Type 4 Tag Operation Specification

Technical Specification

NFC Forum™

T4TOP 2.0

NFCForum-TS-Type-4-Tag_2.0

2011-06-28

RESTRICTIONS ON USE

Contents

Figures

Tables

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 [DIGITAL] and [ACTIVITY].

This specification also defines 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

[ACTIVITY]	NFC Activity Specification, Latest version NFC Forum
[DIGITAL]	NFC Digital Protocol Technical Specification, Version 1.0, NFC Forum
[ISO/IEC_7816-4]	ISO/IEC 7816-4:2005 Identification cards - Integrated circuit cards - Organization, security and commands for interchange, Second edition, January 15, 2005
[NDEF]	NFC Data Exchange Format (NDEF), Version 1.0 NFC Forum
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels, RFC 2119 S. Bradner, March 1997 Harvard University
[T4TOP_V1.0]	Type 4 Tag Operation Technical Specification, Version 1.0, NFC Forum

1.4 Administration

The NFC Forum Type 4 Tag Operation 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: +1 781-610-9864

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

The Devices technical working group maintains this specification. Comments, errors, and other feedback can be submitted at http://www.nfc-forum.org/apps/group_public/document.php?document_id=9801&wg_abbrev=chairs.

1.5 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.

1.6 Intellectual Property

The Type 4 Tag Operation Specification Specification conforms to the Intellectual Property guidelines specified in the NFC Forum's *Intellectual Property Rights Policy*, as outlined in the NFC Forum *Rules of Procedure*. These documents are available on the NFC Forum website.

1.7 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 [RFC2119].

1.8 Notational Conventions

1.8.1 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) on the left, the least significant bit (lsb) on the right, and a “b” added at the end.

  Example: 11110101b

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

  Example: F5h

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

  Example: 245

1.9 Acronyms and Definitions

Table 1: Acronyms and Definitions

Acronym	Definition
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

Acronym	Definition
R-APDU	Response APDU
RF	Radio Frequency
RFU	Reserved for future use

1.10 Glossary

ISO-DEP Protocol

The half-duplex block transmission protocol as defined in [DIGITAL].

NFC Forum Device

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

NFCDevVNo

Mapping version number implemented in the NFC Forum Device.

T4VNo

Mapping version numbers implemented in the Type 4 Tag Platform.

Type 4 Tag Platform

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

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 that provides a file system composed of at least two EF files (see [ISO/IEC_7816-4]): the Capability Container file (CC file, see Section 5.1), and the NDEF file (NDEF file, see Section 5.2).

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 Least Significant Byte (LSB).

As defined by this document, if not otherwise specified, the bit and byte ordering when defining packets and messages follows the big-endian byte order.

3 Framing / Transmission Handling

This section describes the framing (also called packet structures) and the transmission handling of the NFC Forum Device.

[RQ_T4T_FTH_001] The NFC Forum Device SHALL comply with the sequence format, the bit level coding, the frame format, the data and payload format, and the command set as defined in [DIGITAL], including the activation sequence of the ISO-DEP Protocol as defined in [ACTIVITY].

[RQ_T4T_FTH_002] The NFC Forum Device SHALL comply with the ISO-DEP Protocol for exchanging the commands and responses defined in Section 4.

4 Command Set

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

4.1 Activation of the ISO-DEP Protocol

[RQ_T4T_CSE_001] To send any command belonging to the high-level command set (see Section 4.2), the NFC Forum Device activates the ISO-DEP Protocol as described in [ACTIVITY].

4.2 High Level Command Set

[RQ_T4T_CSE_002] The commands that SHALL be supported by the NFC Forum Device are listed in Table 2 and are described in [ISO/IEC_7816-4].

The format of the commands and the relative responses of the command set are described in Section 4.2.1 and Section 4.2.2. To detect and access the NFC Forum data, the specific settings of the command and response fields are described in Section 5.

[RQ_T4T_CSE_003] The commands of the NFC Forum Device SHALL support short length fields.

The command APDUs (C-APDU) are the commands sent from the NFC Forum Device, and the response APDUs (R-APDU) are the responses to a specific command received by the NFC Forum Device.

NOTE [ISO/IEC_7816-4] defines a whole range of commands, where only a few are relevant for a Type 4 Tag Platform implementation.

Table 2: 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.

4.2.1 Format of C-APDU

This section describes the format of the C-APDU that is used in this specification.

[RQ_T4T_CSE_004] The length fields Lc and Le, as well as the data field, are optional. Short length fields (one byte long) SHALL be supported.

Table 3 describes the format of the C-APDU. In Section 5, the structure of Table 3 is used to describe the different C-APDUs.

Table 3: Format of C-APDU

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

[RQ_T4T_CSE_005] *Class Byte:* SHALL be set to 00h (i.e., no secure messaging).

Instruction Byte: Indicates the command to process.

[RQ_T4T_CSE_006] *Parameter Byte 1:* SHALL be set to 00h if no value is specified for instruction use.

[RQ_T4T_CSE_007] *Parameter Byte 2:* SHALL be set to 00h if no value is specified for instruction use.

[RQ_T4T_CSE_008, RQ_T4T_CSE_009] *Data Field Length Lc:* Optional. If it is present, it SHALL contain the number of bytes in the data field of the command and it SHALL NOT be zero.

Data Field: Optional.

Expected Response Length Le: Optional. If present, the response R-APDU (see Section 4.2.2) contains the number of expected bytes.

4.2.2 Format of R-APDU

This section describes the format of the R-APDU that is used in this specification.

Table 4 describes the format of the R-APDU. In Section 5, the structure of Table 4 is used to describe the different R-APDUs.

Table 4: Format of R-APDU

Response Body (optional)	SW1	SW2
Data bytes	Status Word 1	Status Word 2

Response Body: Optional. It carries the data of the R-APDU.

[RQ_T4T_CSE_010] *Response Status Bytes:* Bytes SW1 and SW2 are mandatory.

5 NDEF Detection and Access

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

5.1 NDEF Management

[RQ_T4T_NDA_001] 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/IEC_7816-4]). The NFC Forum Device SHALL accept NDEF Tag Applications having a CC file with a file identifier equal to E103h.

[RQ_T4T_NDA_002] The data structure of the CC file is described in Table 5. The CC file SHALL contain the following fields from offset 0000h to 0006h: CCLEN, Mapping Version, MLe, and MLc. One NDEF File Control TLV SHALL be present at offset 0007h. Zero, one, or more TLV blocks MAY be present from offset 000Fh.

Unless specified otherwise, the term NDEF file in the following sections refers to the NDEF file indicated by the NDEF File Control TLV stored at offset 0007h in the CC file.

Table 5: 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). [RQ_T4T_NDA_003] Valid CCLEN values are between 000Fh and FFFEh. The values between FFFFh and 0000h-000Eh are RFU.
0002h	1	Mapping Version	Indicates the mapping specification version it is compliant to (see Section 5.1.1). 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	Defines the maximum data size that can be read from the Type 4 Tag using a single ReadBinary command. [RQ_T4T_NDA_004] The valid values are MLe = 000Fh-FFFFh. The values between 0000h-0000Eh are RFU.
0005h	2	MLc (bytes); Maximum C-APDU data size	Defines the maximum data size that can be sent to the Type 4 Tag using a single UpdateBinary command. [RQ_T4T_NDA_005] The valid range is MLc = 0001h-FFFFh bytes. The value 0000h is RFU.
0007h	8	NDEF File Control TLV	[RQ_T4T_NDA_006] TLV block that contains information to control and manage the NDEF file (see Section 5.1.2.1)
000Fh	-	TLV Blocks	Zero, one, or more TLV blocks MAY start from offset Fh.

5.1.1 Version Treatment

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: major version number equal to 2h and the minor version number equal to 0h (i.e., Mapping Version 2.0).

If the NFC Forum Device implements Mapping Version 1.0 (see [T4TOP_V1.0]) in addition to this version of the specification, the NFC Forum Device SHALL be compliant with Section 5.5.

[RQ_T4T_NDA_007] 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 6.

Table 6: Handling of the Mapping Document Version Numbers

No.	Version Number Case	Handling Rules
1	If 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 smaller 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. If the NFC Forum Device implements this version and the previous version, it SHALL access the Type 4 Tag Platform. Otherwise, if the NFC Forum Device does not support the previous version, it SHALL reject the Type 4 Tag Platform.

5.1.2 TLV blocks

A TLV block consists of one to three fields:

[RQ_T4T_NDA_008] *T* The tag field identifies the type of the TLV block 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 RFU by the NFC Forum.

[RQ_T4T_NDA_009, RQ_T4T_NDA_010] *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 implement both length field formats as shown in Figure 1. 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 a 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, which indicates that two more bytes are present. The NFC Forum Device SHALL interpret those two 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 RFU.

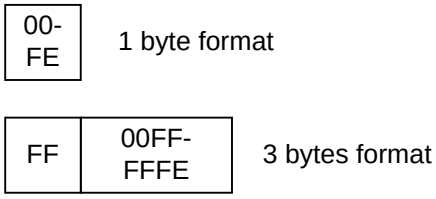

Figure 1: Length Field Formats

- [RQ_T4T_NDA_011, RQ_T4T_NDA_012] *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 a length field and it indicates a length of the value field N bigger than zero (N>0), the value field consists of N consecutive bytes.

Table 7 lists the TLV blocks specified by this document that are described in the following sections.

Table 7: Defined TLV Blocks

TLV block name	Tag Field Value	Short Description
NDEF File Control TLV	04h	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 an EF file containing proprietary data

[RQ_T4T_NDA_013] 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.

5.1.2.1 NDEF File Control TLV

[RQ_T4T_NDA_014] The NDEF File Control TLV is always present inside the CC file and it provides control information about the EF file containing the NDEF message (see Section 5.2). 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 offset 0007h.

[RQ_T4T_NDA_015] The encoding of the 3 fields of NDEF File Control TLV are:

- *T* is equal to 04h (see Table 7).
- *L* is equal to 06h.
- *V* is composed of 6 bytes that specify size, read access conditions, write access conditions, and the EF identifier of the EF file containing the NDEF message. The 6 bytes are encoded as follows:
  - [RQ_T4T_NDA_016] *File Identifier*, 2 bytes. Indicates a valid NDEF file. The valid ranges are 0001h to E101h, E104h to 3EFFh, 3F01h to 3FFEh and 4000h to FFFEh. The values 0000h, E102h, E103h, 3F00h and 3FFFh are reserved (see [ISO/IEC_7816-4]) and FFFFh is RFU.
  - [RQ_T4T_NDA_017] *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 file containing the NDEF message. The valid range is 0005h to FFFEh. The values 0000h-0004h and FFFFh are RFU.
  - [RQ_T4T_NDA_018] *NDEF file read access condition*, 1 byte:
    - 00h indicates read access granted without any security
    - 01h to 7Fh and FFh are RFU
    - 80h to FEh are proprietary
  - [RQ_T4T_NDA_019] *NDEF file write access condition*, 1 byte:
    - 00h indicates write access granted without any security
    - FFh indicates no write access granted at all (read-only)
    - 01h to 7Fh are RFU
    - 80h to FEh are proprietary

NOTE The maximum size of the NDEF file is limited by the Offset and Length Le fields of ReadBinary and NDEF Update C-APDUs (see Table 16, Table 17, Table 22, and Table 23). The maximum size of the NDEF file is reduced to 7FFFh + FFh = 80FEh bytes.

5.1.2.2 Proprietary File Control TLV

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

- *T* is equal to 05h (see Table 7).
- *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 as follows:
  - *File Identifier*, 2 bytes. Indicates a valid Proprietary file. The valid ranges are 0001h to E101h, E104h to 3EFFh, 3F01h to 3FFEh, and 4000h to FFFEh. The values 0000h, E102h, E103h, 3F00h, and 3FFFh are reserved (see [ISO/IEC_7816-4]) and FFFFh is RFU.
  - *Maximum Proprietary file size*, 2 bytes. Maximum size in bytes of the Proprietary file. The valid range is 0003h-FFFEh. The value FFFFh is RFU.
  - *Proprietary file read access condition*, 1 byte:
    - 00h indicates read access granted without any security
    - 01h to 7Fh and FFh are RFU
    - 80h to FEh are proprietary
  - *Proprietary file write access condition*, 1 byte:
    - 00h indicates write access granted without any security
    - FFh indicates no write access granted at all (read-only)
    - 01h to 7Fh are RFU
    - 80h to FEh are proprietary

The proprietary data is stored inside an EF file (see [ISO/IEC_7816-4]) called “Proprietary file” using the data structure described in Table 8. An NDEF Tag Application can have zero, one, or more Proprietary files.

Table 8: 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 RFU.
0002h	x	proprietary data	proprietary data

NOTE x indicates the size of the proprietary data.

5.2 NDEF Storage

[RQ_T4T_NDA_020, RQ_T4T_NDA_021] The data format of the NDEF message is defined in [NDEF]. The NDEF message is stored inside an EF file (see [ISO/IEC_7816-4]) called NDEF file using the data structure described in Table 9. 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 5.4.1).

Table 9: Data Structure of the NDEF File

Offset	Size	Field	Remarks
0h	2	NLEN [bytes]	[RQ_T4T_NDA_022] 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 RFU.
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. The definition of an empty NDEF message is given in Appendix A.

5.3 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 that are capable of writing Type 4 Tag Platforms.

[RQ_T4T_NDA_023] The NFC Forum Device SHALL detect and accept a Type 4 Tag Platform in one of the following states: INITIALIZED, READ/WRITE, and READ-ONLY.

[RQ_T4T_NDA_024] 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.
- The CC File is misconfigured.
- The NDEF file does not allow write operation, if Type 4 Tag Platform is in READ/WRITE state and no other error is detected.
- The NDEF File is misconfigured.

5.3.1 INITIALIZED 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).

[RQ_T4T_NDA_025] The NFC Forum Device SHALL detect a Type 4 Tag Platform in INITIALIZED state when:

- The CC file is set as described in Section 5.1
- The NDEF file, indicated by the File Identifier of the NDEF File Control TLV of the CC file at offset 0007h, is open for read and write access
- The NLEN field of the NDEF file is equal to 0000h

Having detected the INITIALIZED state, the NFC Forum Device MAY modify the content of the NDEF file.

5.3.2 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).

[RQ_T4T_NDA_026] 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 5.1
- The NDEF file, indicated by the File Identifier of the NDEF File Control TLV of the CC file at offset 0007h, is open for read and write access
- The NLEN field of the NDEF file is different from 0000h

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

5.3.3 READ-ONLY State

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

[RQ_T4T_NDA_027] The NFC Forum Device SHALL detect a Type 4 tag in READ-ONLY state when:

- The CC file is set as described in Section 5.1
- The NDEF file, indicated by the File Identifier of the NDEF File Control TLV of the CC file at offset 0007h, is READ-ONLY state
- The NLEN field of the NDEF file is different from 0000h

5.4 Command Sequence Description

Several procedures are described in this section to manage NFC-Forum-defined data (e.g., NDEF message). The different state changes or transitions between the states of the life cycle (see Section 5.3) are shown in detail.

5.4.1 NDEF Detection Procedure

[RQ_T4T_NDA_028] 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 the mandatory NDEF file. The mandatory NDEF file is always indicated by the NDEF File Control TLV located at offset 0007h of the CC file.

[RQ_T4T_NDA_029] The NDEF detection procedure is:

1. Select the NDEF Tag Application (see Section 5.4.2).
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. If the Type 4 Tag Platform is not in a valid state and the NFC Forum Device implements both Mapping Version 1.0 and Mapping Version 2.0, the NFC Forum Device SHALL select the NDEF Tag Application according to [T4TOP_V1.0] Section 6.4.1 (see also Section 5.5).
3. Select the Capability Container (CC) file (see Section 5.4.3).
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 5.4.4) and select the NDEF file (see Section 5.4.5).
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 5.4.6):
    - If NLEN>0000h and NLEN≤Maximum NDEF file size-2, the NDEF message is detected inside the Type 4 Tag Platform
    - 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 INITIALIZED state
    - 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.

5.4.2 NDEF Tag Application Select Procedure

[RQ_T4T_NDA_030] The NFC Forum Device SHALL execute the NDEF Tag Application select procedure to select the NDEF Tag Application.

[RQ_T4T_NDA_031] To perform the NDEF Tag Application select procedure, the NFC Forum Device SHALL send the Select command (see Table 2) in addition to the sequence defined in Section 4.1.

[RQ_T4T_NDA_032] The command parameter of the Select command SHALL be set to select by name. 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.

Table 10 defines the C-APDU of the Select command to select the NDEF Tag Application (called NDEF Tag Application Select).

Table 10: NDEF Tag Application Select – C-APDU

CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	04h	00h	07h	D2760000850101h	00h

Table 11 provides a detailed description of the C-APDU fields for the NDEF Tag Application Select command.

Table 11: NDEF Tag Application Select – Detailed C-APDU Field Description

Byte	Data	Remarks
P1	04h	Select by name
P2	00h	First or only occurrence
Le	00h	Response data field MAY be present

Table 12 provides a detailed description of the R-APDU fields for the NDEF Tag Application Select command.

Table 12: NDEF Tag Application Select – Detailed R-APDU Field Description

Data	SW1	SW2	Remarks
File control information MAY be returned	90h	00h	[RQ_T4T_NDA_034] Command completed; it is optional to return file control information
-	6Ah	82h	NDEF Tag Application not found; no data is returned

NOTE For more return codes and their definitions, refer to [ISO/IEC_7816-4].

5.4.3 Capability Container Select Procedure

[RQ_T4T_NDA_035, RQ_T4T_NDA_036, RQ_T4T_NDA_037] The NFC Forum Device SHALL perform the Capability Container select procedure to select the capability container (CC) file using the Select command (see Table 2). The command parameter of the Select command SHALL be set to select by elementary file (EF).

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.

Table 13 defines the C-APDU of the Select command to select the CC file (called Capability Container Select).

Table 13: Capability Container Select Command – C-APDU

CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	0Ch	02h	E103h	-

Table 14 provides a detailed description of the C-APDU fields for the Capability Container Select command.

Table 14: Capability Container Select Command – Detailed C-APDU Field Description

Byte	Data	Remarks
P1	00h	Select by file identifier
P2	0Ch	First or only occurrence
Lc	02h	2 bytes in data field
Data	E103h	File identifier (EF) of the capability container
Le	-	[RQ_T4T_NDA_038] This field SHALL NOT be present

Table 15 provides a detailed description of the R-APDU fields for the Capability Container Select command.

Table 15: Capability Container Select Command – Detailed R-APDU Field Description

Data	SW1	SW2	Remarks
-	90h	00h	Command completed; no data is returned
-	6Ah	82h	Capability container not found; no data is returned

NOTE For more return codes and their definitions, refer to [ISO/IEC_7816-4].

5.4.4 Capability Container Read Procedure

[RQ_T4T_NDA_040] 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 5.4.3).

[RQ_T4T_NDA_041, RQ_T4T_NDA_042] The Capability Container read procedure is:

1. Read 15 bytes of the CC file (see Table 5) with offset zero in the ReadBinary command (see Table 2).
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.

Table 16 defines the ReadBinary command.

Table 16: ReadBinary Command – C-APDU

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

Table 17 provides a detailed description of the C-APDU fields for the ReadBinary command.

Table 17: 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	-	[RQ_T4T_NDA_043] This field SHALL NOT be present.
Data	-	[RQ_T4T_NDA_044] This field SHALL NOT be present.

Table 18 provides a detailed description of the R-APDU fields for the ReadBinary command.

Table 18: ReadBinary Command – Detailed R-APDU Field Description

Data	SW1	SW2	Remarks
Content read	90h	00h	[RQ_T4T_NDA_045] Command Completed

NOTE For more return codes and their definitions, refer to [ISO/IEC_7816-4].

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

5.4.5 NDEF Select Procedure

[RQ_T4T_NDA_046, RQ_T4T_NDA_047, RQ_T4T_NDA_048] The NFC Forum Device SHALL use the NDEF select procedure to select the NDEF file using the Select command (see Table 2). 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 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 ignore any control information that is returned.

NOTE If 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 5.4.3 and Section 5.4.4).

Table 19 defines the Select command to select the NDEF file (called NDEF Select).

Table 19: NDEF Select Command – C-APDU

CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	0Ch	02h	File ID	-

Table 20 provides a detailed description of the C-APDU fields for the NDEF Select Command.

Table 20: NDEF Select Command – 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	File ID	File identifier of the NDEF file indicated in the homonymous field of the CC file (see Table 5)
Le	-	[RQ_T4T_NDA_049] This field SHALL NOT be present

Table 21 provides a detailed description of the R-APDU fields for the NDEF Select Command.

Table 21: NDEF Select Command – Detailed R-APDU Field Description

Data	SW1	SW2	Remarks
-	90h	00h	[RQ_T4T_NDA_050] Command completed; no data is returned
-	6Ah	82h	NDEF file not found; no data is returned

NOTE For more return codes and their definitions, refer to [ISO/IEC_7816-4].

5.4.6 NDEF Read Procedure

[RQ_T4T_NDA_051] The NFC Forum Device SHALL execute the NDEF read procedure to read the NDEF file.

[RQ_T4T_NDA_052] The NFC Forum Device SHALL complete the following operations before reading the NDEF file:

1. Successfully detect the NDEF file using the NDEF detection procedure (see Section 5.4.1)
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 offset 0007h (see Table 5 and Section 5.1.2.1)
3. Select the NDEF file (see Section 5.4.5)

[RQ_T4T_NDA_053, RQ_T4T_NDA_054, RQ_T4T_NDA_056] The NDEF read procedure is:

1. Read the NLEN (NDEF length) field of NDEF file (see Table 9) using the ReadBinary command, starting from offset zero. The NLEN value MAY be also retrieved from the NDEF detection procedure (see Section 5.4.1).
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 5.4.4.

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

5.4.7 NDEF Update Procedure

[RQ_T4T_NDA_057] The NFC Forum Device SHALL execute the NDEF update procedure to write or update an NDEF message inside the NDEF file.

[RQ_T4T_NDA_058] The NFC Forum Device SHALL complete the following operations before the NDEF update procedure:

1. Successfully detect the NDEF message using the procedure in Section 5.4.1
2. Check that 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 offset 0007h (see Table 5 and Section 5.1.2.1)
3. Select the NDEF file (see Section 5.4.5).

[RQ_T4T_NDA_059] Table 22 defines the NDEF update command (see Table 2) to write or to update the NDEF message inside the NDEF file.

Table 22: NDEF Update – C-APDU

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

Table 23 provides a detailed description of the C-APDU fields.

Table 23: 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.
Lc	Length Lc	The number of bytes written to NDEF file. The valid range is 01h to FFh.
Le	-	[RQ_T4T_NDA_060] This field SHALL NOT be present.

Table 24 provides a description of the data structure of the R-APDU fields.

Table 24: NDEF Update – Data Structure – R-APDU

Data	SW1	SW2	Remarks
-	90h	00h	[RQ_T4T_NDA_061] Command completed; no data is returned

NOTE For more return codes and their definitions, refer to [ISO/IEC_7816-4].

[RQ_T4T_NDA_062] The NDEF update procedure is:

1. If the length of the NDEF message (to be written) is bigger than Maximum NDEF size-2 (see NFC File Control TLV in Section 5.1.2.1), the NDEF update procedure is aborted. Otherwise, go to item 2.
2. Write the value 0000h in the NLEN field (see Table 9) using the NDEF Update command.
3. Write the NDEF message in the NDEF message field (see Table 9) using one or more NDEF Update commands.
4. Write the length of the NDEF message in the NLEN field (see Table 9) using the NDEF Update command.

The NFC Forum Device MAY apply this procedure directly after the NDEF read procedure (see Section 5.4.6) has been executed.

Items 2 and 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.

5.4.8 State Changes

This section describes the possible state changes, also called transition, performed by the NFC Forum Device. Figure 2 describes the transitions of the defined states.

The transition covered by this specification is from INITIALIZED 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.

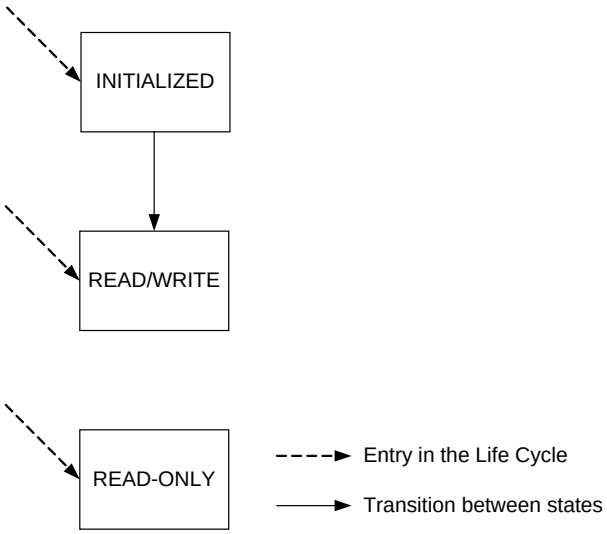

Figure 2: Life Cycle with State Changes

[RQ_T4T_NDA_063] The NFC Forum Device SHALL complete the following operation to perform the transition from INITIALIZED 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 5.4.7).

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

5.5 Coexistence of Type 4 Tag Operation Version 1.0 and Version 2.0

If the NFC Forum Device implements both Command Sequences for Mapping Version 2.0 (see Section 5.4) and Command Sequences for Mapping Version 1.0 (see Section 6.4 of [T4TOP_V1.0]), the NFC Forum Device SHALL execute the Command Sequences for Mapping Version 2.0 first and the Command Sequences for Mapping Version 1.0 second.

The NFC Forum Device SHALL NOT perform the sequence defined in Section 4.1 between the Command Sequences for Mapping Version 2.0 and the Command Sequences for Mapping Version 1.0.

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 an example of NDEF Tag Application stored inside a Type 4 Tag Platform from the NFC Forum Device point of view. Figure 3 provides an overview of the example.

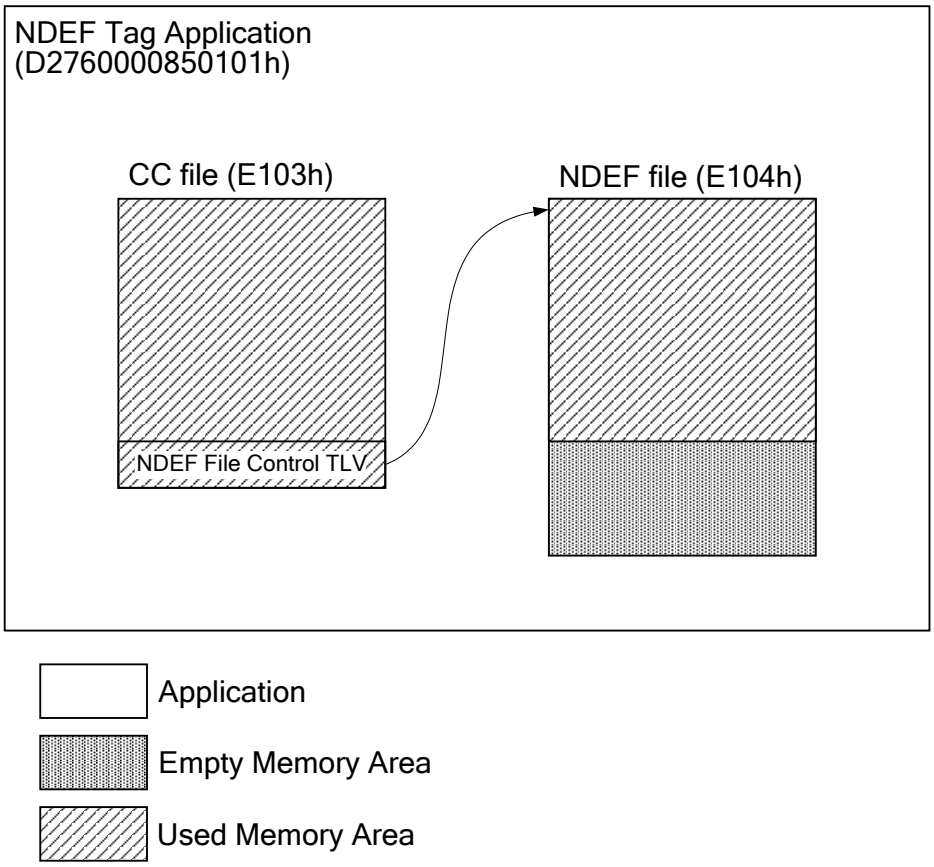

Figure 3: NDEF Tag Application for Mapping Version 2.0 Example

The Capability Container (CC) file is described in detail in Table 25.

Table 25: CC File Example

Offset	Size	Value	Content
0h	2	000Fh	CCLEN (15 bytes)
2h	1	20h	Mapping Version 2.0
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:
		E104h	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 26.

Table 26: 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 appendix 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 properly 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 Appendix 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 5.4.1).

The first command is the NDEF Tag Application select (see Section 5.4.2):

Command: 00 A4 04 00 07 D2 76 00 00 85 01 01 00

- The meanings of the bytes are:
  - 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
  - D2760000850101h NDEF Tag Application name
  - 00h Le field

  Expected Response: UU … UU 90 00
- The meanings of the bytes are:
  - UU…UUh (optional) File Control Information
  - 9000h Status bytes (SW1, SW2), command completed

The second command is the Capability Container select (see Section 5.4.3):

Command: 00 A4 00 0C 02 E1 03

- The meanings of the bytes:
  - 00h Class byte (CLA)
  - A4h Instruction byte (INS) for Select command
  - 00h Parameter byte (P1), select by identifier
  - 0Ch 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:
  - 9000h Status bytes (SW1, SW2), command completed

The third command is ReadBinary data from CC file (see Section 5.4.4):

Command: 00 B0 00 00 0F

- The meanings of the bytes are:
  - 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 20 00 3B 00 34 04 06 E1 04 00 32 00 00 90 00
- The meanings of the bytes are:
  - 00 0Fh CCLEN length of the CC file
  - 20h Mapping Version 2.0
  - 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
  - E104h 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 that the NDEF file has granted read access without any security. This allows 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 5.4.5):

Command: 00 A4 00 0C 02 E1 04

- The meanings of the bytes are:
  - 00h Class byte (CLA)
  - A4h Instruction byte (INS) for Select command
  - 00h Parameter byte (P1), select by identifier
  - 0Ch Parameter byte (P2), first or only occurrence
  - 02h Lc field
  - E104h file identifier of the NDEF file retrieved from the CC file

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

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

Command: 00 B0 00 00 02

- The meanings of the bytes are:
  - 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:
  - 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. Therefore, 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 5.4.6). It is presumed 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:
  - 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:
  - 0003h NLEN length of the NDEF message
  - D00000h NDEF message field, it contains an empty NDEF message (see 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 5.4.7). It is presumed that:

- The NDEF file was previously successfully detected (using the procedure described in Section 5.4.1).
- 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. Because 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:
  - 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:
  - 9000h Status bytes (SW1, SW2), command completed. Five data bytes have been successfully written to the NDEF file starting at offset 0000h.

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

D. Example of Mapping Version 1.0 Detection

This appendix describes how an NFC Forum Device that implements both Command Sequences for Mapping Version 2.0 (see Section 5.4) and Command Sequences for Mapping Version 1.0 (see Section 6.4 of [T4TOP_V1.0]) operates a Type 4 Tag platform supporting Mapping Version 1.0 (see also Section 5.5).

An example of the command flow is given in order to show how a typical interaction on an APDU level is performed by the NFC Forum Device. It is assumed that the Type 4 Tag Platform is configured to support Mapping Version 1.0 and contains a valid NDEF file.

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.

The first command is the NDEF Tag Application select (see Section 5.4.2):

Command: 00 A4 04 00 07 D2 76 00 00 85 01 01 00

- The meanings of the bytes are:
  - 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
  - D2760000850101h NDEF Tag Application name
  - 00h Le field

  Expected Response different from: UU…UU 9000h or 9000h
- The meanings of the bytes:
  - UU…UUh (optional) File Control Information
  - 9000h Status bytes (SW1, SW2), command completed

The first command is the NDEF Tag Application select (see [T4TOP_V1.0], Section 6.4.2 and Appendix C.1):

Command: 00 A4 04 00 07 D2 76 00 00 85 01 00

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

The command sequence continues as indicated in [T4TOP_V1.0], Appendix C.1.

E. Revision History

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

Table 27: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
Type 4 Tag Operation Specification	Version 1.0, July 2007	Final		
Type 4 Tag Operation Specification	Version 2.0, November 2010	Final	Update parameter usage of Select Commands, consistently use defined terms	Version 1.0, July 2007
Type 4 Tag Operation Specification	Version 2.0, June 2011	Final	Editorial Updates	Version 2.0, November 2010