Type 5 Tag

Technical Specification

Version 1.2

2021-09-27

[T5T]

NFC Forum™

Contents

Figures

Tables

Requirements

1 Introduction

The Type 5 Tag Technical Specification is part of the NFC Forum documentation defining NFC Forum Tags.

This specification:

- Defines how a Reader/Writer operates an NFC Forum Type 5 Tag
- Defines how the NFC Forum Type 5 Tag behaves.

The Type 5 Tag, as defined in this specification, is based on the Type 5 Tag defined in [DIGITAL] and [ACTIVITY].

This specification is structured as follows:

- Sections 2 and 3 describe the basic communication interface of a Type 5 Tag, consisting of the Analog layer and Framing and Transmission Handling. These sections reference the definitions in [DIGITAL] and [ACTIVITY] for the Type 5 Tag Platform.
- Section 4 explains the memory structure of a Type 5 Tag, which is a precondition to understand the Type 5 Tag Command Set.
- Section 5 describes the Command Set of a Type 5 Tag.
- Section 6 defines the Type 5 Tag state machine.
- Section 7 specifies the way NDEF is stored and accessed on a Type 5 Tag (the NDEF Mapping).

1.1 Objectives

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

- A Reader/Writer operation and management of a Type 5 Tag
- The behavior of a Type 5 Tag.

This specification also defines the data mapping and how a Reader/Writer in Reader/Writer mode detects, reads, and writes NDEF data on a Type 5 Tag in order to achieve and maintain interchangeability and interoperability. But it defines how to read and write only one NDEF Message.

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
[NDEF]	Data Exchange Format 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
[ISO/IEC 15693]	ISO/IEC 15693-3:2009 Identification cards -- Contactless integrated circuit cards -- Vicinity cards – Part 3: Anticollision and transmission protocol 2009 ISO/IEC

1.3 Administration

The NFC Forum Type 5 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.

1.4 Trademark and Logo Usage

The Near Field Communication Forum's policy regarding the use of trademarks and logos is described in the NFC Forum Brand Identity Guidelines and N-Mark Usage Guidelines, which can be found on the NFC Forum website.

1.5 Intellectual Property

The Type 5 Tag conforms to the Intellectual Property guidelines specified in the NFC Forum's *Intellectual Property Right Policy*, as outlined in the NFC Forum *Rules of Procedures*. 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 document are to be interpreted as described in [RFC2119].

1.7 Notational Conventions

1.7.1 Notations

The notations shown in Table 1 apply in this document.

Table 1: 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: $\lceil 7/8 \rceil = 1$, $\lceil 8/8 \rceil = 1$, $\lceil 9/8 \rceil = 2$
Specially Defined Names	Terms defined in the Glossary or in 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).

1.8 Abbreviations

Table 2 defines the abbreviations used in this document.

Table 2: Abbreviations

Abbreviation	Description
AMS	Address Mode Selector
BLEN	Block Length
BNo	Block Number
CC	Capability Container
lsb	least significant bit
LSB	Least Significant Byte
msb	most significant bit
MSB	Most Significant Byte
MLEN	Memory Length
NB	Number of Blocks
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
OF	Option Flag
RF	Radio Frequency
RFU	Reserved for Future Use (defined in [DIGITAL])
SMS	Select Mode Selector
T5T	Type 5 Tag
TLV	Tag, Length, Value (data format)

1.9 Glossary

Big Endian

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

Block

The smallest data unit written to or read from memory.

Block Number

Identifier of a Block in memory.

Card Emulator

Role of an NFC Forum Device, reached when an NFC Forum Device in Listen Mode has gone through a number of States. In this mode the NFC Forum Device behaves as one of the Technology Subsets.

Command

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

Extended Memory

Offers up to 65536 Blocks ($n \leq 65535$) addressed by two bytes.

Listen Mode

The mode of an NFC Forum Device in which 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

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 an NFC Forum Device.

Poll Mode

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

Poller

An NFC Forum Device in Poll Mode.

Read-Alike Command

Any Command that does not change the persistent State of the Type 5 Tag.

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.

Regular Memory

Offers up to 256 Blocks ($n \leq 255$) addressed by one byte.

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.

State

A state of the Listener.

Technology

A group of transmission parameters defined by the NFC Forum specifications that make a complete communication protocol. A non-exhaustive list of transmission parameters is: RF carrier, communication mode, bit rate, modulation scheme, bit-level coding, frame format, protocol, and Command set. NFC Forum defines four groups and therefore four Technologies: NFC-A, NFC-B, NFC-F and NFC-V. The four Technologies use the same RF carrier (13.56 MHz). Each Technology uses its own modulation scheme, bit-level coding, and frame format, but can have the same protocol and Command set.

Technology Subset

A legacy platform that supports a subset of a Technology. A Technology Subset supports at least the Poll Command of the Technology. The five Technology Subsets are:

- Type 2 Tag Platform, which uses a particular subset of NFC-A, including anti-collision
- Type 3 Tag Platform, which uses a particular subset of NFC-F, including anti-collision
- Type 4 Tag Platform, which uses a particular subset of NFC-A or NFC-B, including anti-collision
- Type 5 Tag Platform, which uses a particular subset of NFC-V, including anti-collision

Type 5 Tag

Role of a Listener when it has gone through a number of States. In this mode the Listener supports the execution of Type 5 Tag Commands to read or write NDEF Messages.

Type 5 Tag Area

Area that is allocated for storing the NDEF Message. The size is declared in the Capability Container (CC) area.

Type 5 Tag Platform

A legacy platform supporting a subset of a Technology (also called a Technology Subset). Type 5 Tag Platform uses a particular subset of NFC – Type V technology including anti-collision. For more information see [DIGITAL].

Write-Alike Command

Any Command that both changes the persistent State of the Type 5 Tag platform and allows a long response time. For more information see [DIGITAL].

2 RF Interface

The RF interface is defined in [ANALOG].

Requirements 1: Analog Interface

Reader/Writer		Type 5 Tag	
2.1.1.1	The Reader/Writer SHALL comply with the analog interface for a Poller using NFC-V, as defined in [ANALOG].	**2.1.1.2**	The T5T SHALL comply with the analog interface for a Listener using NFC-V, as defined in [ANALOG].

3 Framing / Transmission Handling

This section defines the framing and the transmission handling for communication with a Type 5 Tag.

3.1 Communication Protocol

Requirements 2: Communication Protocol

Reader/Writer		Type 5 Tag	
3.1.1.1	The Reader/Writer SHALL comply with the Poll Mode requirements for half-duplex communication protocols, as defined in [DIGITAL].	**3.1.1.2**	The T5T SHALL comply with the Listen Mode requirements for half-duplex communication protocols, as defined in [DIGITAL].

3.2 Frame Structure

Requirements 3: Frame Structure

Reader/Writer		Type 5 Tag	
3.2.1.1	The Reader/Writer SHALL comply with the Sequence Format, Bit Level Coding, Frame Format, and Data and Payload Format defined in [DIGITAL] for the Type 5 Tag Platform (Poll Mode).	**3.2.1.2**	The T5T SHALL comply with the Sequence Format, Bit Level Coding, Frame Format, and Data and Payload Format defined in [DIGITAL] for the Type 5 Tag Platform (Listen Mode).

Requirements 4: Special Frame

Reader/Writer		Type 5 Tag	
3.2.1.3	If the Special_Frame flag within the Capability Container (CC) area is set to 0b, the Reader/Writer SHALL use the standard frame format defined in [DIGITAL] for all Commands and Responses and bit 6 of REQ_FLAG (OF) of Write-Alike Commands SHALL be set to 0b.		
3.2.1.4	If the Special_Frame flag within the CC is set to 1b, the Reader/Writer SHALL use the Special Frame format defined in [DIGITAL] for all Write-Alike Commands and defining the OPTION_FLAG for the use of Special Frames. For these Commands bit 6 of REQ_FLAG (OF) SHALL be set to 1b. The Reader/Writer SHALL use the standard frame format defined in [DIGITAL] for all other Commands and for all Responses.	**3.2.1.5**	If the Special_Frame flag within the CC is set to 1b, the T5T SHALL be able to execute Write-Alike Commands using the Special Frame, as defined in [DIGITAL].

Commands and related Responses are defined in Section 5.

3.3 Activation Sequence

Requirements 5: Activation Sequence

Reader/Writer		Type 5 Tag	
3.3.1.1	The Reader/Writer SHALL comply with the Technology Detection, Collision Resolution and Device Activation activities defined in [ACTIVITY] for NFC-V.	**3.3.1.2**	The T5T SHALL comply with the state machine defined in Section 6.

4 Memory Structure and Management

A Type 5 Tag has memory (that can contain data) and memory control features. The following sections describe the memory addressing, structure and management.

4.1 Memory Addressing

Whatever the memory size, the generic memory structure used by Type 5 Tag is organized by Blocks of fixed size.

Each Block contains either 4, 8, 16 or 32 bytes. The memory is one continuous area which is addressed by the Block number and the byte number within a Block as shown in Table 3.

Table 3: Generic Memory Structure with (n+1) Blocks of BLEN bytes

Block Number	1st byte of Block	2nd byte of Block	…	BLENth byte of Block
0	0	1		BLEN-1
…	…	…		…
x	x*BLEN	x*BLEN+1		x*BLEN+(BLEN-1)
…	…	…	…	…
n	n*BLEN	n*BLEN+1	…	n*BLEN+(BLEN-1)

Regular Memory offers up to 256 Blocks ($n \leq 255$) addressed by one byte.

Extended Memory offers up to 65536 Blocks ($n \leq 65535$) addressed by two bytes.

Memory area starts with byte 0 of the Block 0. The last byte of this memory area is the last byte of the last Block n*BLEN+(BLEN-1), where BLEN is the Block Length. BLEN is equal to the number of DATA bytes in the Response to the READ_SINGLE_BLOCK_REQ Command, as defined in 0.

In this chapter '4 Memory Structure and Management' memory fields defined as 2 or more bytes follow the big-endian byte order.

Requirements 6: Memory Block Length

Reader/Writer		Type 5 Tag	
4.1.1.1	The Reader/Writer SHALL support Block Length (BLEN) values of 4, 8, 16 and 32 bytes.	**4.1.1.2**	The T5T SHALL be formatted with BLEN values of either 4 or 8 or 16 or 32 bytes.
4.1.1.3	The Reader/Writer SHALL retrieve the BLEN from the Response of the Read Single Block of the CC.		

Requirements 7: Addressing

Reader/Writer		Type 5 Tag	
4.1.1.4	The Reader/Writer SHALL support 1-byte Block address Commands.	**4.1.1.5**	The T5T SHALL support 1-byte address mode.
4.1.1.6	The Reader/Writer SHALL support 2-byte Block address Commands.	**4.1.1.7**	The T5T MAY support 2-byte address mode.

4.2 Memory Layout: informative section

This section shows the memory map of a T5T to store and retrieve an NDEF Message.

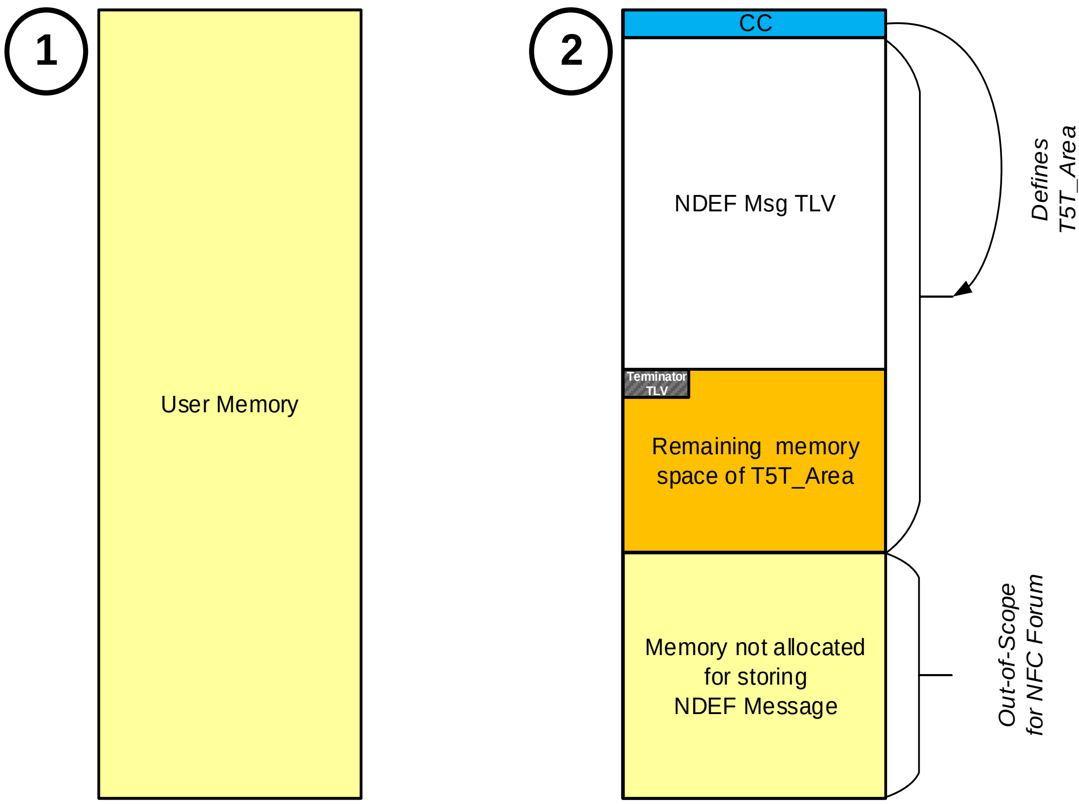

Figure 1: Type 5 Tag Memory map

Figure 1 shows a memory map of a T5T. Layout 1 shows the size of the complete T5T user memory. Layout 2 shows a well formatted T5T containing an NDEF Message in part of the T5T_Area.

The T5T might use only part of the available user memory for storing the NDEF Message. The CC contains information on the part of the memory that is allocated for storing the NDEF Message, in this specification called “T5T_Area”, and information on the access conditions of the NDEF Message.

NOTE Reading and writing to a memory area that is not allocated as T5T_Area is out of scope of this specification.

The NDEF_Msg_TLV (see Section 4.4.3) contains the NDEF Message. The Reader/Write can sequentially write the NDEF Message as part of the NDEF_Msg_TLV to the (remainder of the) T5T_Area.

The Terminator_TLV is a Control TLV written after the last byte of the last TLV.

The Terminator_TLV is always present to indicate the end of the TLVs inside the T5T_Area, unless the last byte of the last TLV is also the last byte of the T5T_Area.

4.3 Memory Structure

The memory area is composed of two fields:

- Capability Container (CC)
- T5T_Area.

The CC begins at the first byte of the memory area and contains 4 or 8 bytes.

The CC is stored on contiguous bytes.

The T5T_Area (i.e., memory containing the NDEF Message) starts with the first byte following the last byte of the CC field. The size of the T5T_Area is defined by the content of the CC field.

4.3.1 Capability Container

The CC field manages the information of the NFC Forum Type 5 Tag.

The four-byte CC field limits the maximum T5T_Area size to 2040 bytes (the Block number is coded in one byte). When it is extended to eight bytes, the CC field allows a larger T5T_Area (the Block number is coded in two bytes).

Table 4: Four Byte Capability Container Field

Byte 0	Byte 1	Byte 2	Byte 3
Magic Number	Version and access condition	MLEN	Additional feature information

Table 5: Eight Byte Capability Container Field

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Magic Number	Version and access condition	00h	Additional feature information	RFU	RFU	MLEN	

Requirements 8: General CC Requirements

Reader/Writer		Type 5 Tag	
4.3.1.1	The Reader/Writer SHALL use READ_SINGLE_BLOCK Command to read the CC in Block 0.	**4.3.1.2**	The T5T SHALL contain a CC starting at the first byte of the memory Block 0.
4.3.1.3	The Reader/Writer SHALL use WRITE_SINGLE_BLOCK Command to update the CC.	**4.3.1.4**	Depending on T5T_Area size, the T5T SHALL code the CC on four or eight bytes.
4.3.1.5	The Reader/Writer SHALL NOT use the CC to store any application related data.		

Requirements 9: Magic Number

Reader/Writer		Type 5 Tag	
4.3.1.6	The Reader/Writer SHALL extract information about 1-byte or 2-byte address mode Commands from the Magic Number (Byte 0) of the CC.	**4.3.1.7**	The T5T SHALL set Magic Number (Byte 0) in CC to E1h, when 1-byte address mode is supported (see 4.1.1.5).
4.3.1.8	The Reader/Writer SHALL NOT use 2-byte address mode Commands, if Magic Number is equal to E1h.	**4.3.1.9**	The T5T SHALL set Magic Number in CC to E2h, if 2-byte address mode is supported (see 4.1.1.5 and 4.1.1.7).

Requirements 10: Version Number and Access Conditions

Reader/Writer		Type 5 Tag	
4.3.1.10	The Reader/Writer SHALL extract version information out of bits b7 to b4 of Byte 1 of the CC.	**4.3.1.11**	The T5T SHALL code the version information in bits b7 to b4 of Byte 1 in the CC, as defined in Table 6.
4.3.1.12	The Reader/Writer SHALL extract Read Access and Write Access conditions out of bits b3 to b0 of Byte 1 of the CC.	**4.3.1.13**	The T5T SHALL code Read Access and Write Access conditions in bits b3 to b0 of Byte 1 in the CC, as defined in Table 6.

Table 6: Byte 1 of Capability Container

b7	b6	b5	b4	b3	b2	b1	b0
Major version 01b: Version 1.x		Minor version 00b: Version y.0		Read Access 00b: Always 01b: RFU 10b: Proprietary 11b: RFU		Write Access 00b: Always 01b: RFU 10b: Proprietary 11b: Never	

NOTE Proprietary access conditions are out of scope of this specification.

Requirements 11: MLEN

Reader/Writer		Type 5 Tag	
4.3.1.14	The Reader/Writer SHALL extract information about the length of the CC from the Byte 2 of the CC.	**4.3.1.15**	If the T5T uses a four-byte CC, Byte 2 of the CC SHALL code the size of the T5T_Area (MLEN).
		4.3.1.16	If the T5T uses an eight byte CC, Byte 2 of the CC SHALL be 00h and Byte 6 and Byte 7 of the CC SHALL code the size of the T5T_Area (MLEN), with the Most Significant Byte of MLEN coded on Byte 6 and the Least Significant Byte of MLEN on Byte 7.
		4.3.1.17	MLEN SHALL be the size of the T5T_Area in bytes divided by 8.

T5T_Area size measured in bytes is equal to 8 * MLEN.

For example, if the T5T_Area has a size of:

- 48 bytes, then the Byte 2 value is 06h.
- 2040 bytes, then the Byte 2 value is FFh.
- 8184 bytes, then the Byte 2 value is 00h, the Byte 6 value is 03h and the Byte 7 value is FFh.

Requirements 12: Additional Feature Information

Reader/Writer		Type 5 Tag	
4.3.1.18	The Reader/Writer SHALL extract information about additional features from Byte 3 of CC.	**4.3.1.19**	The T5T SHALL use Byte 3 of CC to indicate supported additional features, as defined in Table 7.
4.3.1.20	The Reader/Writer MAY use READ_MULTIPLE_BLOCKS or EXTENDED_READ_MULTIPLE_BLOCKS Commands, when b0 of Byte 3 of the CC is set to 1b.	**4.3.1.21**	If the T5T does not support the READ_MULTIPLE_BLOCKS or EXTENDED_READ_MULTIPLE_BLOCKS Commands, then b0 of Byte 3 of the CC SHALL be set to 0b.
4.3.1.22	The Reader/Writer MAY use LOCK_SINGLE_BLOCK or EXTENDED_LOCK_SINGLE_BLOCK Commands, when b3 of Byte 3 of the CC is set to 1b.	**4.3.1.23**	If the T5T does not support the LOCK_SINGLE_BLOCK or the EXTENDED_LOCK_SINGLE_BLOCK Commands, then b3 of Byte 3 of the CC SHALL be set to 0b.
4.3.1.24	The Reader/Writer SHALL use the Special Frame format defined in [DIGITAL] for Write-Alike Commands when b4 of Byte 3 of the CC is set to 1b.	**4.3.1.25**	If the T5T needs the Special Frame format defined in [DIGITAL] for Write-Alike Commands, then b4 of Byte 3 of the CC SHALL be set to 1b.

Table 7: Byte 3 of Capability Container

b7	b6	b5	b4	b3	b2	b1	b0
RFU			Special Frame	Lock Block	RFU		MBREAD

Table 8, Table 9 and Table 10 show examples of coding the CC, respectively coded in 4 and 8 bytes, including a message smaller or bigger than 255 data Blocks:

- With NFC Forum defined data (Byte 0 = E1h)
- Supporting version 1.0 (major number 1h, minor number 0h) of the mapping document and granting the Read and Write access (Byte 1 = 40h)
- With 256 bytes of the T5T_Area size (Byte 2 = 20h)
- No specific feature granted (Byte 3 = 00h)

Table 8: Example of 4 Byte Coding of the CC

Byte 0	Byte 1	Byte 2	Byte 3
E1h	40h	20h	00h

- With NFC Forum defined data (Byte 0 = E1h), NDEF Message readable using the READ_SINGLE_BLOCK_REQ Command
- Supporting version 1.0 (major number 1h, minor number 0h) of the mapping document and granting the Read and Write access (Byte 1 = 40h),
- Byte 2 = 00h defining a CC of 8 bytes.
- READ_MULTIPLE_BLOCK_REQ or EXTENDED_READ_MULTIPLE_BLOCK_REQ being command supported Byte 3 = 01h
- Byte 4 and Byte 5 being RFU and equal to 00h
- Byte 6 and Byte 7 defining the T5T_Area size with 8184 bytes.

Table 9: Example of 8 Byte Coding of the CC, Indicating the 1-Byte Address Mode

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
E1h	40h	00h	01h	00h	00h	03h	FFh

- With NFC Forum defined data (Byte 0 = E2h), NDEF Message readable using the READ_SINGLE_BLOCK_REQ Command if less than 256 Blocks or using the EXTENDED_READ_SINGLE_BLOCK_REQ Command
- Supporting the version 1.0 (major number 1h, minor number 0h) of the mapping document and granting the Read and Write access (Byte 1 = 40h),
- Byte 2 = 00h defining a CC of 8 bytes
- READ_MULTIPLE_BLOCK_REQ or EXTENDED_READ_MULTIPLE_BLOCK_REQ being command supported Byte 3 = 01h
- Byte 4 and Byte 5 being RFU and equal to 00h
- Byte 6 and Byte 7 defining the T5T_Area size with 8184 bytes.

Table 10: Example of 8 Byte Coding of the CC, Indicating the 2-Byte Address Mode

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
E2h	40h	00h	01h	00h	00h	03h	FFh

4.3.2 T5T_Area

The T5T_Area directly follows the CC in the memory of the Type 5 Tag.

Requirements 13: T5T_Area

Reader/Writer	Type 5 Tag	
	4.3.2.1	The T5T_Area SHALL start from the first byte following the CC.

4.4 TLV Structure Definition

This section describes the TLV structure and content used on a T5T.

4.4.1 TLV Structure

A TLV Block consists of one to three fields:

- *T:* The tag field (T-field) encodes the type of the TLV structure in one byte.
  - Table 11 defines the values for the encoding of the T-field that are used in this specification.
- *L:* The length field (L-field) encodes the size of the V-field in bytes.
  Depending on the value of the T-field, the L-field may or may not be present.
  If the L-field is present, it may contain one or three bytes:
  - One byte if the length to encode is between 00h and FEh.
    The value FFh for the first byte encodes a three-byte format.
  - Three bytes if the length to encode is between 00FFh and FFFEh.
    The three-byte value FFFFFFh is RFU.

  Figure 2 shows the format of the L-field.

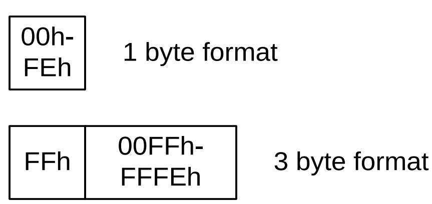

Figure 2: Length Field Formats

- *V:* The Value field (V-field), if present, contains the data of the TLV.
  The V-field is not present if the L-field has the value 00h or if the L-field is not present.
  If the L-field is present and has a value N (with N larger than zero), the V-field consists of N consecutive bytes.

4.4.2 Defined TLV Structures for the T5T

Table 11 lists the TLVs that are defined for the T5T.

Table 11: Defined TLV Structures

TLV structure name	Tag field Value	Short Description
	00h-02h	RFU
NDEF Message TLV	03h	Contains the NDEF Message, see [NDEF]
	04h-FDh	RFU
Terminator TLV	FEh	Last TLV Block in the T5T_Area
	FFh	RFU

TLVs with a Tag Field value of RFU are not to be interpreted by the Reader/Writer.

Requirements 14: TLV Blocks

Reader/Writer		Type 5 Tag	
4.4.2.1	If an RFU TLV is detected in the T5T_Area, the Reader/Writer SHALL check whether there are other TLVs present after the RFU TLV.	**4.4.2.2**	The T5T SHALL not use a Tag Field value that is defined as RFU.
4.4.2.3	The Reader/Writer SHALL not change the values of any bytes that belong to RFU TLVs.		

NOTE Future definitions of TLV structures composed of only the Tag Field are not backward compatible with this NFC Forum specification, since a Reader/Writer requires the length field to jump over the unknown TLVs.

4.4.3 NDEF Message TLV

A T5T always has an NDEF Message TLV to store and retrieve an NDEF Message.

The encoding of the TLV fields of an NDEF Message TLV is:

Table 12: NDEF Message TLV

Field	Length	Value	Description	
T	1 Byte	03h	Indicates the NDEF Message TLV	
L	1 Byte	N	Encodes the size of the V-field or the size of the L-field	
			Value	**Description**
	or		00h	Indicates an empty NDEF Message TLV The V-field is not present
			01h-FEh	Length of the V-field (NDEF Message)
			FFh	Indicates the use of 3 bytes for the L-field
	3 Bytes	N	The first byte (FFh) indicates that the size of the V-field for N > 254 bytes is encoded in three bytes	
			Value	**Description**
			FF0000h-FFFFFEh	The 2 Least Significant Bytes indicate the length of the V-field (NDEF Message)
			FFFFFFh	RFU
V	N Bytes		If present, it contains the NDEF Message, as defined in [NDEF]	

Requirements 15: NDEF Message TLV

Reader/Writer		Type 5 Tag	
4.4.3.1	The Reader/Writer SHALL be able to read and process at least the first NDEF Message TLV formatted as defined in Table 12.	**4.4.3.2**	A T5T SHALL contain at least one NDEF Message TLV.

4.4.4 Terminator TLV

The Terminator TLV is the last TLV structure in the T5T_Area. Table 13 shows the encoding of the Terminator TLV.

Table 13: Terminator TLV

Field	Length	Value	Description
T	1 Byte	FEh	Indicates the Terminator TLV
L	None	None	-
V	None	None	-

Requirements 16: Terminator TLV

Reader/Writer		Type 5 Tag	
4.4.4.1	The Reader/Writer SHALL be able to read and process the Terminator TLV.	**4.4.4.2**	If the last byte of the last TLV in the T5T_Area is not the last byte of the T5T_Area, the T5T SHALL have a Terminator TLV formatted as defined in Table 13, immediately following the last TLV in the T5T_Area.
4.4.4.3	If the last byte of the NDEF Message TLV in the T5T_Area is not the last byte of the T5T_Area, the Reader/Writer SHALL write the Terminator TLV immediately following the NDEF Message TLV in the T5T_Area.		

5 Command Set

Table 14 lists the Commands that are available for communication with an NFC Forum Type 5 Tag. For each Command the corresponding Response from the Type 5 Tag is indicated.

Table 14: Command Set

Poll Mode (Command)	Listen Mode (Response)	Read-Alike	Write-Alike	Special Frame support
READ_SINGLE_BLOCK _REQ	READ_SINGLE_BLOCK_RES	X		
WRITE_SINGLE_BLOCK_R EQ	WRITE_SINGLE_BLOCK_RES		X	X
LOCK_SINGLE_BLOCK_R EQ	LOCK_SINGLE_BLOCK_RES		X	X
READ_MULTIPLE_BLOCK _REQ	READ_MULTIPLE_BLOCK_R ES	X		
EXTENDED_READ_SINGL E_REQ	EXTENDED_READ_SINGLE_ BLOCK_RES	X		
EXTENDED_WRITE_SING LE_BLOCK _REQ	EXTENDED_WRITE_SINGLE _BLOCK_RES		X	X
EXTENDED_LOCK_SINGL E_BLOCK _REQ	EXTENDED_LOCK_SINGLE_ BLOCK_RES		X	X
EXTENDED_READ_MULTI PLE_BLOCK _REQ	EXTENDED_READ_MULTIPL E_BLOCK_RES	X		
SELECT_REQ	SELECT_RES	X		
SLPV_REQ	–	X		

All Commands for which the use of the Special Frame is possible are Write-Alike Commands. All other Commands are Read-Alike Commands.

Other optional or custom Commands of the related standard are out of scope of this specification but may be supported by the Reader/Writer or the NFC Forum Type 5 Tag.

UID, CRC, BNo, NB, Mask values are considered as multi-byte fields. BNo and NB are multi-byte fields only in the case of the extended command set (Command starting with EXTENDED_)

5.1 Generic Command Response Structure

NFC Forum is specifying a subset of [ISO/IEC 15693].

NOTE In [ISO/IEC 15693] bit numbering starts from 1.

5.1.1 Generic Command Structure

Table 15 specifies the generic command structure for all Commands described in this section.

Table 15: Generic Command Structure

Length	1 byte	1 byte	8 bytes (Optional)	N bytes (Conditional)	Up to 32 bytes (Conditional)
Parameter	REQ_FLAG	CMD_CODE	UID	PARAMETER	DATA

UID is coded on 8 bytes.

If present, the PARAMETER and DATA fields are defined in each Command.

DATA fields contain the content of one memory Block and places the first byte of the Block as the first transmitted byte of the DATA field and the BLENth byte of the Block as the last transmitted byte of the DATA field.

The REQ_FLAG is coded as specified in Table 16:

Table 16: REQ_FLAG

b7	b6	b5	b4	b3	b2	b1	b0	Meaning
0								RFU
	x							OPTION_FLAG (OF): The meaning of this flag is Command specific
		x						Address mode Selector (AMS)
			x					Select mode Selector (SMS)
				0				0b: No protocol format extension
					0			0b: No Inventory Command
						1		1b: High Data Rate
							0	0b: Single Subcarrier

Commands can be sent in Select mode. REQ_FLAG bit b4 “SMS” is the Select Mode Selector.

When b4 is equal to 1b, the Command is executed only by the Type 5 Tag in its **SELECTED** State.

When b4 is equal to 0b, the command execution depends on the AMS bit and the UID field.

Commands can be sent in Addressed or Non-addressed mode. REQ_FLAG bit b5 “AMS” is the Address mode selector.

When b5 is equal to 1b, the Command is sent in Addressed mode; only the Type 5 Tag with its UID matching the UID of the Command will execute the Command.

When b5 is equal to 0b, the Command is sent in a Non-addressed mode; all Type 5 Tags in either **READY** State or **SELECTED** State will execute the Command.

Requirements 17: Generic Command

Reader/Writer		**Type 5 Tag**	
5.1.1.1	The Reader/Writer SHALL set the REQ_FLAG according to Table 16.	**5.1.1.2**	If bit 0 of REQ_FLAG is received as 0b, the T5T SHALL respond with single subcarrier modulation.
		5.1.1.3	If bit 1 of REQ_FLAG is received as 1b, the T5T SHALL respond with High Data Rate.
		5.1.1.4	A T5T MAY respond to other settings of the REQ_FLAG according to [ISO/IEC 15693].
5.1.1.5	To use the addressed mode the Reader/Writer SHALL set bit b5 (“AMS”) of REQ_FLAG to 1b, and SHALL set bit b4 (“SMS”) of REQ_FLAG to 0b and SHALL set the UID field with the UID of the targeted tag.	**5.1.1.6**	The T5T SHALL process the Command with bit b5 (“AMS”) of REQ_FLAG set to 1b and bit b4 (“SMS”) of REQ_FLAG set to 0b if its UID matches the UID field in the Command. The T5T SHALL ignore the Command if its UID does not match the UID field in the Command.
5.1.1.7	To use the select mode the Reader/Writer SHALL set bit b5 (“AMS”) of REQ_FLAG to 0b, SHALL set bit b4 (“SMS”) of REQ_FLAG to 1b and SHALL NOT include any UID field in the Command.	**5.1.1.8**	The T5T in `SELECTED` State SHALL process the Command with bit b5 (“AMS”) of REQ_FLAG set to 0b and bit b4 (“SMS”) of REQ_FLAG set to 1b and with no UID field. In any other States the T5T SHALL ignore the Command.
5.1.1.9	To use the non-addressed mode the Reader/Writer SHALL set bit b5 (“AMS”) of REQ_FLAG to 0b and SHALL set bit b4 (“SMS”) of REQ_FLAG to 0b and SHALL NOT include any UID field in the Command.	**5.1.1.10**	The T5T SHALL process the Command with bit b5 (“AMS”) of REQ_FLAG set to 0b and bit b4 (“SMS”) of REQ_FLAG set to 0b and with no UID field.
5.1.1.11	The Reader/Writer SHALL NOT simultaneously in a single command set both bit b5 (“AMS”) of REQ_FLAG to 1b and bit b4 (“SMS”) of REQ_FLAG to 1b.	**5.1.1.12**	The T5T MAY ignore or respond with an error code, if bit b5 (“AMS”) of REQ_FLAG is set to 1b and bit b4 (“SMS”) of REQ_FLAG is set to 1b simultaneously in the received Command.
5.1.1.13	The Reader/Writer SHALL transmit and interpret received Multi-byte fields as defined in [DIGITAL].	**5.1.1.14**	The T5T SHALL transmit and interpret received Multi-byte fields as defined in [DIGITAL].

NOTE In non-addressed mode all Type 5 Tags in the operating volume will execute the received Command, e.g. Write-Alike Commands might lead to unintended data corruption.

5.1.2 Generic Response Structure

Table 17 specifies the generic Response structure for all Commands described in this section.

Table 17: Generic Response Structure

Length		1 byte	N bytes (Conditional)
Responses	Normal case	RES_FLAG	[Status (conditional)+ DATA]
	Error case	RES_FLAG	ERROR CODE (1 byte)

If present, DATA is defined in each Response.

The DATA field is the content of one Block of memory. The first received byte of the Block in the DATA field is the first byte of the Block and the last received byte of the DATA field is the BLENth byte of the Block.

The coding of the RES_FLAG is specified by Table 18.

Table 18: Format of RES_FLAG

b7	b6	b5	b4	b3	b2	b1	b0	Meaning
0	0	0	0	0	0	0		RFU
							X	ERROR

If bit b0 of the RES_FLAG “ERROR” is equal to 0b, DATA defined for each Response can follow the RES-FLAG.

If bit b0 of the RES_FLAG “ERROR” is equal to 1b, an error has occurred and RES-FLAG is always followed by a single data byte indicating the error code, as defined in Table 19.

Table 19: Error Codes

Error Code	Meaning
01h	Command not supported
02h	Command not recognized
03h	Command option not supported
0Fh	No information or a specific error code is not supported
10h	Specified Block doesn’t exist
11h	Specified Block already locked and cannot be locked again
12h	Specified Block already locked and cannot be changed
13h	Specified Block was not successfully programmed
14h	Specified Block was not successfully locked
15h	Specified Block is protected
40h	Generic Cryptographic Error
A0h – DFh	Proprietary error code
Other values	RFU

Requirements 18: Generic Response

Reader/Writer		**Type 5 Tag**	
5.1.2.1	The Reader/Writer SHALL be ready to receive a Response as specified in Table 17.	**5.1.2.2**	When processing a correctly formatted Command the T5T SHALL construct its Response according to Table 17.
5.1.2.3	When no Response is received, the Reader/Writer SHALL treat this as a timeout error; see [DIGITAL].	**5.1.2.4**	When it receives an incorrectly formatted Command, the T5T MAY ignore the Command or MAY respond with an error code, as defined in Table 19.

5.2 READ_SINGLE_BLOCK

The READ_SINGLE_BLOCK Command is used to read the content of one memory Block and derive the BLEN from the Response.

This Command has the command code 20h and a 1-byte Block Number (BNo) as PARAMETER, but no DATA field. The Reader/Writer can request the Type 5 Tag to return in its Response the Block security status byte, by setting bit 6 (OF) of REQ_FLAG to 1b.

Table 20 specifies the READ_SINGLE_BLOCK Command. Table 21 specifies the corresponding Response.

Table 20: Format of READ_SINGLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	1 Byte
REQ_FLAG	20h	UID	BNo

Table 21: Format of READ_SINGLE_BLOCK_RES

1 Byte	1 Byte (optional)	BLEN Bytes
RES_FLAG	Block security status	DATA

Requirements 19: READ_SINGLE_BLOCK Command

Reader/Writer		Type 5 Tag	
5.2.1.1	The Reader/Writer SHALL construct the READ_SINGLE_BLOCK_REQ Command, as defined in Table 20.	**5.2.1.2**	When it receives a correctly formed READ_SINGLE_BLOCK_REQ Command, the T5T SHALL respond with the READ_SINGLE_BLOCK_RES Response, as defined in Table 21.
5.2.1.3	The Reader/Writer SHALL use BNo to address the desired Block.	**5.2.1.4**	The T5T SHALL respond with data from the addressed Block with Block number BNo.
5.2.1.5	The Reader/Writer SHALL be ready to receive a READ_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte and BLEN Data bytes, if bit 6 of REQ_FLAG (OF) was set to 0b.	**5.2.1.6**	When bit 6 of REQ_FLAG is set to 0b, the T5T SHALL populate the payload of its READ_SINGLE_BLOCK_RES Response with a RES_FLAG byte and BLEN Data bytes.
5.2.1.7	The Reader/Writer SHALL be ready to receive a READ_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte, Block security status and BLEN Data bytes, if bit 6 of REQ_FLAG (OF) was set to 1b. The Reader/Writer SHALL not consider the value of the Block security status byte for the NDEF identification and access procedures (see Section 7).	**5.2.1.8**	When bit 6 of REQ_FLAG is set to 1b, the T5T SHALL populate the payload of its READ_SINGLE_BLOCK_RES Response with a RES_FLAG byte, the Block security status and BLEN Data bytes.

5.3 WRITE_SINGLE_BLOCK

The WRITE_SINGLE_BLOCK_REQ Command is used to update the content of one memory Block.

This Command has the command code 21h and a 1-byte Block Number (BNo) as PARAMETER. The DATA field contains the new content of the Block and has a length of BLEN bytes. The first DATA field byte represents the first byte of the Block to update and the last DATA field byte represents the last byte of the Block.

Table 22 specifies the WRITE_SINGLE_BLOCK Command and Table 23 the corresponding Response.

Table 22: Format of WRITE_SINGLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	1 Byte	BLEN Bytes
REQ_FLAG	21h	UID	BNo	DATA

Table 23: Format of WRITE_SINGLE_BLOCK_RES

1 Byte
RES_FLAG

Requirements 20: WRITE_SINGLE_BLOCK Command

Reader/Writer		Type 5 Tag	
5.3.1.1	The Reader/Writer SHALL construct the WRITE_SINGLE_BLOCK_REQ Command, as defined in Table 22.	**5.3.1.2**	When it receives a correctly formed WRITE_SINGLE_BLOCK_REQ Command, the T5T SHALL respond with the WRITE_SINGLE_BLOCK_RES Response, as defined in Table 23.
5.3.1.3	The Reader/Writer SHALL use BNo to address desired Block.	**5.3.1.4**	When it receives a WRITE_SINGLE_BLOCK_REQ Command, the T5T SHALL update the data of the Block addressed by the Block number with the data from the DATA field.
5.3.1.5	The Reader/Writer SHALL use exact BLEN data bytes in the DATA field to update Block BNo.		
5.3.1.6	The Reader/Writer SHALL set b6 of REQ_FLAG (OF) in accordance with Section 3.2.	**5.3.1.7**	The T5T SHALL respond in accordance with Section 3.1.
5.3.1.8	The Reader/Writer SHALL be ready to receive a WRITE_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte.	**5.3.1.9**	When it sends a WRITE_SINGLE_BLOCK_RES Response, the T5T SHALL populate the payload with a RES_FLAG byte.

5.4 LOCK_SINGLE_BLOCK

The LOCK_SINGLE_BLOCK_REQ Command is used to lock the content of one memory Block.

This Command has the command code 22h and a 1-byte Block Number (BNo) as PARAMETER but no DATA field.

Table 24 specifies the LOCK_SINGLE_BLOCK_REQ Command and Table 25 the corresponding Response.

Table 24: Format of LOCK_SINGLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	1 Byte
REQ_FLAG	22h	UID	BNo

Table 25: Format of LOCK_SINGLE_BLOCK_RES

1 Byte
RES_FLAG

Requirements 21: LOCK_SINGLE_BLOCK Command

Reader/Writer		Type 5 Tag	
5.4.1.1	The Reader/Writer SHALL construct the LOCK_SINGLE_BLOCK_REQ Command, as defined in Table 24.	**5.4.1.2**	When it receives a correctly formed LOCK_SINGLE_BLOCK_REQ Command, the T5T SHALL respond with the LOCK_SINGLE_BLOCK_RES Response, as defined in Table 25.
5.4.1.3	The Reader/Writer SHALL use BNo to address desired Block.	**5.4.1.4**	The T5T SHALL lock the data of the Block addressed by Block number BNo.
5.4.1.5	The Reader/Writer SHALL set b6 of REQ_FLAG (OF) in accordance with Section 3.2.	**5.4.1.6**	The T5T SHALL respond in accordance with Section 3.1.
5.4.1.7	The Reader/Writer SHALL be ready to receive a LOCK_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte.	**5.4.1.8**	When it sends a LOCK_SINGLE_BLOCK_RES Response, the T5T SHALL populate the payload with a RES_FLAG byte.

5.5 READ_MULTIPLE_BLOCK

The READ_MULTIPLE_BLOCK Command is used to read the content of one or more memory Blocks.

This Command has the command code 23h and a 1-byte Block Number (BNo) specifying the first Block to read followed by one byte specifying the number of additional Blocks to read (NB) as PARAMETER but no DATA field. The Reader/Writer can request the Type 5 Tag to return in its Response the Block security status byte, by setting bit 6 (OF) of REQ_FLAG to 1b.

Table 26 specifies the READ_MULTIPLE_BLOCK Command and Table 27 the corresponding Response. The number of data Blocks (and optionally Block security bytes) returned by the Type 5 Tag in its Response is (NB +1).

Table 26: Format of READ_MULTIPLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	1 Byte	1 Byte
REQ_FLAG	23h	UID	BNo	NB

Table 27: Format of READ_MULTIPLE_BLOCK_RES

1 Byte	(NB+1)*	[1 Byte (optional)	BLEN Bytes]
RES_FLAG		Block security status	DATA

Requirements 22: READ MULTIPLE Command

Reader/Writer		Type 5 Tag	
5.5.1.1	The Reader/Writer SHALL construct the READ_MULTIPLE_BLOCK_REQ Command, as described in Table 26.	**5.5.1.2**	When it receives a correctly formed READ_MULTIPLE_BLOCK_REQ Command, the T5T SHALL respond with the READ_MULTIP¨LE_BLOCK_RES Response, as defined in Table 27.
5.5.1.3	The Reader/Writer SHALL use BNo to address the first desired Block.	**5.5.1.4**	The T5T SHALL respond with the content of the Block starting from the Block addressed by Block number BNo up to Block number (BNo + NB).
5.5.1.5	The Reader/Writer SHALL use NB to indicate how many adjacent Blocks in addition SHALL be read.		
5.5.1.6	The Reader/Writer SHALL be ready to receive a READ_MULTIPLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte and (NB+1)*BLEN Data bytes, if bit 6 of REQ_FLAG (OF) was set to 0b.	**5.5.1.7**	When bit 6 of REQ_FLAG is set to 0b, the T5T SHALL populate the payload of its READ_MULTIPLE_BLOCK_RES Response with a RES_FLAG byte and (NB+1)*BLEN data bytes.
5.5.1.8	The Reader/Writer SHALL be ready to receive a READ_MULTIPLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte and (NB + 1) tuples of Block security status byte and BLEN data bytes, if bit 6 of REQ_FLAG (OF) was set to 1b. The Reader/Writer SHALL not consider the value of the Block security status bytes for the NDEF identification and access procedures (see Section 7).	**5.5.1.9**	When bit 6 of REQ_FLAG is set to 1b, the T5T SHALL populate the payload of its READ_MULTIPLE_BLOCK_RES Response with a RES_FLAG byte, and (NB + 1) tuples of Block security status byte and BLEN data bytes.

5.6 EXTENDED_READ_SINGLE_BLOCK

The EXTENDED_READ_SINGLE_BLOCK Command is used to read the content of one memory Block from an Extended Memory.

This Command has the command code 30h and a 2-byte Block Number (BNo) as PARAMETER, but no DATA field. The Reader/Writer can request the Type 5 Tag to return in its Response the Block security status byte by setting bit 6 (OF) of REQ_FLAG to 1b.

Table 28 specifies the EXTENDED_READ_SINGLE_BLOCK Command and Table 29 the corresponding Response.

Table 28: Format of EXTENDED_READ_SINGLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	2 Bytes
REQ_FLAG	30h	UID	BNo

Table 29: Format of EXTENDED_READ_SINGLE_BLOCK_RES

1 Byte	1 Byte (optional)	BLEN Bytes
RES_FLAG	Block security status	DATA

Requirements 23: EXTENDED_READ_SINGLE_BLOCK Command

Reader/Writer		Type 5 Tag	
5.6.1.1	The Reader/Writer SHALL construct the EXTENDED_READ_SINGLE_BLOCK_REQ Command, as defined in Table 28.	**5.6.1.2**	When it receives a correctly formed EXTENDED_READ_SINGLE_BLOCK_REQ Command, the T5T SHALL respond with the EXTENDED_READ_SINGLE_BLOCK_RES Response, as defined in Table 29.
5.6.1.3	The Reader/Writer SHALL use 2-byte coding for BNo to address desired Block.	**5.6.1.4**	The T5T SHALL respond with data from addressed Block that has Block number BNo.
5.6.1.5	The Reader/Writer SHALL be ready to receive an EXTENDED_READ_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte and BLEN Data bytes, if bit 6 of REQ_FLAG (OF) was set to 0b.	**5.6.1.6**	When bit 6 of REQ_FLAG is set to 0b, the T5T SHALL populate the payload of its EXTENDED_READ_SINGLE_BLOCK_RES Response with a RES_FLAG byte and BLEN Data bytes.
5.6.1.7	The Reader/Writer SHALL be ready to receive an EXTENDED_READ_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte, Block security status and BLEN Data bytes, if bit 6 of REQ_FLAG (OF) was set to 1b. The Reader/Writer SHALL not consider the value of the Block security status byte for the NDEF identification and access procedures (see Section 7).	**5.6.1.8**	When bit 6 of REQ_FLAG is set to 1b, the T5T SHALL populate the payload of its EXTENDED_READ_SINGLE_BLOCK_RES Response with a RES_FLAG byte, the Block security status and BLEN Data bytes.

5.7 EXTENDED_WRITE_SINGLE_BLOCK

The EXTENDED_WRITE_SINGLE_BLOCK Command is used to update the content of one Block in Extended Memory.

This Command has the command code 31h and a 2-byte Block Number (BNo) as PARAMETER. The DATA field contains the new content of the Block and has a length of BLEN bytes. The first DATA field byte represents the first byte of the Block to update and the last DATA field byte represents the last byte of the Block.

Table 30 specifies the EXTENDED_WRITE_SINGLE_BLOCK Command and Table 31 the corresponding Response.

Table 30: Format of EXTENDED_WRITE_SINGLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	2 Bytes	BLEN bytes
REQ_FLAG	31h	UID	BNo	DATA

Table 31: Format of EXTENDED_WRITE_SINGLE_BLOCK_RES

1 Byte
RES_FLAG

Requirements 24: EXTENDED_WRITE_SINGLE_BLOCK Command

Reader/Writer		Type 5 Tag	
5.7.1.1	The Reader/Writer SHALL construct the EXTENDED_WRITE_SINGLE_BLOCK_REQ Command, as defined in Table 30.	**5.7.1.2**	When it receives a correctly formed EXTENDED_WRITE_SINGLE_BLOCK_REQ Command, the T5T SHALL respond with the WRITE_SINGLE_BLOCK_RES Response, as defined in Table 31.
5.7.1.3	The Reader/Writer SHALL use 2-byte coding for BNo to address the desired Block.	**5.7.1.4**	When it receives an EXTENDED _WRITE_SINGLE_BLOCK_REQ Command, the T5T SHALL update the data of the Block addressed by the Block number with the data from the DATA field.
5.7.1.5	The Reader/Writer SHALL set b6 of REQ_FLAG (OF) in accordance with Section 3.2		
5.7.1.6	The Reader/Writer SHALL be ready to receive an EXTENDED_WRITE_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte.	**5.7.1.7**	When it sends an EXTENDED_WRITE_SINGLE_BLOCK_RES Response, the T5T SHALL populate the payload with a RES_FLAG byte.

5.8 EXTENDED_LOCK_SINGLE_BLOCK

The EXTENDED_LOCK_SINGLE_BLOCK Command is used to lock the content of one Block in an Extended Memory.

This Command has the command code 32h and a 2-byte Block Number (BNo) as PARAMETER but no DATA field.

Table 32 specifies the EXTENDED_LOCK_SINGLE_BLOCK Command and Table 33 the corresponding Response.

Table 32: Format of EXTENDED_LOCK_SINGLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	2 Bytes
REQ_FLAG	32h	UID	BNo

Table 33: Format of EXTENDED_LOCK_SINGLE_BLOCK_RES

1 Byte
RES_FLAG

Requirements 25: EXTENDED_LOCK_SINGLE_BLOCK Command

Reader/Writer		Type 5 Tag	
5.8.1.1	The Reader/Writer SHALL construct the EXTENDED_LOCK_SINGLE_BLOCK_REQ Command, as defined in Table 32.	**5.8.1.2**	When it receives a correctly formed EXTENDED_LOCK_SINGLE_BLOCK_REQ Command, the T5T SHALL respond with the EXTENDED_LOCK_SINGLE_BLOCK_RES Response, as defined in Table 33.
5.8.1.3	The Reader/Writer SHALL use BNo to address the desired Block.	**5.8.1.4**	The T5T SHALL lock the data of the addressed Block with Block number BNo.
5.8.1.5	The Reader/Writer SHALL set b6 of REQ_FLAG (OF) in accordance with Section 3.2.	**5.8.1.6**	The T5T SHALL respond in accordance with Section 3.1.
5.8.1.7	The Reader/Writer SHALL be ready to receive an EXTENDED_LOCK_SINGLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte.	**5.8.1.8**	When it sends an EXTENDED_LOCK_SINGLE_BLOCK_RES Response, the T5T SHALL populate the payload with a RES_FLAG byte.

5.9 EXTENDED_READ_MULTIPLE_BLOCK

The EXTENDED_READ_MULTIPLE_BLOCK Command is used to read the content of one or more Blocks in an Extended Memory.

This Command has the command code 33h and a 2-byte Block Number (BNo) specifying the first Block to read, followed by one byte specifying the number of additional Blocks to read (NB) as PARAMETER but no DATA field. The Reader/Writer can request the Type 5 Tag to return in its Response the Block security status, by setting bit 6 (OF) of REQ_FLAG to 1b.

Table 34 specifies the EXTENDED_READ_MULTIPLE_BLOCK Command and Table 35 the corresponding Response. The number of data Blocks (and optionally Block security bytes) returned by the Type 5 Tag in its Response is (NB +1).

Table 34: Format of EXTENDED_READ_MULTIPLE_BLOCK_REQ

1 Byte	1 Byte	8 Bytes (optional)	2 Bytes	2 Bytes
REQ_FLAG	33h	UID	BNo	NB

Table 35: Format of EXTENDED_READ_MULTIPLE_BLOCK_RES

1 Byte	(NB+1)*	[1 Byte (optional)	BLEN Bytes]
RES_FLAG		Block security status	DATA

Requirements 26: EXTENDED_READ_MULTIPLE_BLOCK_REQ Command

Reader/Writer		Type 5 Tag	
5.9.1.1	The Reader/Writer SHALL construct the EXTENDED_READ_MULTIPLE_BLOCK_REQ Command, as defined in Table 34.	**5.9.1.2**	When it receives a correctly formed EXTENDED_READ_MULTIPLE_BLOCK_REQ Command, the T5T SHALL respond with the EXTENDED_READ_MULTIP¨LE_BLOCK_RES Response, as defined in Table 35.
5.9.1.3	The Reader/Writer SHALL use 2-byte coding for BNo to address the first desired Block.	**5.9.1.4**	The T5T SHALL respond with data starting from addressed Block with Block number BNo up to Block number BNo + NB.
5.9.1.5	The Reader/Writer SHALL use NB to indicate how many adjacent Blocks in addition SHALL be read.		
5.9.1.6	The Reader/Writer SHALL be ready to receive an EXTENDED_READ_MULTIPLE_BLOCK_RES Response with a payload composed of a RES_FLAG byte and (NB+1) * BLEN Data bytes, if b6 (OF) of REQ_FLAG was set to 0b.	**5.9.1.7**	When it sends an EXTENDED_READ_MULTIPLE_BLOCK_RES Response, the T5T SHALL populate the payload with a RES_FLAG byte and (NB+1) * BLEN Data bytes, if b6 (OF) of REQ_FLAG was set to 0b.
5.9.1.8	The Reader/Writer SHALL be ready to receive an EXTENDED_READ_MULTIPLE_BLOCK_RES Response with a payload composed of an RES_FLAG byte and (NB + 1) tuples of Block security status byte and BLEN data bytes, if bit 6 of REQ_FLAG (OF) was set to 1b. The Reader/Writer SHALL not consider the value of the Block security status bytes for the NDEF identification and access procedures (see Section 7).	**5.9.1.9**	When bit 6 of REQ_FLAG is set to 1b, the T5T SHALL populate the payload of its EXTENDED_READ_MULTIPLE_BLOCK_RES Response with a RES_FLAG byte, and (NB + 1) tuples of the Block security status byte and BLEN data bytes.

5.10 SELECT

The SELECT_REQ Command is used to set one Type 5 Tag to the **SELECTED** State to, e.g., optimize transaction times.

This Command has the command code 25h and an 8-byte UID as a Parameter, but no DATA field. This Command is always sent in addressed mode.

Table 36 specifies the SELECT Command and Table 37 the corresponding Response.

Table 36: Format of the SELECT_REQ Command

1 Byte	1 Byte	8 Bytes
REQ_FLAG	25h	UID

Table 37: Format of SELECT_RES

1 Byte
RES_FLAG

Requirements 27: Select Command

Reader/Writer		Type 5 Tag	
5.10.1.1	The Reader/Writer SHALL construct the SELECT_REQ Command, as defined in Table 36.	**5.10.1.2**	When it receives a correctly formed SELECT_REQ Command and the **`SELECTED`** State is supported, the T5T SHALL respond with the SELECT_RES Response, as defined in Table 37.
5.10.1.3	The Reader/Writer SHALL set b5 (AMS) of REQ_FLAG to 1b.		
5.10.1.4	The Reader/Writer SHALL be ready to receive a SELECT_RES Response with a payload composed of a RES_FLAG byte.	**5.10.1.5**	When it sends a SELECT_RES Response, the T5T SHALL populate the payload with a RES_FLAG byte.

5.11 SLPV_REQ

The SLPV_REQ Command is used to set a Type 5 Tag to the **`QUIET`** State to, e.g., process further tags.

This Command has the command code 02h and an 8-byte UID as a Parameter, but no DATA field. This Command is always sent in addressed mode.

Table 38 specifies the SLPV_REQ Command. A Type 5 Tag does not respond to a SLPV_REQ Command.

Table 38: Format of the SLPV_REQ Command

1 Byte	1 Byte	8 Bytes
REQ_FLAG	02h	UID

Requirements 28: SLPV_REQ

Reader/Writer		Type 5 Tag	
5.11.1.1	The Reader/Writer SHALL construct the SLPV_REQ Command, as defined in Table 38.	**5.11.1.2**	When it receives a correctly formed SLPV_REQ Command, the T5T SHALL not respond.
5.11.1.3	The Reader/Writer SHALL set b5 (AMS) of REQ_FLAG to 1b.		
5.11.1.4	The Reader/Writer SHALL treat the SLPV_REQ Command as acknowledged by the device without receiving a Response within $FDT_{V,POLL}$. $FDT_{V,POLL}$ is defined in [DIGITAL].		
5.11.1.5	Following the end of the SLPV_REQ Command the Reader/Writer SHALL wait at least $FDT_{V,POLL}$ before it sends the next Command. $FDT_{V,POLL}$ is defined in [DIGITAL].	**5.11.1.6**	The T5T SHALL be ready to receive the next Command no later than $FDT_{V,POLL}$ after the end of the SLPV_REQ Command. $FDT_{V,POLL}$ is defined in [DIGITAL].

5.12 Timing Requirements

The T5T use NFC-V Frame Delay Times. See [DIGITAL] for further details.

Requirements 29: Timing Requirements

Reader/Writer		Type 5 Tag	
5.12.1.1	The Reader/Writer SHALL comply with the NFC-V Timing Requirements for the Poller, as defined in [DIGITAL].	**5.12.1.2**	The T5T SHALL comply with the NFC-V Timing Requirements for the Listener, as defined in [DIGITAL].

5.13 Checking the Presence of a Type 5 Tag

The Reader/Writer can check whether a T5T 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 T5T that confirms its presence.

Requirements 29: Presence Check Procedure

Reader/Writer	
5.13.1.1	The Reader/Writer MAY check whether a T5T is still present in the Operating Field by sending a READ_SINGLE_BLOCK Command with BNo = 0 in accordance with section 5.1.1 and section 5.2, waiting for at least 200 ms between consecutive Commands.

If the Reader/Writer receives a valid READ_SINGLE_BLOCK_RES, the T5T is still present in the Operating Field and ready to receive another Type 5 Tag Command.

6 Type 5 Tag State Machine

6.1 General State Requirements

This section defines the State diagram of a Type 5 Tag and the Commands used by a Reader/Writer to manage State transitions.

A Type 5 Tag operates in three States called "READY", "QUIET" and "SELECTED".

In **READY** State all Commands where the SMS is not set are processed.

In **QUIET** State the Type 5 Tag does not process the INVENTORY Command. Any other Commands where the AMS is set are processed.

In **SELECTED** State every Command where SMS is set is processed. There can be at most one Type 5 Tag in **SELECTED** State (the one addressed by the last SELECT_REQ Command).

Requirements 30: General Requirements for All States

Reader/Writer		Type 5 Tag	
6.1.1.1	To bring the T5T to **POWER_OFF** State the Reader/Writer SHALL switch the Operating Field from Operating Field On to Operating Field Off for at least **t_{FIELD_OFF}**, as defined in [ACTIVITY].	**6.1.1.2**	When the Remote Field changes from Remote Field On to Remote Field Off, the T5T SHALL enter the **POWER_OFF** State no later than **t_{FIELD_OFF}**, as defined in [ACTIVITY].
		6.1.1.3	The T5T SHALL support **POWER_OFF**, **READY** and **QUIET** States.
		6.1.1.4	The T5T MAY support **SELECTED** State.

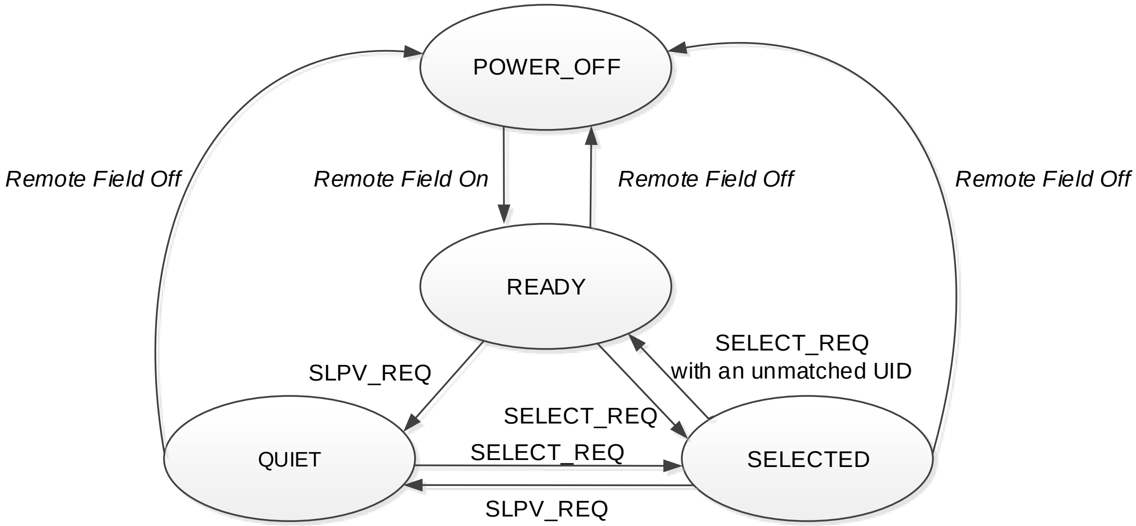

Figure 3: Type 5 Tag State Diagram

6.2 POWER_OFF State

The requirements in this section apply to the Type 5 Tag in its **POWER_OFF** State.

Requirements 31: POWER_OFF State

Reader/Writer		Type 5 Tag	
6.2.1.1	To bring the T5T to **READY** State, the Reader/Writer SHALL switch the RF field On.	**6.2.1.2**	In the **POWER_OFF** State, if the RF field is On, then the T5T SHALL enter the **READY** State.

6.3 READY State

The requirements in this section apply to Type 5 Tag in the **READY** State.

Requirements 32: READY State

Reader/Writer		Type 5 Tag	
6.3.1.1	To bring the T5T to a **SELECTED** State, the Reader/Writer SHALL send a SELECT_REQ Command with the UID of the targeted tag.	**6.3.1.2**	If the SELECT_REQ is supported, the T5T SHALL enter the **SELECTED** State after it receives a valid SELECT_REQ Command with the matching UID.
6.3.1.3	To bring the T5T to the **QUIET** State, the Reader/Writer SHALL send a SLPV_REQ Command with the UID of the targeted tag.	**6.3.1.4**	The T5T SHALL enter the **QUIET** State after it receives a valid SLPV_REQ Command with the matching UID.

6.4 SELECTED State

The requirements in this section apply to Type 5 Tag in the **SELECTED** State.

Requirements 33: SELECTED State

Reader/Writer		Type 5 Tag	
6.4.1.1	To bring the T5T to the **QUIET** State, the Reader/Writer SHALL send a SLPV_REQ Command with the UID of the targeted tag.	**6.4.1.2**	The T5T SHALL enter the **QUIET** State after it receives a valid SLPV_REQ Command with the matching UID.
6.4.1.3	To address a T5T in **SELECTED** State, the Reader/Writer SHALL set the REQ_FLAG b4 (SMS) to 1b and SHALL set b2 (Inventory) to 0b.	**6.4.1.4**	The T5T in **SELECTED** State SHALL process received Commands that contain a REQ_FLAG with b4 (SMS) set to 1b and b2 (Inventory) set to 0b.
		6.4.1.5	The T5T SHALL enter the **READY** State after it receives a valid SELECT_REQ Command with an unmatched UID.

6.5 QUIET State

The requirements in this section apply to Type 5 Tag in the **QUIET** State.

Requirements 34: QUIET State

Reader/Writer		Type 5 Tag	
6.5.1.1	To bring the T5T to **SELECTED** State, the Reader/Writer SHALL send a SELECT_REQ Command with the UID of the targeted tag.	**6.5.1.2**	The T5T SHALL enter the **SELECTED** State after it receives a valid SELECT_REQ Command with the matching UID.
6.5.1.3	To address a T5T in **QUIET** State, the Reader/Writer SHALL set the REQ_FLAG's b5 (AMS) to 1b and SHALL set b2 (Inventory) to 0b.	**6.5.1.4**	The T5T in **QUIET** State SHALL only process received Commands that contain a REQ_FLAG with b5 (AMS) set to 1b and b2 (Inventory) set to 0b.

..

7 NDEF Identification and Access

This section describes how the Reader/Writer treats the different versions of the T5T and how it stores and accesses the NDEF data in the T5T.

A Reader/Writer is able to read [ISO/IEC 15693] VICCs formatted as defined in, and compliant to, this specification.

7.1 NDEF Identification

A Reader/Writer can identify the T5T by reading the CC to detect the mapping version and the access information of the NDEF data.

7.2 Version Treating

Byte 1 of the CC contains the Mapping Version of this T5T. The mapping version is encoded with a major version number and minor version number.

The handling of the different NDEF mapping version numbers applied to the Type 5 Tag (called T5VNo) and the one implemented in the Reader/Writer (called T5TR/W_VNo) is explained in the following requirements.

Requirements 35: Handling of the Mapping Version Numbers

Reader/Writer	
7.2.1.1	If the Major T5TR/W_VNo is equal to major T5VNo, and minor T5TR/W_VNo is bigger than or equal to minor T5VNo, the Reader/Writer SHALL access the T5T and SHALL use all features of the applied mapping to this T5T.
7.2.1.2	If major T5TR/W_VNo is equal to major T5VNo, and minor T5TR/W_VNo is lower than minor T5VNo then possibly not all features of the T5T can be accessed. The Reader/Writer SHALL use all its features and SHALL access this T5T.
7.2.1.3	If major T5TR/W_VNo is smaller than major T5VNo the data format is incompatible. The Reader/Writer SHALL reject this T5T.
7.2.1.4	If major T5TR/W_VNo is bigger than major T5VNo, the Reader/Writer might implement the support for previous versions of this specification in addition to its main version. In case the Reader/Writer support the version announced by the T5T, it SHALL access the T5T. In case the Reader/Writer does not support the version announced by the T5T, it SHALL reject the T5T.

Future versions of this specification have to define the allowed actions to a Type 5 Tag with a version number lower than the version number of the Reader/Writer (e.g., whether it is allowed to upgrade the Type 5 Tag to the new version).

7.3 NDEF Storage

The data format of the NDEF Message is defined in [NDEF]. The NDEF Message is stored inside the V-field of the NDEF Message TLV (see section 4.4.3) in the T5T_Area.

7.4 Life Cycle

7.4.1 Type 5 Tag Life Cycle States

A Reader/Writer can detect a T5T in different Life Cycle States.

The CC and the length field of the NDEF message determine the Life Cycle State.

Table 39 describes the valid States of a T5T.

Table 39: Type 5 Tag Life Cycle States

State	Description
`INITIALIZED`	The T5T is formatted with a CC and an empty NDEF Message TLV; the access conditions allow both Reading and Writing data.
`READ/WRITE`	The T5T is formatted with a CC and a non-empty NDEF Message TLV; the access conditions allow both Reading and Writing data.
`READ-ONLY`	The T5T is formatted with a CC and a non-empty NDEF Message TLV; the access conditions are restricted to Read-Only.

NOTE The Terminator TLV is used as defined in 4.4.4.

Requirements 36: Type 5 Tag Life Cycle States

Reader/Writer		Type 5 Tag	
7.4.1.1	The Reader/Writer MAY apply a repair mechanism, if the T5T is not in one of the three valid Life Cycle States, as defined in Table 39.	**7.4.1.2**	The T5T SHALL be in one of three valid Life Cycle States, as defined in Table 39.

7.4.2 INITIALIZED State

In the **INITIALIZED** State the CC and the T5T_Area are accessible for reading and writing data.

Requirements 37: INITIALIZED State

Reader/Writer		Type 5 Tag	
7.4.2.1	To identify that a T5T is in the **INITIALIZED** State, the Reader/Writer SHALL check that: • The CC area is encoded as described in section 4.3.1, with b3 to b0 of byte 1 equal to 0000b (read/write access granted). • The T5T_Area contains an NDEF Message TLV. • The L-field of the NDEF Message TLV is equal to 00h.	**7.4.2.2**	A T5T in the **INITIALIZED** State SHALL conform to: • The CC area is encoded as described in section 4.3.1, with b3 to b0 of byte 1 equal to 0000b (read/write access granted). • The T5T_Area contains an NDEF Message TLV. • The L-field of the NDEF Message TLV is equal to 00h.

7.4.3 READ/WRITE State

In the **READ/WRITE** State the CC and the T5T_Area are accessible for reading and writing data.

Requirements 38: READ/WRITE State

Reader/Writer		Type 5 Tag	
7.4.3.1	To identify that a T5T is in **READ/WRITE** State, the Reader/Writer SHALL check that: • The CC area is encoded as described in section 4.3.1, with b3 to b0 of byte 1 equal to 0000b (read/write access granted). • The T5T_Area contains an NDEF Message TLV. • The L-field of the NDEF Message TLV is not equal to 00h.	**7.4.3.2**	A T5T in **READ/WRITE** State SHALL conform to: • The CC area is encoded as described in section 4.3.1, with b3 to b0 of byte 1 equal to 0000b (read/write access granted). • The T5T_Area contains an NDEF Message TLV. • The L-field of NDEF Message TLV is not equal to 00h and equal to the actual length of the NDEF Message in the V-field.

7.4.4 READ-ONLY State

In the **READ-ONLY** State, the CC and the T5T_Area are set to read-only.

Requirements 39: READ-ONLY State

Reader/Writer		Type 5 Tag	
7.4.4.1	To identify that a T5T is in **READ-ONLY** State, the Reader/Writer SHALL check that: • The CC area is encoded as described in section 4.3.1, with b3 and b2 of byte 1 equal to 00b and b1 and b0 of byte 1 not equal to 00b (only read access granted). • The T5T_Area contains an NDEF Message TLV. • The L-field of NDEF Message TLV is not equal to 00h.	**7.4.4.2**	A T5T in **READ-ONLY** State SHALL conform to: • The CC area SHALL be encoded as described in Table 6 of section 4.3.1, with b3 and b2 of byte 1 equal to 00b and b1 and b0 of byte 1 equal to 10b or 11b. • The T5T_Area SHALL contain an NDEF Message TLV. • The L-field of NDEF Message TLV SHALL NOT be equal to 00h and SHALL be equal to the actual length of the NDEF Message in the V-field. • All Blocks in the CC area and in the T5T_Area SHALL be locked.

7.5 NDEF Procedures

7.5.1 General requirements

The data format of the NDEF Message is defined in [NDEF]. The NDEF Message is stored inside the V-field of the NDEF Message TLV (see Section 4.4.3) in the TLVs_Area.

Each NDEF procedure defines a sequence of Commands to manage the NDEF data on the T5T.

The NDEF procedures assume that the Reader/Writer has performed the Technology Detection, Collision Detection and Device Activation activities, as documented in the [ACTIVITY] specification. The NDEF procedures are performed by the Reader/Writer during the Data Exchange activity, as defined in [ACTIVITY]. A Reader/Writer can put a T5T in **SELECTED** State before executing any NDEF procedure defined below.

NOTE Communicating with a T5T in **SELECTED** State reduces the transaction time because it avoids the inclusion of the 8 byte UID in each command that uses the addressed mode during the NDEF procedures.

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 this procedure can also be used to change the lifecycle State (see section 7.6).

Requirements 40: NDEF Procedures – Type 5 Tag

Type 5 Tag	
7.5.1.1	A T5T SHALL NOT change the content of its CC during Operating Field On condition, except when requested by a Reader/Writer. The T5T MAY change the content of its CC during Operating Field Off condition.
7.5.1.2	A T5T SHALL NOT change the NDEF Message TLV start position in the T5T_Area during Operating Field On, except when requested by a Reader/Writer. The T5T MAY change the NDEF Message TLV start position in the TLVs_Area during 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 40.

Table 40: NDEF Procedures – Greedy Collection

Name	Format	Size (bytes)	Description
GRE_CC_CONTENT	byte	8	Value of the Capability Container to store a value of 4 or 8 bytes
GRE_NDEF_TLV_POS_BN	byte	2	Value of the Block number which contains the start of the NDEF Message TLV
GRE_NDEF_TLV_POS_BY	byte	1	Value of the byte number which contains the start of the NDEF Message TLV
GRE_NDEF_TLV_LENGTH	byte	2	Value of the NDEF Message TLV length

Requirements 41: Greedy Collection

Reader/Writer	
7.5.2.1	The Reader/Writer SHALL delete the Greedy collection parameters by setting all bytes to 00h before using the NDEF detection procedure.

7.5.3 NDEF Detection Procedure

This section describes how the Reader/Writer can detect the presence of an NDEF Message (see [NDEF]).

The detection procedure is based on the coding of the CC, and the presence of an NDEF Message TLV that can contain an NDEF Message.

The NDEF detection procedure determines whether the Tag is configured for NDEF data and if yes, retrieves the data to fill the Greedy collection parameters from the T5T.

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

Figure 4 shows the procedure to detect the NDEF Message.

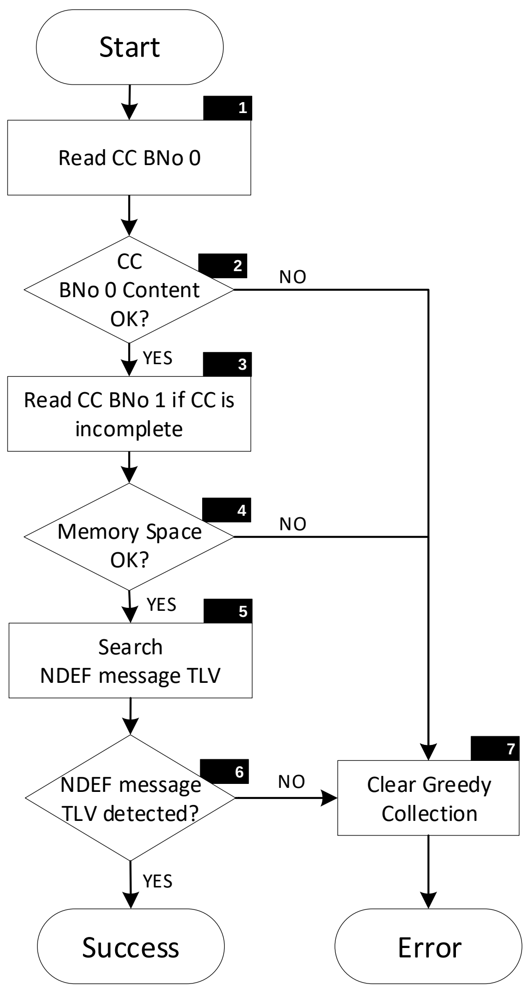

Figure 4: NDEF Detection Flowchart

Requirements 42: NDEF Detection Procedure

Reader/Writer

7.5.3.1 Symbol 1

The Reader/Writer SHALL read the CC Block BNo 0 using one READ_SINGLE_BLOCK Command defined in section 5.2 and fill the parameter GRE_CC_CONTENT bytes 0 to 3.

Afterwards the Reader/Writer SHALL proceed to Symbol 2.

7.5.3.2 Symbol 2

The Reader/Writer SHALL verify in the content of GRE_CC_CONTENT that:

- GRE_CC_CONTENT [0] (Magic Number) SHALL be equal to E1h (only 1-byte address mode is supported) or E2h (2-byte address mode is supported).
- GRE_CC_CONTENT [1] (Version and Access conditions) SHALL contain the correct version number, with b7 to b4 indicating a supported version number (see section 4.3.1), and b3b2 is 00b (Read Access Always allowed). The Reader/Writer MAY continue the NDEF read procedure independent of the Read Access conditions.

If the Reader/Writer intends to perform the NDEF write procedure afterwards and b1b0 of CC_CONTENT [1] (Access Conditions for Write) is different from 00b, the Reader/Writer SHOULD proceed to Symbol 7.

NOTE A Reader/Writer compliant to previous versions of this specification can behave differently than recommended above.

If the GRE_CC_CONTENT is not correct as described by the bullets, the Reader/Writer SHALL proceed to Symbol 7. Otherwise the Reader/Writer SHALL proceed to Symbol 3.

7.5.3.3 Symbol 3

If the read of the CC is complete with BNo = 0, the Reader/Writer SHALL proceed to Symbol 4

Otherwise the Reader/Writer SHALL read Block BNo 1 and fill the parameter GRE_CC_CONTENT bytes 4 to 7 using either the READ_SINGLE_BLOCK (see section 5.2), EXTENDED_READ_SINGLE_BLOCK (see section 5.6), READ_MULTIPLE_BLOCK (see section 5.5) or EXTENDED_READ_MULTIPLE_BLOCK (see section 5.9) commands depending of the capability of the tag.

For READ_MULTIPLE_BLOCK and EXTENDED_READ_MULTIPLE_BLOCK commands, the READER/WRITER SHALL set either BNo = 0 and NB = 1, or BNo = 1 and NB = 0.

Afterwards the Reader/Writer SHALL proceed to Symbol 4.

7.5.3.4 Symbol 4

If the Reader/Writer intends to perform the NDEF write procedure afterwards and the NDEF Message to be written is larger than the available space in the T5T_Area, then the Reader/Writer SHOULD proceed to Symbol 7.

Otherwise, the Reader/Writer SHALL proceed to Symbol 5

7.5.3.5 Symbol 5

If the Reader/Writer did not detect the NDEF Message TLV in Symbol 1, the Reader/Writer SHALL read the T5T_Area sequentially using the READ_SINGLE_BLOCK (see section 5.2), EXTENDED_READ_SINGLE_BLOCK (see section 5.6), READ_MULTIPLE_BLOCK (see section 5.5) or EXTENDED_READ_MULTIPLE_BLOCK (see section 5.9) Commands searching for an NDEF Message TLV:

- The Reader/Writer SHALL start to read from Block 1 or 2 depending on the Blocks read in Symbols 1 and 3.
- The Reader/Writer MAY skip reading memory Blocks that are used completely by the V-field of a TLV having a Tag of an RFU value.
- The Reader/Writer SHALL stop searching as soon as the T-field and L-field of the NDEF Message TLV is found or if the end of the T5T_Area has been reached.

If the T-field and L-field of the NDEF Message TLV is found, the Reader/Writer SHALL fill the parameters GRE_NDEF_TLV_POS_BN, GRE_NDEF_TLV_POS_BY and GRE_NDEF_TLV_LENGTH and SHALL proceed to Symbol 6.

NOTE The Reader/Writer can detect the NDEF Message TLV in Symbol 1 only if BLEN is larger than 8. If BLEN is 4, the Reader/Writer cannot detect the NDEF Message TLV in Symbol 1 because Block 0 only contains bytes 0 to 3 of the Capability Container. If BLEN is 8, the Reader/Writer cannot detect the NDEF Message TLV in Symbol 1 because the Capability Container has a length of eight Bytes to ensure that T5T_Area ends on a Block boundary.

7.5.3.6 Symbol 6

If GRE_NDEF_TLV_POS_BN is equal to 0000h and GRE_NDEF_TLV_POS_BY is equal to 0000h then the Reader/Writer SHALL proceed to Symbol 7.

If GRE_NDEF_TLV_LENGTH is equal to 0000h and the Reader/Writer intends to perform the NDEF read procedure afterwards, then the Reader/Writer SHALL proceed to Symbol 7.

Otherwise the Reader/Writer SHALL conclude with Success.

7.5.3.7 Symbol 7

The Reader/Writer SHALL clear the entire Greedy Collection and afterwards SHALL conclude with Error.

NOTE The NDEF detection procedure only reads the length of the stored NDEF Message TLV and does not parse the data.

7.5.4 NDEF Read Procedure

The following figure shows the procedure to read the NDEF Message from a Type 5 Tag.

The purpose of this procedure is to read the Blocks containing NDEF data to assemble the NDEF Message stored on the T5T.

Figure 5 provides the flowchart for the NDEF read procedure:

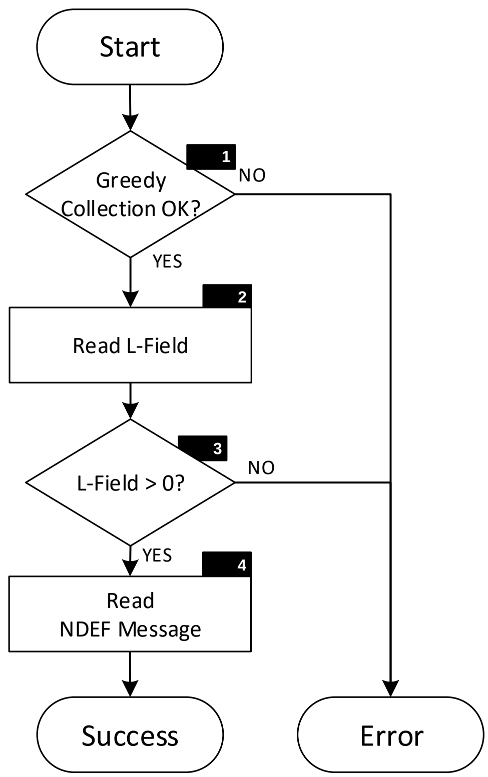

Figure 5: NDEF Read Procedure Flowchart

Requirements 43: NDEF Read Procedure

Reader/Writer	
7.5.4.1	**Symbol 1** If GRE_NDEF_TLV_POS_BN and GRE_NDEF_TLV_POS_BY are both equal to 0000h then 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 the L-field of the NDEF Message TLV using READ_SINGLE_BLOCK (see section 5.2), EXTENDED_READ_SINGLE_BLOCK (see section 5.6), READ_MULTIPLE_BLOCK (see section 5.5) or EXTENDED_READ_MULTIPLE_BLOCK (see section 5.9) Commands. The Reader/Writer SHALL fill the parameter GRE_NDEF_TLV_LENGTH and SHALL proceed to Symbol 3.
7.5.4.3	**Symbol 3** If the length of the NDEF Message TLV is equal to 0000h, then the Reader/Writer SHALL conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 4.
7.5.4.4	**Symbol 4** The Reader/Writer SHALL use one or more READ_SINGLE_BLOCK (see section5.2), EXTENDED_READ_SINGLE_BLOCK (see section 5.6), READ_MULTIPLE_BLOCK (see section 5.5) or EXTENDED_READ_MULTIPLE_BLOCK (see section 5.9) Commands to sequentially read the entire NDEF Message from the NDEF Message TLV, including optionally T-field and/or the L-field using the information contained in GRE_NDEF_TLV_POS_BN, GRE_NDEF_TLV_POS_BY and GRE_NDEF_TLV_LENGTH. Afterwards the Reader/Writer SHALL proceed to Success.

7.5.5 NDEF Write Procedure

The purpose of this procedure is to write an NDEF Message to a T5T. Any existing NDEF data on the T5T will be overwritten. It is designed to avoid corrupting the T5T content if the communication gets lost in the middle of the WRITE procedure: the NDEF Message TLV is therefore first "re-initialized" to an empty NDEF Message TLV by setting the Length field to zero. At the end of the procedure, after the NDEF Message has been written properly, the Length field is updated to reflect the actual length of the NDEF Message.

Figure 6 provides the flowchart for the NDEF write procedure:

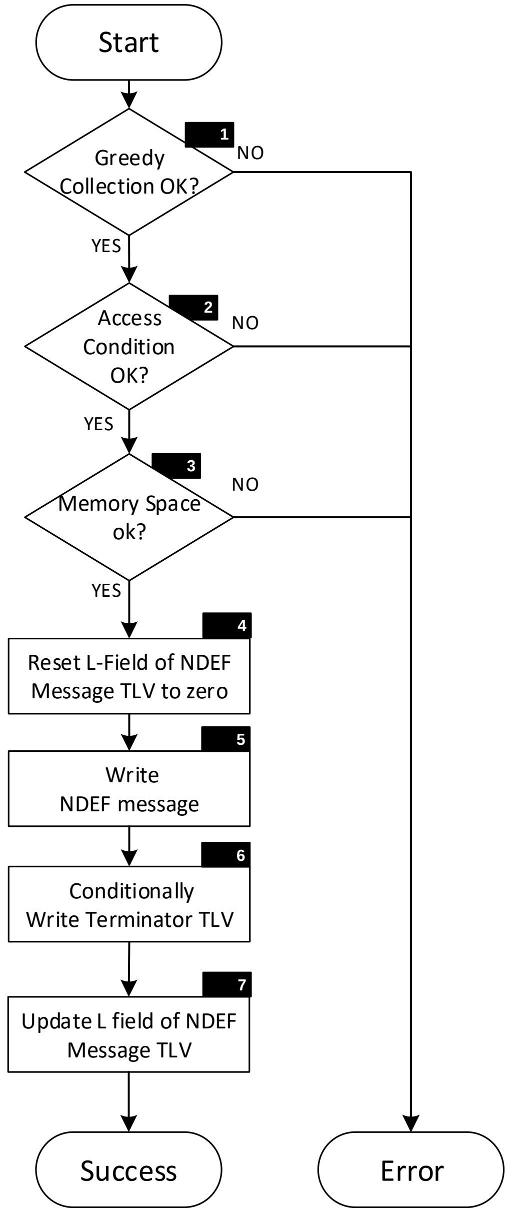

Figure 6: NDEF Write Procedure Flowchart

Requirements 44: NDEF Write Procedure

Reader/Writer	
7.5.5.1	During the NDEF write procedure the Reader/Writer SHALL NOT change the content of the CC field.
7.5.5.2	**Symbol 1** If GRE_NDEF_TLV_POS_BN and GRE_NDEF_TLV_POS_BY are both equal to 0000h then the Reader/Writer SHALL conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 2.
7.5.5.3	**Symbol 2** The Reader/Writer SHALL verify the content of GRE_CC_CONTENT: • b1b0 of GRE_CC_CONTENT[1] (Access Conditions for Write) SHALL be different from 11b (**READ-ONLY** State). If the GRE_CC_CONTENT is not correct as described above, the Reader/Writer SHALL conclude with Error. If b1b0 of CC_CONTENT [1] (Access Conditions for Write) is different from 00b, the Reader/Writer MAY conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 3.
7.5.5.4	**Symbol 3** The Reader/Writer SHALL verify that the available space in the T5T_Area is large enough to contain the NDEF Message TLV. If there is not enough space in the T5T_Area, then the Reader/Writer SHALL conclude with Error. Otherwise the Reader/Writer SHALL proceed to Symbol 4.
7.5.5.5	**Symbol 4** The Reader/Writer SHALL use a WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK commands to reset the NDEF Message TLV to "empty" by setting the value of the L-field to a value of 0. Afterwards the Reader/Writer SHALL proceed to Symbol 5. Optionally, before setting the value of the L-field to 0, if the T-field of the NDEF Message TLV does not start at the 1st byte of a Block, the Reader/Writer MAY use one or more READ_SINGLE_BLOCK (see section 5.2), EXTENDED_READ_SINGLE_BLOCK (see section 5.6), READ_MULTIPLE_BLOCK (see section 5.5) or EXTENDED_READ_MULTIPLE_BLOCK (see section 5.9) commands to read the Blocks containing the NDEF T-field and/or L-field of the NDEF Message TLV. If the Reader/Writer uses the READ_MULTIPLE_BLOCK or EXTENDED_READ_MULTIPLE_BLOCK command, it SHALL set the parameter NB to 0 or 1.
NOTE	The L-field value of 0 can be either coded as 1 byte 00h, or 3 bytes 00h 00h 00h.

7.5.5.6 Symbol 5

The Reader/Writer SHALL use one or more WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK commands to sequentially write the new data to the V-field of the NDEF Message TLV, starting immediately after the Length field:

- The 2nd available byte after the T-field, if the new NDEF Message length is smaller than 255 bytes (1-byte Length field)
- The 4th available byte after the T-field, if the new NDEF Message length is larger than 254 bytes (3-byte Length field).

For optimizations the Reader/Writer MAY combine Symbols 4 and 5, thus MAY write the L-field of 00h and the first part of the V-Field in one WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK Command

Furthermore, if the last byte of the NDEF Message TLV does not end at the last byte of the T5T_Area and is not located at the last byte of the last Block, then the Reader/Writer MAY write the last bytes of the NDEF Message TLV and the Terminator TLV using a single WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK Command.

Afterwards the Reader/Writer SHALL proceed to Symbol 6.

7.5.5.7 Symbol 6

If the last byte of the NDEF Message TLV does not end at the last byte of the T5T_Area and the Reader/Writer has not written the Terminator TLV in Symbol 5, the Reader/Writer SHALL use a WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK command to write the Terminator TLV immediately after the NDEF Message TLV.

If the Terminator TLV is not the BLENth byte of a Block, the Reader/Writer SHALL set the remaining Bytes of the same Block after the Terminator TLV to 00h.

7.5.5.8	**Symbol 7**
	The Reader/Writer SHALL use a WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK command to update 1 or 3 bytes for the L-field to set the length of the new NDEF Message. If the entire NDEF Message TLV and, if applicable, the optional terminator TLV, fit into a single Block, the Reader/Writer MAY write the entire NDEF Message TLV and the optional terminator TLV using a single WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK Command. In this case the reset of the L-Field MAY be skipped (Symbol 4) If the Terminator TLV is not the BLENth byte of a Block, the Reader/Writer SHALL set the remaining Bytes of the same Block after the Terminator TLV to 00h. Optionally, before setting the L-field to the length of the new NDEF Message, the Reader/Writer MAY use one or more READ_SINGLE_BLOCK (see section5.2), EXTENDED_READ_SINGLE_BLOCK (see section 5.6), READ_MULTIPLE_BLOCK (see section 5.5) or EXTENDED_READ_MULTIPLE_BLOCK (see section 5.9) commands to read the Blocks containing the NDEF T-field and/or L-field of the NDEF Message TLV before updating the L-field of the NDEF Message TLV. If the Reader/Writer uses the READ_MULTIPLE_BLOCK or EXTENDED_READ_MULTIPLE_BLOCK command, it SHALL set the parameter NB to 0 or 1.
NOTE	When the Reader/Writer writes the L-field and the NDEF Message TLV in one single WRITE_SINGLE_BLOCK or EXTENDED_WRITE_SINGLE_BLOCK Command, in case of a power loss, the Reader/Writer SHALL not assume the T5T will remain in a valid State.
	The Reader/Writer SHALL set GRE_NDEF_TLV_LENGTH equal to the value corresponding to the length of the new NDEF Message. Afterwards the Reader/Writer SHALL conclude with Success.

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 T5T.

Requirements 45: Single NDEF Read Operation

Reader/Writer	
7.5.6.1	To read a single NDEF Message from a T5T 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 T5T.

Requirements 46: Single NDEF Write Operation

Reader/Writer	
7.5.7.1	To write a single NDEF Message to a T5T 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 transitions in State that the Reader/Writer can perform. Figure 7 shows the States and possible transitions.

An NFC Forum Type 5 Tag platform might be issued in any valid State. So, a Type 5 Tag platform might be issued in **INITIALIZED** State, **READ/WRITE** State or even in **READ-ONLY** State having a predefined NDEF Message stored on it.

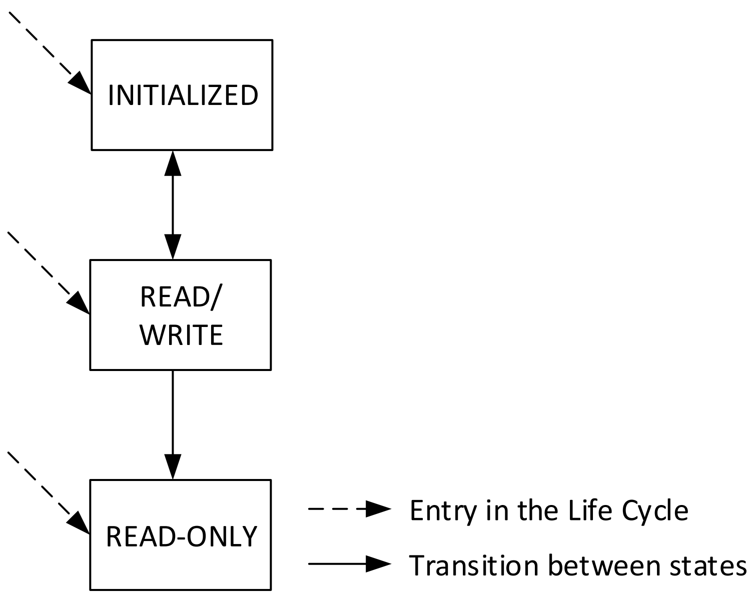

Figure 7: Life Cycle with State Transitions

Table 41 shows the possible transitions that the Reader/Writer can perform:

Table 41: Type 5 Tag State Transitions

From	-	To
INITIALIZED	-	**READ/WRITE**
READ/WRITE	-	**INITIALIZED**
READ/WRITE	-	**READ-ONLY**

7.6.2 State Transitions Support

Requirements 47: State Transitions

Reader/Writer	
7.6.2.1	The Reader/Writer SHALL be able to perform the transitions listed in Table 41.

NOTE The requirements for the State transitions below assume that the Reader/Writer has determined the current Life Cycle State of the T5T.

NOTE If the T5T is not in a valid State, the Reader/Writer can try to repair the T5T.

7.6.3 Transition from INITIALIZED to READ/WRITE

Requirements 48: 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 perform the transition from INITIALIZED to READ/WRITE by replacing the empty NDEF Message TLV with a non-empty NDEF Message TLV.

7.6.4 Transition from READ/WRITE to INITIALIZED

Requirements 49: Transition from READ/WRITE to INITIALIZED

Reader/Writer	
7.6.4.1	The Reader/Writer SHALL use the NDEF write procedure (see section 7.5.5) to perform the transition from **`READ/WRITE`** State to **`INITIALIZED`** State, by setting the L-field of the NDEF Message TLV to 0.

7.6.5 Transitions from READ/WRITE to READ-ONLY

Requirements 50: Transitions from READ/WRITE to READ-ONLY

Reader/Writer	
7.6.5.1	The Reader/Writer SHALL perform the transition of a T5T from the **`READ/WRITE`** State to the **`READ-ONLY`** State by: • Setting bits b1b0 of byte 1 of the CC to 11b (see Table 6). The Reader/Writer SHOULD use the minimum number of WRITE_SINGLE_BLOCK commands to update these bits and SHALL limit the writing to the Blocks that contain the CC. The Reader/Writer SHALL NOT change the value of any other bit. • Locking all Blocks in the CC area and in the T5T_Area by using the LOCK_SINGLE_BLOCK or EXTENDED_LOCK_SINGLE_BLOCK Commands.

The Reader/Writer cannot perform State transition from **READ/WRITE** State to **READ-ONLY** State when the T5T does not support either the LOCK_SINGLE_BLOCK or EXTENDED_LOCK_SINGLE_BLOCK Commands.

A. Exhibit A

No items have been included in Exhibit A.

B. 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 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 its value is D0h 00h 00h.

C. Examples

This annex provides informative examples indicating how to detect, read and write NDEF Messages. The method of parsing these NDEF Messages is out of scope of these examples.

C.1 NDEF Message Detection

To detect the NDEF Message the NDEF detection procedure is used (see section 7.5.3).

C.1.1 Reading the Capability Container

A READ_SINGLE_BLOCK_REQ Command is used to read Block 0 (CC) of the T5T.

Table 42 shows the READ Command to read the CC.

Table 42: READ_SINGLE_BLOCK_REQ

REQ_FLAG	Command Code	BNo
02h	20h	00h

Table 43: READ_SINGLE_BLOCK_RES

		Description
00h	RES_FLAG	No error
E1h or E2h	DATA	Byte 0: if E1h T5T supports 1-byte address mode if E2h T5T supports 2-byte address mode
40h		Byte 1 b7-b4: Mapping Version 1.0 Byte 1 b3-b0: read and write access granted
06h		Byte 2: T5T_Area consists of 48 bytes
00h		Byte 3: no special features granted

C.1.2 Verifying the Capability Container

There is important information in the CC:

- BLEN is 4 bytes, because the DATA field of the Response to the READ_SINGLE_BLOCK_REQ contains 4 bytes.
- Magic Word is E1h; this means only 1-byte address mode is supported by the T5T.
- Magic Word is E2h; this means 1-byte and 2-byte address mode is supported by the T5T.
- Version Info indicates that the Reader/Writer is facing a Mapping Version 1.0 T5T.
- Access Conditions are set in such a way that the Reader/Writer has full read and write access.
- The size of the T5T_Area is 48 bytes.
- The T5T does not support any special features.

C.1.3 Reading T5T_Area

The T5T_Area starts immediately after the Capability Container. In this case the NDEF Message TLV starts at the first byte of Block 1. Consequently, the next step is to read Block 1 and obtain the information about the size of the NDEF Message.

Table 44: READ_SINGLE_BLOCK_REQ

REQ_FLAG	Command code	BNo
02h	20h	01h

Table 45: EXTENDED_READ_SINGLE_BLOCK_REQ

REQ_FLAG	Command code	BNo
02h	30h	0001h

Table 46: READ_SINGLE_BLOCK_RES

		Description
00h	RES_FLAG	No error
03h	DATA	Tag indicating an NDEF Message TLV
03h		Length of the NDEF Message is 3 bytes
D0h		Begin of NDEF Message
00h		Second byte of NDEF Message

C.1.4 Check Presence of NDEF Message TLV

The Tag value of the TLV is 03h and indicates the presence of an NDEF Message TLV.

C.1.5 Check Length of NDEF Message

The Length field of the NDEF Message TLV is 03h.

C.1.6 Empty NDEF Message TLV

Since the length is 03h, there is a non-empty NDEF Message TLV stored on the T5T.

C.1.7 NDEF Message Detected

The T5T is in **READ/WRITE** State.

C.2 NDEF Read Procedure

After it detects a T5T in **READ/WRITE** State, the Reader/Writer may start reading the NDEF Message.

In the NDEF detection procedure the first two bytes of the NDEF Message are already read. In this example the length of the NDEF Message is 3 bytes. The only possible NDEF Message with a length of 3 bytes is the empty NDEF Message. So in this example no further action is needed.

If the length of the NDEF Message is greater than 3 bytes, the next step is to continue reading the complete NDEF Message TLV and Terminator TLV.

C.3 Writing a New NDEF Message TLV to the T5T

This example shows how to write the URL "nfc-forum.org/" and the (optional) description "NFC Forum" in the NDEF Message TLV.

The NDEF detection procedure already proved that the T5T is in the **READ/WRITE** State and that sufficient memory is available.

Table 47 shows the corresponding NDEF Message (see [NDEF]).

Table 47: NDEF Message with NFC Forum URL

Offset	Hexadecimal values	ASCII
00h	D1 02 23 53 70 91 01 0F	..#Sp...
08h	55 03 6E 66 63 2D 66 6F	U.nfc-fo
10h	72 75 6D 2E 6F 72 67 2F	rum.org/
18h	51 01 0C 54 02 65 6E 4E	Q..T.enN
20h	46 43 20 46 6F 72 75 6D	FC Forum

C.3.1 Set NDEF Message TLV Length Field to ZERO

The first WRITE_SINGLE_BLOCK_REQ sets the length of NDEF Message to ZERO and writes the first two bytes of the NDEF Message.

Table 48: WRITE_SINGLE_BLOCK_REQ

REQ_FLAG	Command Code	BNo	DATA
02h	21h	01h	03h 00h D1h 02h

Table 49: EXTENDED_WRITE_SINGLE_BLOCK_REQ

REQ_FLAG	Command Code	BNo	DATA
02h	31h	0001h	03h 00h D1h 02h

Table 50: WRITE_SINGLE_BLOCK_RES

RES_FLAG
00h

C.3.2 Write Remaining Bytes of NDEF Message

Table 51 shows the remaining WRITE_SINGLE_BLOCK_REQ or EXTENDED_WRITE_SINGLE_BLOCK_REQ Commands to write the complete NDEF Message into the T5T_Area. The last line of the table contains the last bytes of the NDEF Message and the Terminator TLV.

The WRITE_SINGLE_BLOCK_RES is always RES_FLAG = 00h.

Table 51: WRITE_SINGLE_BLOCK_REQ Commands

REQ_FLAG	Command Code	BNo	DATA			
02h	21h	02h	23h	53h	70h	91h
02h	21h	03h	01h	0Fh	55h	03h
02h	21h	04h	6Eh	66h	63h	2Dh
02h	21h	05h	66h	6Fh	72h	75h
02h	21h	06h	6Dh	2Eh	6Fh	72h
02h	21h	07h	67h	2Fh	51h	01h
02h	21h	08h	0Ch	54h	02h	65h
02h	21h	09h	6Eh	4Eh	46h	43h
02h	21h	0Ah	20h	46h	6Fh	72h
02h	21h	0Bh	75h	6Dh	FEh	00h

Table 52: EXTENDED_WRITE_SINGLE_BLOCK_REQ Commands

REQ_FLAG	Command Code	BNo	DATA			
02h	31h	0002h	23h	53h	70h	91h
02h	31h	0003h	01h	0Fh	55h	03h
02h	31h	0004h	6Eh	66h	63h	2Dh
02h	31h	0005h	66h	6Fh	72h	75h
02h	31h	0006h	6Dh	2Eh	6Fh	72h
02h	31h	0007h	67h	2Fh	51h	01h
02h	31h	0008h	0Ch	54h	02h	65h
02h	31h	0009h	6Eh	4Eh	46h	43h
02h	31h	000Ah	20h	46h	6Fh	72h
02h	31h	000Bh	75h	6Dh	FEh	00h

C.3.3 Update NDEF Message TLV Length Field

As the last step the correct length of the NDEF Message needs to be set in the Length field of the NDEF Message TLV.

Table 53: WRITE_SINGLE_BLOCK_REQ

REQ_FLAG	Command Code	BNo	DATA
02h	21h	01h	03h 28h D1h 02h

Table 54: EXTENDED_WRITE_SINGLE_BLOCK_REQ

REQ_FLAG	Command Code	BNo	DATA
02h	31h	0001h	03h 28h D1h 02h

Table 55: WRITE_SINGLE_BLOCK_RES

RES_FLAG
00h

D. Revision History

Table 56 outlines the revision history of the Type 5 Tag Technical Specification.

Table 56: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
Type 5 Tag Technical Specification	Version 1.0 September, 2017	Final	Initial publication.	
Type 5 Tag Technical Specification	Version 1.0 April, 2018	Final	Clarifications on endianness and minor corrections; editorial update	Version 1.0 Sep. 2017
Type 5 Tag Technical Specification	Version 1.1, December 2019	Final	Technical updates to support TNEP and to optimize the command sequences. Minor editorial changes.	Version 1.0 April, 2018
Type 5 Tag Technical Specification	Version 1.1, January 2020	Final	Editorial change of copyright notice and Word section changes.	Version 1.1 December 2019
Type 5 Tag Technical Specification	Version 1.1, May 2020	Final	Clarifications added to increase flexibility for Reader/Writer implementations. Further editorial updates.	Version 1.1 January 2020
Type 5 Tag Technical Specification	Version 1.1, June 2020	Final	Format and style updates using Word 2019.	Version 1.1 May 2020
Type 5 Tag Technical Specification	Version 1.2, September 2021	Final	Removal Type 1 Tag; NDEF length update at NDEF Read; Clarifications for SELECTED/READ-ONLY states. Editorial: inserted missing pointers to section 5.2 and updated the copyright dates.	Version 1.1 June 2020

.