented in the NFC Forum device (called NFCDevVNo) is explained in the 4 cases of Table 5.

Table 5: Handling of the mapping document version numbers

No	Version Number Case	Handling Rules
1	Major NFCDevVNo is equal to major T4VNo, and minor NFCDevVNo is bigger than or equal to minor T4VNo	The NFC Forum device SHALL access the Type 4 tag and SHALL use all features of the applied mapping document to this Type 4 tag platform.
2	If major NFCDevVNo is equal to major T4VNo, and minor NFCDevVNo is lower than minor T4VNo	Possibly not all features of the Type 4 tag platform can be accessed. The NFC Forum device SHALL use all its features and SHALL access this Type 4 tag platform.
3	If major NFCDevVNo is smaller than major T4VNo	Incompatible data format. The NFC Forum device cannot understand the Type 4 tag platform data. The NFC Forum device SHALL reject this Type 4 tag platform.
4	If major NFCDevVNo is bigger than major T4VNo	The NFC Forum device MAY implement the support for previous versions of this specification in addition to its main version. In case the NFC Forum device has the support from previous version, it SHALL access the Type 4 tag platform. On the contrary, in case the NFC Forum device has not the support from previous version, it SHALL reject the Type 4 tag platform.

NOTE Future versions of this specification have to define the allowed actions to an NFC Forum Type 4 tag platform with a version number lower than the version number of the NFC Forum device (e.g. whether it is allowed to upgrade the tag to the new version).

6.1.2 TLV blocks

A TLV block consists of one to three fields:

- *T* The tag field identifies the type of the TLV block, see Table 6, and consists of a single byte encoding a number from 00h to FEh. The tag field values from 00h to 03h and from 06h to FFh are reserved for future use by the NFC Forum.
- *L* The length field provides the size in bytes of the value field. It has two different formats composed of one or three bytes. The NFC Forum device SHALL understand all two length field formats. Figure 1 shows the two different length field formats. However depending on the tag field value, the length field may not be present.
  - *One byte format:* The NFC Forum device SHALL use the one byte format to code the length of the value field between 00h and FEh bytes. The NFC Forum device SHALL interpret this byte as a cardinal if the value is between 00h and FEh. If it contains FFh, the NFC Forum device SHALL interpret the value as flag that specifies that the length field is composed of more than one byte.
  - *Three consecutive bytes format:* The NFC Forum device SHALL use this format to code the length of the value field between 00FFh and FFFEh bytes. The first byte is assumed to be a flag equal to FFh indicating that two more bytes are present. The NFC Forum device SHALL interpret the two more bytes as a word. The NFC Forum device SHALL interpret this word as a cardinal if the value is between 00FFh and FFFEh. The value FFFFh is reserved for future use (RFU).

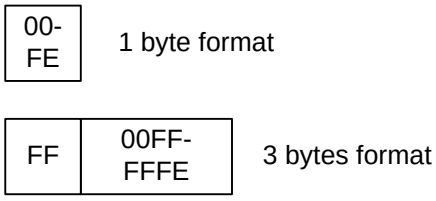

Figure 1: Length Field Formats

- *V* If the length field is equal to 00h or there is no length field, the value field is not present, i.e. the TLV block is empty. If there is the length field and indicates a length of the value field N bigger than zero (N>0), the value field consists of N consecutive bytes.

Table 6 lists the TLV blocks specified by this document that are described in the following sections.

Table 6: Defined TLV blocks

TLV block name	Tag Field Value	Short Description
NDEF File Control TLV	04h	It contains control information concerning the EF file containing the NDEF message
Proprietary File Control TLV	05h	Contains control information concerning a 'Proprietary file', which is a EF file containing proprietary data

NFC Forum Devices SHALL ignore and jump over those TLV blocks that make use of reserved tag field values. To jump over a TLV block with reserved tag field values, the NFC Forum device SHALL read the length field to understand the length of the value field.

NOTE Future definitions of TLV blocks composed of only the tag field are not backward compatible with this NFC Forum specification.

6.1.2.1 NDEF File Control TLV

The NDEF File Control TLV is always present inside the CC file. It provides control information about the EF file containing the NDEF message, see section 6.2. The NFC Forum device SHALL be able to read and process the NDEF File Control TLV. The NFC Forum device SHALL check that the CC file contains an NDEF File Control TLV at the offset 0007h.

Below the encoding of the 3 TLV fields of NDEF File Control TLV are shown:

- T is equal to 04h (see Table 6).
- L is equal to 06h.
- V is composed of 6 bytes that specifies size, read access conditions, write access conditions, and EF identifier of the EF file containing the NDEF message. The 6 bytes are encoded in the following way:
  - File Identifier, 2 bytes. It indicates a valid NDEF file. The valid range is 0000h-FFFEh, except the values E102h, E103h, 3F00h, and 3FFFh. The values 3F00h, 3FFFh are reserved and FFFFh is reserved for future use (see [ISO 7816-4]).
  - Maximum NDEF file size, 2 bytes. Maximum size in bytes of the NDEF file. This size does not reflect the size of the contained NDEF message as such but rather the size of the created file containing the NDEF message. The valid range is 0005h-FFFEh. The values 0000h-0004h, and FFFFh are reserved for future use.
  - NDEF file read access condition, 1 byte:
    - The value 00h indicates read access granted without any security,
    - The values from 01h to 7Fh and FFh are reserved for future use, and
    - The values from 80h to FEh are proprietary.
  - NDEF file write access condition, 1 byte:
    - The value 00h indicates write access granted without any security.
    - The value FFh indicates no write access granted at all (read-only).
    - The values from 01h to 7Fh are reserved for future use, and
    - The values from 80h to FEh are proprietary.

NOTE The maximum size of the NDEF file is not limited by the Maximum NDEF file size but Offset and Length Le fields of Read Binary and NDEF Update C-APDUs (see Table 15, Table 16, Table 21 and Table 22). The maximum size of the NDEF file is so reduced to 7FFFh + FFh = 80FEh bytes.

6.1.2.2 Proprietary File Control TLV

The Proprietary File Control TLV contains control information about the Proprietary file. The Proprietary file is an EF file that contains proprietary information (see below). The CC file contains zero, one or more Proprietary File Control TLV. The NFC Forum device SHALL be able to read/process this TLV block. The NFC Forum device MAY ignore the data contained in the Proprietary File Control TLV. Below the encoding of the 3 TLV fields of Proprietary TLV are shown:

- T is equal to 05h (see Table 6).
- L is equal to 06.
- V is composed of 6 bytes that specifies size, read access conditions and write access conditions, and EF identifier of the EF file containing the proprietary data. The 6 bytes are encoded in the following way:
    - File Identifier, 2 bytes. It indicates a valid Proprietary file. The valid range is 0000h-FFFEh, except the values E102h, E103h, 3F00h, and 3FFFh. The values 3F00h, 3FFFh are reserved and FFFFh are reserved for future use (see [ISO 7816-4]).
    - Maximum Proprietary file size, 2 bytes. Maximum size in bytes of the Proprietary file. The valid range is 0003h-FFFEh. The value FFFFh is reserved for future use.
    - Proprietary file read access condition, 1 byte:
        - The value 00h indicates read access granted without any security,
        - The values from 01h to 7Fh and FFh are reserved for future use, and
        - The values from 80h to FEh are proprietary.
    - Proprietary file write access condition, 1 byte:
        - The value 00h indicates write access granted without any security.
        - The value FFh indicates no write access granted at all (read-only).
        - The values from 01h to 7Fh are reserved for future use, and
        - The values from 80h to FEh are proprietary.

The proprietary data is stored inside an EF file (see [ISO 7816-4]) called Proprietary file using the data structure described in Table 7. An NDEF Tag Application can have zero, one or more Proprietary files.

Table 7: Data Structure of the Proprietary File

Offset (bytes)	Size (bytes)	Field	Remarks
0000h	2	PLEN [bytes]	The Proprietary Length field (PLEN) indicates the size of the proprietary data stored in the Proprietary file. Valid PLEN values are between 0000h and FFFEh, FFFFh is reserved for future use.
0002h	x	proprietary data	proprietary data

NOTE x indicates the size of the proprietary data.

6.2 NDEF Storage

The data format of the NDEF message is defined in [NDEF]. The NDEF message is stored inside an EF file (see [ISO 7816-4]) called NDEF file using the data structure described in Table 8. The NFC Forum device SHALL check that the NDEF file specified in the mandatory NDEF File Control TLV is present in the NFC Forum application (see section 6.4.1).

Table 8: Data Structure of the NDEF file

Offset	Size	Field	Remarks
0h	2	NLEN [bytes]	The NDEF Length field (NLEN) indicates the size of the NDEF message stored in the NDEF file. Valid NLEN values are between 0000h and FFFEh, FFFFh is reserved for future use.
2h	x	NDEF message	NDEF message, see [NDEF].

NOTE x indicates the size of the NDEF message.

The NDEF file contains either an empty or a non-empty NDEF message. Below the definition of an empty NDEF message is given in Appendix A.

6.3 Life Cycle

The NFC Forum device classifies a Type 4 tag platform into several states. The state is reflected by the content of the NDEF Tag Application. Every state has its own valid operations.

The transition states are only relevant for NFC Forum devices, which are capable of writing Type 4 tag platforms.

The NFC Forum device SHALL detect and accept a Type 4 tag platform in one of the following states: INITIALISED, READ/WRITE and READ-ONLY.

The NFC Forum device SHALL ignore Type 4 tag platform not in a valid state. The reasons MAY be:

- The NDEF Tag Application, the CC file, or the NDEF file is not present in the Type 4 tag platform.
- Misconfigured CC file.
- Not allowed write operation on the NDEF file, if Type 4 tag platform is in READ/WRITE state and no other error is detected.
- Misconfigured NDEF file.

6.3.1 INITIALISED State

In this state the NFC Forum device MAY modify the content of the NFC Forum defined data i.e. NDEF Message TLV in the Type 4 tag platform.

The NFC Forum device SHALL detect a Type 4 tag platform in INITIALISED state when:

- The CC file is set as described in section 6.1,
- The NDEF file, indicated by the File Identifier of the NDEF File Control TLV of the CC file at the offset 0007h, is open for read and write access, and
- The NLEN field of the NDEF file is equal to 0000h.

Having detected the INITIALISED state, the NFC Forum device MAY modify the content of the NDEF file.

6.3.2 READ/WRITE State

In this state the NFC Forum device MAY modify the content of the NFC Forum defined data i.e. NDEF Message TLV in the Type 4 tag platform.

The NFC Forum device SHALL detect a Type 4 tag platform in READ/WRITE state when:

- The CC file is set as described in section 6.1,
- The NDEF file, indicated by the Filed Identifier of the NDEF File Control TLV of the CC file at the offset 0007h, is open for read and write access, and
- The NDEF file contains the NLEN field different from 0000h.

Having detected the READ/WRITE state, the NFC Forum device MAY modify the content of the NDEF file.

6.3.3 READ-ONLY State

In this state the NDEF file is set to read-only.

The NFC Forum device SHALL detect a Type 4 tag in READ-ONLY state when:

- The CC file is set as described in section 6.1,
- The NDEF file, indicated by the Filed Identifier of the NDEF File Control TLV of the CC file at the offset 0007h, is READ-ONLY state, and
- The NDEF file contains the NLEN field different from 0000h.

6.4 Command Sequence Description

In this section several procedures are described to manage NFC Forum defined data e.g. NDEF message. Finally the different state changes or transitions between the states of the life cycle (see section 6.3) are shown in detail.

6.4.1 NDEF Detection Procedure

The NFC Forum device SHALL use the NDEF detection procedure to detect the NDEF message inside a Type 4 tag platform.

The NDEF file that is found by the NDEF detection procedure is also called mandatory NDEF file. The mandatory NDEF file is always indicated by the NDEF File Control TLV located at the offset 0007h of the CC file.

The NDEF detection procedure is the following:

1. Select the NDEF Tag Application (see section 6.4.2).
2. If the NDEF Tag Application is successfully selected then go to item 3. Otherwise the Type 4 tag platform is not in a valid state.
3. Select the Capability Container (CC) file (see section 6.4.3).
4. If the CC file is successfully selected then go to item 5. Otherwise the Type 4 tag platform is not in a valid state.
5. Read the CC file (see section 6.4.4), and select the NDEF file (see section 6.4.5).
6. If the CC file is successfully read, and the NDEF file has read access without any security, and is successfully selected then go to item 7. Otherwise the Type 4 tag platform is not in a valid state.
7. Read NLEN (NDEF length) from NDEF file (see section 6.4.6):
    1. If NLEN>0000h and NLEN≤Maximum NDEF file size-2, the NDEF message is detected inside the Type 4 tag platform.,
    2. If NLEN is equal to 0000h, no NDEF message is detected in the Type 4 tag platform. The Type 4 tag platform might be in INITIALISED state, and
    3. If NLEN is bigger than Maximum NDEF size-2, the Type 4 tag platform is not in a valid state.

NOTE The NDEF detection procedure does not relate to a valid NDEF message (see [NDEF]). It reads the length of the store data from the NLEN field and does not parse the data itself from the NDEF message field.

6.4.2 NDEF Tag Application Select Procedure

The NFC Forum device SHALL execute the NDEF Tag Application select procedure to select the NDEF Tag Application.

To perform the NDEF Tag Application select procedure, the NFC Forum device SHALL send the Select command (see Table 1) in addition to the sequence defined in section 5.1.

The command parameter of the Select command SHALL be set to select by name. Only when this command returns command completed in the R-APDU the NDEF Tag Application is selected. File control information that is possibly returned MAY be ignored.

The following table defines the C-APDU of the Select command to select the NDEF Tag Application (called NDEF Tag Application Select).

Table 9: NDEF Tag Application Select – C-APDU

CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	04h	00h	07h	D2760000850100h	-

The table below provides a detailed description of the C-APDU fields.

Table 10: NDEF Tag Application Select – Detailed C-APDU Field Description

Byte	Data	Remarks
P1	04h	Select by name
P2	00h	First or only occurrence
Le	-	This field SHALL NOT be present

Table 11: NDEF Tag Application Select – R-APDU

Data	SW1	SW2	Remarks
File control information MAY be returned	90h	00h	Command completed, it is optional to return file control information
-	6Ah	82h	NDEF Tag Application not found, no data is returned

NOTE For further return codes and definitions of those refer to [ISO 7816-4].

6.4.3 Capability Container Select Procedure

The NFC Forum device SHALL perform the Capability Container select procedure to select the capability container (CC) file using the Select command (see Table 1). The command parameter of the Select command SHALL be set to select by elementary file (EF).

Only when this command returns command completed in the R-APDU the CC file is selected. File control information that is possibly returned in the R-APDU MAY be ignored.

The following table defines the C-APDU of the Select command to select the CC file (called Capability Container Select).

Table 12: Capability Container Select – C-APDU

CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	00h	02h	E103h	-

The following table provides a detailed description of the C-APDU fields.

Table 13: Capability Container Select – Detailed C-APDU Field Description

Byte	Data	Remarks
P1	00h	Select by file identifier
P2	00h	First or only occurrence
Lc	02h	2 bytes in data field
Data	E103h	File identifier (EF) of the capability container
Le	-	This field SHALL NOT be present

Table 14: Capability Container Select – R-APDU

Data	SW1	SW2	Remarks
File control information MAY be returned	90h	00h	Command completed, it is optional to return file control information
-	6Ah	82h	Capability container not found, no data is returned

NOTE For further return codes and definitions of those refer to [ISO 7816-4].

6.4.4 Capability Container Read Procedure

The NFC Forum device SHALL use the Capability Container read procedure to read the data from the Capability Container (CC) file after having previously selected it (see section 6.4.3).

The Capability Container read procedure is the following:

1. Read 15 bytes of the CC file (see Table 4) with offset zero in the ReadBinary command (see Table 1).
2. If CLEN<000Fh or read access without any security to the CC file is not granted, the CC file is not valid and the Type 4 tag platform is not in a valid state of the life cycle.

The following table defines the ReadBinary command.

Table 15: ReadBinary Command – C-APDU

CLA	INS	P1	P2	Lc	Data	Le
00h	B0h	Offset		-	-	Length Le

The following table provides a detailed description of the C-APDU fields.

Table 16: ReadBinary Command – Detailed C-APDU Field Description

Byte	Data	Remarks
P1/P2	Offset	File offset where to start reading data, valid range is 0000h-7FFFh
Le	Length Le	The number of bytes to be read from file. The valid range is 01h to FFh.
Lc	-	This field SHALL NOT be present
Data	-	This field SHALL NOT be present.

Table 17: ReadBinary Command – R-APDU

Data	SW1	SW2	Remarks
Content read	90h	00h	Command Completed

NOTE For further return codes and definitions of those refer to [ISO 7816-4].

The NFC Forum device MAY read the data of the CC file after the offset 000Fh using one or more ReadBinary command.

6.4.5 NDEF Select Procedure

The NFC Forum device SHALL use the NDEF select procedure to select the NDEF file using the Select command (see Table 1). The parameter File ID of the Select command SHALL be equal to the File Identifier of the NDEF File Control TLV contained in the CC file at the offset 0007h.

The NFC Forum device successfully selects an NDEF file when the status in the R-APDU is equal to command completed. The NFC Forum device File MAY be ignored control information that is possibly returned.

NOTE In case the Type 4 tag platform supports an ISO file system the NDEF file is located in the same DF of the CC file.

The NDEF select procedure MAY be done directly after selecting and reading the CC file (see section 6.4.3 and 6.4.4).

The following table defines the Select command to select the NDEF file (called NDEF Select).

Table 18: NDEF Select – C-APDU

CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	00h	02h	File ID	-

The following tables provide a detailed description of the C-APDU fields.

Table 19: NDEF Select– Detailed C-APDU Field Description

Byte	Data	Remarks
P1	00h	Select by file identifier
P2	02h	First or only occurrence
Lc	02h	2 bytes in data field
Data	File ID	File identifier of the NDEF file indicated in the homonymous field of the CC file (see Table 4)
Le	-	This field SHALL NOT be present

Table 20: NDEF Select – R-APDU

Data	SW1	SW2	Remarks
File control information MAY be returned	90h	00h	Command completed, it is optional to return file control information
-	6Ah	82h	NDEF file not found, no data is returned

NOTE For further return codes and definitions of those refer to [ISO 7816-4].

6.4.6 NDEF Read Procedure

The NFC Forum device SHALL execute the NDEF read procedure to read the NDEF file.

The NFC Forum device SHALL do the following operations before reading the NDEF file:

1. Detect successfully the NDEF file using the NDEF detection procedure (see section 6.4.1),
2. Check that the read access without any security is granted for the NDEF file from the information provided by NDEF File Control TLV in the CC file at the offset 0007h (see Table 4 and section 6.1.2.1), and
3. Select the NDEF file (see section 6.4.5).

The NDEF read procedure is the following:

1. Read the NLEN (NDEF length) field of NDEF file (see Table 8) using the ReadBinary command starting from offset zero. The NLEN value MAY be also retrieved from the NDEF detection procedure (see section 6.4.1).
2. Read the NDEF message that starts at offset 0002h of the NDEF file using one or more ReadBinary commands.

The details of the ReadBinary command are described in section 6.4.4.

NOTE The read access without any security is only granted when the NDEF file read access condition indicated in the CC file is set to 00h.

6.4.7 NDEF Update Procedure

The NFC Forum device SHALL execute the NDEF update procedure to write or update an NDEF message inside the NDEF file.

The NFC Forum device SHALL do the following operations before the NDEF update procedure:

1. Detect successfully the NDEF message using the procedure in section 6.4.1,
2. Check that the write access without any security is granted for the NDEF file from the information provided by NDEF File Control TLV in the CC file at the offset 0007h (see Table 4 and section 6.1.2.1), and
3. Select the NDEF file (see section 6.4.5).

The following table defines the NDEF update command (see Table 1) to write or to update the NDEF message inside the NDEF file.

Table 21: NDEF Update – C-APDU

CLA	INS	P1	P2	LC	Data	Le
00h	D6h	Offset		Length Lc	Data to be written in the NDEF file	-

The following table provides a detailed description of the C-APDU fields.

Table 22: NDEF Update– Detailed C-APDU Field Description

Byte	Data	Remarks
P1/P2	Offset	Offset in NDEF file where starting to write data. The valid range is 0000h-7FFFh, see [ISO4]
LC	Length Lc	The number of bytes written to NDEF file. The valid range is 01h to FFh.
Le	-	This field SHALL NOT be present.

Table 23: NDEF Update data structure – R-APDU

Data	SW1	SW2	Remarks
-	90h	00h	Command Completed, no data is returned

NOTE For further return codes and definitions of those refer to [ISO 7816-4].

The NDEF update procedure is the following:

1. If the length of the NDEF message (to be written) is bigger than Maximum NDEF size-2 (see NFC File Control TLV in the section 6.1.2.1) the NDEF update procedure is aborted. Otherwise go to item 2.
2. Write the value 0000h in the NLEN field (see Table 8) using the NDEF Update command,
3. Write the NDEF message in the NDEF message field (see Table 8) using one or more NDEF Update commands, and
4. Write the length of the NDEF message in the NLEN field (see Table 8) using the NDEF Update command.

The NFC Forum device MAY apply this procedure directly after the NDEF read procedure (see section 6.4.6) has been executed.

The item 2 to 3 MAY be done using a single NDEF Update command if the NLEN field and the NDEF message field fit inside the data field of the NDEF Update command.

6.4.8 State Changes

This section describes the possible state changes or also called transition performed by the NFC Forum device. Figure 2 shows the states and the transitions.

The transition covered by this specification is from INITIALISED to READ/WRITE.

NOTE A Type 4 tag platform can be issued in any valid state. A Type 4 tag platform can be issued in READ/WRITE state or even in READ-ONLY state having a predefined NDEF message stored on it.

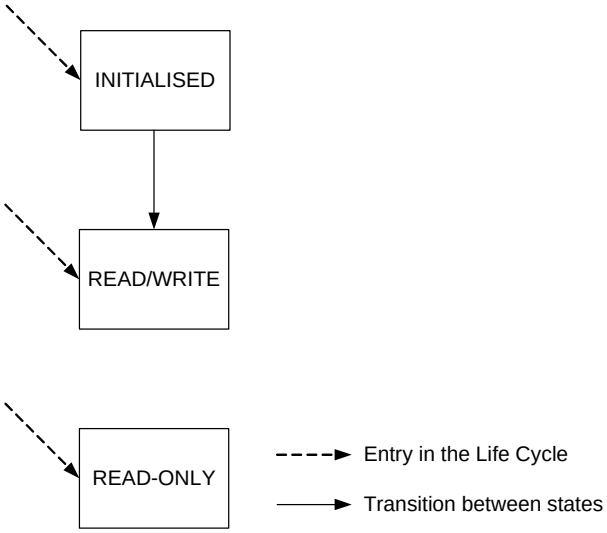

Figure 2: Life Cycle with State Changes

The NFC Forum device SHALL do the following operation to perform the transition from INITIALISED to READ/WRITE: an NDEF message (see [NDEF]) is written in the NDEF Message field of the mandatory NDEF file using the NDEF update procedure (see section 6.4.7).

The NFC Forum device MAY replace a non-empty NDEF message with an empty NDEF message (see Appendix A).

A. Empty NDEF Message

An empty NDEF message (see [NDEF]) is defined as an NDEF message composed of one NDEF record. The NDEF record uses the NDEF short-record layout (SR=1b) with: Type Name Format (TNF) field value equal to 00h (empty, TYPE_LENGTH=00h, PAYLOAD_LENGTH=00h), no ID_LENGTH field (IL=0b), MB=1b, ME=1b, CF=0b. The empty NDEF record (i.e. the empty NDEF message) is composed of 3 bytes and it is equal to D00000h.

B. Example of NDEF Tag Application

This appendix describes from the NFC Forum device point of view an example of NDEF Tag Application (see Figure 3) stored inside a Type 4 tag platform.

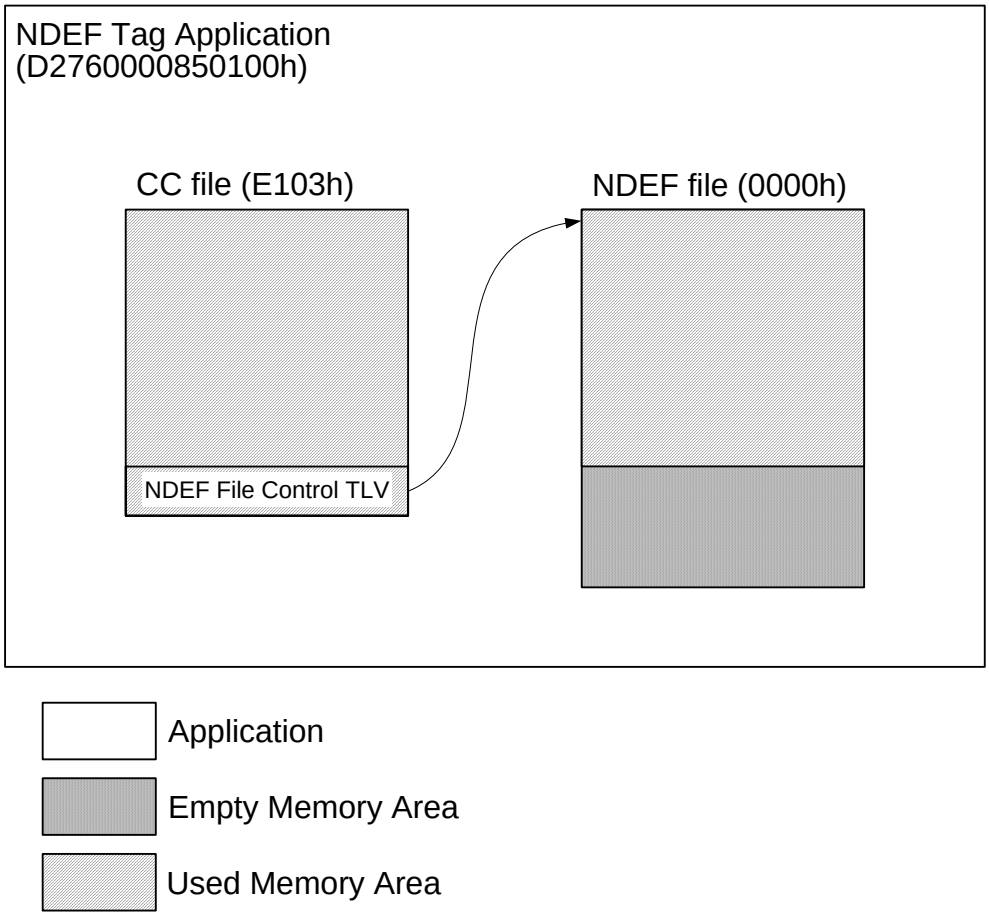

Figure 3: NDEF Tag Application Example

The Capability Container (CC) file is described in detail in Table 24.

Table 24: CC File Example

Offset	Size	Value	Content
0h	2	000Fh	CCLEN (15 bytes)
2h	1	10h	Mapping Version 1.0 (this specification)
3h	2	003Bh	MLe (49 bytes); Maximum R-APDU data size
5h	2	0034h	MLc (52 bytes); Maximum C-APDU data size
7h	1	04h	T field of the NDEF File Control TLV
	1	06h	L field of the NDEF File Control TLV
	6		V field of the NDEF File Control TLV:
		0000h	File Identifier
		0032h	Maximum NDEF size (50 bytes)
		00h	NDEF file read access condition, read access without any security
		00h	NDEF file write access condition, write access without any security

The NDEF file is described in detail in Table 25.

Table 25: NDEF File Example

Offset	Size	Value	Content
0h	2	0003h	NLEN, NDEF length 3 bytes
2h	3	D00000h	Empty NDEF message

C. Example Command Flow

This chapter provides some examples of the command flow in order to show how a typical interaction on an APDU level could be performed by the NFC Forum device. It is assumed that the Type 4 tag platform is configured in a proper way and contains a valid NDEF file. The examples do not cover any check of the NDEF file, and they are related to the NDEF Tag Application described in B.

The commands and the responses are written in hexadecimal form with a space between each byte without the "h" character at the end e.g. 30 F3 AB 9C. The left-most byte is the first byte to be sent, and the right-most byte is the last one to be sent. The special acronyms Byte5 and Byte14 are used to indicate a group of bytes with a specific meaning indicated later on in the description of the command or of the response.

C.1 Detection of NDEF message

The example in this section detects the NDEF message applying the NDEF detection procedure (see section 6.4.1).

The first command is the NDEF Tag Application select (see section 6.4.2).
Command: 00 A4 04 00 07 D2 76 00 00 85 01 00

- The meanings of the bytes are the following ones:
  - 00h Class byte (CLA),
  - A4h Instruction byte (INS) for Select Command,
  - 04h Parameter byte (P1), select by name,
  - 00h Parameter byte (P2), first or only occurrence,
  - 07h Lc field,
  - D2760000850100h NDEF Tag Application name.

  Expected Response: 90 00
- The meanings of the bytes are the following ones:
  - 9000h Status bytes (SW1, SW2), command completed.

The second command is the Capability Container select (see section 6.4.3):
Command: 00 A4 00 00 02 E1 03

- The meanings of the bytes are the following ones:
  - 00h Class byte (CLA),
  - A4h Instruction byte (INS) for Select command,
  - 00h Parameter byte (P1), select by identifier,
  - 00h Parameter byte (P2), first or only occurrence,
  - 02h Lc field,
  - E103h file identifier of the CC file.

  Expected Response: 9000

- The meanings of the bytes are the following ones:
  - 9000h Status bytes (SW1, SW2), command completed.

The third command is ReadBinary data from CC file (see section 6.4.4):

Command: 00 B0 00 00 0F

- The meanings of the bytes are the following ones:
  - 00h Class byte (CLA),
  - B0h Instruction byte (INS) for ReadBinary command,
  - 0000h Parameter byte (P1, P2), offset inside the CC file,
  - 0Fh Le field,

  Expected Response: 00 0F 10 00 3B 00 34 04 06 00 00 00 32 00 00 90 00
- The meanings of the bytes are the following ones:
  - 00 0Fh CCLEN length of the CC file,
  - 10h Mapping Version 1.0 (this specification),
  - 003Bh MLe maximum 59 bytes R-APDU data size,
  - 0034h MLc maximum 52 bytes C-APDU data size,
  - NDEF File Control TLV
    - 04h T field of the NDEF File Control TLV,
    - 06h L field of the NDEF File Control TLV,
    - V field of the NDEF File Control TLV,
      - 0000h File Identifier of NDEF file,
      - 0032h Maximum NDEF file size of 50 bytes,
      - 00h read access without any security,
      - 00h write access without any security.

From the response it is possible to understand that the NDEF file identifier is 0000h, and the NDEF file has granted read access without any security. This allows the following two operations: the selection of the NDEF file and the read of the NDEF message.

The fourth command is the NDEF Select command (see section 6.4.5):

Command: 00 A4 00 00 02 00 00

- The meanings of the bytes are the following ones:
  - 00h Class byte (CLA),
  - A4h Instruction byte (INS) for Select command,
  - 00h Parameter byte (P1), select by identifier,
  - 00h Parameter byte (P2), first or only occurrence,
  - 02h Lc field,

- 0000h file identifier of the NDEF file retrieved from the CC file.

  Expected Response: 90 00

- The meanings of the bytes are the following ones:
  - 9000h Status bytes (SW1, SW2), command completed.

The fifth command is the Read Binary command. The command reads the NLEN field of the NDEF file:

Command: 00 B0 00 00 02

- The meanings of the bytes are the following ones:
  - 00h Class byte (CLA),
  - B0h Instruction byte (INS) for ReadBinary command,
  - 0000h Parameter byte (P1, P2), offset inside the CC file,
  - 02h Le field,

  Expected Response: 00 0F 90 00

- The meanings of the bytes are the following ones:
  - 000Fh NLEN length of the NDEF message,
  - 9000h Status bytes (SW1, SW2), command completed.

NLEN is smaller than the Maximum NDEF file size-2 (equal to 50-2=48 bytes), and bigger than 0000h, hence the NDEF message is successfully detected inside the NDEF file.

C.2 Read data from the NDEF file

To read the NDEF file the NDEF read procedure is applied (see section 6.4.6). It is supposed that the NDEF file was previously successfully detected, and the NDEF file is correctly selected.

The command ReadBinary reads 15 bytes from the NDEF file:
Command: 00 B0 00 00 0F

- The meanings of the bytes are the following ones:
  - 00h Class byte (CLA),
  - B0h Instruction byte (INS) for ReadBinary command,
  - 0000h Parameter byte (P1, P2), offset inside the CC file,
  - 0Fh Le field,

  Expected Response: 00 03 D0 00 00 Byte5…Byte14 90 00

- The meanings of the bytes are the following ones:
  - 0003h NLEN length of the NDEF message,
  - D00000h NDEF message field, it contains an empty NDEF message (see section Appendix A).
  - Byte5…Byte14 data that belong to the NDEF file that is ignored.
  - 9000h Status bytes (SW1, SW2), command completed.

C.3 Write data in the NDEF file

To write the NDEF file the NDEF update procedure is applied (see section 6.4.7). It is supposed that:

- The NDEF file was previously successfully detected (using the procedure described in section 6.4.1),
- The NDEF file has write access without any security granted,
- The NDEF file is correctly selected,
- The NDEF file size -2 (see Maximum NDEF file field of the CC file) is bigger than the NDEF message that has to be written into the NDEF file. In this example the NDEF message is 3 bytes long and the Maximum NDEF file = 50 bytes, hence 50-2≥3 it is allowed to write the NDEF message in the NDEF file.

NDEF Update command to write data into the NDEF file:

Command: 00 D6 00 00 05 00 03 D0 00 00

- The meanings of the bytes are the following ones:
  - 00h Class byte (CLA),
  - D6h Instruction byte (INS) for UpdateBinary command,
  - 00 00h Parameter byte (P1, P2), offset inside the CC file,
  - 05h Lc field,
  - 0003h NLEN, length of the NDEF message.
  - D00000h NDEF message composed of one empty record.

  Expected Response: 9000
- The meanings of the bytes are the following ones:
  - 9000h Status bytes (SW1, SW2), command completed. Five data bytes have been successfully written to the NDEF file stating at the offset 0000h.

The writing of the NLEN and NDEF message field of the NDEF file has been done using one single NDEF Update command because the NLEN field and the NDEF message field are small enough (2 byets + 3 bytes = 5 bytes) to be contained in the data field of the NDEF Update command (MLe=59 bytes).

D. Revision History

The following table outlines the revision history of Type 4 Tag Operation.

Table 26: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
NFCForum-TS-Type-4-Tag_1.0	1.0, July 2007	Final		