Type 4 Tag

Technical Specification

Version 1.2

2022-08-16

[T4T]

NFC Forum™

Contents

Figures

Tables

Requirements

1 Introduction

This document is part of the NFC Forum specifications defining NFC Forum Tags.

The Type 4 Tag, as it is defined in this specification, is based on the Type 4 Tag Platform defined in [DIGITAL] and [ACTIVITY].

Since the Type 4 Tag can be based either on NFC-A or NFC-B RF technologies, the following names are used in this specification:

- Type 4 Tag, or T4T, when the statement applies for both RF technologies
- Type 4A Tag, when the statement only applies to a T4T based on NFC-A
- Type 4B Tag, when the statement only applies to a T4T based on NFC-B.

1.1 Objectives

The purpose of this specification is to define the requirements and to define, with a set of rules and guidelines:

- The Reader/Writer operation and management of a T4T
- The behavior of a T4T.

NOTE In this specification a T4T card emulation is considered identical to a T4T.

This specification also defines data mapping and how a Reader/Writer detects, reads and writes NDEF data on the Type 4 Tag in order to achieve and maintain interchangeability and interoperability.

1.2 Applicable Documents or References

[ACTIVITY]	Activity Technical Specification, NFC Forum
[ANALOG]	Analog Technical Specification, NFC Forum
[DIGITAL]	Digital Protocol Technical Specification, NFC Forum
[ISO/IEC_7816-4]	ISO/IEC 7816-4:2013, Identification cards. Integrated circuit cards. Organization, security and commands for interchange, 2013 ISO/IEC
[ISO/IEC_8825-1]	ISO/IEC 8825-1:2008, Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 2008, ISO/IEC
[NDEF]	NFC Data Exchange Format (NDEF) Technical Specification, NFC Forum

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels, RFC 2119
S. Bradner,
March 1997
Internet Engineering Task Force

[T4TOP_v1.0] Type 4 Tag Operation Technical Specification,
Version 1.0
NFC Forum

NOTE The NFC Forum [T4TOP_v1.0] is not current anymore, but the document is still available at http://members.nfc-forum.org/apps/org/workgroup/allmembers/download.php/11072/NFCForum-NCTS-Type-4-Tag_1.0.pdf.

1.3 Administration

The NFC Type 4 Tag 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 NFC Forum, Inc. maintains this specification. Comments, errors, and other feedback can be submitted at http://nfc-forum.org/our-work/specifications-and-application-documents/feedback-on-technical-specifications/.

1.4 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 only to designated members and only during the period of time for which their membership dues are paid, as stipulated in the most recent Membership Privileges document.
- Each member's distributors and sales representatives MAY use the NFC Forum logo in promoting the member's products that are 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.
- 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 NFC Forum name or logo appears:

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

1.5 Intellectual Property

The Type 4 Tag 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.

1.6 Special Word Usage

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT" and "MAY" in this specification are to be interpreted as described in [RFC2119].

1.7 Requirement Numbering

Requirements in this document are uniquely numbered with the number appearing next to each requirement. Table 1 shows an example.

Table 1: Sample Requirement

1.7.1.1	**A car SHALL have four wheels.**

A requirement can have different numbers in different versions of the specifications. Hence, all references to a requirement SHALL include the version of the document as well as the requirement's number.

1.8 Notational Conventions

1.8.1 Notations

Table 2: Notational Conventions

Notation	Description
XYh	Hexadecimal notation. Hexadecimal numbers are represented using the numbers 0 - 9 and the characters A – F. An "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
xyb	Binary notation. Binary numbers are represented by strings of the digits 0 and 1, shown with the most significant bit (msb) on the left and the least significant bit (lsb) on the right. A "b" is added at the end. Example: 11110101b
xy	Decimal notation Decimal numbers are represented without any tailing character. Example: 245
⌈..⌉	A roundup integer function is expressed by the brackets ⌈..⌉ Example: ⌈7/8⌉ = 1, ⌈8/8⌉ = 1, ⌈9/8⌉ = 2
Specially Defined Names	Terms defined in the Glossary or other NFC Technical Specification Glossaries are written with initial capital letters.
STATE	Names of defined States are written in bold all-capital **COURIER FONT** letters.
COMMAND and RESPONSE	The defined Command and Response names are written in non-bold all-capital letters.
PARAMETER	Parameter names are written in non-bold all-capital letters. Parameter names start with the following prefix: GRE_ Prefix for variables used in the Greedy Collection (e.g., GRE_POLL_A).

1.9 Abbreviations

Acronym	Definition
AID	Application Identifier
AID_NDEF	Application ID of NDEF Tag application, value D2760000850101h
APDU	Application Protocol Data Unit
C-APDU	Command APDU
CC File	Capability Container file
DDO	Discretionary Data Object (see [ISO/IEC_7816-4])
DF	Dedicated file
EF	Elementary file
FID_CC-File	File identifier of CC Files, value E103h
kb	kilobit (1024 bits)
KB	Kilobyte (1024 bytes)
Lc	Length field (value is the number of bytes in the Command Data field)
Le	Length expected (value is the maximum number of bytes expected in the Response Data field)
lsb	least significant bit
LSB	Least Significant Byte
MLc	Maximum data size that can be written by the Type 4 Tag in one WRITE Command
MLe	Maximum data size that can be read from the Type 4 Tag in one READ Command
msb	most significant bit
MSB	Most Significant Byte
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
ODO	Offset Data Object (see [ISO/IEC_7816-4])
R-APDU	Response APDU
RF	Radio Frequency
RFU	Reserved for Future Use (defined in [DIGITAL])
T4T	Type 4 Tag
TLV	Tag, Length, Value (data format)
VNo	Version number

1.10 Glossary

Application IDentifier (AID)

Defined in [ISO/IEC_7816-4], this is a specific type of Dedicated File (DF) name that is used in a SELECT Command to identify applications.

Big Endian

A method of recording or transmitting numerical data of more than one byte, with the most significant byte placed at the beginning.

Command

An instruction transmitted from one device to another device in order to move the other device through a state machine.

Correct Frame

A frame without Transmission Error.

File identifier

Data element (two bytes) used to address a file.

ISO-DEP Protocol

Half-duplex block transmission protocol defined in [DIGITAL].

Listen Mode

The mode of an NFC Forum Device where it receives Commands and sends Responses.

Listener

An NFC Forum Device in Listen Mode.

NDEF Message

The basic message construct defined by this specification. An NDEF Message contains one or more NDEF Records.

NDEF Record

An NDEF Record contains a payload described by a type, a length, and an optional identifier

NDEF Tag application

An application with AID D2760000850101h that contains all information related to storing and retrieving the NDEF Message.

NFC Forum Device

A device that supports at least one communication protocol for at least one communication mode defined by the NFC Forum specifications. Currently the following NFC Forum Devices are defined:
NFC Universal Device, NFC Tag Device and NFC Reader Device.

NFC Reader Device

An NFC Forum Device that supports the following Modus Operandi: Reader/Writer. It can also support Initiator.

NFC Tag Device

An NFC Forum Device that supports at least one communication protocol for Card Emulator and NDEF.

NFC Universal Device

An NFC Forum Device that supports the following Modus Operandi: Initiator, Target, and Reader/Writer. It can also support Card Emulator.

Operating Field

The radio frequency field created by the NFC Forum Device.

Poll Mode

The mode of an NFC Forum Device where it sends Commands and receives Responses.

Poller

An NFC Forum Device in Poll Mode.

Protocol Error

A Semantic Error or Syntax Error.

Reader/Writer

Role of a Poller when it has gone through a number of Activities. In this mode the Poller communicates with Type 2 Tags, Type 3 Tags, Type 4 Tags or Type 5 Tags.

Remote Field

The radio frequency field generated by a remote device and sensed by the NFC Forum Device.

Remote Field On

A condition of the Remote Field being stable and strong enough to put the NFC Forum Device in a state that it can operate in Passive Communication Mode. Defined in [ANALOG].

Response

Information sent from one device to another device upon receipt of a Command. The information received by the other device allows it to continue the data exchange.

R/W_VNo

Mapping Version number implemented in the NFC Forum Device.

Semantic Error

A Correct Frame with no Syntax Error is received when it is not expected.

State

A state of the Listener.

Syntax Error

A Correct Frame is received with invalid content. In this case the coding of the Command or the block within the frame is not consistent with [DIGITAL].

T4T_VNo

Mapping Version number implemented in the Type 4 Tag.

Type 4 Tag

Role of a Listener when it has gone through a number of States. In this mode the Listener supports the execution of Type 4 Tag Commands to read or write NDEF Messages.

Type 4 Tag Platform

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

Valid Command

A Command without Protocol Error within a Correct Frame.

2 Radio Frequency (RF) Interface

The RF interface is defined in [ANALOG].

Requirements 1: Analog Interface

Reader/Writer		Type 4 Tag	
2.1.1.1	The Reader/Writer SHALL comply with the analog interface for a Polling Device using NFC-A, as defined in [ANALOG].	**2.1.1.2**	The Type 4A Tag SHALL comply with the analog interface for a Listening Device using NFC-A, as defined in [ANALOG].
2.1.1.3	The Reader/Writer SHALL comply with the analog interface for a Polling Device using NFC-B, as defined in [ANALOG].	**2.1.1.4**	The Type 4B Tag SHALL comply with the analog interface for a Listening Device using NFC-B, as defined in [ANALOG].

3 Framing / Transmission Handling

This section describes the frames and transmission handling for communication with a Type 4 Tag.

3.1 Frame Structure

Requirements 2: Frame Structure

Reader/Writer		Type 4 Tag	
3.1.1.1	The Reader/Writer SHALL comply with the Sequence Format, the Bit Level coding, Frame Format, Data and Payload Format defined in [DIGITAL] for the Type 4A Tag Platform (Poll Mode).	**3.1.1.2**	The Type 4A Tag SHALL comply with the Sequence Format, the Bit Level coding, Frame Format, Data and Payload Format defined in [DIGITAL] for the Type 4A Tag Platform (Listen Mode).
3.1.1.3	The Reader/Writer SHALL comply with the Sequence Format, the Bit Level coding, Frame Format, and Data and Payload Format defined in [DIGITAL] for the Type 4B Tag Platform (Poll Mode).	**3.1.1.4**	The Type 4B Tag SHALL comply with the Sequence Format, the Bit Level coding, Frame Format, and Data and Payload Format defined in [DIGITAL] for the Type 4B Tag Platform (Listen Mode).

NOTE The activation of bitrates higher than 106 kb/s of a T4T is out of scope of the NFC Forum specifications.

3.2 Communication Protocol

This section contains the requirements for the communication protocol.

Requirements 3: Communication Protocol

Reader/Writer		Type 4 Tag	
3.2.1.1	The Reader/Writer SHALL comply with the Poll Mode requirements given in [DIGITAL] for half-duplex communication protocols.	**3.2.1.2**	The T4T SHALL comply with the Listen Mode requirements given in [DIGITAL] for half-duplex communication protocols.
3.2.1.3	The Reader/Writer SHALL comply with the ISO-DEP protocol, Poll side, defined in [DIGITAL].	**3.2.1.4**	The T4T SHALL comply with the ISO-DEP protocol, Listen Mode side, defined in [DIGITAL].

4 Memory configuration of the Type 4 Tag

4.1 File System Structure

This section describes the logical data structure for storing an NFC Forum NDEF Message on a T4T. The data structure is an [ISO/IEC_7816-4] compliant file system. This file system uses a tree structure. There are elementary files (EFs) that contain data and dedicated files (DFs) that contain EFs. All files belong to an Application Identifier (AID) and can be addressed by a two byte file identifier.

NOTE The actual storage location of data in the physical memory of the T4T is out of scope of this specification.

All relevant information related to the NDEF Tag Application is stored in the application DF with the AID_NDEF.

The data for the NDEF Tag Application is stored in two EF files:

- The Capability Container File (called “CC File” in this specification)
- The NDEF File that contains the NDEF Message.

The T4T can also contain additional files, which, if present, are listed in the CC File.

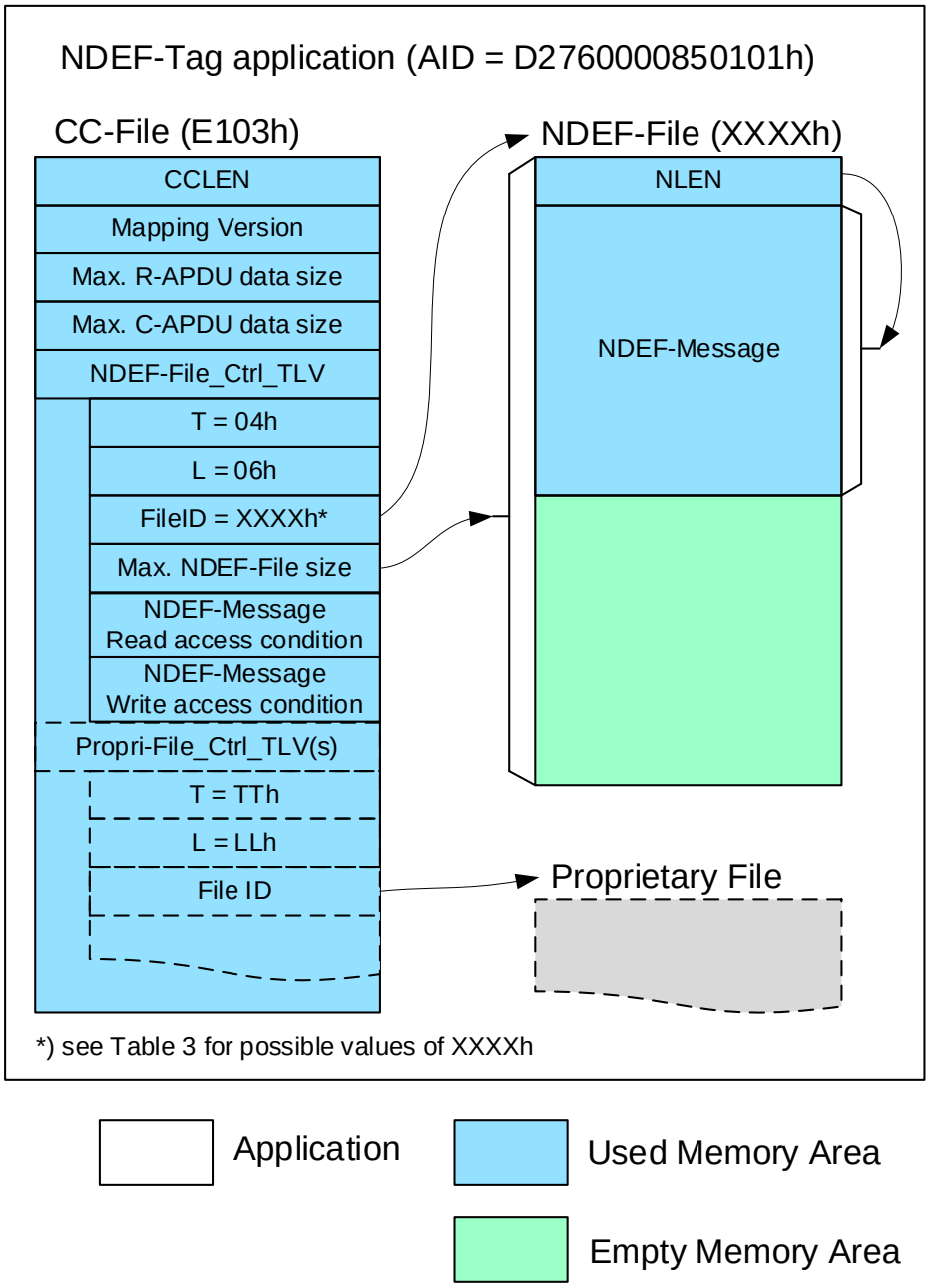

Figure 1: Example Application and File System Structure

NOTE This specification does not define the formatting of the T4T to install the required file structure (application DF with the AID_NDEF, CC File, NDEF File and optional proprietary files).

For a detailed description of the CC File see Section 4.4.

For a detailed description of the NDEF File see Section 4.5.

For a detailed description of the Proprietary File see Section 4.6.

4.2 File Identifiers and Access Conditions

Table 3 lists the file identifiers that are available in an [ISO/IEC_7816-4] compliant file system.

Table 3: File Identifiers

Value	Description
0000h	Reserved ([ISO/IEC_7816-4])
0001h – 3EFFh	Valid range
3F00h	Reserved ([ISO/IEC_7816-4])
3F01h - 3FFEh	Valid range
3FFFh	Reserved ([ISO/IEC_7816-4])
4000h – E101h	Valid Range
E102h	Reserved
E103h	FID_CC-File
E104h - FFFEh	Valid range
FFFFh	RFU ([ISO/IEC_7816-4])

Table 4 lists the READ access conditions for files within the NFC Forum application with AID_NDEF. This includes the READ access conditions for the NDEF Message.

Table 4: File READ Access Conditions

Value	Description
00h	READ access granted without any security
01h - 7Fh	RFU
80h - FEh	Limited READ access, granted based on proprietary methods
FFh	RFU

Table 5 lists the WRITE access conditions for files within the NFC Forum application with AID_NDEF. This includes the WRITE access conditions for the NDEF Message.

Table 5: File WRITE Access Conditions

Value	Description
00h	WRITE access granted without any security
01h - 7Fh	RFU
80h - FEh	Limited WRITE access, granted based on proprietary methods
FFh	No WRITE access granted at all (i.e., Read only)

4.3 Memory Mapping Versions

There are two memory Mapping Versions defined. Mapping Version 2.0 supports T4Ts with a maximum memory size of 32 KB. Mapping Version 3.0 supports T4Ts with more than 32 KB of memory.

4.3.1 Mapping Versions 2.0 and 3.0

Figure 2 shows the data structures of the NDEF File for Mapping Versions 2.0 and 3.0.

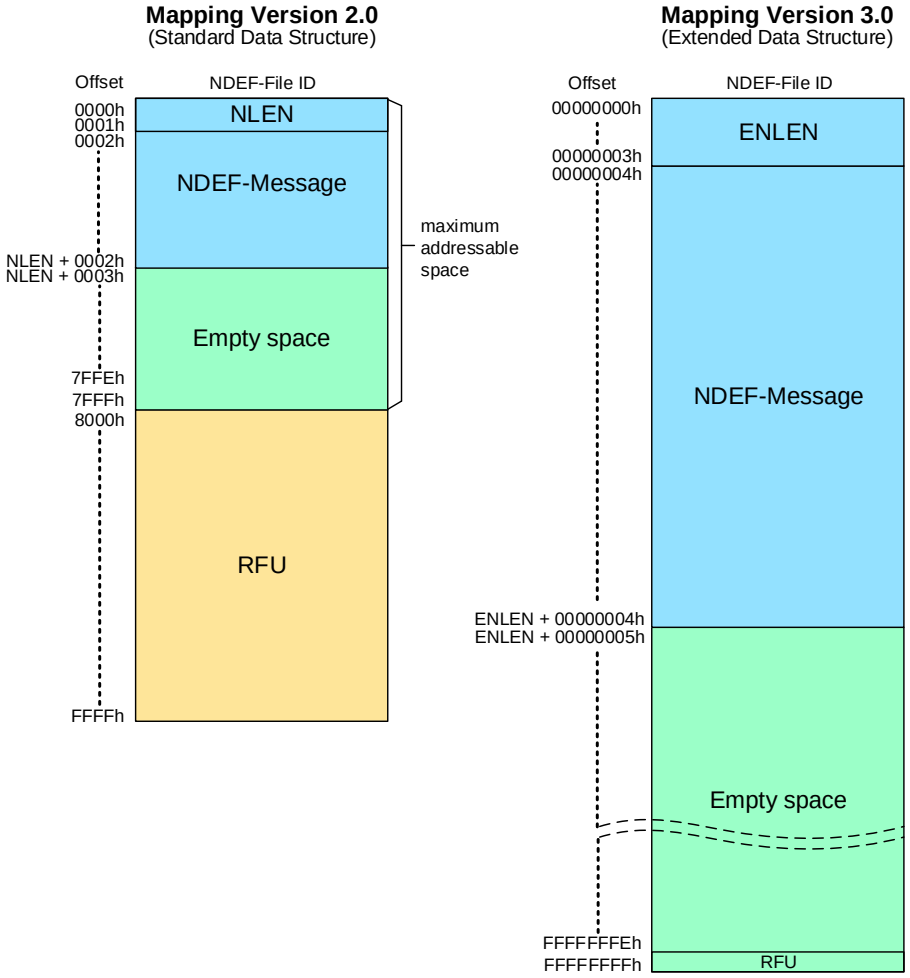

Figure 2: Memory Mapping Versions 2.0 and 3.0

The CC File contains a Mapping Version field with the Mapping Version that the T4T has implemented.

Requirements 4: Implementation of Mapping Version

Tag 4 Type	
4.3.1.1	The T4T SHALL implement Mapping Version 2.0 if the maximum size of the NDEF File is smaller than or equal to 7FFFh bytes and Mapping Version 3.0 if the maximum size of the NDEF File is equal to or larger than 8000h bytes.

NOTE The maximum size of the NDEF File depends on the maximum size that the NDEF Message can reach during the lifetime of the T4T. When the size of the NDEF Message can increase over time (for example because the NDEF Tag application adds log data to the NDEF Message), the initial NDEF File needs to be formatted with the Mapping Version that can support the maximum size of the NDEF Message.

4.3.2 Version Treatment

The Mapping Version field in the CC File contains the Mapping Version of the data structure that is implemented in the NDEF File of the T4T. The Mapping Version consists of a major version number in the four msbs and a minor version number in the four lsbs.

Table 6: Mapping Version Values

Major Version	Minor Version	Description
1	0	This version can be supported by a Reader/Writer for T4T, but this version is outdated and therefore no longer maintained. See Section 4.3.3.
1	1-F	RFU
2	0	Defined in this specification, for T4T using the Standard Data Structure of the NDEF File (See Section 4.5)
2	1-F	RFU
3	0	Defined in this specification, for T4T using the Extended Data Structure of the NDEF File (See Section 4.5)
3	1-F	RFU
4-F	0-F	RFU

Requirements 5: Compliance to Mapping Version 1.0

Reader/Writer	
4.3.2.1	If the Reader/Writer implements Mapping Version 1.0 (see [T4TOP_v1.0]), the Reader/Writer SHALL be compliant with Section 4.3.3.

An NFC Forum Reader/Writer has implemented support for a certain Mapping Version, indicated by R/W_T4T_VNo in this specification. T4T_VNo is the indication for the Mapping Version that is implemented on the T4T (as specified by the Mapping Version field of the CC File).

The following table lists the requirements for the Reader/Writer to handle the different versions of the T4Ts.

Requirements 6: Treating the Mapping Version Numbers

Reader/Writer	
4.3.2.2	If major T4T_R/W_VNo is larger than or equal to major T4T_VNo AND minor T4T_R/W_VNo is larger than or equal to minor T4T_VNo, the Reader/Writer SHALL access the T4T and SHALL use all features of the T4T_VNo defined by this T4T specification.
4.3.2.3	If major T4T_R/W_VNo is equal to major T4T_VNo) AND minor T4T_R/W_VNo is smaller than minor T4T_VNo) then possibly not all features of the T4T can be accessed. The Reader/Writer SHALL use all its features and SHALL access this T4T.
4.3.2.4	If major T4T_R/W_VNo is smaller than major T4T_VNo the data formats are incompatible. The Reader/Writer cannot understand the T4T data. The Reader/Writer SHALL conclude communication with this T4T.

4.3.3 Coexistence of Mapping Version 1.0 and Mapping Version 2.0 or Higher

Requirements 7: Coexistence of Type 4 Tag with MV 1.0 and MV 2.0 or Higher

Reader/Writer	
4.3.3.1	If the Reader/Writer implements both Command Sequences for Mapping Version 2.0 (or higher, see Section 7.5) and Command Sequences for Mapping Version 1.0 (see Section 6.4 of [T4TOP_v1.0]), the Reader/Writer SHALL execute the Command Sequences for Mapping Version 2.0 (or higher) first and the Command Sequences for Mapping Version 1.0 second.

NOTE The difference between [T4TOP_v1.0] and this specification is the definition of the value for the AID of the NDEF Tag application.
For Mapping Version 1.0 the value of the AID is specified as “D2760000850100h” and the Le field is not present in the select NDEF Tag application C-APDU.
For Mapping Version 2.0 the value of the AID is specified as “D2760000850101h” and the Le field can be present in the select NDEF Tag application C-APDU, allowing file control information to be returned in the Data field.

4.4 CC File

The CC File contains all information for reading and writing an NDEF Message. It is stored as a read-only EF with FID_CC-File.

Table 7 defines the data structure of the CC File.

Table 7: Data Structure of the CC File

Offset (bytes)	Size (bytes)	Field	Description	
0000h	2	CCLEN (bytes)	Indicates the size of this CC (including this field).	
			Value	**Description**
			0000h-000Eh	RFU
			000Fh-7FFFh	Valid range
			8000h-FFFFh	RFU
0002h	1	T4T_VNo	Indicates the Mapping Version that is implemented on the T4T (see Section 4.5). Section 4.3.2 defines how the Reader/Writer needs to handle different Mapping Versions.	
			Value	**Description**
			20h	Mapping Version 2.0 (with the Standard Data Structure)
			30h	Mapping Version 3.0 (with the Extended Data Structure)
0003h	2	MLe (bytes); Maximum R-APDU data size	Defines the maximum data size that can be read from the T4T using a single READ_BINARY Command.	
			Value	**Description**
			0000h-000Eh	RFU
			000Fh-FFFFh	Valid range
0005h	2	MLc (bytes); Maximum C-APDU data size	Defines the maximum data size that can be sent to the T4T using a single Command.	
			Value	**Description**
			0000h-000Ch	RFU
			000Dh-FFFFh	Valid range (Needs at least to be able to send the Select NDEF Tag Application C-APDU to the T4T.)

Offset (bytes)	Size (bytes)	Field	Description
0007h	8	NDEF-File_Ctrl_TLV	Section 4.7.3 specifies the content of the NDEF-File_Ctrl_TLV block that contains information to control and manage the NDEF File for Mapping Version 2.0 (see Section 4.5).
	or	or	or
	10	ENDEF-File_Ctrl_TLV	Section 4.7.4 specifies the content of the ENDEF-File_Ctrl_TLV block that contains information to control and manage the NDEF File for Mapping Version 3.0 (see Section 4.5).
-	-	Control TLV(s)	Zero or more TLV blocks that contain information to control and manage proprietary files (see Section 4.6). Sections 4.7.5 and 4.7.6 specify the content of the Proprietary-File_Ctrl_TLV and the EProprietary-File_Ctrl_TLV, respectively.

Unless defined otherwise, the term “NDEF File” in the following sections refers to the NDEF File indicated by the NDEF-File_Ctrl_TLV or the ENDEF-File_Ctrl_TLV.

Requirements 8: CC File

Reader/Writer		Type 4 Tag	
4.4.1.1	The Reader/Writer SHALL read the CC File, with FID_CC-File.	**4.4.1.2**	The T4T SHALL contain a Read-Only CC File, with FID_CC-File.
4.4.1.3	The Reader/Writer SHALL parse the mandatory fields of the CC File.	**4.4.1.4**	The T4T SHALL have the CC File with the mandatory fields: CCLEN, T4T_VNo, MLe, MLc, and an NDEF File_Ctrl_TLV, as specified in Table 7.
4.4.1.5	The Reader/Writer SHALL respect the values of the MLe and MLc fields for communication with the T4T.	**4.4.1.6**	T4T_VNo SHALL have the value 20h or 30h.
4.4.1.7	The Reader/Writer SHALL verify that T4T_VNo has value 20h or 30h.		
4.4.1.8	The Reader/Writer MAY parse the remainder of the CC File.	**4.4.1.9**	The T4T MAY have additional Control TLV blocks located after the mandatory fields listed in 4.4.1.4.

4.5 NDEF File

The NDEF File stores the length and the content of the NDEF Message, which is defined by [NDEF].

The data structure of the NDEF File is defined by the Mapping Version:

- Table 8 defines Mapping Version 2.0 with the Standard Data Structure, to store an NDEF Message with a maximum size between 3h and 7FFDh bytes.
- Table 9 defines Mapping Version 3.0 with the Extended Data Structure, to store an NDEF Message that has a maximum size between 3h and FFFFFFFAh bytes.

The CC File indicates the Mapping Version that the T4T has implemented.

NOTE In this specification the bit and byte ordering in the definitions of multi-byte data structures, packets and Messages follows the Big Endian byte order, unless they are defined otherwise.

Table 8: NDEF File with Mapping Version 2.0; Standard Data Structure

Offset (bytes)	Size (bytes)	Field	Description	
0h	2	NLEN	The NLEN (NDEF length) field indicates the size of the NDEF Message stored in the NDEF File, in bytes.	
			Value	**Description**
			0000h	T4T in **`INITIALIZED`** State
			0001h - 0002h	RFU
			0003h - 7FFDh	Valid range
			7FFEh-FFFFh	RFU
2h	NLEN	NDEF Message	NDEF Message (see [NDEF]).	

Table 9: ENDEF File with Mapping Version 3.0; Extended Data Structure

Offset (bytes)	Size (bytes)	Field	Description	
0h	4	ENLEN	The ENLEN (Extended NDEF length) field indicates the size of the NDEF Message stored in the ENDEF File, in bytes.	
			Value	**Description**
			00000000h	T4T in **`INITIALIZED`** State
			00000001h - 00000002h	RFU
			00000003h - FFFFFFFAh	Valid range
			FFFFFFFBh - FFFFFFFFh	RFU
4h	ENLEN	NDEF Message	NDEF Message (see [NDEF]).	

NOTE The minimum size of the NDEF Message is three bytes, for the empty NDEF Message. See Appendix B for the definition of the empty NDEF Message.

Requirements 9: NDEF File

Type 4 Tag	
4.5.1.1	A T4T with value 20h in T4T_VNo SHALL contain an NDEF File using the Standard Data Structure defined in Table 8.
4.5.1.2	A T4T with value 30h in T4T_VNo SHALL contain an NDEF File using the Extended Data Structure defined in Table 9.

4.6 Proprietary File

The Proprietary File is an EF file (see [ISO/IEC_7816-4]) that contains the length and the content of proprietary data. A T4T can contain zero or more Proprietary Files.

The Proprietary File contains one of the following two data structures:

- Table 10 defines Mapping Version 2.0 with the Standard Data Structure, to store proprietary data with a size between 0001h and 7FFDh bytes.
- Table 11 defines Mapping Version 3.0 with the Extended Data Structure, to store proprietary data, with a size between 0001h and FFFFFFFAh bytes.

The CC File indicates the Mapping Version that a T4T supports.

Table 10: Proprietary File with Mapping Version 2.0; Standard Data Structure

Offset (bytes)	Size (bytes)	Field	Description	
0h	2	PLEN	The Proprietary Length field (PLEN) indicates the size of the proprietary data, in bytes.	
			Value	**Description**
			0000h	RFU
			0001h- 7FFDh	Valid range
			7FFEh-FFFFh	RFU
2h	PLEN	Proprietary data	Proprietary data	

Table 11: EProprietary File with Mapping Version 3.0; Extended Data Structure

Offset (bytes)	Size (bytes)	Field	Description	
0h	4	EPLEN	The Extended Proprietary Length field (EPLEN) indicates the size of the proprietary data, in bytes.	
			Value	**Description**
			00000000h	RFU
			00000001h-FFFFFFFAh	Valid range
			FFFFFFFBh-FFFFFFFFh	RFU
4h	EPLEN	Proprietary Data	Proprietary data	

Requirements 10: Proprietary File

Reader/Writer		Type 4 Tag	
4.6.1.1	The Reader/Writer MAY access the content of Proprietary File(s), if present on the T4T.	**4.6.1.2**	A T4T with value 20h in its T4T_VNo MAY contain a Proprietary File using the Standard Data Structure defined in Table 10.
		4.6.1.3	A T4T with the value 30h in its T4T_VNo MAY contain an EProprietary File using the Extended Data Structure defined in Table 11.

4.7 File Control TLVs

This specification defines several Control TLVs that can be used on a T4T.

4.7.1 File Control TLV structure

The Control TLVs for this specification are defined as SIMPLE-TLV data objects (see [ISO/IEC_7816-4]) and consist of three fields:

- T The Tag field (T-field) encodes the type of the TLV structure in one byte. Table 12 defines the values for the encoding of the T-field for this specification.
- L The Length field (L-field) encodes the size of the V-field in one byte.
- V The Value field (V-field) contains the data for the Control TLV.

If the L-field has value N, then the V-field consists of N consecutive bytes.

4.7.2 List of File Control TLVs

Table 12 lists the Control TLVs that are defined for the T4T.

Table 12: File Control TLVs Defined in this Specification

TLV structure name	Tag Field Value	Short Description
	00h-03h	RFU
NDEF-File_Ctrl_TLV	04h	Encodes the use of the Standard Data Structure for the NDEF File.
Proprietary-File_Ctrl_TLV	05h	Encodes the use of the Standard Data Structure for the Proprietary File
ENDEF-File_Ctrl_TLV	06h	Encodes the use of the Extended Data Structure for the NDEF File.
EProprietary-File_Ctrl_TLV	07h	Encodes the use of the Extended Data Structure for the Proprietary File.
	08h-FFh	RFU

Requirements 11: File Control TLVs

Reader/Writer		Type 4 Tag	
4.7.2.1	The Reader/Writer MAY ignore the content of the V-field of the Control TLVs using a T-field with the values 05h, 07h or any value defined as RFU, and jump over these Control TLVs to check if there are other TLVs present in the CC File.	**4.7.2.2**	A T4T with value 20h in T4T_VNo SHALL contain an NDEF-File_Ctrl_TLV (defined in Table 13), positioned at offset 07h in the CC File.
		4.7.2.3	A T4T with value 30h T4T_VNo SHALL contain an ENDEF-File_Ctrl_TLV (defined in Table 14), positioned at offset 07h in the CC File.

4.7.3 NDEF-File_Ctrl_TLV

The NDEF-File_Ctrl_TLV is present inside the CC File of a T4T that indicates support for Mapping Version 2.0. Table 13 shows the encoding of the three fields of the NDEF-File_Ctrl_TLV:

Table 13: NDEF-File_Ctrl_TLV

Field	Length	Value	Description		
T	1 byte	04h	Indicates the NDEF-File_Ctrl_TLV		
L	1 byte	06h	The length of the V-field is 6 bytes		
V	6 bytes	**Parameter**	**Length**	**Value**	**Description**
		NDEF File Identifier	2 bytes	See Table 3	File identifier of the NDEF File
		NDEF File Size	2 bytes	0000h - 0004h	RFU
				0005h - 7FFFh	Valid range
				8000h - FFFFh	RFU
		NDEF File READ Access Condition	1 byte	See Table 4	See Table 4
		NDEF File WRITE Access Condition	1 byte	See Table 5	See Table 5

NOTE The NDEF File Size parameter contains the size of the memory allocated to the storage of the NDEF Message, not the size of the NDEF Message itself.

4.7.4 ENDEF-File_Ctrl_TLV

The Extended NDEF-File_Ctrl_TLV is present inside the CC File of a T4T that indicates support for Mapping Version 3.0. Table 14 shows the encoding of the three fields of the ENDEF-File_Ctrl_TLV.

Table 14: ENDEF-File_Ctrl_TLV

Field	Length	Value	Description		
T	1 byte	06h	Indicates the ENDEF-File_Ctrl_TLV		
L	1 byte	08h	The length of the V-field is 8 bytes		
V	8 bytes	**Parameter**	**Length**	**Value**	**Description**
		ENDEF File Identifier	2 bytes	See Table 3.	File identifier of the ENDEF File
		ENDEF File Size	4 bytes	00000000h - 00000006h	RFU
				00000007h - FFFFFFFEh	Valid range
				FFFFFFFFh	RFU
		ENDEF File READ Access Condition	1 byte	See Table 4.	See Table 4.
		ENDEF File WRITE Access Condition	1 byte	See Table 5.	See Table 5.

NOTE The ENDEF File Size parameter contains the size of the memory allocated to the storage of the NDEF Message, not the size of the NDEF Message itself.

4.7.5 Proprietary-File_Ctrl_TLV

The Proprietary-File_Ctrl_TLV is present inside the CC File of a T4T that indicates support for Mapping Version 2.0. Table 15 shows the encoding of the three fields of the Proprietary-File_Ctrl_TLV.

Table 15: Proprietary-File_Ctrl_TLV

Field	Length	Value	Description		
T	1 byte	05h	Indicates the Proprietary-File_Ctrl_TLV		
L	1 byte	06h	The length of the V-field is 6 bytes		
V	6 bytes	**Parameter**	**Length**	**Value**	**Description**
		Proprietary File Identifier	2 bytes	See Table 3	See Table 3
		Proprietary File Size	2 bytes	0000h - 0002h	RFU
				0003h - 7FFFh	Valid range
				8000h - FFFFh	RFU
		Proprietary File READ Access Condition	1 byte	See Table 4	See Table 4
		Proprietary File WRITE Access Condition	1 byte	See Table 5	See Table 5

NOTE The Proprietary File Size parameter contains the size of the memory allocated to the storage of proprietary data, not the size of the proprietary data itself.

4.7.6 EProprietary-File_Ctrl_TLV

The EProprietary-File_Ctrl_TLV is present inside the CC File of a T4T that indicates support for Mapping Version 3.0. Table 16 shows the encoding of the three fields of the EProprietary_File_Ctrl_TLV.

Table 16: EProprietary-File_Ctrl_TLV

Field	Length	Value	Description		
T	1 byte	07h	Indicates the EProprietary-File_Ctrl_TLV		
L	1 byte	08h	The length of the V-field is 8 bytes		
V	8 bytes	**Parameter**	**Length**	**Value**	**Description**
		EProprietary File Identifier	2 bytes	See Table 3	File identifier of the EProprietary File
		EProprietary File Size	4 bytes	00000000h - 00000004h	RFU
				00000005h - FFFFFFFEh	Valid range
				FFFFFFFFh	RFU
		EProprietary File READ Access Condition	1 byte	See Table 4	See Table 4
		EProprietary File WRITE Access Condition	1 byte	See Table 5	See Table 5

NOTE The EProprietary File Size parameter contains the size of the memory allocated to the storage of proprietary data, not the size of the proprietary data itself.

5 Command Set

NOTE This section describes the Commands to access the NDEF Message.

5.1 Basic Command Set

5.1.1 Selection of ISO/IEC 7816-4 Commands

The Reader/Writer can access the EF Files using a subset of the [ISO/IEC_7816-4] Commands. Table 17 lists the basic Command set, Table 18 lists the Command set that can be used to access the data in the file using Offset-Data-Object (ODO).

Table 17: Basic Command Set

Command	Instruction Code	Description
SELECT	A4h	Selection of applications or files
READ_BINARY	B0h	Read data from file
UPDATE_BINARY	D6h	Update (erase and write) data to file

Table 18: Command Set for File Access using ODO

Command	Instruction Code	Description
READ_BINARY	B1h	Read data from file using ODO
UPDATE_BINARY	D7h	Update (erase and write) data to file using ODO

NOTE Instruction codes B1h and D7h use the BER-TLV to encapsulate the data and Response according to [ISO/IEC_7816-4].

The Commands from the Reader/Writer to the T4T are called Command-APDUs (referred to as C-APDUs in this specification). Section 5.1.2 defines the generic format.

The Responses from the T4T to the Reader/Writer are called Response-APDUs (referred to as R-APDUs in this specification). Section 5.1.3 defines the generic format.

Section 7 defines the Commands and Responses to detect and access the data.

Requirements 12: File Access Command

Reader/Writer		Type 4 Tag	
5.1.1.1	To access files on a T4T with Mapping Version 2.0 the Reader/Writer SHALL use only the Commands listed in Table 17.	**5.1.1.2**	A T4T with value 20h in T4T_VNo SHALL support all of the Commands listed in Table 17 and SHALL implement the Commands defined in [ISO/IEC_7816-4].
5.1.1.3	To access files on a T4T with Mapping Version 3.0 the Reader/Writer SHALL use only the Commands listed in Table 17 or Table 18.	**5.1.1.4**	A T4T with value 30h in T4T_VNo SHALL support all of the Commands listed in Table 17 and Table 18 and SHALL implement the Commands defined in [ISO/IEC_7816-4].
5.1.1.5	When it sends a Command to the T4T, the Reader/Writer SHALL be able to receive from the T4T a Response that contains an error indication in the status bytes defined in [ISO/IEC_7816-4] (even when they are not defined in this specification) and continue operation.	**5.1.1.6**	After it receives a Command, the T4T SHALL respond with an error indication in the status bytes defined in [ISO/IEC_7816-4] when it does not support the Command or the content in one of the Command's parameters.

5.1.2 Format of the Command-APDU

Table 19 defines the format of the C-APDU.

Table 19: 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 (CLA): Contains 00h, because this specification does not use secure messaging.

Instruction byte (INS): Encodes the Command to process.

Parameter byte 1 (P1): Contains 00h, if no other value is specified for the instruction.

Parameter byte 2 (P2): Contains 00h, if no other value is specified for the instruction.

Data field length (Lc): optional. If Lc is present, it encodes the number of bytes in the Data field of the Command, encoded as defined in Table 20.

Data field: Optional.

Expected Response Length (Le): optional. If Le field is absent, then there are no data bytes expected in the Response Body field of the R-APDU. If Le is present, it encodes the maximum length of the Response Body field in the Response R-APDU (see Section 5.1.3), encoded as defined in Table 21. The use of Short field coding or Extended Field coding is defined by the coding used for Lc (see Sections 5.1.2.6 and 5.1.2.12).

Table 20: Coding of Lc field

Lc Coding	Length	Value	Description
Lc absent	0	–	The number of bytes in the Command Data field zero.
Short Field coding	1 byte	00h	Reserved for Extended Field coding.
		01h - FFh	Valid range: encodes the number of bytes in the Data field between 1 and 255.
Extended Field coding	3 bytes	00h	First byte is always 00h.
		0001h - FFFFh	Valid range: encodes the number of bytes in the Data field between 1 and 65535.

Table 21: Coding of Le field

Le Coding	Length	Value	Description
Le absent	0	–	The maximum number of bytes expected in the Response Data field is zero
Short Field coding	1 byte	01h - FFh	Valid range: encodes the maximum number of bytes expected between 1 and 255
		00h	Encodes the maximum number of bytes expected equal to 256
Extended Field coding with extended Lc field present	2 bytes	0001h-FFFFh	Valid range: encodes the maximum number of bytes expected between 1 and 65535
		0000h	Encodes the maximum number of bytes expected equal to 65536
Extended Field coding with absent Lc field	3 bytes	00h	First byte is always 00h
		0001h-FFFFh	Valid range: encodes the maximum number of bytes expected between 1 and 65535.
		0000h	Encodes the maximum number of bytes expected equal to 65536

Requirements 13: C-APDU

Reader/Writer	
5.1.2.1	The Reader/Writer SHALL set the Class byte to 00h (no secure messaging).
5.1.2.2	If the Data field length (Lc) is present, the Reader/Writer SHALL set the value of Lc to the number of bytes in the Data field, as defined in Table 20.
5.1.2.3	If the Expected Response Length (Le) is present, the Reader/Writer SHALL set the value of Le to the maximum number of data bytes in the Response Body field of the R-APDU (see Section 5.1.3), as defined in Table 21.
5.1.2.4	If the T4T responds to a READ_BINARY Command with the error indication SW1 = 6Ch and the SW2 byte set to any value in the status bytes (as defined in [ISO/IEC_7816-4]), the Reader/Writer SHALL repeat the Command with the same parameters, except that the byte Le MAY be set to any value.
5.1.2.5	If the T4T responds to a READ_BINARY Command with the error indication SW1 = 67h and SW2 = 00h in the status bytes (as defined in [ISO/IEC_7816-4]), the Reader/Writer SHALL repeat the Command once and SHALL set Le to a value not exceeding the number of bytes left in the selected File from the File offset specified in the READ_BINARY Command.

Requirements 14: C-APDU for Mapping Version 2.0

Reader/Writer		Type 4 Tag	
5.1.2.6	If the Reader/Writer communicates with a T4T that has implemented Mapping Version 2.0, it SHALL support Short Field coding for Lc and Le, and it MAY support Extended Field coding for Lc and Le.	**5.1.2.7**	A T4T with value 20h in T4T_VNo SHALL support Short Field coding for Lc and Le.
		5.1.2.8	A T4T that supports only Short Field coding SHALL assign a value for MLc smaller than or equal to a Data field length of 255 bytes and a value for MLe smaller than or equal to an expected Response length of 256 bytes.
		5.1.2.9	A T4T that assigns a value for MLc larger than a Data field length of 255 bytes and/or a value for MLe larger than an expected Response length of 256 bytes SHALL additionally support Extended Field coding.

Requirements 15: C-APDU for Mapping Version 3.0

Reader/Writer		Type 4 Tag	
5.1.2.10	If the Reader/Writer communicates with a T4T that has implemented Mapping Version 3.0, it SHALL support Short and it MAY support Extended Field coding for Lc and Le.	**5.1.2.11**	A T4T with value 30h in the Mapping Version field of the CC File SHALL support Short and it MAY support Extended Field coding for Lc and Le.
5.1.2.12	If the Reader/Writer communicates with a T4T that has implemented Mapping Version 3.0, it SHALL use the same field coding (i.e., Short or Extended) for the Lc and Le fields within a single C-APDU.	**5.1.2.13**	A T4T with value 30h in the Mapping Version field of the CC File SHALL reject a C-APDU which does not use the same field coding (i.e., Short or Extended) for the Lc and Le fields by sending an error condition in the status bytes.
5.1.2.14	If the Reader/Writer communicates with a T4T that has implemented Mapping Version 3.0, it SHALL NOT use the Extended Field coding for Lc and Le until it has read the values of MLc and MLe parameters in the CC File, and these values are above 255 bytes for MLc and above 256 bytes for MLe.	**5.1.2.15**	A T4T with value 30h in the Mapping Version field of the CC File MAY report a value above 255 bytes for MLc and a value above 256 bytes for MLe in the CC File.

5.1.3 Format of Response-APDU

Table 22 defines the format of the R-APDU for this specification.

Table 22: Format of R-APDU

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

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

Response Status bytes: bytes SW1 and SW2 are mandatory.

NOTE Return code 90h for SW1 and 00h for SW2 indicates success.
Other return codes and their definitions are defined in [ISO/IEC_7816-4].

5.2 Select Data Commands

5.2.1 Select NDEF Tag Application

Table 23 defines the C-APDU to select the NDEF Tag application.

Table 23: C-APDU to Select NDEF Tag Application

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

Table 24 provides a description of the C-APDU fields.

Table 24: C-APDU Fields for Select NDEF Tag Application

Field	Data	Remarks
P1	04h	Select by name
P2	00h	First or only occurrence
Lc	07h	7 bytes in Data field
Data	D2760000850101h	AID_NDEF
Le	00h	Request to return all bytes that are available

Table 25 provides a description of the R-APDU fields.

Table 25: R-APDU Fields for Select NDEF Tag Application

Data	SW1	SW2	Remarks
File control information can be returned	90h	00h	Command completed; it is optional to return file control information in the Data field
-	6Ah	82h	NDEF Tag Application not found; no data returned.

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

Requirements 16: Select NDEF Tag Application

Reader/Writer		Type 4 Tag	
5.2.1.1	The Reader/Writer SHALL use the Command defined in Table 23 to select the NDEF Tag Application.	**5.2.1.2**	The T4T SHALL respond to a Select the NDEF Tag Application Command with an R-APDU, as defined in Table 25.
5.2.1.3	The receipt of the 'Command Completed' R-APDU, as defined in Table 25, is the confirmation that the NDEF Tag Application has been selected.		

5.2.2 Select CC File

Table 26 defines the C-APDU to select the CC File.

Table 26: C-APDU to Select CC File

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

Table 27 provides a description of the C-APDU fields.

Table 27: C-APDU Fields for Select CC File

Field	Data	Remarks
P1	00h	Select by file identifier
P2	0Ch	First or only occurrence
Lc	02h	2 bytes in Data field
Data	E103h	FID_CC-File
Le	-	Not present

Table 28 provides a description of the R-APDU fields.

Table 28: R-APDU Fields for Select CC File

Data	SW1	SW2	Remarks
-	90h	00h	Command completed; no data returned.
-	6Ah	82h	CC File not found; no data returned.

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

Requirements 17: Select CC File

Reader/Writer		Type 4 Tag	
5.2.2.1	To Select the CC File, the Reader/Writer SHALL use the Command defined in Table 26.	**5.2.2.2**	The T4T SHALL respond to a Valid Command to Select the CC File with an R-APDU that is indicating 'Command completed', as defined in Table 28.
5.2.2.3	The receipt of the 'Command completed' R-APDU, as defined in Table 28, is the confirmation that the CC File has been selected.		

5.2.3 Select NDEF File

Table 29 defines the Command to select the NDEF File.

Table 29: C-APDU to Select NDEF File

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

Table 30 provides a description of the C-APDU fields.

Table 30: C-APDU Fields for Select NDEF File

Byte	Data	Remarks
P1	00h	Select by file identifier
P2	0Ch	First or only occurrence
Lc	02h	2 bytes in Data field
Data	File identifier	File identifier of the NDEF File, as indicated by offset 0009h in the CC File.
Le	-	Not present

Table 31 provides a description of the R-APDU fields.

Table 31: R-APDU Fields for Select NDEF File

Data	SW1	SW2	Remarks
-	90h	00h	Command completed; no data returned.
-	6Ah	82h	NDEF File not found; no data returned.

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

Requirements 18: Select NDEF File

Reader/Writer		Type 4 Tag	
5.2.3.1	The Reader/Writer SHALL use the Command defined in Table 29, to select the NDEF File indicated by offset 0009h in the CC File.	**5.2.3.2**	The T4T SHALL respond with an R-APDU that is indicating 'Command completed' (as defined in Table 31**)** to a Select NDEF File C-APDU.
5.2.3.3	The receipt of the 'Command completed' R-APDU, as defined in Table 31, is the confirmation that the NDEF File has been selected.		

5.3 Read Data Commands

5.3.1 Read Data from File with Mapping Version 2.0

Table 32 defines the READ_BINARY Command to read the data from a file with Mapping Version 2.0.

Table 32: C-APDU for READ_BINARY Command

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

Table 33 provides a description of the C-APDU fields.

Table 33: C-APDU Fields for READ_BINARY Command

Field	Data	Remarks
P1/P2	Offset	File offset of where to start reading data; valid range is 0000h-7FFFh.
Lc	-	Not present.
Data	-	Not present.
Le	Length	Expected Response Length encoded as defined in Table 21.

NOTE The CC File contains the value for MLe. This implies that the first READ_BINARY Command to read the CC File has to use short length field encoding.

Table 34 provides a description of the R-APDU fields.

Table 34: R-APDU Fields for the READ_BINARY Command

Data	SW1	SW2	Remarks
Content read	90h	00h	Command completed.
-	67h	00h	Wrong length; no further indication.
-	6Ch	XXh	Wrong Le field; SW2 encodes the exact number of available data bytes.

Requirements 19: Reading Data from a File with Mapping Version 2.0

Reader/Writer		Type 4 Tag	
5.3.1.1	When using the READ_BINARY Command, the Reader/Writer SHALL format it according to Table 32 and Table 33.	**5.3.1.2**	The T4T SHALL respond to a Valid READ_BINARY Command with an R-APDU that contains one of the options that are defined in Table 34.
5.3.1.3	The Reader/Writer SHALL configure the number of data bytes to be read within a single READ_BINARY Command equal to or smaller than MLe (i.e. Le ≤ MLe).		

5.3.2 Read Data from File with Mapping Version 3.0

When the T4T contains an NDEF File with the Extended Data Structure, the Reader/Writer can use the Offset Data Object feature of the READ_BINARY Command, as defined in Table 35.

Table 35: C-APDU for READ_BINARY Command with ODO

CLA	INS	P1	P2	Lc	Data	Le
00h	B1h	00h	00h	Length Data	54 03 xxyyzz	Length

Table 36 provides a description of the C-APDU.

Table 36: C-APDU Fields for READ_BINARY Command with ODO

Field	Data	Remarks
Lc	Length Data	The number of bytes in the Data field, encoded as defined in Table 20.
Data	5403xxyyzz	This field contains the Offset Data Object, which includes a 3-byte offset value indicated as xxyyzz. The valid range is 000000h - FFFFFEh.
Le	Length Le	Expected Response Length encoded as defined in Table 21.

Table 37 provides a description of the R-APDU fields.

Table 37: R-APDU Fields for READ_BINARY Command with ODO

Data	SW1	SW2	Remarks
[53h, length of Content read, Content read]	90h	00h	Command completed.

NOTE If INS = 'B1', the content data must be encapsulated in a DDO (tag 53h), see [ISO/IEC_7816-4].

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

Requirements 20: Reading Data from a File with Mapping Version 3.0

Reader/Writer		Type 4 Tag	
5.3.2.1	When using the READ_BINARY Command with ODO, the Reader/Writer SHALL format it according to Table 35 and Table 36.	**5.3.2.2**	A T4T with the value 30h in the Mapping Version field of the CC File SHALL respond to a Valid READ_BINARY Command with ODO with an R-APDU, as defined in Table 37.
5.3.2.3	The Reader/Writer SHALL NOT use the READ_BINARY Command with ODO unless the T4T has the value 30h in the Mapping Version field of the CC File.		
5.3.2.4	The Reader/Writer SHALL configure the number of data bytes to be read within a single READ_BINARY Command with ODO equal to or smaller than MLe (i.e., Le ≤ MLe).		

5.4 Write Data Commands

5.4.1 Write Data to NDEF File with Mapping Version 2.0

NOTE This section describes how to write data to the NDEF File, but the Commands can also be used to write data to the Proprietary File that has been selected with a Select NDEF File Command.

Table 38 defines the UPDATE_BINARY Command to write the NDEF Message with a Standard Data structure.

Table 38: C-APDU for the UPDATE_BINARY Command

CLA	INS	P1	P2	Lc	Data	Le
00h	D6h	[Offset]		Length Data	Data to be written	–

Table 39 provides a description of the C-APDU fields.

Table 39: C-APDU Fields for the UPDATE_BINARY Command

Field	Data	Remarks
P1/P2	Offset	Offset in bytes from the beginning of the NDEF File to start writing the data. The valid range is 0000h to 7FFFh (0 to 32768).
Lc	Length Data	One byte encoding the number of bytes in the Data field using the Short length coding of Lc, as defined in Table 20.
Data	Data to be written	Data to be written to the NDEF File.
Le	–	Not present.

Table 40 provides a description of the R-APDU fields for the UPDATE_BINARY Command.

Table 40: R-APDU Fields for the UPDATE_BINARY Command

Data	SW1	SW2	Remarks
-	90h	00h	Command completed; no data returned.

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

Requirements 21: Writing Data to the NDEF File with Mapping Version 2.0

Reader/Writer		Type 4 Tag	
5.4.1.1	When using the UPDATE_BINARY Command, the Reader/Writer SHALL format it according to Table 38 and Table 39.	**5.4.1.2**	The T4T SHALL respond to a Valid UPDATE_BINARY Command with an R-APDU that is indicating 'Command completed', as defined in Table 40.
5.4.1.3	The Reader/Writer SHALL set the number of data bytes, to be written with a single UPDATE_BINARY Command, smaller than or equal to MLc.		

5.4.2 Write Data to NDEF File with Mapping Version 3.0

NOTE This section describes how to write data to the NDEF File, but the Commands can also be used to write data to the EProprietary File that has been selected with a Select NDEF File Command.

When the T4T has an NDEF File using the Extended Data structure, the Reader/Writer is unable to access the whole NDEF File, since the offset and the number of bytes that can be written from this offset are limited. In that case, the Reader/Writer can then use the Offset Data Object and Discretionary Data Object features of the UPDATE_BINARY Command, as defined in Table 41.

Table 41: C-APDU for the UPDATE_BINARY Command with ODO and DDO

CLA	INS	P1	P2	Lc	Data	Le
00h	D7h	00h	00h	Length Data	54 03 xxyyzz 53 Ld {Data to be written to the ENDEF File}	-

Table 42 provides a description of the C-APDU fields.

Table 42: C-APDU Fields for UPDATE_BINARY with ODO and DDO

Field	Data	Remarks
Lc	Length Data	Contains the number of bytes in the Data field using the encoding of Lc defined in Table 20.
Data	54 03 xxyyzz 53 Ld {data to be written to the ENDEF File}	The Data field contains: 1. Offset Data Object (tag ´54´ see [ISO/IEC_7816-4]) with a 3-byte Offset xxyyzz that has 000000h-FFFFFEh as a valid range. 2. Discretionary Data Object (tag ´53´, see [ISO/IEC_7816-4]), which contains the data to be written in the NDEF File. The length Ld indicates the length of the data and can be one or more bytes long.
Le	-	Not present.

Table 43 provides a description of the R-APDU fields for the UPDATE_BINARY Command with ODO.

Table 43: R-APDU Fields for the UPDATE_BINARY Command with ODO and DDO

Data	SW1	SW2	Remarks
-	90h	00h	Command completed; no data returned.

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

Requirements 22: Writing Data to the NDEF File with Mapping Version 3.0

Reader/Writer		Type 4 Tag	
5.4.2.1	When using the UPDATE_BINARY Command with ODO, the Reader/Writer SHALL format it according to Table 41 and Table 42.	**5.4.2.2**	A T4T with the value 30h in the Mapping Version field of the CC File SHALL respond to a valid UPDATE_BINARY Command with ODO with an R-APDU that is indicating 'Command completed', as defined in Table 43**.**
5.4.2.3	The Reader/Writer SHALL NOT use the UPDATE_BINARY Command with ODO unless the T4T implements Mapping Version 3.0.		
5.4.2.4	The Reader/Writer SHALL set the number of data bytes, to be written with a single UPDATE_BINARY Command with ODO and DDO, smaller than or equal to MLc.		

5.5 Checking the Presence of the Type 4 Tag

The Reader/Writer can check whether a T4T is still present in the Operating Field with the Presence Check procedure. This procedure sends a Command with the sole purpose of getting a Response from a T4T that confirms its presence.

Requirements 23: Presence Check Procedure

Reader/Writer	
5.5.1.1	Before the first I-block exchange with a T4T (see ISO-DEP in [DIGITAL]), to check if this T4T is still present in the Operating Field, the Reader/Writer SHALL send an R(NACK) block (with block number 0) and expect to receive an R(ACK) block (with block number 1) from the T4T.
5.5.1.2	After the first I-block exchange with a T4T (see ISO-DEP in [DIGITAL]), to check if this T4T is still present in the Operating Field, the Reader/Writer SHALL send an R(NACK) block (with current block number) and expect to receive an R(ACK) block from the T4T, in which case the Reader/Writer SHALL NOT retransmit its last I-block.

If the Reader/Writer receives a Response as described in Requirements 5.5.1.1 and 5.5.1.2, the T4T is still present in the Operating Field and ready to receive another SELECT, READ_BINARY or UPDATE_BINARY Command.

6 Type 4 Tag State Machine

This section refers to the Listen Mode state machine and the related requirements in [ACTIVITY].

Requirements 24: Type 4 Tag Generic State Machine

Type 4 Tag	
6.1.1.1	The start State of the T4T is the **NO_REMOTE_FIELD** State.
6.1.1.2	If, during a single period of Remote Field On, the T4T responds only to a single Technology and answers corresponding Poll Commands with a single Response, then the T4T SHALL maintain a single state machine.
6.1.1.3	If, during a single period of Remote Field On, the T4T responds to multiple Poll Commands in different Technologies and/or to a single Poll Command with multiple Responses, then the T4T SHALL maintain the equivalent number of independent state machines (i.e. a state machine for each Response).
6.1.1.4	When the T4T leaves the Remote Field, the T4T SHALL conclude the state machine within a delay not greater than t_{FIELD_OFF}.

NOTE The state machine for T4T follows the Listen Mode state machine defined in [ACTIVITY], with the exception that the State names "**CARD_EMULATOR_4A**" and "**CARD_EMULATOR_4B**" in this specification are renamed to "**PROTOCOL_4A**" and "**PROTOCOL_4B**", respectively.

Requirements 25: Type 4 Tag Activation Sequence

Reader/Writer		Type 4 Tag	
6.1.1.5	The Reader/Writer SHALL comply with the Technology Detection, Collision Resolution and Device Activation activities defined in [ACTIVITY] for NFC-A.	**6.1.1.6**	The Type 4A Tag SHALL comply with the Listen Mode state machine defined in [ACTIVITY], from States **IDLE** or **SLEEP_A**, through States **READY_A**/**READY_A***, (**READY_A'**/**READY_A'***,) (**READY_A''**/**READY_A''***,) **ACTIVE_A**/**ACTIVE_A***, up to the **PROTOCOL_4A** State, including all transitions between these States.
6.1.1.7	The Reader/Writer SHALL comply with the Technology Detection, Collision Resolution, Device Activation activities defined in [ACTIVITY] for NFC-B.	**6.1.1.8**	The Type 4B Tag SHALL comply with the Listen Mode state machine defined in [ACTIVITY], from State **IDLE**, through States **READY_B_REQU**, **READY_B_DECL**, **SLEEP_B**, up to the **PROTOCOL_4B** State, including all transitions between these States.

7 NDEF Identification and Access

This section describes how the NFC-Forum-defined data are written to or read from the T4T.

Unless specified otherwise, the term “NDEF File” in the following sections refers to the NDEF File indicated by the NDEF-File_Ctrl_TLV or the ENDEF-File_Ctrl_TLV stored at offset 0007h in the CC File.

7.1 NDEF Identification

A Reader/Writer can identify the T4T by reading the CC File to detect the Mapping Version and the access information of the NDEF data.

7.2 Version Treatment

Section 4.3.2 defines the encoding of the Mapping Version and how the Reader/Writer deals with it.

7.3 NDEF Storage

The data format of the NDEF Message is defined in [NDEF]. The NDEF Message is stored in the NDEF File.

7.4 Life Cycle

7.4.1 Type 4 Tag States

At personalization, the Reader/Writer can put a T4T into several States. The State of the T4T can be derived from the content of the CC File. Every State has its own valid operations.

Table 44 shows the list of valid States, together with a short description of each.

Table 44: Type 4 Tag States

State	Description
INITIALIZED	The T4T contains the CC File and an empty NDEF File in the NDEF Tag application; the access conditions of the NDEF File allow for both READ and WRITE.
READ/WRITE	The T4T contains the CC File and a non-empty NDEF File in the NDEF Tag application; the access conditions of the NDEF File allow for both READ and WRITE.
READ-ONLY	The T4T contains the CC File and a non-empty NDEF File in the NDEF Tag application; the access conditions of the NDEF File are restricted to READ ONLY.

A Reader/Writer can initiate transitions between these States.

Requirements 26: Type 4 Tag States

Reader/Writer		Type 4 Tag	
7.4.1.1	The Reader/Writer SHALL conclude communication with any T4T that is not in one of the three valid States: **`INITIALIZED`**, **`READ/WRITE`** and **`READ-ONLY`**.	**7.4.1.2**	The T4T SHALL be in one of the three valid States listed in Table 44.

The T4T might be in an invalid State because:

- The NDEF Tag Application or the CC File is missing in the Type 4 Tag.
- The CC File is not configured as defined in Section 4.4.
- The NDEF File with the file identifier indicated by the NDEF-File_Ctrl_TLV (see Section 4.7.3) or the ENDEF-File_Ctrl_TLV (see Section 4.7.4) is not present in the NDEF Tag Application.
- The NDEF File does not allow write operation, if the T4T is in the **`READ/WRITE`** State and no other error is detected.
- The NDEF File is not configured as defined in this specification.

7.4.2 INITIALIZED State

Requirements 27 define the conditions for a T4T to be in the **INITIALIZED** State.

Requirements 27: INITIALIZED State

Reader/Writer		Type 4 Tag	
7.4.2.1	To identify that a T4T is in the **INITIALIZED** State, the Reader/Writer SHALL verify that T4T meets all of the conditions listed in Requirements 7.4.2.2, 7.4.2.3, 7.4.2.4 and 7.4.2.5.	**7.4.2.2**	The CC File SHALL be configured as defined in Table 7.
		7.4.2.3	The NDEF File READ access condition SHALL be set to 00h or 80h - FEh and the NDEF File WRITE access condition SHALL be set to 00h or 80h - FEh (see Table 4 and Table 5).
		7.4.2.4	A T4T with value 20h in T4T_VNo SHALL set the value of the NLEN field in the NDEF File to 0000h.
		7.4.2.5	A T4T with value 30h in T4T_VNo SHALL set the value of the ENLEN field in the ENDEF File to 00000000h.

Once it has detected the **INITIALIZED** State, the Reader/Writer can modify the content of the NDEF file.

7.4.3 READ/WRITE State

The following requirements define the conditions for a T4T to be in the **READ/WRITE** State.

Requirements 28: READ/WRITE State

Reader/Writer		Type 4 Tag	
7.4.3.1	To determine that a T4T is in the **READ/WRITE** State, the Reader/Writer SHALL verify that the T4T meets all of the conditions listed in the Requirements 7.4.3.2, 7.4.3.3 and 7.4.3.4.	**7.4.3.2**	The CC File SHALL be configured as defined in Table 7.
		7.4.3.3	The NDEF File READ access condition SHALL be set to 00h or 80h - FEh and the NDEF File WRITE access condition SHALL be set to 00h or 80h - FEh (see Table 4 and Table 5).
		7.4.3.4	If the NDEF-File_Ctrl_TLV (see Section 4.7.3) is used, the value of the NLEN field SHALL be larger than 0004h; if the ENDEF-File_Ctrl_TLV (see Section 4.7.4) is used, the value of the ENLEN field SHALL be larger than 00000006h.

Once it has detected the **READ/WRITE** State, the Reader/Writer can modify the content of the NDEF File.

7.4.4 READ-ONLY State

Requirement 24 describes the conditions for a T4T to be in the **READ-ONLY** State.

Requirements 29: READ-ONLY State

Reader/Writer		Type 4 Tag	
7.4.4.1	To determine that a T4T is in the **READ-ONLY** State, the Reader/Writer SHALL verify that the T4T meets all of the conditions listed in the requirements 7.4.4.2, 7.4.4.3 and 7.4.4.4.	**7.4.4.2**	The CC File SHALL be configured as defined in Section 4.4.
		7.4.4.3	The NDEF File READ access condition SHALL be set to 00h or 80h - FEh and the NDEF File WRITE access condition SHALL be set to a value FFh (see Table 4 and Table 5).
		7.4.4.4	If the NDEF-File_Ctrl_TLV (see Section 4.7.3) is used, the value of the NLEN field SHALL be larger than 0004h; if the ENDEF-File_Ctrl_TLV (see Section 4.7.4) is used, the value of the ENLEN field SHALL be larger than 00000006h.

7.5 NDEF Procedures

7.5.1 General Requirements

The NDEF procedures assume that the Reader/Writer has performed the Technology Detection, Collision Detection and Device Activation activities, as documented in [ACTIVITY].

The Reader/Writer performs the actions during the Data Exchange activity, as defined in [ACTIVITY].

Each NDEF procedure defines a sequence of Commands to manage the NDEF data on the T4T.

The NDEF procedures defined in this section consist of the NDEF detection procedure, the NDEF read procedure and the NDEF write procedure.

This section also defines two sequences of NDEF procedures, the single NDEF read operation and the single NDEF write operation.

Note that these procedures can also be used to change the Life Cycle States (see section 7.6).

Requirements 30: NDEF Procedures – Type 4 Tag

Type 4 Tag	
7.5.1.1	A T4T SHALL NOT change the select status of the NDEF Tag Application, the CC File or the NDEF File during the Operating Field On condition, except when requested by a Reader/Writer.
7.5.1.2	A T4T SHALL NOT change the content of its CC File during the Operating Field On condition. The T4T MAY change the content of its CC File during the Operating Field Off condition.

7.5.2 Greedy Collection

The NDEF procedures use the Greedy Collection, as defined in [ACTIVITY]. The parameters used by the NDEF procedures are listed in Table 45.

Table 45: NDEF Procedures – Greedy Collection

Name	Format	Size	Description
GRE_T4T_VNo	byte	1	Mapping Version of the T4T
GRE_NDEF_File_Identifier	byte	2	File identifier of the NDEF File
GRE_NDEF_File_Selected	bit	1	Indicator if NDEF File is selected 0b … not selected 1b … selected
GRE_NDEF_File_Size	byte	4	Size of the NDEF File
GRE_NDEF_File_READ_Access	byte	1	NDEF File READ Access Condition
GRE_NDEF_File_WRITE_Access	byte	1	NDEF File WRITE Access Condition
GRE_NDEF_Length	byte	4	Length of NDEF Message

Requirements 31: Greedy Collection

Reader/Writer	
7.5.2.1	The Reader/Writer SHALL set all of the Greedy Collection parameters that are used by the NDEF procedures to 0 before starting the NDEF detection procedure.

7.5.3 NDEF Detection Procedure

The NDEF detection procedure determines whether the Tag is configured for NDEF data and, if so, retrieves the data to fill the Greedy Collection parameters (as listed in Table 45.) from the T4T.

The NDEF detection procedure has been successful if the Greedy Collection parameters have a value different than 0. Otherwise the Tag is not configured for NDEF or is not valid.

The NDEF detection procedure flow chart is shown in Figure 3.

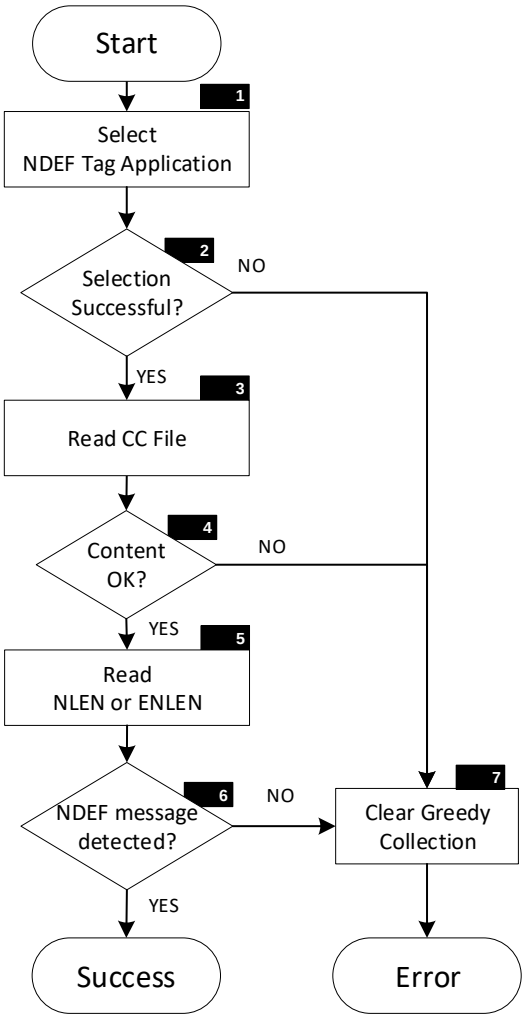

Figure 3: NDEF Detection Procedure Flowchart

Requirements 32: NDEF Detection Procedure

Reader/Writer	
7.5.3.1	Symbol 1 The Reader/Writer SHALL select the NDEF Tag Application, as defined in Section 5.2.
7.5.3.2	Symbol 2 The Reader/Writer SHALL proceed to Symbol 7 if the T4T returns an error Response on the selection of the NDEF Tag Application. NOTE If the T4T returns an error and if the Reader/Writer has also implemented Mapping Version 1.0, the Reader/Writer can proceed to select the NDEF Tag Application according to [T4TOP_v1.0], Section 6.4.1 (see also Section 4.3.3). Otherwise the Reader/Writer SHALL proceed to Symbol 3.
7.5.3.3	Symbol 3 The Reader/Writer SHALL: • Select the CC File, as defined in Section 5.2.2. • Read the CC File with the READ_BINARY Command using short field coding (see Section 5.3.1) • Fill the parameters GRE_T4T_VNo, GRE_NDEF_File_Identifier, GRE_NDEF_File_Size, GRE_NDEF_File_READ_Access and GRE_NDEF_File_WRITE_Access. After it completes these steps the Reader/Writer SHALL proceed to Symbol 4.
7.5.3.4	Symbol 4 The Reader/Writer SHALL proceed to Symbol 7 if any of the following is true: • The T4T returns an error Response on the selection of the CC File. • The read Commands caused an error Response. • The CC File is NOT configured as defined in Table 7. • The field values of the CC File are NOT valid. • The Reader/Writer cannot support the T4T_VNo. • The values of MLe and MLc are NOT within the valid range. The Reader/Writer MAY continue the NDEF read procedure independent of the NDEF File READ Access conditions (see Table 4). If the Reader/Writer intends to perform the NDEF write procedure later and the NDEF File WRITE Access condition (GRE_NDEF_File_Write_access) is different from 00h (see Table 4), the Reader/Writer MAY proceed to Symbol 7. If the Reader/Writer intends to perform the NDEF write procedure later and the NDEF Message to be written is larger than the available space in the NDEF File (GRE_NDEF_File_Size), then the Reader/Writer MAY proceed to Symbol 7. Otherwise the Reader/Writer SHALL proceed to Symbol 5.

7.5.3.5	Symbol 5 The Reader/Writer SHALL: • Select the NDEF File, as defined in Section 5.2.3. • Read the NLEN (2 Bytes length field) of the NDEF Message if the T4T uses Mapping Version 2.0 (GRE_T4T_VNo) • Read the ENLEN (4 Bytes length field) of the NDEF Message if the T4T uses Mapping Version 3.0 (GRE_T4T_VNo). The Reader/Writer SHALL fill the parameter GRE_NDEF_Length and SHALL proceed to Symbol 6.
7.5.3.6	Symbol 6 The Reader/Writer SHALL proceed to Symbol 7 if any of the following is true: • The T4T returns an error Response on the selection of the NDEF File. • The read Commands caused an error Response. • GRE_NDEF_Length is equal to 00000000h and the Reader/Writer intends to perform the NDEF read procedure later. Otherwise the Reader/Writer SHALL set GRE_NDEF_File_Selected to 1b and conclude with Success.
7.5.3.7	Symbol 7 The Reader/Writer SHALL clear the entire Greedy Collection and after that SHALL conclude with Error.

NOTE The NDEF detection procedure does not verify the validity of the NDEF Message. It reads the length of the stored data from the NLEN or ENLEN field and does not parse the data in the NDEF Message field.

7.5.4 NDEF Read Procedure

Figure 4 shows the procedure for the Reader/Writer to READ the NDEF File.

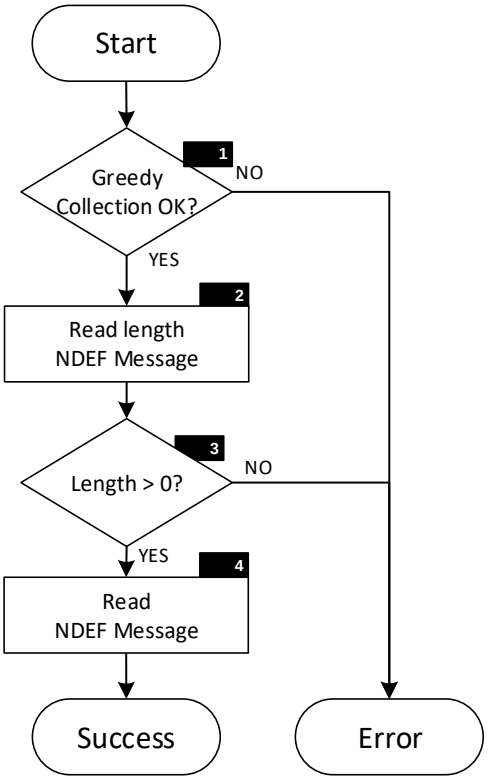

Figure 4: NDEF Read Procedure Flowchart

Requirements 33: NDEF Read Procedure

Reader/Writer	
7.5.4.1	Symbol 1 If GRE_NDEF_File_Selected is 0b, the Reader/Writer SHALL conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 2.
7.5.4.2	Symbol 2 If an NDEF procedure other than the NDEF detection procedure has been performed immediately before the NDEF read procedure, the Reader/Writer SHALL read NLEN or ENLEN, depending on the T4T Mapping Version (GRE_T4T_VNo) using the READ_BINARY Command defined in Section 5.3.1. The Reader/Writer SHALL fill the parameter GRE_NDEF_Length and SHALL proceed to Symbol 3. Otherwise the Reader/Writer SHALL proceed to Symbol 3.
7.5.4.3	Symbol 3 If GRE_NDEF_Length is equal to 00000000h, then the Reader/Writer SHALL conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 4.
7.5.4.4	Symbol 4 If the T4T uses Mapping Version 2.0 (GRE_T4T_VNo is 20h), the Reader/Writer SHALL sequentially read the NDEF Message (NLEN bytes) from the NDEF File, starting at offset two, using one or more READ_BINARY Commands (see Section 5.3.1). If the T4T uses Mapping Version 3.0 (GRE_T4T_VNo is 30h), the Reader/Writer SHALL sequentially read the NDEF Message (ENLEN bytes) from the ENDEF File, starting at offset four, using one or more READ_BINARY Commands (see Section 5.3.2). After the NDEF Message is read the Reader/Writer SHALL conclude with Success.

7.5.5 NDEF Write Procedure

Figure 5 shows the procedure for the Reader/Writer to WRITE the NDEF Message in the NDEF File on a T4T.

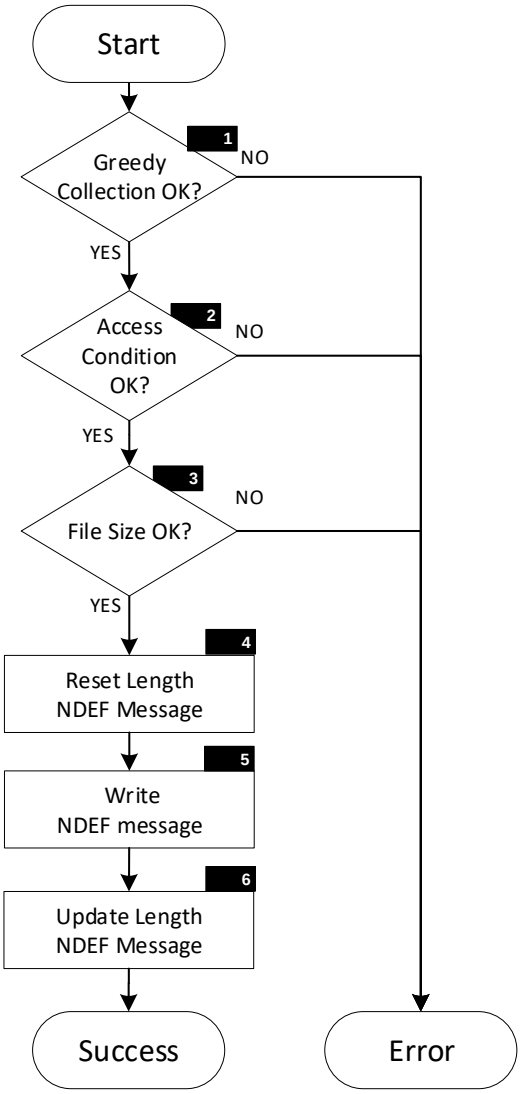

Figure 5: NDEF Write Procedure Flowchart

Requirements 34: NDEF Write Procedure

Reader/Writer	
7.5.5.1	Symbol 1 If GRE_NDEF_File_Selected is 0b, the Reader/Writer SHALL conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 2.
7.5.5.2	Symbol 2 If the content of GRE_NDEF_File_WRITE_Access (Access Conditions for Write) is equal to FFh (**`READ-ONLY`** State, see Section 4.5) Reader/Writer SHALL conclude with Error. If GRE_NDEF_File_WRITE_Access (Access Conditions for Write) is different from 00h, the Reader/Writer MAY conclude with Error. NOTE If the tag indicated a proprietary write access condition, the Reader/Writer needs to perform a proprietary procedure before continuing the NDEF write procedure, which is out of scope of this specification. Otherwise the Reader/Writer SHALL proceed to Symbol 3.
7.5.5.3	Symbol 3 If the NDEF Message to be written is larger than the available space in the NDEF File, then the Reader/Writer SHALL conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 4.
7.5.5.4	Symbol 4 If the entire NDEF Message can be written with a single UPDATE_BINARY Command, the Reader/Writer MAY write NLEN and ENLEN (Symbol 6), as well as the entire NDEF Message (Symbol 5) using a single UPDATE_BINARY Command. In this case the Reader/Writer SHALL proceed to Symbol 5 and merge Symbols 5 and 6 operations into a single UPDATE_BINARY Command. NOTE When the Reader/Writer writes the NLEN and the NDEF Message in one single UPDATE_BINARY Command with a power loss, the Reader/Writer SHALL not assume that the T4T will remain in a consistent State. If GRE_T4T_VNo is 20h and NLEN field is larger than 0000h, the Reader/Writer SHALL use a WRITE Command (see Section 5.4.1) to reset NLEN to 0000h. If GRE_T4T_VNo is 30h and ENLEN field is larger than 00000000h, the Reader/Writer SHALL use a WRITE Command (see Section 5.4.2) to reset ENLEN to 0000 0000h. The Reader/Writer MAY write the first part of the NDEF Message in the same WRITE Command as used for the reset of NLEN or ENLEN field. Afterwards the Reader/Writer SHALL proceed to Symbol 5.

7.5.5.5	Symbol 5 The Reader/Writer SHALL sequentially write the NDEF Message in the NDEF Message field (see Table 8 and Table 9) using one or more UPDATE_BINARY Commands, according to either Section 5.4.1 or Section 5.4.2, if required (see Section 5.1.2). The Reader/Writer SHALL start to write at offset 2 if GRE_T4T_VNo is 20h or write at offset 4 if GRE_T4T_VNo is 30h. Afterwards the Reader/Writer SHALL proceed to Symbol 6.
7.5.5.6	Symbol 6 If GRE_T4T_VNo is 20h, the Reader/Writer SHALL write the length of the NDEF Message in the NLEN field (see Table 8) using the UPDATE_BINARY Command (see Section 5.4.1). If GRE_T4T_VNo is 30h, the Reader/Writer SHALL write the length of the NDEF Message in the ENLEN field (see Table 8) using the UPDATE_BINARY Command (see Section 5.4.2). The Reader/Writer SHALL set GRE_NDEF_Length equal to the value corresponding to the length of the new NDEF Message. Afterwards the Reader/Writer SHALL conclude with Success.

NOTE The Reader/Writer MAY replace a non-empty NDEF Message with an empty NDEF Message (see Appendix B).

7.5.6 Single NDEF Read Operation

The single NDEF read operation defines the sequence of procedures to be used if the Reader/Writer solely intends to read a single NDEF Message from the T4T.

Requirements 35: Single NDEF Read Operation

Reader/Writer	
7.5.6.1	To read a single NDEF Message from a T4T, the Reader/Writer SHALL perform the NDEF detection procedure followed by the NDEF read procedure.

7.5.7 Single NDEF Write Operation

The Single NDEF Write Operation defines the sequence of procedures to be used if the Reader/Writer solely intends to write a single NDEF Message to the T4T.

Requirements 36: Single NDEF Write Operation

Reader/Writer	
7.5.7.1	To write a single NDEF Message to a T4T, the Reader/Writer SHALL perform the NDEF detection procedure followed by the NDEF write procedure.

7.6 State Transitions

7.6.1 Introduction

This section describes the possible State transitions that the Reader/Writer can initiate. Figure 6 shows the States and the possible State transitions.

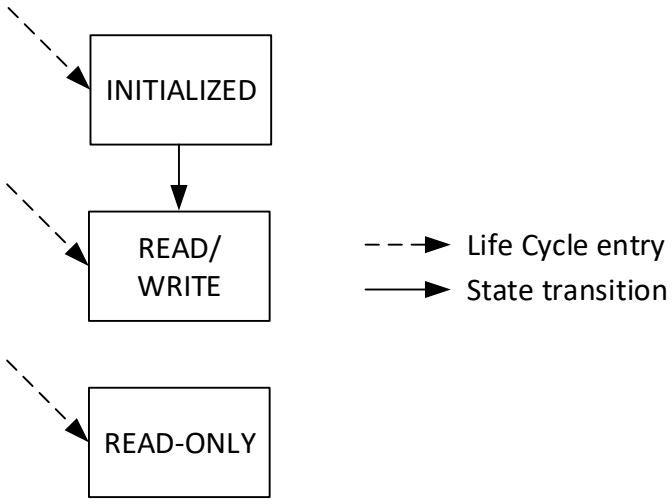

Figure 6: Life Cycle with State Transitions

Table 46 shows the possible transitions that the Reader/Writer can initiate.

Table 46: Type 4 Tag State Transitions

From	-	To
`INITIALIZED`	-	`READ/WRITE`

7.6.2 State Transition Support

Requirements 37: State Transitions

Reader/Writer	
7.6.2.1	The Reader/Writer SHALL be able to perform the State transitions listed in Table 46.

7.6.3 Transition from INITIALIZED to READ/WRITE

Requirements 38: Transition from INITIALIZED to READ/WRITE

Reader/Writer	
7.6.3.1	The Reader/Writer SHALL use the NDEF write procedure (see Section 7.5.5) to initiate the transition from **`INITIALIZED`** State to **`READ/WRITE`** State by writing an NDEF Message into the NDEF File.

A. Exhibit A

No items have been included in Exhibit A.

B. Empty NDEF Message

An empty NDEF Message contains only one NDEF Record (see [NDEF]).

B.1 Record Definition Empty NDEF Message

The single NDEF Record uses the NDEF short-record layout and contains the following three bytes:

Byte 1: Record Header byte with value D0h, configured as:

- 1b Message Begin flag
- 1b Message End flag
- 0b Chunk flag
- 1b Short Record flag
- 0b ID_LENGTH bit
- 000b Type Name Format field value (empty).

Byte 2: TYPE_LENGTH byte with value 00h.

Byte 3: PAYLOAD_LENGTH byte with value 00h.

B.2 NDEF File with Empty NDEF Message

The NDEF File consists of an NLEN (NDEF length) field and an NDEF Message.

The NLEN field contains two bytes with the value: 0003h.

The empty NDEF Message contains three bytes with the value: D00000h.

Therefore, this NDEF File contains five bytes with the value: 0003D00000h.

B.3 ENDEF File with Empty NDEF Message

The ENDEF File consists of an ENLEN (ENDEF length) field and an NDEF Message.

The ENLEN field contains four bytes with the value: 00000003h.

The empty NDEF Message contains three bytes with the value: D00000h.

Therefore, this ENDEF File contains seven bytes with the value: 00000003D00000h.

C. Example of NDEF Tag Mapping Version 2.0

This appendix describes an example of NDEF Tag Application stored inside a T4T from the Reader/Writer point of view. Figure 1 provides an overview of the example.

The content of the CC File for Mapping Version 2.0 is described in Table 47.

Table 47: CC File Example for NDEF File with Mapping Version 2.0

Offset	Size	Value	Content
0h	2	000Fh	CCLEN (15 bytes)
2h	1	20h	Mapping Version 2.0
3h	2	003Bh	MLe (59 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_Ctrl_TLV
	1	06h	L-field of the NDEF-File_Ctrl_TLV
	6		V-field of the NDEF-File_Ctrl_TLV:
		E104h	File identifier
		0032h	NDEF File 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 48.

Table 48: NDEF File Example

Offset	Size	Value	Content
0h	2	0003h	NLEN (NDEF length) 3 bytes
2h	3	D00000h	Empty NDEF Message

D. Example of NDEF Tag Mapping Version 3.0

This appendix describes an example of NDEF Tag Application stored inside a T4T from the Reader/Writer point of view. Figure 1 provides an overview of the example.

The content of the CC File for Mapping Version 3.0 is described in Table 49.

Table 49: CC File Example of NDEF File with Mapping Version 3.0

Offset	Size	Value	Content
0h	2	0011h	CCLEN (17 bytes).
2h	1	30h	Mapping Version 3.0.
3h	2	003Bh	MLe (59 bytes); maximum R-APDU data size.
5h	2	0034h	MLc (52 bytes); maximum C-APDU data size.
7h	1	06h	T-field of the ENDEF-File_Ctrl_TLV
	1	08h	L-field of the ENDEF-File_Ctrl_TLV
	8		V-field of the ENDEF-File_Ctrl_TLV:
		E104h	File identifier.
		00100000h	NDEF File size (1048576 bytes).
		00h	NDEF File READ access condition; READ access without security.
		00h	NDEF File WRITE access condition; WRITE access without security.

The NDEF File is described in detail in Table 50.

Table 50: ENDEF File Example

Offset	Size	Value	Content
0h	4	00000003h	NLEN (NDEF length) 3 bytes
4h	3	D00000h	Empty NDEF Message

E. Example of Mapping Version 2.0 Command Flow

This appendix contains an example of the Command flow of a typical interaction on APDU level. This example assumes that the T4T is configured properly and contains a valid NDEF File. The example does not check the content of the NDEF File.

The example is related to the NDEF application that is described in Appendix C.

E.1 Detection of the NDEF Message

In this section the NDEF Message is detected by applying the NDEF detection procedure (see Section 7.5.3).

E.1.1 First Command: to Select the NDEF Tag Application

Table 51 shows the Command to select the NDEF Tag Application (see Section 5.2.1).

Table 51: Command to Select the NDEF Tag Application

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

Table 52 shows the expected Response.

Table 52: Expected Response of the Command to Select the NDEF Tag Application

Offset (bytes)	Size (bytes)	Field	Description
0000h	NNh	XX…XXh	(Optional) bytes with File Control Information.
NNh	2	9000h	Status bytes (SW1, SW2), Command completed.

E.1.2 Second Command: to Select the CC File

Table 53 shows the Command to select the CC File (see Section 5.2.2).

Table 53: Command to Select CC File

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

Table 54 shows the expected Response.

Table 54: Expected Response of the Command to Select the CC File

Offset (bytes)	Size (bytes)	Field	Description
0000h	2	9000h	Status bytes (SW1, SW2), Command completed.

E.1.3 Third Command: to Read the CC File

Table 55 shows the Command to read data from the CC File (see Section 5.3).

Table 55: Command to Read the CC File

CLA	INS	P1	P2	Lc	Data	Le
00h	B0h	00h	00h	-	-	0Fh

Table 56 shows the expected Response.

Table 56: Response with the Data Structure of the CC File

Offset (bytes)	Size (bytes)	Field	Description
0000h	2	000Fh	Size of the CC File.
0002h	1	20h	Mapping Version 2.0.
0003h	2	003Bh	MLe, maximum size of R-APDU is 59 bytes.
0005h	2	0034h	MLc, maximum size of C-APDU is 52 bytes.
0007h	1	04h	Identifier for NDEF-File_Ctrl_TLV.
0008h	1	06h	The length of the V-field.
0009h	2	E104h	NDEF File identifier.
000Bh	2	0032h	NDEF File size of 50 bytes.
000Dh	1	00h	NDEF File READ access condition (no protection).
000Eh	1	00h	NDEF File WRITE access condition (no protection).

The Response allows the NDEF File to be selected and the NDEF Message to be read.

E.1.4 Fourth Command: to Select the NDEF File

Table 57 shows the Command to select the NDEF File (see Section 5.2.3).

Table 57: Command to Select the NDEF File

CLA	INS	P1	P2	Lc	Data	Le
00h	A4h	00h	0Ch	02h	E104h	-

Table 58 shows the expected Response.

Table 58: Expected Response of the Command to Select the NDEF File

Offset (bytes)	Size (bytes)	Field	Description
0000h	2	9000h	Status bytes (SW1, SW2), Command completed.

E.1.5 Fifth Command: to Read the Length of the NDEF File

Table 59 shows the READ_BINARY Command to read the NLEN field of the NDEF File (see Section 5.3.1).

Table 59: Command to Read Length of NDEF File

CLA	INS	P1	P2	Lc	Data	Le
00h	B0h	00h	00h	-	-	02h

Table 60 shows the expected Response.

Table 60: Expected Response of the Command to Read the Length of the NDEF File

Offset (bytes)	Size (bytes)	Field	Description
0000h	2	0003h	NLEN; length of the NDEF Message.
0002h	2	9000h	Status bytes (SW1, SW2), Command completed.

NLEN is smaller than the NDEF File size - 2 (equal to 50 - 2=48 bytes) and larger than 0000h. Therefore, the NDEF Message is successfully detected inside the NDEF File.

E.2 Read Data from the NDEF File

To read the NDEF File the Reader/Writer performs the NDEF read procedure (see Section 7.5.4).

It is presumed that:

- The NDEF File was previously successfully detected.
- The NDEF File is correctly selected.

E.2.1 Read Data from the NDEF File

Table 61 shows the READ_BINARY Command to read 3 bytes from the NDEF File (see Section 5.3.1).

Table 61: Command to Read Data from the NDEF File

CLA	INS	P1	P2	Lc	Data	Le
00h	B0h	00h	02h	-	-	03h

Table 62 shows the expected Response with the data structure of the NDEF File.

Table 62: Data Structure of the NDEF File

Offset (bytes)	Size (bytes)	Field	Description
0002h	3	D00000h	NDEF Message field (contains an empty NDEF Message, see Appendix B).
000Fh	2	9000h	Status bytes, Command completed.

E.3 Write Data to the NDEF File

It is presumed that:

- The NDEF File was previously successfully detected (using the procedure described in Section 7.5.3).
- The NDEF File has WRITE access without any security granted.
- The NDEF File is correctly selected.
- The (NDEF File size) - 2 (see the NDEF File Size parameter of the CC File) is larger than the NDEF Message that is to be written into the NDEF File.
  In this example the NDEF Message is 3 bytes long and the NDEF File size is 50 bytes.
  Because $(50 - 2) \geq 3$, the NDEF Message can be written in the NDEF File.

E.3.1 Command to Write Data to the NDEF File

Table 63 shows the UPDATE_BINARY Command to write the length and an empty NDEF Message into the NDEF File (see Section 5.4.1).

Table 63: Command to Write Data to the NDEF File

CLA	INS	P1	P2	Lc	Data	Le
00h	D6h	00h	00h	05h	0003D00000h	-

Table 64 shows the expected Response.

Table 64: Expected Response of Writing Data to NDEF File

Offset (bytes)	Size (bytes)	Field	Description
0000h	2	9000h	Status bytes (SW1, SW2), Command completed. Five data bytes have been successfully written to the NDEF File starting at offset 0000h.

F. Revision History

Table 65 outlines the revision history of the Type 4 Tag Technical Specification.

Table 65: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
Type 4 Tag Technical Specification	Version 1.0, August 2017	Final	Initial publication.	Type 4 Tag Operation Version 2.0
Type 4 Tag Technical Specification	Version 1.1, December 2019	Final	Technical updates to support TNEP and to optimize the command sequences. Minor editorial changes.	Type 4 Tag Technical Specification Version 1.0 August 2017
Type 4 Tag Technical Specification	Version 1.1, January 2020	Final	Editorial change to copyright notice.	Type 4 Tag Technical Specification Version 1.1 December 2019
Type 4 Tag Technical Specification	Version 1.2, August 2022	Final	Removal of Type 1 Tag; NDEF length update at NDEF Read; update copyright.	Type 4 Tag Technical Specification Version 1.1 January 2020