Type 1 Tag Operation

Technical Specification

Version 1.2

2014-01-28

[T1TOP]

NFC Forum™

Contents

Figures

Tables

1 Introduction

This specification is part of the NFC Forum documentation about tag types that an NFC Forum Device needs to support in NFC Forum Reader/Writer Mode.

This specification documents how an NFC Forum Device SHALL operate an NFC Forum Type 1 Tag. This is not a specification of the NFC Forum Type 1 Tag itself.

1.1 Objectives

The purpose of this specification is to document the requirements and to specify, with a set of rules and guidelines, the NFC Forum Device operation and management of the Type 1 Tag.

This specification assumes that the Collision Detection and Device Activation activities have been performed as documented in the [ACTIVITY] and [DIGITAL] specifications and these have been completed up to the level of making a single Type 1 Tag identifier (UID) available.

This specification also defines the data mapping and how the NFC Forum Device detects, reads, and writes NDEF data into the Type 1 Tag in order to achieve and maintain interchangeability and interoperability.

The Type 1Tag Operation specifies how to read and write one NDEF Message TLV only.

1.2 Applicable Documents or References

[ACTIVITY] Activity Technical Specification,
NFC Forum

[DIGITAL] Digital Protocol Technical Specification,
NFC Forum

[ISO/IEC_14443] Identification cards – Contactless integrated circuit cards – Proximity cards

Includes:

[ISO/IEC 14443-1:2008], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 1: Physical characteristics

[ISO/IEC 14443-2:2010], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 2: Radio frequency power and signal balance

[ISO/IEC 14443-3:2001], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 3: Initialization and anticollision

[ISO/IEC_14443-3:2001/Amd.1], Identification cards -- Contactless integrated circuit(s) cards -- Proximity cards -- Part 3: Initialization and Anti-collision, 1 February 2001 with Amendment 1: Bit rates of fc/64, fc/32 and fc/16, 15 June 2005; Amendment 3: Handling of reserved fields and values, 22 March 2006; and Corrigendum 1: Amendment 1 - Corrigendum, 29 August 2006

[ISO/IEC 14443-4:2008], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol

ISO/IEC

[NDEF] NFC 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

1.3 Administration

The NFC Type 1 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 to designated members only as stipulated on the most recent Membership Privileges document, during the period of time for which their membership dues are paid.
- Member's distributors and sales representatives MAY use the NFC Forum logo in promoting member's products sold under the name of the member.
- The logo SHALL be printed in black or in color as illustrated on the Logo Page that is available from the NFC Forum at the address above. The aspect ratio of the logo SHALL be maintained, but the size MAY be varied. Nothing MAY be added to or deleted from the logos.
- Since the NFC Forum name is a trademark of the Near Field Communication Forum, the following statement SHALL be included in all published literature and advertising material in which the name or logo appears:

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

1.5 Intellectual Property

The Type 1 Tag Operation Specification conforms to the Intellectual Property guidelines specified in the NFC Forum's *Intellectual Property Rights Policy* (http://nfc-forum.org/wp-content/uploads/2013/11/NFC-Forum-IPR-Policy.pdf), as outlined in the NFC Forum *Rules of Procedure* (http://nfc-forum.org/wp-content/uploads/2013/11/NFC-Forum-Rules-of-Procedure.pdf).

1.6 Special Word Usage

The key words “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT” and “MAY” in this document with the exception of the RESTRICTION ON USE section are to be interpreted as described in [RFC2119].

1.7 Convention and Notations

1.7.1 Representation of Numbers

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

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

  Example: 11110101b

- Hexadecimal numbers are represented 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 and the least significant byte (LSB) on the right.

  Example: F5h

- Decimal numbers are represented as is (without any trailing character).
  Example: 245

1.8 Abbreviations

The abbreviations as used in this document are defined in Table 1.

Table 1: Abbreviations

Abbreviation	Description
ALL_REQ	ALL NFC-A REQuest
CC	Capability Container
CE	Command End
CRC	Cyclic Redundancy Check
HR0	Header ROM byte 0
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
lsb	least significant bit
LSB	Least Significant Byte
msb	most significant bit
MSB	Most Significant Byte
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
NMN	NDEF Magic Number

Abbreviation	Description
RALL	Read All Command
READ	Read Command
RFU	Reserved for Future Use
RID	Read ID Command
RTD	Record Type Description
ROM	Read Only Memory
RWA	Read Write Access
SENS_REQ	Sense Request Command Type A
TLV	Tag, Length, Value
TMS	Tag Memory Size
UID	Unique IDentification
URI	Uniform Resource Identifier
VNo	Version Number
WRITE-E	Write with Erase Command
WRITE-NE	Write no Erase Command

1.9 Glossary

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

Mandatory NDEF Message TLV, first NDEF Message TLV

NDEF Message TLV detected by the NDEF detection procedure

NFC Forum Device

A device that supports the following Modus Operandi: Initiator, Target and Reader/Writer. It may also support Card Emulation Mode.

NFC Forum Reader/Writer Mode

In NFC Forum Reader/Writer Mode, the NFC Forum Device starts the Master/Slave Communication and sends commands to an NFC Forum Tag or contactless card. The communication for this mode is abbreviated as RW.

Type 1 Tag Platform

A legacy platform supporting a subset of a Technology (also called Technology Subset). Type 1 Tag Platform uses a particular subset of NFC – Type A technology (for more information, see [DIGITAL]).

2 Memory Structure and Management

2.1 General

The NFC Forum Type 1 Tag utilizes a simple memory model.

[RQ_T1T_MEM_001] There SHALL be two memory model mappings depending on the memory size of the tag:

- Static memory structure applies for a tag with physical memory size equal to 120 bytes,
- Dynamic memory model applies for a tag with physical memory size larger than 120 bytes.

[RQ_T1T_MEM_002] The memory SHALL be considered as being divided into blocks containing 8 bytes each.

Each block is numbered from 0 to 14 (Eh) for static memory structure or from 0 to k for dynamic memory structure. The number associated to a block is called the 'block number'.

The 8 bytes inside each block are numbered from 0 to 7, where byte 0 is the LSB and byte 7 is the MSB of the block.

For the complete tag address space, byte 0 of block 0 corresponds to ByteAddr = 0 as the LSB.

Byte 7 of block Eh for static memory structure or byte 7 of block k for dynamic memory structure indicates the MSB.

Unless otherwise stated, within this document the byte ordering when defining packets and messages follows the little-endian byte order.

The next two sections described in detail the two memory structures.

2.2 Static Memory Structure

2.2.1 Memory Map

The static memory map of the NFC Forum Type 1 Tag, with HR0 = 11h, is shown in Figure 1.

HR0	HR1
11h	xxh

EEPROM Memory Map										
Type	**Block No.**	**Byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7 (MSB)**	**Lockable**
UID	0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
Data	1	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Yes
Data	2	Data8	Data9	Data10	Data11	Data12	Data13	Data14	Data15	Yes
Data	3	Data16	Data17	Data18	Data19	Data20	Data21	Data22	Data23	Yes
Data	4	Data24	Data25	Data26	Data27	Data28	Data29	Data30	Data31	Yes
Data	5	Data32	Data33	Data34	Data35	Data36	Data37	Data38	Data39	Yes
Data	6	Data40	Data41	Data42	Data43	Data44	Data45	Data46	Data47	Yes
Data	7	Data48	Data49	Data50	Data51	Data52	Data53	Data54	Data55	Yes

EEPROM Memory Map										
Type	**Block No.**	**Byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7 (MSB)**	**Lockable**
Data	8	Data56	Data57	Data58	Data59	Data60	Data61	Data62	Data63	Yes
Data	9	Data64	Data65	Data66	Data67	Data68	Data69	Data70	Data71	Yes
Data	A	Data72	Data73	Data74	Data75	Data76	Data77	Data78	Data79	Yes
Data	B	Data80	Data81	Data82	Data83	Data84	Data85	Data86	Data87	Yes
Data	C	Data88	Data89	Data90	Data91	Data92	Data93	Data94	Data95	Yes
Reserved	D									
Lock/Reserved	E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	

Reserved for internal use
User Block Lock & Status
OTP bits

Figure 1: Static Memory Map of the Base NFC Forum Type 1 Tag

2.2.2 Header ROM Format

The NFC Forum Type 1 Tag includes two bytes of fixed header ROM called HR0 and HR1, as shown in Figure 1. These are not individually addressable by a Read command.

The contents are automatically included in the response packet to certain commands.

[RQ_T1T_MEM_003] HR0 Upper nibble = 0001b SHALL determine that it is a Type 1, NDEF capable tag.

[RQ_T1T_MEM_004] HR0 Lower nibble = 0001b SHALL determine static memory map.

[RQ_T1T_MEM_005] HR0 Lower nibble ≠ 0001b SHALL determine the dynamic memory map.

[RQ_T1T_MEM_006] HR1 = xxh is undefined and SHALL be ignored.

2.2.3 UID Format

Block 0 is reserved for the read-only Unique Identification (UID) number.

Byte 7 is reserved for future use.

Byte 6 is the manufacturer’s identification code.

Bytes 5, 4, 3, 2, 1, 0 are unique numbers.

2.2.4 Main Read/Write Memory Format

The 12 blocks numbered as 1h to Ch contain 96 bytes of general read/write memory.

Each block is individually lockable to become read-only by use of the relevant bits within the lock control bytes, as described in Section 2.2.6.

2.2.5 Block Dh

The block numbered as Dh is read-only and reserved for internal use.

2.2.6 Lock Control/Status Bytes

Bytes 0 and 1 of block Eh function as the lock controls for the various memory blocks.

They operate in a bit-wise one-time-programmable fashion.

Figure 2 shows the factory default settings for a Type 1 Tag with static memory map.

The individual locking bits can be set to 1b by using a suitable bit mask via a standard write command to the relevant bytes in block number Eh.

This process is irreversible: if one bit of the lock bytes is set to 1b, it cannot be changed back to 0b again.

LOCK-0 (Byte 0 of Block Eh)								LOCK-1 (Byte 1 of Block Eh)							
b7	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**	**b7**	**b6**	**B5**	**b4**	**b3**	**b2**	**b1**	**b0**
0b = BLOCK-7 Unlocked	0b = BLOCK-6 Unlocked	0b = BLOCK-5 Unlocked	0b = BLOCK-4 Unlocked	0b = BLOCK-3 Unlocked	0b = BLOCK-2 Unlocked	0b = BLOCK-1 Unlocked	1b = BLOCK-0 Locked	Not used	1b = BLOCK-E Locked	1b = BLOCK-D Locked	0b = BLOCK-C Unlocked	0b = BLOCK-B Unlocked	0b = BLOCK-A Unlocked	0b = BLOCK-9 Unlocked	0b = BLOCK-8 Unlocked

Figure 2: Lock Control/Status Bytes

2.2.7 OTP Bytes

Bytes 2 – 7 of block Eh are allocated as One Time Programmable (OTP) bits and are not defined for NFC Forum purposes.

2.3 Dynamic Memory Structure

2.3.1 Dynamic Memory Map

[RQ_T1T_MEM_007] The NFC Forum Type 1 Tag with dynamic memory map is indicated by HR0 = 1yh, where y ≠ 1. In this case, a capability container SHALL be included in the tag memory containing information about the physical memory size. See Section 5.1.4.

An example of the dynamic memory map representation of the NFC Forum Type 1 Tag with HR0 = 1yh, where y ≠ 1, is shown in Figure 3.

HR0	HR1
1yh	xxh

EEPROM Memory Map										
Type	**Block No.**	**Byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7 (MSB)**	**Lockable**
UID	0h	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
Data	1h	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Yes
Data	2h	Data8	Data9	Data10	Data11	Data12	Data13	Data14	Data15	Yes
Data	3h	Data16	Data17	Data18	Data19	Data20	Data21	Data22	Data23	Yes
Data	4h	Data24	Data25	Data26	Data27	Data28	Data29	Data30	Data31	Yes

EEPROM Memory Map										
Type	**Block No.**	**Byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7 (MSB)**	**Lockable**
Data	5h	Data32	Data33	Data34	Data35	Data36	Data37	Data38	Data39	Yes
Data	6h	Data40	Data41	Data42	Data43	Data44	Data45	Data46	Data47	Yes
Data	7h	Data48	Data49	Data50	Data51	Data52	Data53	Data54	Data55	Yes
Data	8h	Data56	Data57	Data58	Data59	Data60	Data61	Data62	Data63	Yes
Data	9h	Data64	Data65	Data66	Data67	Data68	Data69	Data70	Data71	Yes
Data	Ah	Data72	Data73	Data74	Data75	Data76	Data77	Data78	Data79	Yes
Data	Bh	Data80	Data81	Data82	Data83	Data84	Data85	Data86	Data87	Yes
Data	Ch	Data88	Data89	Data90	Data91	Data92	Data93	Data94	Data95	Yes
Reserved	Dh									
Lock/Reserved	Eh	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	
Lock/Reserved	Fh	LOCK-2	LOCK-3							
Data	10h	Data96	Data97	Data98	Data99	Data100	Data101	Data102	Data103	Yes
Data	11h	Data104	Data105	Data106	Data107	Data108	Data109	Data110	Data111	Yes
Data	12h	Data112	Data113	Data114	Data115	Data116	Data117	Data118	Data119	Yes
Data	13h	Data120	Data121	Data122	Data123	Data124	Data125	Data126	Data127	Yes
Data	14h	Data128	Data129	Data130	Data131	Data132	Data133	Data134	Data135	Yes
Data	15h	Data136	Data137	Data138	Data139	Data140	Data141	Data142	Data143	Yes
Data	.	Data144	Data145	Data146	Data147	Data148	Data149	Data150	Data151	Yes
Data	.	Data152	Data153	Data154	Data155	Data156	Data157	Data158	Data159	Yes
Data	.	Data160	Data161	Data162	Data163	Data164	Data165	Data166	Data167	Yes
Data	.	Data168	Data169	Data170	Data171	Data172	Data173	Data174	Data175	Yes
Data	.	Data176	Data177	Data178	Data179	Data180	Data181	Data182	Data183	Yes
Data	.	Data184	Data185	Data186	Data187	Data188	Data189	Data190	Data191	Yes
Lock/Reserved	.	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	
Lock/Reserved	k	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	LOCK-x	

Figure 3: Example Dynamic Memory Map of NFC Forum Type 1 Tag

In Figure 3, each memory block is numbered from 0 to k.

[RQ_T1T_MEM_008] Dynamic lock bytes and reserved bytes MAY be located at any byte address in between or at the end of the data area starting from block 0Fh.

[RQ_T1T_MEM_009] Compared to the static memory structure, the dynamic memory structure SHALL contain configuration information to describe the details of dynamic lock bits and to identify reserved memory areas in the data area using the Lock Control TLV and the Memory Control TLV.

The capability container and TLV data areas are not shown in Figure 3. For an example, refer to B.2.

2.3.2 Dynamic Memory Reserved Bytes

[RQ_T1T_MEM_010] These bytes belong to Reserved memory areas and SHALL be ignored / jumped over during read and write parsing operations of NFC Forum data.

[RQ_T1T_MEM_011] The location of Reserved bytes SHALL be identified by one or more Memory Control TLV blocks, as described in Section 2.4.

2.3.3 Dynamic Memory Lock Bytes

A tag with a dynamic memory structure contains two kinds of lock bytes:

- Static lock bytes as specified in Section 2.2.6.
- Dynamic memory lock bytes.

[RQ_T1T_MEM_012] The position of the dynamic memory lock bytes within the tag memory MAY change.

2.3.4 Dynamic Memory Area

The additional dynamic memory area is located from block Fh onward.

The available data area for the dynamic memory structure is contained from block 1 up to the last block of the memory, including the 96 bytes of the static memory structure and excluding static and dynamic lock bytes and reserved bytes.

Addressing of memory blocks is relative to and includes Block 0.

The available data area capacity in bytes is equal to:

$$8 \cdot (k - 3) - DynamicLockBytes - DynamicReservedBytes$$

This calculation includes the data area of the static memory structure equal to 96 bytes and discounts blocks 0, Dh, and Eh.

2.4 TLV Blocks

2.4.1 Format

A TLV block consists of one to three fields:

- **T** (tag field or T field) identifies the type of the TLV block and consists of a single byte encoding a number from 00h to FFh. The tag values 04h to FCh and FFh are reserved for future use by the NFC Forum.
- **L** (length field or L field) provides the size in bytes of the value field. It has two different formats composed of one or three bytes.

[RQ_T1T_MEM_013] The NFC Forum device SHALL understand both length field formats.

Figure 4 shows the two different length field structures.

[RQ_T1T_MEM_014] However, depending on the tag field value, the length field MAY not be present.

One byte format:

[RQ_T1T_MEM_015] The NFC Forum device SHALL use the one byte format to code the length of the value field between 00h and FEh bytes.

[RQ_T1T_MEM_016] The NFC Forum device SHALL interpret this byte as a cardinal if the value is between 00h and FEh.

[RQ_T1T_MEM_017] If it contains FFh, the NFC Forum device SHALL interpret the value as flag that specifies that the length field is composed of more than one byte.

Three consecutive bytes format:

[RQ_T1T_MEM_018] The NFC Forum device SHALL use this format to code the length of the value field between 00FFh and FFFEh bytes.

[RQ_T1T_MEM_019] The first byte is assumed to be a flag equal to FFh, indicating that two more bytes are present. The NFC Forum device SHALL interpret the two more bytes as a word.

[RQ_T1T_MEM_020] The NFC Forum device SHALL interpret this word as a cardinal if the value is between 00FFh and FFFEh.

The value FFFFh is reserved for future use (RFU).

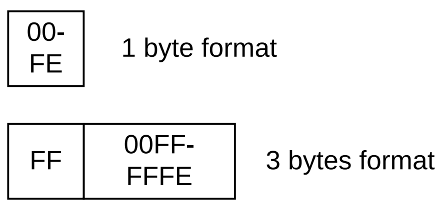

Figure 4: Length Field Formats

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

Table 2 lists the TLV blocks defined by this document that are described in the following sections.

Table 2: Defined TLV blocks

TLV Block Name	Tag Field Value	Short Description
NULL TLV	00h	[RQ_T1T_MEM_021] Might be used for padding of memory areas and the NFC Forum Device SHALL ignore this
Lock Control TLV	01h	Defines details of the lock bytes
Memory Control TLV	02h	Identifies reserved memory areas
NDEF Message TLV	03h	Contains the NDEF message
Proprietary TLV	FDh	Tag proprietary information
Terminator TLV	FEh	Last TLV block in the data area

2.4.2 Location

[RQ_T1T_MEM_022] The NFC Forum device SHALL recognize and interpret the TLV blocks in a specific order inside the data area according to the following rules:

- NDEF Message TLVs and Proprietary TLVs are present after all Lock Control TLVs and Memory Control TLVs.
- If present, the Terminator TLV is the last TLV block on the Type 1 Tag.

NULL TLV and Terminator TLV are the only TLV blocks that are 1 byte long (e.g., composed of only the Tag field. See the NOTE below).

[RQ_T1T_MEM_023] NFC Forum Devices SHALL ignore and jump over those TLV blocks that make use of reserved Tag field values.

[RQ_T1T_MEM_024] To jump over a TLV block with reserved Tag field values, the NFC Forum device SHALL read the length field to understand the length of the value field.

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

2.4.3 Lock Control TLV

[RQ_T1T_MEM_025] The Lock Control TLV can be present inside the Type 1 Tag. An NFC Forum Device SHALL be able to read and process it.

The Lock Control TLV provides control information about the lock areas where the dynamic lock bytes are located.

Each Lock Control TLV indicates a single lock area. More lock areas are indicated using more Lock Control TLV blocks. The encoding of the 3 TLV fields of the Lock Control TLV is as follows:

- **T** is equal to 01h.
- **L** is equal to 03h.
- **V** is composed of 3 bytes that uniquely identify the position and the size of the lock area and the number of bytes locked by each bit of the dynamic lock bytes. The 3 bytes are encoded as follows:

- Position, MSB. It codes the position inside the tag memory of the lock area. The position byte consists of 2 parts (to calculate the bytes address from the position byte, see the equation below):
- PagesAddr. Most significant nibble (4 bits), coded as number of pages (0h=0…Fh=15) and
- ByteOffset. Least significant nibble, coded as number of bytes (0h=0…Fh=15).
- Size. Middle byte, coded as number of bits (01h=1…FFh=255, 00h=256). It indicates the size in bits of the lock area (i.e., the number of dynamic lock bits). If the number of dynamic lock bits is not a multiple of 8, they are stored inside the dynamic lock bytes as explained in the description of the default setting of the dynamic lock bits.
- Page control, LSB. The page control provides general control information: the size in bytes of a page and the number of bytes that each dynamic lock bit is able to lock. Page control byte is split up into two nibbles of 4 bits each:
  - BytesPerPage: Least significant nibble, coded as 2^n (0h=RFU, 1h=1…Fh=15). It indicates the number of bytes per page.
  - BytesLockedPerLockBit: Most significant nibble, coded as 2^n (0h=RFU, 1h=1…Fh=15). It indicates the number of bytes that each dynamic lock bit is able to lock.

[RQ_T1T_MEM_026] The NFC Forum device SHALL calculate the byte address (ByteAddr) of the beginning of the lock area as follows:

$$ByteAddr = PageAddr \cdot 2^{BytesPerPage} + ByteOffset$$

The ByteAddr is calculated from the beginning of the overall memory of the tag (i.e., Byte 0 of Block 0 is indicated by ByteAddr equal to 0).

The ByteAddr is used to read and write the relative lock area using the appropriate tag access commands. The page definition has nothing to do with the block definition used by tag access commands.

An example of the BytesLockedPerLockBit is: If the memory area locked by a single dynamic lock bit is 8 bytes, then the BytesLockedPerLockBit is equal to 3 (i.e., $2^{BytesLockedPerLockBit}=2^3=8$ bytes).

NOTE The Lock Control TLV might be skipped if a Type 1 Tag is in READ-ONLY state. Lock Control TLV blocks can be replaced by Reserved Memory Control TLV indicating the same memory areas for Type 1 Tag in READ-ONLY state.

2.4.4 Reserved Memory Control TLV

[RQ_T1T_MEM_027] The Reserved Memory Control TLV can be present inside the Type 1 Tag and an NFC Forum Device SHALL be able to read and process it.

It provides control information about the location and the size of the reserved byte area.

[RQ_T1T_MEM_028] If the vendor delivers the Type 1 Tag in the READ-ONLY state, the NFC Forum device MAY use the Reserved Memory Control TLV to indicate control information for a mix of reserved and lock areas.

The encoding of the 3 TLV fields of the Reserved Memory Control TLV is:

- **T** is equal to 02h.
- **L** is equal to 03h.
- **V** is composed of 3 bytes that uniquely identify the position and the size of the reserved area. The 3 bytes are encoded as follows:
  - Position, MSB. It codes the position inside the tag of the reserved area. The Position byte consists of 2 parts (to calculate the bytes address from the position byte, see below):
  - PagesAddr. Most significant nibble, coded as number of pages (0h=0…Fh=15)
  - ByteOffset. Least significant nibble, coded as number of bytes (0h=0…Fh=15)
  - Size. Middle byte, coded as number of bytes (1h=1, FFh=255, 0h=256). It indicates the size in bytes of the reserved area.
  - Partial Page Control, LSB. The partial page control provides the size in bytes of a page. It is split up into two nibbles of 4 bits each:
    - Least significant nibble (BytesPerPage nibble), coded as 2^n (0h=RFU, 1h=1…Fh=15). It indicates the number of bytes per page.
    - Most significant nibble is RFU.

[RQ_T1T_MEM_029] The NFC Forum device SHALL calculate the byte address (ByteAddr) of each reserved area as follows:

$$ByteAddr = PageAddr \cdot 2^{BytesPerPage} + ByteOffset$$

The ByteAddr is calculated from the beginning of the overall memory of the tag (i.e., Byte 0 of Block 0 is indicated by ByteAddr equal to 0).

The page definition has nothing to do with the block definition used by tag access commands.

2.4.5 NDEF Message TLV

The NDEF Message TLV is always present inside the Type 1 Tag. It stores the NDEF message inside the Value field.

[RQ_T1T_MEM_030] The NFC Forum device SHALL be able to read and process the first (or mandatory) NDEF message.

Further NDEF Message TLV blocks can be present.

In case of multiple NDEF Message TLVs, the first (or mandatory) NDEF message TLV is the NDEF Message TLV starting at the smaller block number in the memory area.

When writing an NDEF Message TLV the first (or mandatory) NDEF message TLV is the one going to be written.

The encoding of the 3 TLV fields of the NDEF Message TLV is:

- **T** is equal to 03h.
- **L** is equal to the size in bytes of the stored NDEF message.
- **V** stores the NDEF message (see [NDEF]).

An empty NDEF Message TLV is defined as an NDEF Message TLV with L field equal to 00h and no V field (i.e., no NDEF message is present in the V field; see [NDEF]).

2.4.6 Proprietary TLV

The Proprietary TLV contains proprietary information.

[RQ_T1T_MEM_031] A Type 1 Tag contains zero, one, or more Proprietary TLV. The NFC Forum device MAY ignore the data contained in this TLV block.

The encoding of the 3 TLV fields of the Proprietary TLV is:

- **T** is equal to FDh.
- **L** is equal to the size in bytes of the proprietary data in the Value field.
- **V** contains any proprietary data.

2.4.7 NULL TLV

The NULL TLV can be used for padding of the data area.

[RQ_T1T_MEM_032] A Type 1 Tag contains zero, one, or more NULL TLV. The NFC Forum device SHALL ignore and jump over this TLV block.

The NULL TLV is composed of 1 byte tag field.

The encoding of the T field of the NULL TLV is:

- **T** is equal to 00h.
- **L** is not present.
- **V** is not present.

2.4.8 Terminator TLV

[RQ_T1T_MEM_033] The Terminator TLV can be present inside the Type 1 Tag and an NFC Forum device SHALL be able to read and process it.

The Terminator TLV is the last TLV block in the data area. Terminator TLV is composed of 1-byte tag field.

The encoding of the T field of the Terminator TLV is:

- **T** is equal to FEh.
- **L** is not present.
- **V** is not present.

3 Framing and Transmission Handling

3.1 Frame Formats

The frame formats used by the NFC Forum device to operate with the NFC Forum Type 1 Tag are given in [DIGITAL].

3.2 Transmission Handling

[RQ_T1T_FTH_001] The NFC Forum device SHALL implement the transmission handling of the commands/responses used for initialization, collision detection, device activation activities, and selection of the Type 1 Tag as defined in [DIGITAL].

The Type 1 Tag platform Commands and Responses are to be transmitted according to the rules defined for half-duplex protocols in Section 7 of [DIGITAL].

[RQ_T1T_FTH_001_1] The Type 1 Tag platform Command Set usage SHALL be compliant with the requirements defined for half-duplex protocols in Section 7 of [DIGITAL].

Command/responses for operation according to this specification are given in Section 4.

4 Command Set

4.1 State Diagram (Informative)

This section shows the state diagram of a Type 1 Tag. This section is informative only.

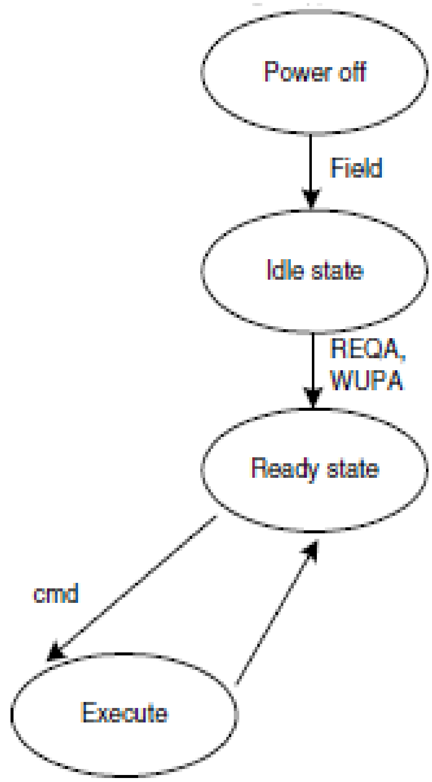

Figure 5: Type 1 Tag State Diagram

4.2 Tag Command and Response Set

4.2.1 Static Memory Model

[RQ_T1T_CSE_001] Commands used for the Type 1 Tag with the static memory map SHALL generate a response comprised of a number of bytes as shown in Table 3.

Table 3: Command-Response Byte Count (Static Memory Model)

Command	Command bytes	Response bytes
RALL	9	124
READ	9	4
WRITE-E	9	4
WRITE-NE	9	4

Details of the sequence of Command and Response bytes for the operation of the Type 1 Tag with the static memory map are shown in Table 4.

Table 4: Command-Response Summary (Static Memory Model)

Command-Response Summary Table																			
[RQ_T1T_CSE_002] Greyed-out frames are dummy frames – their data content SHALL be 00h																			
Command									Response										
RALL	00h	00h	UID 0	UID 1	UID 2	UID 3	CR C1	CR C2	HR0	HR1	UID 0	….	….	….	….	….	OTP 5	CR C1	CR C2
READ	AD D	00h	UID 0	UID 1	UID 2	UID 3	CR C1	CR C2	ADD	DAT	CR C1	CR C2							
WRIT E-E	AD D	DAT	UID 0	UID 1	UID 2	UID 3	CR C1	CR C2	ADD	DAT	CR C1	CR C2							
WRIT E-NE	AD D	DAT	UID 0	UID 1	UID 2	UID 3	CR C1	CR C2	ADD	DAT	CR C1	CR C2							

[RQ_T1T_CSE_003] A two-byte CRC, as defined in [DIGITAL], SHALL be appended to the end of commands and responses as shown in Table 4.

4.2.2 Dynamic Memory Model

The additional Command-Response bytes required for access to the dynamic memory model are shown in Table 5 and Table 6.

Table 5: Command-Response Byte Count (Dynamic Memory Model)

Command	Command bytes	Response bytes
RSEG	16	131
READ8	16	11
WRITE-E8	16	11
WRITE-NE8	16	11

Table 6: Command-Response Summary (Dynamic Memory Model)

Command															
RSEG	ADDS	00h	00h	00h	00h	00h	00h	00h	00h	UID0	UID1	UID2	UID3	CRC1	CRC2
READ8	ADD8	00h	00h	00h	00h	00h	00h	00h	00h	UID0	UID1	UID2	UID3	CRC1	CRC2
WRITE-E8	ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	UID0	UID1	UID2	UID3	CRC1	CRC2
WRITE-NE8	ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	UID0	UID1	UID2	UID3	CRC1	CRC2

Response												
ADDS	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	...	DAT 127	CRC1	CRC2
ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	CRC1	CRC2		
ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	CRC1	CRC2		
ADD8	DAT0	DAT1	DAT2	DAT3	DAT4	DAT5	DAT6	DAT7	CRC1	CRC2		

4.3 Command Format

4.3.1 Command List

Table 7: List of Commands (Static Memory Model)

Command		Command Code (7-bits) msb b7	b6	b5	b4	b3	b2	lsb b1	Comment (all commands are independent)
RALL	00h	0	0	0	0	0	0	0	Read All (all bytes)
READ	01h	0	0	0	0	0	0	1	Read (a single byte)
WRITE-E	53h	1	0	1	0	0	1	1	Write-with-erase (a single byte)
WRITE-NE	1Ah	0	0	1	1	0	1	0	Write-no-erase (a single byte)

[RQ_T1T_CSE_004] The Type 1 Tag with the static memory model SHALL ignore any command code bit patterns other than those commands shown in Table 7.

Table 8: List of Additional Commands (Dynamic Memory Model)

Command		Command Code (7-bits) msb b7	b6	b5	b4	b3	b2	lsb b1	Comment (all commands are independent)
RSEG	10h	0	0	1	0	0	0	0	Read Segment
READ8	02h	0	0	0	0	0	1	0	Read (eight bytes)
WRITE-E8	54h	1	0	1	0	1	0	0	Write-with-erase (eight bytes)
WRITE-NE8	1Bh	0	0	1	1	0	1	1	Write-no-erase (eight bytes)

The Type 1 Tag with the dynamic memory model will ignore any command code bit patterns other than those commands shown in Table 7 and Table 8.

4.3.2 Command-Response Format

[RQ_T1T_CSE_005] 8-bit operand and data frames, as defined in [DIGITAL], SHALL follow all commands listed in Table 7 and Table 8.

4.3.3 Address Operand

[RQ_T1T_CSE_006] The format of the address operand ‘ADD’ for the READ, WRITE-E, and WRITE-NE commands of the Type 1 Tag with static memory model SHALL be as shown in Table 9.

Table 9: Format of Address Operand ADD (Static Memory Structure)

Address operand ‘ADD’ Block = select one of blocks 0h – Eh Byte = select one of bytes 0 – 7							
msb					lsb		
b8	b7	b6	b5	b4	b3	b2	b1
0b	Static Block				Byte		

[RQ_T1T_CSE_007] The format of the address operand 'ADDS' for the RSEG command of the Type 1 Tag with the dynamic memory model SHALL be as shown in Table 10.

Table 10: Format of Address Operand ADDS (Dynamic Memory Model)

Address operand 'ADDS' Segment = select one of the Segments 0h – Fh							
msb						lsb	
b8	b7	b6	b5	b4	b3	b2	b1
Segment				0b	0b	0b	0b

[RQ_T1T_CSE_008] The format of the block address operand 'ADD8' for the READ8, WRITE-E8, and WRITE-NE8 commands of the Type 1 Tag with the dynamic memory model SHALL be as shown in Table 11.

Table 11: Format of Address Operand ADD8 (Dynamic Memory Model)

Address operand 'ADD8' Block = select one of the 8-byte blocks 00h – FFh							
msb						lsb	
b8	b7	b6	b5	b4	b3	b2	b1
Global Block							

4.3.4 CRC

[RQ_T1T_CSE_009] The CRC operation SHALL be as defined in [DIGITAL].

4.3.5 UID Echo

[RQ_T1T_CSE_010] The NFC Forum Device in NFC Forum Reader/Writer Mode SHALL execute a single Type 1 Tag selection feature as defined in [DIGITAL].

[RQ_T1T_CSE_011] This SHALL result in provision of a single identifier comprised of the lower four bytes of UID.

[RQ_T1T_CSE_012] All subsequent commands used to communicate with this Type 1 Tag for operation as described in this specification SHALL include these lower four bytes of UID as part of the proprietary Read and Write commands.

[RQ_T1T_CSE_013] If the four lower bytes of UID do not match, then the Type 1 Tag SHALL halt operation and remain in 'READY' state, as defined in [DIGITAL], waiting for the next valid command.

4.4 Command Details

4.4.1 Detailed Timing

The detailed command timing of a single bit period is defined in [DIGITAL].

4.4.2 Timing Definitions

The timing definitions for commands covered by this document are given in Table 12 below.

Table 12: Timing Definitions

Timing & Description Definitions (Used in the Command Sequence Descriptions)		
Name	**Description**	**Specification**
RRDD	Reader-Reader Data Delay As defined in [DIGITAL]	Minimum: [DIGITAL] • ≥ $RRDD_{T1T,MIN,BIT0}$ when last bit was 0 • ≥ $RRDD_{T1T,MIN,BIT1}$ when last bit was 1 Maximum: • None
DRD	Type1 tag Device Response Delay (Frame Delay Time) The time between the end of the last pause transmitted by the Reader/Writer and the first modulation edge within the start bit transmitted by the Type 1 Tag (taken from the FDT definition in [ISO/IEC_14443])	FDT timing from [ISO/IEC_14443], where n: • For RALL, READ, RSEG, and READ8: n=9 • For WRITE_E and WRITE_E8: n=554 • For WRITE_NE and WRITE_NE8: n=281 With tolerance for Digital & Analog elements of ± 6.5 clock cycles (13.56MHz).
RRD	Reader Response Delay Delay time Type 1 Tag to Reader/Writer (i.e., the time between the last modulation transmitted by the Type 1 Tag and the first gap transmitted by the Reader/Writer)	[ISO/IEC_14443]1172/fc ≅ 86 µs
CE	Command End	
UID-echo	The four least significant UID bytes from block 0 (LSB first)	

Table 13: FDT Timing Calculations

Timing Table			
Command	**n**	**FDT_{bit-1} = 128n + 84**	**FDT_{bit-0} = 128n + 20**
RALL, READ, RSEG, and READ8	9	1236/fc ≈ 91 µs	1172/fc ≈ 86 µs
WRITE-E and WRITE_E8	554	70996/fc ≈ 5236 µs	70932/fc ≈ 5231 µs
WRITE-NE and WRITE_NE8	281	36052/fc ≈ 2659 µs	35988/fc ≈ 2654 µs

NOTE The diagrams in the following sections do not show lead-in, start, and end of frame bits.

4.5 SENS_REQ and ALL_REQ

These commands are defined in [DIGITAL].

4.6 Read Identification (RID)

This command is defined in [DIGITAL].

4.7 Read All Blocks 0-Eh (RALL)

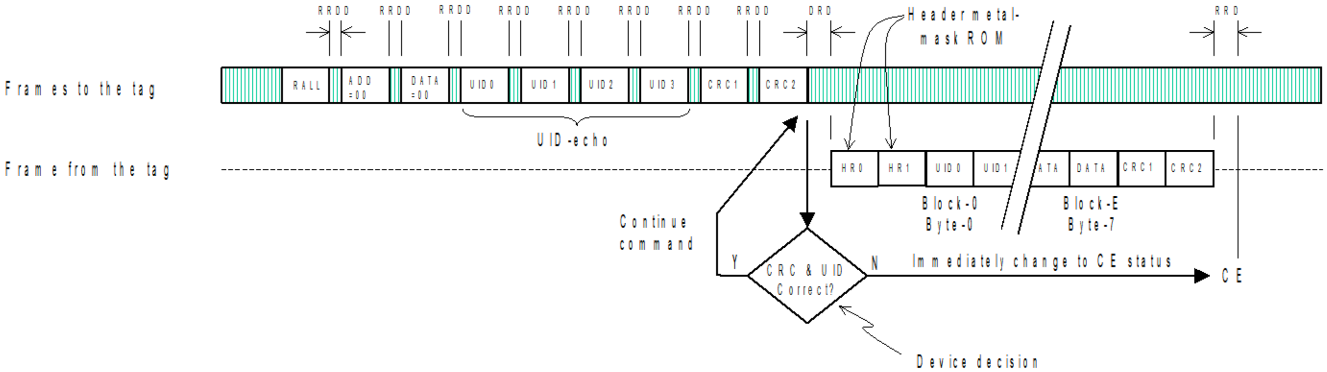

Figure 6: RALL Command/Response Diagram

The RALL command reads-out the two Header ROM bytes and all of the static memory blocks 0-Eh.

[RQ_T1T_CSE_014] The Command frame, then Address frame, Data-byte frame, UID-echo frames (with UID data received from previous RID command), and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

[RQ_T1T_CSE_015] However, the Address and Data-bytes SHALL be set to zero.

If the UID and CRC are valid, the HR0 and HR1 bytes followed by the contents of memory blocks 0-Eh and the frame CRC bytes will be sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

4.8 Read Byte (READ)

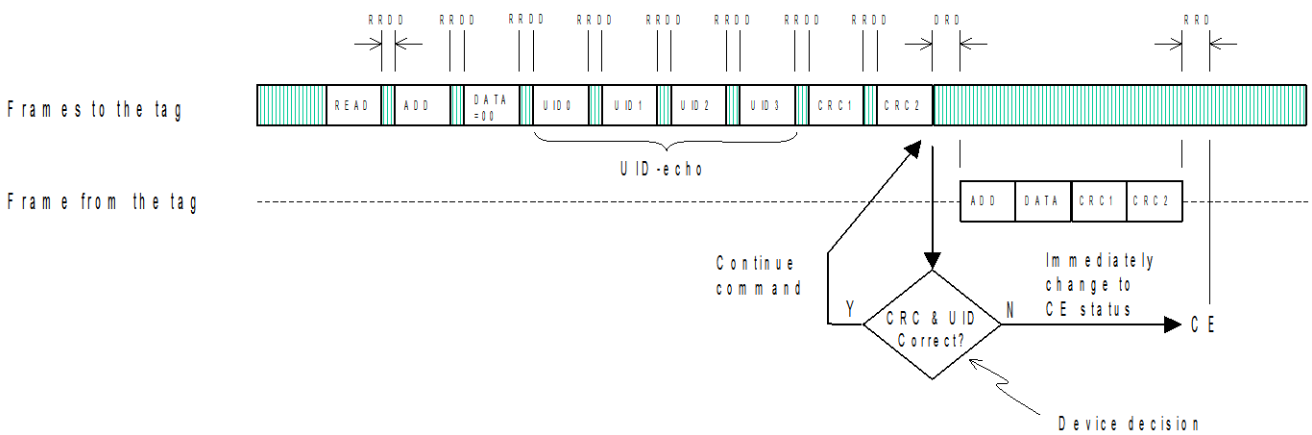

Figure 7: READ Command/Response Diagram

[RQ_T1T_CSE_016] The READ command relates to a single EEPROM memory byte within the static memory model area of blocks 0-Eh. The byte address (Block number and Byte number), as defined in Table 9, SHALL be sent with the command.

[RQ_T1T_CSE_017] The command frame, then Address frame, Data-byte frame, UID-echo frames (with UID data received from previous RID command) and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

[RQ_T1T_CSE_018] However, the Data-byte SHALL be set to zero.

If the CRC and UID are valid, the requested memory data byte is read from memory. The Address, followed by the read data byte and the frame CRC bytes, will be sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

4.9 Write-Erase Byte (WRITE-E)

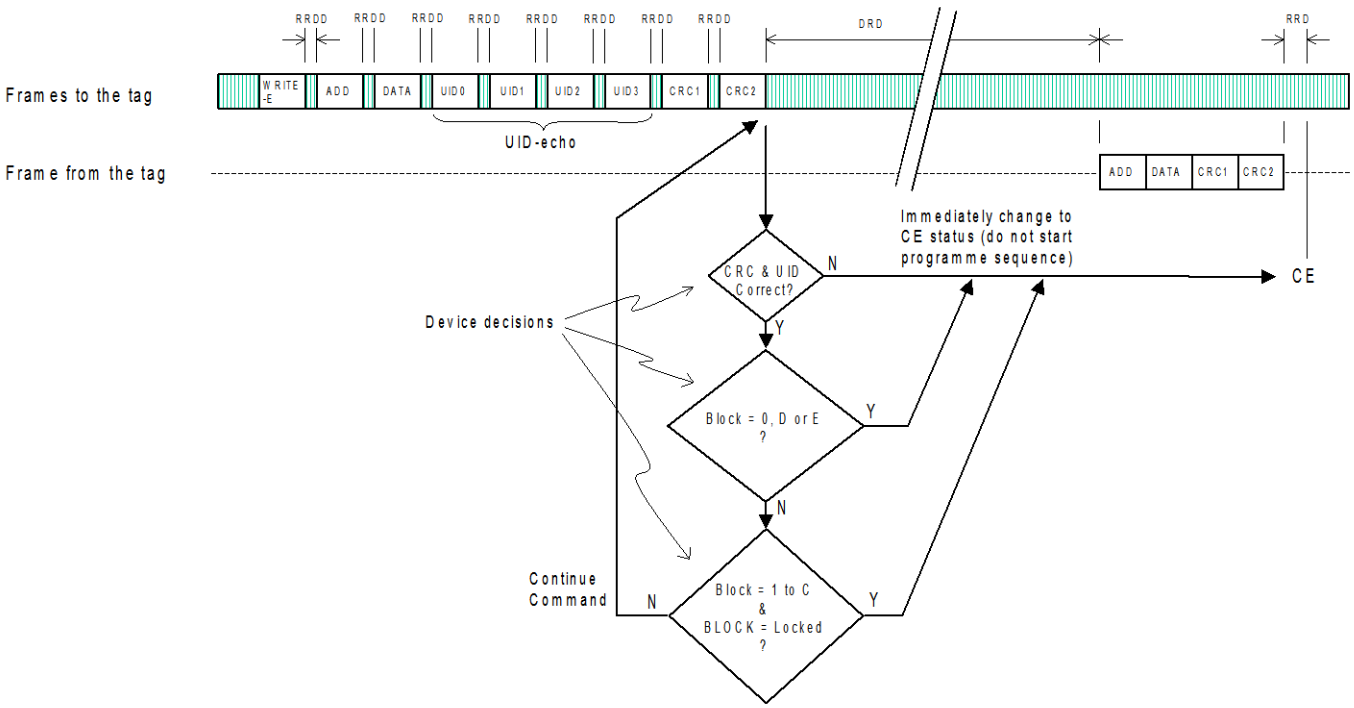

Figure 8: WRITE-E Command/Response Diagram

[RQ_T1T_CSE_019] The WRITE-E (Write–Erase) command relates to an individual memory byte within the static memory model area of blocks 0-Eh. The target byte address (Block number and Byte number), as defined in Table 9, SHALL be sent with the command.

This command performs the ‘normal’ erase-write cycle (i.e., it erases the target byte before it writes the new data).

If any of BLOCK-0 to BLOCK-D is locked, then WRITE-E is barred from those blocks. Additionally, WRITE-E is always barred from Blocks 0, D, and E because these are automatically in the locked condition.

[RQ_T1T_CSE_020] The Command frame, then Address frame, Data-byte frame, UID-echo frames (with UID data received from previous RID command), and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

If the UID and CRC are valid (and WRITE-E is not barred), the EE memory erase-write cycle is carried out. The byte is then read back from the EE memory. The address, followed by the data byte and the frame CRC bytes, are then sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

If WRITE-E is barred, the erase-write cycle is skipped—no write operation occurs and the tag will enter READY status waiting for a new command.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

4.10 Write-No-Erase Byte (WRITE-NE)

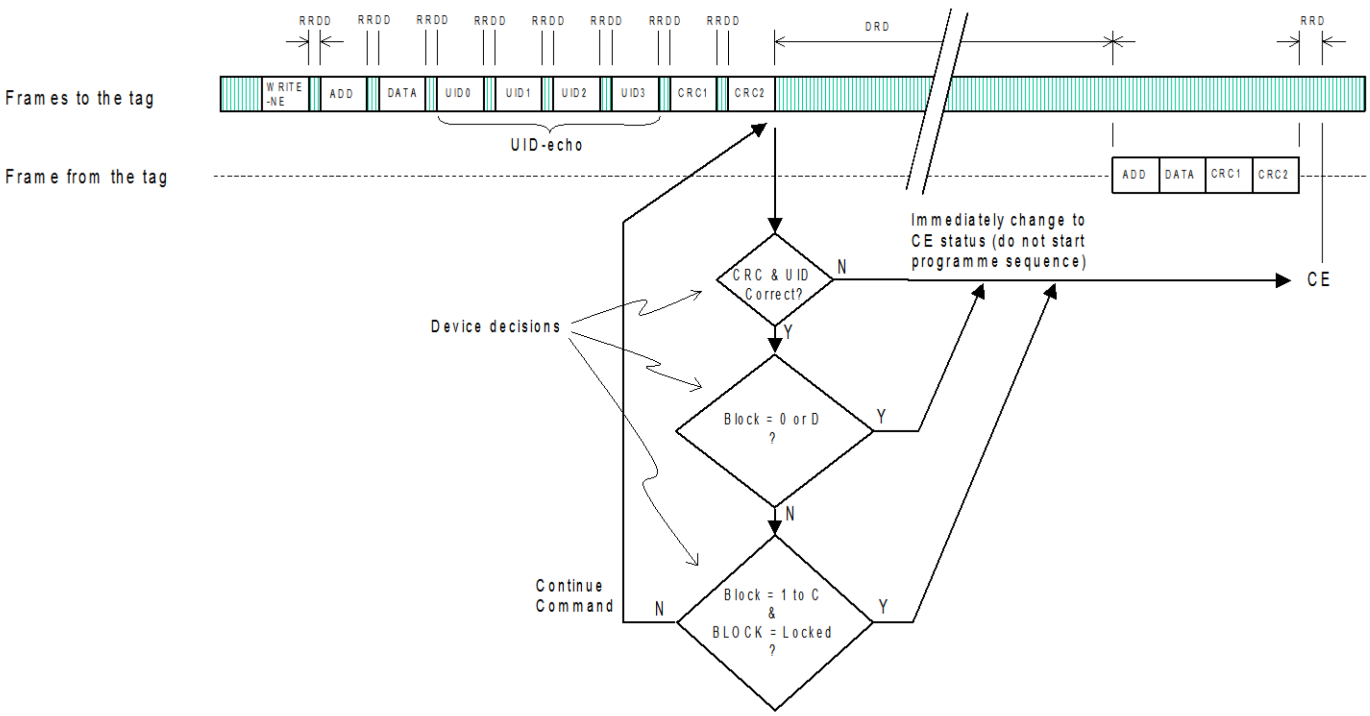

Figure 9: WRITE-NE Command/Response Diagram

[RQ_T1T_CSE_021] The WRITE-NE (Write-no-erase) command relates to an individual memory byte within the static memory model area of blocks 0-Eh. The target byte address (Block number and Byte number), as defined in Table 9, SHALL be sent with the command.

This command does not erase the target byte before writing the new data, and the execution time is approximately half that of the 'normal' write command (WRITE-E). Bits can be set but not reset (i.e., data bits previously set to a '1' cannot be reset to a '0').

The WRITE-NE command has three main purposes:

- Lock – to set the 'lock bit' for a block.
- OTP – to set One-Time-Programmable bits (bytes 2 – 7 of Block-E), where between one and eight OTP bits can be set with a single WRITE-NE command.
- A fast-write in order to reduce overall time to write data to memory blocks for the first time given that the original condition of memory is zero.

If any of BLOCK-1 to BLOCK-C are locked, then WRITE-E is barred from that block.

WRITE-NE is not barred from BLOCK-E to allow setting of lock and OTP bits.

[RQ_T1T_CSE_022] The Command frame, then Address frame, Data-byte frame, UID-echo frames (with UID data received from previous RID command), and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

If the UID and CRC are valid (and WRITE-NE is not barred), the EE memory write-no-erase cycle is carried out. The byte is then read back from the EE memory. The Address, followed by the Data byte and the frame CRC bytes, are then sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

If WRITE-NE is barred, the write-no-erase cycle is skipped—no write operation occurs and the tag will return to the “READY” state and wait for a new command.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

4.11 Locking

All twelve of the memory blocks 1h to Ch are separately lockable.

When a block’s ‘lock-bit’ is set to a 1, that block becomes irreversibly frozen as ‘read-only’.

The lock-bits are stored in the Bytes 0 and 1 of BLOCK-Eh.

[RQ_T1T_CSE_023] The WRITE-NE command with appropriate data pattern SHALL be used by the NFC Forum Device in NFC Forum Reader/Writer Mode to set individual lock-bits.

A single WRITE-NE command can be used to set between one and eight lock-bits.

4.12 Read Segment (RSEG)

The RSEG command reads-out a complete segment of memory. A segment consists of 16 blocks (i.e., 128 bytes of memory).

The command frames to the Type 1 Tag are similar to the RALL command, with the ADD replaced by ADDS (Address Segment) to select the required segment with the format as defined in Table 10.

The Command-Response summary is given in Table 6.

[RQ_T1T_CSE_024] The Command frame, then Address frame, eight data-byte frames, UID-echo frames (with UID data received from previous RID command), and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

[RQ_T1T_CSE_025] However, the eight data-bytes SHALL be set to zero.

If the UID and CRC are valid, then the ADDS, followed by the 128 byte contents of that segment and the frame CRC bytes, will be sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

4.13 Read 8 Bytes (READ8)

The READ8 command reads-out a block of memory.

The command frames to the Type 1 Tag are similar to the single byte READ command, with the ADD replaced by ADD8 (Address 8) to select the required block with the format as defined in Table 11.

The Command-Response summary is given in Table 6.

[RQ_T1T_CSE_026] The Command frame, then Address frame, eight data-byte frames, UID-echo frames (with UID data received from previous RID command), and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

[RQ_T1T_CSE_027] However, the eight data-bytes SHALL be set to zero.

If the UID and CRC are valid, then the ADD8, followed by the 8 data-bytes contents read from that block and the frame CRC bytes, will be sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

4.14 Write-Erase 8 Bytes (WRITE-E8)

The WRITE-E8 command writes with erase to a block of memory.

The command frames to the Type 1 Tag are similar to the single byte WRITE-E command, with the ADD replaced by ADD8 (Address 8) to select the required block with the format as defined in Table 11.

The Command-Response summary is given in Table 6.

[RQ_T1T_CSE_028] The Command frame, then Address frame, eight data-byte frames for the data to be written, UID-echo frames (with UID data received from previous RID command), and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

If the UID and CRC are valid, then the ADD8, followed by the 8 data-bytes contents just written to that block and the frame CRC bytes, will be sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

4.15 Write-No-Erase 8 Bytes (WRITE-NE8)

The WRITE-E8 command writes with no erase to a block of memory.

The command frames to the Type 1 Tag are similar to the single byte WRITE-NE command, with the ADD replaced by ADD8 (Address 8) to select the required block with the format as defined in Table 11.

The Command-Response summary is given in Table 6.

[RQ_T1T_CSE_029] The Command frame, then Address frame, eight data-byte frames for the data to be written, UID-echo frames (with UID data received from previous RID command), and CRC frames SHALL be sent by the NFC Forum Device in NFC Forum Reader/Writer Mode to the tag.

If the UID and CRC are valid, then the ADD8, followed by the 8 data-bytes contents just written to that block and the frame CRC bytes, will be sent back to the NFC Forum Device in NFC Forum Reader/Writer Mode.

As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state.

5 NDEF Detection and NDEF Access

5.1 NDEF Management

5.1.1 Identification as NFC Forum Type 1 Tag

The Type 1 Tag has a fixed Header ROM byte called HR0.

[RQ_T1T_NDA_001] To identify the Type 1 Tag, the high nibble of HR0 SHALL be equal to 0001b.

[RQ_T1T_NDA_002] When the NFC Forum Device operating in NFC Forum Reader/Writer Mode encounters a tag working to the proprietary protocol as used by the Type 1 Tag, then it SHALL use this HR0 value to identify if the tag is capable of carrying an NDEF message.

[RQ_T1T_NDA_003] The HR0 value SHALL be made available from the output of the collision detection, device activation, and single identifier activities of the Mode Switch as defined in [DIGITAL].

5.1.2 Write Permission

[RQ_T1T_NDA_004] An NFC Forum Device in NFC Forum Reader/Writer Mode SHALL not attempt to write to a tag unless confirmed by HR0 = 1xh. This pre-qualification SHALL be used to protect accidental writing and corruption of a non-NDEF application tag, such as a transit ticket based on an IC operating with the same proprietary protocol but with different HR0 value.

5.1.3 Confirmation of Presence of NDEF Message in Type 1 Tag

[RQ_T1T_NDA_005] An actual NDEF message MAY be present, although the qualification of the HR0 value identifies the tag encountered as a Type 1 Tag and therefore capable of carrying an NDEF message.

[RQ_T1T_NDA_030] To qualify that a valid NDEF message is actually present, a Capability Container (CC) SHALL be used.

[RQ_T1T_NDA_006] The CC SHALL contain NFC Forum management data.

[RQ_T1T_NDA_007] The CC SHALL be assigned to be in the first four bytes of memory block 1.

5.1.4 Capability Container

[RQ_T1T_NDA_008] The CC memory area SHALL not be used to store any application related data.

[RQ_T1T_NDA_009] Byte 0, when equal to E1h (NDEF Magic Number), SHALL indicate that NFC Forum defined NDEF Message data is stored in the data area.

[RQ_T1T_NDA_010] Byte 1 SHALL carry the Version Number (VNo) of this document as supported by the Type 1 Tag. The most significant nibble (the 4 most significant bits) SHALL indicate the major version number, and the least significant nibble (the 4 least significant bits) SHALL indicate the minor version number.

[RQ_T1T_NDA_012] Byte 2 SHALL indicate the physical tag memory size (TMS) of the Type 1 Tag as multipliers of (8 bytes) * (n+1). Examples:

- 120 bytes are indicated by 0Eh

- 256 bytes are indicated by 1Fh
- 2048 bytes are indicated by FFh

[RQ_T1T_NDA_013] Byte 3 SHALL indicate the read and write access (RWA) capability of the CC and data area of the Type 1 Tag.

[RQ_T1T_NDA_014] The most significant nibble (the 4 most significant bits) SHALL indicate the read access condition:

- The value 0h indicates read access granted without any security.
- Any other value is reserved for future use.

[RQ_T1T_NDA_015] The least significant nibble (the 4 least significant bits) SHALL indicate the write access condition:

- The value 0h indicates write access granted without any security.

[RQ_T1T_NDA_016] The value Fh SHALL indicate that no write access is granted.

- Any other value is reserved for future use.

Table 14 shows an example coding of the CC bytes. This example is related to a Type 1 Tag:

- With NFC Forum defined data (byte 0 = E1h)
- Supporting version 1.2 (major number 1h, minor number 2h) of the mapping document (byte 1 = 12h)
- With 120 bytes of memory size (byte 2 = 0Eh)
- With read and write access granted without any security (byte 3 = 00h)

Table 14: Example Coding of the CC Bytes of Block 1

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
NDEF “Magic Number”	Version Number	Tag Memory Size	Read/Write Access	Start of TLV and NDEF Message data area			
NMN	VNo	TMS	RWA	Octet 1	Octet 2	Octet 3	Octet 4
E1h	12h	0Eh	00h	-	-	-	-

5.2 Version Treatment

[RQ_T1T_NDA_017] Byte 1 of the CC SHALL contain the Version number (VNo) of this document as applied to the storage of NDEF Message data within Type 1 Tag.

This SHALL be indicated with two numbers: major number version and minor version number.

[RQ_T1T_NDA_018] The document version numbers applied to the Type 1 Tag (called T1VNo) and the one implemented in the NFC Forum device (called NFCDevVNo) SHALL be defined as shown in Table 15.

Table 15: Rules for Handling of the Version Number

No	Version Number Case	Handling
1	Major NFCDevVNo is equal to major T1VNo, and minor NFCDevVNo is bigger than or equal to minor T1VNo	The NFC Forum device SHALL access the Type 1 Tag and SHALL use all features of the applied mapping document to this Type 1 Tag.
2	If major NFCDevVNo is equal to major T1VNo, and minor NFCDevVNo is lower than minor T1VNo	Possibly not all features of the Type 1 Tag can be accessed. The NFC Forum device SHALL use all its features and SHALL access this Type 1 Tag.
3	If major NFCDevVNo is smaller than major T1VNo	Incompatible data format. The NFC Forum device cannot understand the Type 1 Tag data. The NFC Forum device SHALL reject this Type 1 Tag.
4	If major NFCDevVNo is bigger than major T1VNo	The NFC Forum device might implement the support for previous versions of this specification in addition to its main version. In case the NFC Forum device has the support from previous version, it SHALL access the Type 1 Tag. On the contrary, in case the NFC Forum device does not have the support from the previous version, it SHALL reject the Type 1 Tag.

NOTE Future versions of this specification have to define the allowed actions with an NFC Forum Type 1 Tag with a version number lower than the version number of the NFC Forum Device (i.e., whether it is allowed to upgrade the tag to the new version).

5.3 NDEF Storage

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

[RQ_T1T_NDA_019] The NDEF Message SHALL be stored inside the value field of the NDEF Message TLV in the data area of the Type 1 Tag as shown in Figure 10.

HR0	HR1
11h	xxh

EEPROM Memory Map									
Memory Block	**Byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7 (MSB)**	**Lockable**
0	**UID-0**	**UID-1**	**UID-2**	**UID-3**	**UID-4**	**UID-5**	**UID-6**		**Locked**
1	CC0 (NMN) =E1h	CC1 (Vno) =10h	CC2 (TMS) =0Eh	CC3 (RWA) =00h	NDEF Message TLV T=03h	NDEF Message TLV L=5Ah	Octet1	Octet2	Yes
2	Octet3	Octet4	Octet5	Octet6	Octet7	Octet8	Octet9	Octet10	Yes
3	Octet11	Octet12	Octet13	Octet14	Octet15	Octet16	Octet17	Octet18	Yes
4	Octet19	Octet20	Octet21	Octet22	Octet23	Octet24	Octet25	Octet26	Yes
5	Octet27	Octet28	Octet29	Octet30	Octet31	Octet32	Octet33	Octet34	Yes
6	Octet35	Octet36	Octet37	Octet38	Octet39	Octet40	Octet41	Octet42	Yes
7	Octet43	Octet44	Octet45	Octet46	Octet47	Octet48	Octet49	Octet50	Yes
8	Octet51	Octet52	Octet53	Octet54	Octet55	Octet56	Octet57	Octet58	Yes
9	Octet59	Octet60	Octet61	Octet62	Octet63	Octet64	Octet65	Octet66	Yes
A	Octet67	Octet68	Octet69	Octet70	Octet71	Octet72	Octet73	Octet74	Yes
B	Octet75	Octet76	Octet77	Octet78	Octet79	Octet80	Octet81	Octet82	Yes
C	Octet83	Octet84	Octet85	Octet86	Octet87	Octet88	Octet89	Octet90	Yes
D									Locked
E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	OTP

	Reserved for internal use
	User Block Lock & Status
	OTP bits

Figure 10: Location of NDEF Message

[RQ_T1T_NDA_020] The TLV and NDEF Message storage SHALL start from byte 4 of memory block 1 onward, up to the maximum capacity of the memory.

5.4 Life Cycle

5.4.1 General

An NFC Forum Type 1 Tag can be classified to exist in several states. The state is reflected by the contents of the tag as perceived by the NFC Forum Device in NFC Forum Reader/Writer Mode.

[RQ_T1T_NDA_031] Each state SHALL have its own set of valid operations depending on the current context of the NFC Forum Device. By context, it is meant whether the application is expecting to read from a tag or expecting to write NDEF message onto a tag.

The following sections specify the life-cycle relevant to the NFC Forum Type 1 Tag.

5.4.2 Overview of Life-Cycle States

[RQ_T1T_NDA_021] The NFC Forum Device in NFC Forum Reader/Writer Mode SHALL interpret the NFC Forum Type 1 Tag to be in one of the following states: INITIALIZED, READ/WRITE, and READ-ONLY.

[RQ_T1T_NDA_032] The state SHALL be reflected by the content of the tag.

The state transitions are only relevant for NFC Forum Devices, which are capable of writing Type 1 Tags.

The states represented in this section are not related to tag command states as shown in [DIGITAL].

5.4.3 INITIALIZED State

[RQ_T1T_NDA_022] A Type 1 Tag SHALL be considered to be in INITIALIZED state when not in the READ/WRITE or READ ONLY states.

[RQ_T1T_NDA_033] The Capability container as defined in Section 5.1.4 SHALL be present in the Initialized state.

[RQ_T1T_NDA_034] In the case of tags with memory size >120 Bytes (i.e., the Dynamic Memory structure), then the Lock Control and Memory Control TLV blocks as defined in Section 2.4 SHALL also be present in the Initialized state.

See the example in B.2.

5.4.4 READ/WRITE State

The tag is considered to already contain a valid NDEF message content. It is available for read and re-write access.

[RQ_T1T_NDA_023] This state SHALL provide the ability to read the NDEF message and also to modify it (i.e., completely overwrite the existing NDEF message with a new NDEF message).

[RQ_T1T_NDA_024] This state SHALL be reached via the INITIALIZED state.

[RQ_T1T_NDA_025] A Type 1 Tag SHALL be detected in READ/WRITE state when:

- CC has byte 0 equal to E1h, and
- CC has byte 1 with value according to version handling rules of Section 5.2, and
- CC has byte 3 equal to 00h (read/write access granted), and
- The data area contains an NDEF Message TLV, and
- The length field of the NDEF Message TLV is different from zero and equal to the actual length of the NDEF message in the value field

5.4.5 READ ONLY State

[RQ_T1T_NDA_026] This state SHALL be reached via the READ/WRITE or INITIALIZED state. In this configuration, the CC and the whole data area SHALL be set to read-only. The Type 1 Tag SHALL stay in READ-ONLY state for the remaining life cycle.

[RQ_T1T_NDA_027] The tag SHALL contain a valid NDEF message and SHALL have read-only access.

[RQ_T1T_NDA_028] A Type 1 Tag SHALL be detected in READ-ONLY state when:

- CC has byte 0 equal to E1h, and
- CC has byte 1 with value according to version handling rules of Section 5.2, and
- CC area has byte 3 equal to 0Fh (only read access granted), and
- The data area contains an NDEF Message TLV, and
- The length field of the NDEF Message TLV SHALL be different from zero and equal to the actual length of the NDEF message in the value field, and
- The lock bits related to the memory area of the CC and the NDEF message are in the locked state.

In this state, the memory area is set to read-only (i.e., locked). This process is irreversible because setting the appropriate lock bits to 1 performs the transition from READ/WRITE to READ ONLY.

5.4.6 Determination of Life Cycle State

[RQ_T1T_NDA_029] Before attempting a read or write operation, the NFC Forum Device in NFC Forum Reader/Writer Mode SHALL determine the state of a tag.

Generally, the most efficient approach is to read the complete tag contents and to buffer the data in its memory for analysis and parsing as follows:

- RID to capture HR0, to qualify it as an NDEF capable tag and to capture UID0-3
- RALL using UID0-3, to capture tag contents into local memory buffer
- Analyze CC and Lock Status bytes to determine the state

5.5 Rules for Life Cycle Operation

5.5.1 Detect NDEF on Tag

[RQ_T1T_NDA_035] Having determined the Life Cycle State of the tag as described in Section 5.4, the contents of the memory buffer SHALL be further analyzed to detect the presence of a valid NDEF message as follows:

1. If byte 0 of block 1 is equal to E1h and byte 1 describes the right version number (see Section 5.2) and the most significant nibble of byte 3 is equal to 0h, then go to item 2. Otherwise, no NDEF data is detected in the Type 1 Tag.

1. Parse the static data area contents already in memory and read the dynamic memory data areas if relevant.

5.5.2 Read NDEF Message

[RQ_T1T_NDA_036] The rules for how an NFC Forum Device in NFC Forum Reader/Writer Mode SHALL operate for the “read” context are as follows:

INITIALIZED

No read of NDEF message is possible.

READ/WRITE

[RQ_T1T_NDA_037] The memory contents SHALL be parsed to pass on the NDEF message to the application. If relevant, then the dynamic memory data areas SHALL also be read.

READ ONLY

[RQ_T1T_NDA_038] The memory contents SHALL be parsed to pass on the NDEF message to the application. If relevant, then the dynamic memory data areas SHALL also be read.

5.5.3 Write NDEF Message

[RQ_T1T_NDA_039] The rules for how an NFC Forum Device in NFC Forum Reader/Writer Mode SHALL operate for the “write” context are as follows:

- INITIALIZED

  The writing of a new NDEF message SHALL occur as follows:

  - Write NMN = 00h to indicate that no valid NDEF message is present during writing to allow error detection in the event that the tag is removed from the field prior to completion of operation.
  - Write VNo and RWA if required
  - Write NDEF Message TLV
  - Write NDEF Message data
  - Write NMN = E1h as the last byte to be written

- READ/WRITE

  The overwriting of a new NDEF message SHALL occur as follows:

  - Write NMN = 00h to invalidate an existing NDEF message during writing to allow error detection in the event that the tag is removed from the field prior to completion of operation.
  - Write VNo and RWA if required
  - Write NDEF Message TLV
  - Write NDEF Message data
  - Write NMN = E1h as the last byte to be written

- READ ONLY

  Write of NDEF message is not possible.

A. Exhibit A

Exhibit A is left blank intentionally.

B. Example NDEF Mappings

B.1 Example NDEF Mapping (Static Memory Model)

The contents of B.1 are only considered to be informative.

The following example NDEF message, which is copied from the Smartposter RTD draft specification document, has a total length of 23 bytes (=17h).

Table 16: Example Smartposter NDEF Message

Offset	Content	Length	Explanation
0	0xD1	1	NDEF header. TNF = 0x01 (Well Known Type). SR=1, MB=1, ME=1
1	0x02	1	Record name length (2 bytes)
2	0x12	1	Length of the Smart Poster data (18 bytes)
3	“Sp”	2	The record name
5	0xD1	1	NDEF header. TNF = 0x01, SR=1, MB=1, ME=1
6	0x01	1	Record name length (1 byte)
7	0x0E	1	The length of the URI payload (14 bytes)
8	“U”	1	Record type: “U”
9	0x01	1	Abbreviation: “http://www.”
10	“nfc-forum.org”	13	The URI itself.

After mapping onto the NFC Forum Type 1 Tag, the example NDEF message of Table 16 would look like the memory map of Figure 11 below.

HR0	HR1
11h	xxh

EEPROM Memory Map									
Block No.	Byte-0 (LSB)	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7 (MSB)	Lockable
0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
1	CC0 (NMN) =E1h	CC1 (Vno) =12h	CC2 (TMS) =0Eh	CC3 (RWA) =00h	NDEF Message TLV T=03h	NDEF Message TLV L=17h	D1	02	Yes
2	12	‘S’	‘p’	D1	01	0E	‘U’	01	Yes
3	‘n’	‘f’	‘c’	‘-‘	‘f’	‘o’	‘r’	‘u’	Yes
4	‘m’	‘.’	‘o’	‘r’	‘g’	Terminator TLV T=FEh			Yes
5									Yes
6									Yes
7									Yes
8									Yes
9									Yes
A									Yes
B									Yes
C									Yes
D									Locked
E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	

Figure 11: Memory Map of Example Smartposter NDEF Message

B.2 Example NDEF Mapping (Dynamic Memory Model)

The contents of B.2 are considered to be informative.

Figure 12 shows an example Type 1 Tag with a dynamic memory of 32 blocks (k=1Fh=31).

There is no NDEF Message present in this example.

HR0	HR1
12h	xxh

EEPROM Memory Map										
Type	**Block No.**	**Byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7 (MSB)**	**Lockable**
UID	0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
Data	1	CC0 (NMN) =00h	CC1 (VNo) =12h	CC2 (TMS) =1Fh	CC3 (RWA) =00h	Lock ControlTLV T=01h	Lock ControlTLV L=03h	Lock ControlTLV V0=F0h	Lock ControlTLV V1=10h	Yes
Data	2	Lock ControlTLV V2=33h	Memory Control TLV T=02h	Memory ControlTLV L=03h	Memory Control TLV V0=F2h	Memory ControlTLV V1=06h	Memory Control TLV V2=03h			Yes
Data	3									Yes
Data	4									Yes
Data	5									Yes
Data	6									Yes
Data	7									Yes
Data	8									Yes
Data	9									Yes
Data	A									Yes
Data	B									Yes
Data	C									Yes
Reserved	D									
Lock/Reserved	E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	
Lock/Reserved	F	LOCK-2	LOCK-3	Reserved -0	Reserved -1	Reserved -2	Reserved -3	Reserved -4	Reserved -5	
Data	10									Yes
Data	11									Yes
Data	12									Yes
Data	13									Yes
Data	14									Yes
Data	15									Yes
Data	16									Yes
Data	17									Yes
Data	18									Yes

EEPROM Memory Map										
Type	**Block No.**	**Byte-0 (LSB)**	**Byte-1**	**Byte-2**	**Byte-3**	**Byte-4**	**Byte-5**	**Byte-6**	**Byte-7 (MSB)**	**Lockable**
Data	19									Yes
Data	1A									Yes
Data	1B									Yes
Data	1C									Yes
Data	1D									Yes
Data	1E									Yes
Data	1F									Yes

Figure 12: Example Dynamic Memory Map

The tag is in INITIALIZED state and the main memory area is set as following:

- All lock bits are set to 0b: Lock0 = Lock1 = Lock2 = Lock3 = 00h
- The CC is set as follows:
  - CC0 = 00h to indicate that NDEF data is not present
  - CC1 = 12h to indicate support of the version 1.2 (major number 1h, minor number 2h) of this operation document
  - CC2 = 1Fh to indicate 32 blocks or 256 bytes of memory size
  - CC3 = 00h to indicated read and write access granted without any security
- The data area contains four TLV blocks in the following order:
  - Lock Control TLV:

    T = 01h

    L = 03h

    V = F0 10 33h indicates that each lock bit locks 1 page, each page is 8 bytes, and the lock area is 16 bits long, starting at the byte address 120 as calculated by the formula;

    ByteAddr = PageAddr*2$^{\mathrm{BytesPerPage}}$ **+ ByteOffset = 15 * 2**^{3} **+ 0 = 120** where:

    - Position = F0h contains PageAddr = Fh = 15 and ByteOffset = 0h
    - Size = 10h = 16 bits
    - PageControl = 33h contains BytesPerPage = 3h (2^3 = 8 bytes) and BytesLockedPerLockBit = 3h (2^3 = 8 bytes).

- Reserved Memory Control TLV:

  T = 02h

  L = 03h

  V = F20603h indicates that the reserved area is 6 bytes long starting at the byte address 122 as calculated by the formula;

  $$\mathbf{ByteAddr = PageAddr*2^{BytesPerPage} + ByteOffset = 15 * 2^{3} + 2 = 122}$$

  where:

  - Position = F2h contains PageAddr = Fh = 15 and ByteOffset = 2h
  - Size = 06h
  - PageControl = 03h contains BytesPerPage = 3h, as the least significant nibble.

    The most significant nibble is ignored

C. Revision History

The following table outlines the revision history of Type 1 Tag Operation.

Table 17: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
Type 1 Tag Operation	Version 1.0, July 2007	Final	None	
Type 1 Tag Operation	Version 1.1, January 2011	Final	Corrected informative appendix, added requirements numbering.	Version 1.0, July 2007
Type 1 Tag Operation	Version 1.1, April 2011	Final	Added CR.	Version 1.1, January 2011
Type 1 Tag Operation	Version 1.2, January 2014	Final	Added CR	Version 1.1, April 2011