RESTRICTIONS ON USE

This specification is copyright © 2005-2011 by the NFC Forum, and was made available pursuant to a license agreement entered into between the recipient (Licensee) and NFC Forum, Inc. (Licensor) and may be used only by Licensee, and in compliance with the terms of that license agreement (License). If you are not the Licensee, you may read this Specification, but are not authorized to implement or make any other use of this specification. However, you may obtain a copy of this Specification and implementation rights at the following page of Licensor's website: http://www.nfc-forum.org/specs/spec_license after entering into and agreeing to such license terms as Licensor is then requiring. On the date that this specification was downloaded by Licensee, the non-implementation terms of that license were as follows:

1. LICENSE GRANT.

Licensor hereby grants Licensee the right, without charge, to copy (for internal purposes only) and share this Specification with Licensee's members, employees and (to the extent related to Licensees use of this Specification) consultants. This license grant does not include the right to sublicense, modify or create derivative works based upon the Specification.

2. NO WARRANTIES.

THE SPECIFICATION IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY, COMPLETENESS AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL LICENSOR, ITS MEMBERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE SPECIFICATION.

3. THIRD PARTY RIGHTS.

Without limiting the generality of Section 2 above, LICENSOR ASSUMES NO RESPONSIBILITY TO COMPILE, CONFIRM, UPDATE OR MAKE PUBLIC ANY THIRD PARTY ASSERTIONS OF PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS THAT MIGHT NOW OR IN THE FUTURE BE INFRINGED BY AN IMPLEMENTATION OF THE SPECIFICATION IN ITS CURRENT, OR IN ANY FUTURE FORM. IF ANY SUCH RIGHTS ARE DESCRIBED ON THE SPECIFICATION, LICENSOR TAKES NO POSITION AS TO THE VALIDITY OR INVALIDITY OF SUCH ASSERTIONS, OR THAT ALL SUCH ASSERTIONS THAT HAVE OR MAY BE MADE ARE SO LISTED.

4. TERMINATION OF LICENSE.

In the event of a breach of this Agreement by Licensee or any of its employees or members, Licensor shall give Licensee written notice and an opportunity to cure. If the breach is not cured within thirty (30) days after written notice, or if the breach is of a nature that cannot be cured, then Licensor may immediately or thereafter terminate the licenses granted in this Agreement.

5. MISCELLANEOUS.

All notices required under this Agreement shall be in writing, and shall be deemed effective five days from deposit in the mails. Notices and correspondence to the NFC Forum address as it appears below. This Agreement shall be construed and interpreted under the internal laws of the United States and the Commonwealth of Massachusetts, without giving effect to its principles of conflict of law.

NFC Forum, Inc.
401 Edgewater Place, Suite 600
Wakefield, MA, USA 01880
Contents

1 Introduction ................................................................................................................................. 1
  1.1 Objectives ............................................................................................................................... 1
  1.2 Applicable Documents or References .................................................................................... 1
  1.3 Administration ......................................................................................................................... 2
  1.4 Name and Logo Usage ............................................................................................................. 2
  1.5 Intellectual Property ................................................................................................................ 3
  1.6 Special Word Usage ................................................................................................................ 3
  1.7 Convention and Notations ...................................................................................................... 3
    1.7.1 Representation of Numbers .............................................................................................. 3
  1.8 Abbreviations .......................................................................................................................... 3
  1.9 Glossary .................................................................................................................................... 5

2 Memory Structure and Management ......................................................................................... 6
  2.1 General ...................................................................................................................................... 6
  2.2 Static Memory Structure ......................................................................................................... 6
    2.2.1 Memory Map ..................................................................................................................... 6
    2.2.2 Header ROM Format ......................................................................................................... 7
    2.2.3 UID Format ....................................................................................................................... 7
    2.2.4 Main Read/Write Memory Format ...................................................................................... 7
    2.2.5 Block Dh ............................................................................................................................ 7
    2.2.6 Lock Control/Status Bytes ............................................................................................... 8
    2.2.7 OTP Bytes .......................................................................................................................... 8
  2.3 Dynamic Memory Structure .................................................................................................... 8
    2.3.1 Dynamic Memory Map ..................................................................................................... 8
    2.3.2 Dynamic Memory Reserved Bytes .................................................................................... 10
    2.3.3 Dynamic Memory Lock Bytes .......................................................................................... 10
    2.3.4 Dynamic Memory Area ................................................................................................... 10
  2.4 TLV Blocks ................................................................................................................................ 10
    2.4.1 Format .................................................................................................................................. 10
    2.4.2 Location .............................................................................................................................. 12
    2.4.3 Lock Control TLV ............................................................................................................. 12
    2.4.4 Reserved Memory Control TLV .......................................................................................... 13
    2.4.5 NDEF Message TLV ......................................................................................................... 14
    2.4.6 Proprietary TLV ................................................................................................................. 15
    2.4.7 NULL TLV ........................................................................................................................... 15
    2.4.8 Terminator TLV .................................................................................................................. 15

3 RF Interface .............................................................................................................................. 16

4 Framing and Transmission Handling ....................................................................................... 17
  4.1 Frame Formats ......................................................................................................................... 17
  4.2 Transmission Handling ............................................................................................................ 17

5 Command Set ........................................................................................................................... 18
  5.1 State Diagram ......................................................................................................................... 18
  5.2 Tag Command and Response Set ............................................................................................ 18
    5.2.1 Static Memory Model ....................................................................................................... 18
    5.2.2 Dynamic Memory Model ................................................................................................. 18
  5.3 Command Format .................................................................................................................... 19
    5.3.1 Command List ................................................................................................................... 19
5.3.2  Command-Response Format ............................................................... 20
5.3.3  Address Operand .............................................................................. 20
5.3.4  CRC ..................................................................................................... 21
5.3.5  UID Echo ........................................................................................... 21
5.4  Command Details .................................................................................. 21
5.4.1  Detailed Timing .................................................................................. 21
5.4.2  Timing Definitions ............................................................................. 21
5.5  SENS_REQ and ALL_REQ ................................................................. 22
5.6  Read Identification (RID) ..................................................................... 22
5.7  Read All Blocks 0-Eh (RALL) ............................................................... 23
5.8  Read Byte (READ) ................................................................................ 23
5.9  Write-Erase Byte (WRITE-E) ................................................................. 24
5.10  Write-No-Erase Byte (WRITE-NE) ..................................................... 25
5.11  Locking ............................................................................................... 26
5.12  Read Segment (RSEG) ....................................................................... 26
5.13  Read 8 Bytes (READ8) ....................................................................... 27
5.14  Write-Erase 8 Bytes (WRITE-E8) ....................................................... 27
5.15  Write-No-Erase 8 Bytes (WRITE-NE8) .............................................. 27

6  NDEF Detection and NDEF Access .................................................. 29
6.1  NDEF Management ............................................................................. 29
6.1.1  Identification as NFC Forum Type 1 Tag .......................................... 29
6.1.2  Write Permission ............................................................................. 29
6.1.3  Confirmation of Presence of NDEF Message in Type 1 Tag .......... 29
6.1.4  Capability Container ....................................................................... 29
6.2  Version Treatment .............................................................................. 30
6.3  NDEF Storage ..................................................................................... 32
6.4  Life Cycle ........................................................................................... 33
6.4.1  General ........................................................................................... 33
6.4.2  Overview of Life-Cycle States ........................................................... 33
6.4.3  INITIALIZE State .......................................................................... 33
6.4.4  READ/WRITE State ........................................................................ 33
6.4.5  READ ONLY State .......................................................................... 34
6.4.6  Determination of Life Cycle State .................................................... 34
6.5  Rules for Life Cycle Operation .......................................................... 35
6.5.1  Detect NDEF on tag ......................................................................... 35
6.5.2  Read NDEF Message ...................................................................... 35

A. Appendix A .......................................................................................... 37
A.1  Example NDEF Mapping (Static Memory Model) ................................ 37
A.2  Example NDEF Mapping (Dynamic Memory Model) ....................... 39

B. Revision History .................................................................................. 42
Figures

Figure 1: Static Memory Map of the Base NFC Forum Type 1 Tag.............................................. 7
Figure 2: Lock Control/Status Bytes............................................................................................ 8
Figure 3: Example Dynamic Memory Map of NFC Forum Type 1 Tag.......................................... 9
Figure 4: Length Field Formats.................................................................................................. 11
Figure 5: RALL Command/Response Diagram.......................................................................... 23
Figure 6: READ Command/Response Diagram.......................................................................... 23
Figure 7: WRITE-E Command/Response Diagram...................................................................... 24
Figure 8: WRITE-NE Command/Response Diagram................................................................. 25
Figure 9: Location of NDEF Message......................................................................................... 32
Figure 10: Memory Map of Example Smartposter NDEF Message............................................ 38
Figure 11: Example Dynamic Memory Map.............................................................................. 40

Tables

Table 1: Abbreviations................................................................................................................. 3
Table 2: Defined TLV blocks........................................................................................................ 12
Table 3: Command-Response Byte Count (Static Memory Model)........................................... 18
Table 4: Command-Response Summary (Static Memory Model)............................................... 18
Table 5: Command-Response Byte Count (Dynamic Memory Model)....................................... 19
Table 6: Command-Response Summary (Dynamic Memory Model)........................................ 19
Table 7: List of Commands (Static Memory Model)..................................................................... 19
Table 8: List of Additional Commands (Dynamic Memory Model)........................................ 20
Table 9: Format of Address Operand ADD (Static Memory Structure)....................................... 20
Table 10: Format of Address Operand ADDS (Dynamic Memory Model)................................. 20
Table 11: Format of Address Operand ADD8 (Dynamic Memory Model)................................. 21
Table 12: Timing Definitions....................................................................................................... 22
Table 13: FDT Timing Calculations............................................................................................ 22
Table 14: Example Coding of the CC Bytes of Block 1............................................................ 30
Table 15: Rules for Handling of the Version Number............................................................... 31
Table 16: Example Smartposter NDEF Message........................................................................ 37
Table 17: Revision History......................................................................................................... 42
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], [DIGITAL], and [ANALOG] 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.

1.2 Applicable Documents or References


[ANALOG] NFC Analog, In progress, NFC Forum


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

Includes:


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/


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, as outlined in the NFC Forum Rules of Procedure. These documents are available on the NFC Forum website.

1.6 Special Word Usage
The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

1.7 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>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL_REQ</td>
<td>ALL NFC-A REQuest</td>
</tr>
<tr>
<td>CC</td>
<td>Capability Container</td>
</tr>
<tr>
<td>CE</td>
<td>Command End</td>
</tr>
<tr>
<td><strong>Abbreviation</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>HR0</td>
<td>Header ROM byte 0</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commission</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>lsb</td>
<td>least significant bit</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Byte</td>
</tr>
<tr>
<td>msb</td>
<td>most significant bit</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Byte</td>
</tr>
<tr>
<td>NDEF</td>
<td>NFC Data Exchange Format</td>
</tr>
<tr>
<td>NFC</td>
<td>Near Field Communication</td>
</tr>
<tr>
<td>NMN</td>
<td>NDEF Magic Number</td>
</tr>
<tr>
<td>UID</td>
<td>Unique IDentifier</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>RALL</td>
<td>Read All Command</td>
</tr>
<tr>
<td>READ</td>
<td>Read Command</td>
</tr>
<tr>
<td>RFU</td>
<td>Reserved for Future Use</td>
</tr>
<tr>
<td>RID</td>
<td>Read ID Command</td>
</tr>
<tr>
<td>RTD</td>
<td>Record Type Description</td>
</tr>
<tr>
<td>ROM</td>
<td>Read Only Memory</td>
</tr>
<tr>
<td>RWA</td>
<td>Read Write Access</td>
</tr>
<tr>
<td>SENS_REQ</td>
<td>Sense Request Command Type A</td>
</tr>
<tr>
<td>TLV</td>
<td>Tag, Length, Value</td>
</tr>
<tr>
<td>TMS</td>
<td>Tag Memory Size</td>
</tr>
<tr>
<td>UID</td>
<td>Unique IDentification</td>
</tr>
<tr>
<td>VNo</td>
<td>Version Number</td>
</tr>
<tr>
<td>WRITE-E</td>
<td>Write with Erase Command</td>
</tr>
<tr>
<td>WRITE-NE</td>
<td>Write no Erase Command</td>
</tr>
</tbody>
</table>
1.9 Glossary

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

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.
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.

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

EEPROM Memory Map

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

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 6 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 Control/Status Bytes](image)

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 6.1.4.

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.

![Dynamic Memory Map](image)

**EEPROM Memory Map**

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

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

**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 might be located at any byte address in between or at the end of the data area starting from block 0Fh.
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 A.2.

### 2.3.2 Dynamic Memory Reserved Bytes

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

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.

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.

The NFC Forum device SHALL understand both length field formats.
Figure 4 shows the two different length field structures.

However, depending on the tag field value, the length field may not be present.

**One byte format:**

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

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

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

**Three consecutive bytes format:**

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

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>00-</th>
<th>FE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FF</td>
<td>00FF-</td>
</tr>
</tbody>
</table>
|   |     | FFFE | 3 bytes format

**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

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

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:
Memory Structure and Management

- 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:

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

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., $2^{\text{BytesLockedPerLockBit}} = 2^3 = 8$ 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:

\[
\text{ByteAddr} = \text{PageAddr} \cdot 2^{\text{BytesPerPage}} + \text{ByteOffset}
\]

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 (see [NDEF]).

[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.
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]).

A non-empty NDEF Message TLV can contain either empty or non-empty NDEF messages.

### 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 RF Interface

[RQ_T1T_RFI_001] The RF interface of the NFC Forum Device required for operation in NFC Forum NFC Forum Reader/Writer Mode is defined in [ANALOG].
4 Framing and Transmission Handling

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

4.2 Transmission Handling
[RQ_T1T_FTH_001] The transmission handling of the commands/responses used for initialization, collision detection, device activation activities, and selection of the Type 1 Tag by the NFC Forum device are defined in [DIGITAL].

Command/responses for operation according to this specification are given in Section 5.
5 Command Set

5.1 State Diagram
The basic state chart for operation of the NFC Forum Type 1 Tag is shown in [DIGITAL].

5.2 Tag Command and Response Set

5.2.1 Static Memory Model

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>
<thead>
<tr>
<th>Command</th>
<th>Command bytes</th>
<th>Response bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RALL</td>
<td>9</td>
<td>124</td>
</tr>
<tr>
<td>READ</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>WRITE-E</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>WRITE-NE</td>
<td>9</td>
<td>4</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Command-Response Summary Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RQ_T1T_CSE_002] Greyed-out frames are dummy frames – their data content SHALL be 00h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Response</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RALL</td>
<td>00h</td>
<td>HR0</td>
<td>CR C1</td>
</tr>
<tr>
<td></td>
<td>UID 0</td>
<td>HR1</td>
<td>CR C2</td>
</tr>
<tr>
<td></td>
<td>UID 1</td>
<td>UID 0</td>
<td>AD D</td>
</tr>
<tr>
<td></td>
<td>UID 2</td>
<td>UID 1</td>
<td>AD D</td>
</tr>
<tr>
<td></td>
<td>UID 3</td>
<td>UID 2</td>
<td>AD D</td>
</tr>
<tr>
<td></td>
<td>CR C1</td>
<td>UID 3</td>
<td>ADD</td>
</tr>
<tr>
<td></td>
<td>CR C2</td>
<td>CR C1</td>
<td>DAT</td>
</tr>
<tr>
<td>READ</td>
<td>AD D</td>
<td>CR C2</td>
<td>CR C2</td>
</tr>
<tr>
<td>WRITE-E</td>
<td>AD D</td>
<td>CR C2</td>
<td>CR C2</td>
</tr>
<tr>
<td>WRITE-NE</td>
<td>AD D</td>
<td>CR C2</td>
<td>CR C2</td>
</tr>
</tbody>
</table>

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

5.2.2 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)

<table>
<thead>
<tr>
<th>Command</th>
<th>Command bytes</th>
<th>Response bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSEG</td>
<td>16</td>
<td>131</td>
</tr>
<tr>
<td>READ8</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>WRITE-E8</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>WRITE-NE8</td>
<td>16</td>
<td>11</td>
</tr>
</tbody>
</table>

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

5.3 Command Format

5.3.1 Command List

Table 7: List of Commands (Static Memory Model)

<table>
<thead>
<tr>
<th>Command</th>
<th>Command Code (7-bits)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>msb</td>
<td>Isb</td>
</tr>
<tr>
<td>RALL</td>
<td>00h</td>
<td>b7</td>
</tr>
<tr>
<td>READ</td>
<td>01h</td>
<td>0</td>
</tr>
<tr>
<td>WRITE-E</td>
<td>53h</td>
<td>1</td>
</tr>
<tr>
<td>WRITE-NE</td>
<td>1Ah</td>
<td>0</td>
</tr>
</tbody>
</table>

[RQ_T1T_CSE_004] The Type 1 Tag with the static memory model will ignore any command code bit patterns other than those commands shown in Table 7.
Table 8: List of Additional Commands (Dynamic Memory Model)

<table>
<thead>
<tr>
<th>Command</th>
<th>Command Code (7-bits)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>msb b7 b6 b5 b4 b3 b2 b1</td>
<td>(all commands are independent)</td>
</tr>
<tr>
<td>RSEG</td>
<td>10h</td>
<td>Read Segment</td>
</tr>
<tr>
<td>READ8</td>
<td>02h</td>
<td>Read (eight bytes)</td>
</tr>
<tr>
<td>WRITE-E8</td>
<td>54h</td>
<td>Write-with-erase (eight bytes)</td>
</tr>
<tr>
<td>WRITE-NE8</td>
<td>1Bh</td>
<td>Write-no-erase (eight bytes)</td>
</tr>
</tbody>
</table>

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.

5.3.2 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.

5.3.3 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)

<table>
<thead>
<tr>
<th>Address operand ‘ADD’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block = select one of blocks 0h – Eh</td>
</tr>
<tr>
<td>Byte = select one of bytes 0 – 7</td>
</tr>
<tr>
<td>msb b8 b7 b6 b5 b4 b3 b2 b1</td>
</tr>
<tr>
<td>lsb</td>
</tr>
</tbody>
</table>

[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)

<table>
<thead>
<tr>
<th>Address operand ‘ADDS’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment = select one of the Segments 0h – Fh</td>
</tr>
<tr>
<td>msb b8 b7 b6 b5 b4 b3 b2 b1</td>
</tr>
<tr>
<td>lsb Segment 0b 0b 0b 0b</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>Address operand ‘ADD8’</th>
<th>Block = select one of the 8-byte blocks 00h – FFh</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>msb</td>
</tr>
<tr>
<td></td>
<td>b8</td>
</tr>
<tr>
<td></td>
<td>b6</td>
</tr>
<tr>
<td></td>
<td>b4</td>
</tr>
<tr>
<td></td>
<td>b2</td>
</tr>
</tbody>
</table>

Global Block

5.3.4 CRC

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

5.3.5 UID Echo

The NFC Forum Device in NFC Forum Reader/Writer Mode SHALL execute a single Type 1 Tag selection feature as defined in [DIGITAL]. This SHALL result in provision of a single identifier comprised of the lower four bytes of UID.

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.

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

5.4 Command Details

5.4.1 Detailed Timing

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

5.4.2 Timing Definitions

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRDD</td>
<td>Reader-Reader Data Delay</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The time between the end of the last pause of a frame</td>
<td></td>
</tr>
<tr>
<td></td>
<td>transmitted by the Reader/Writer and the first pause</td>
<td>Minimum:</td>
</tr>
<tr>
<td></td>
<td>of the next frame to be transmitted by the Reader/Writer</td>
<td>• ≥ 28 µs when last bit was 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ≥ 23 µs when last bit was 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• None</td>
</tr>
<tr>
<td>DRD</td>
<td>Type-1 Tag Device Response Delay</td>
<td>FDT timing from [ISO/IEC_14443], where n:</td>
</tr>
<tr>
<td></td>
<td>(Frame Delay Time)</td>
<td>• For RALL and READ: n=9</td>
</tr>
<tr>
<td></td>
<td>The time between the end of the last pause</td>
<td>• For WRITE_E: n=554</td>
</tr>
<tr>
<td></td>
<td>transmitted by the Reader/Writer and the first</td>
<td>• For WRITE_NE: n=281</td>
</tr>
<tr>
<td></td>
<td>modulation edge within the start bit transmitted by the Type-1 Tag</td>
<td>With tolerance for Digital &amp; Analogue elements of ± 6.5</td>
</tr>
<tr>
<td></td>
<td>(taken from the FDT definition in [ISO/IEC_14443])</td>
<td>clock cycles (13.56MHz).</td>
</tr>
<tr>
<td>RRD</td>
<td>Reader Response Delay</td>
<td>[ISO/IEC_14443]1172/fc ≈ 86 µs</td>
</tr>
<tr>
<td></td>
<td>Delay time Type-1 Tag to Reader/Writer (i.e., the time between the last</td>
<td></td>
</tr>
<tr>
<td></td>
<td>modulation transmitted by the Type-1 Tag and the first gap transmitted by</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the Reader/Writer)</td>
<td></td>
</tr>
<tr>
<td>CE</td>
<td>Command End</td>
<td></td>
</tr>
<tr>
<td>UID-echo</td>
<td>The four least significant UID bytes from block 0 (LSB first)</td>
<td></td>
</tr>
</tbody>
</table>

Table 13: FDT Timing Calculations

<table>
<thead>
<tr>
<th>Command</th>
<th>n</th>
<th>FDT_{\text{FALL}} = 128n + 84</th>
<th>FDT_{\text{MLE}} = 128n + 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>RALL and READ</td>
<td>9</td>
<td>1236/fc ≈ 91 µs</td>
<td>1172/fc ≈ 86 µs</td>
</tr>
<tr>
<td>WRITE-E</td>
<td>554</td>
<td>70996/fc ≈ 5236 µs</td>
<td>70932/fc ≈ 5231 µs</td>
</tr>
<tr>
<td>WRITE-NE</td>
<td>281</td>
<td>36052/fc ≈ 2659 µs</td>
<td>35988/fc ≈ 2654 µs</td>
</tr>
</tbody>
</table>

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

5.5 SENS_REQ and ALL_REQ
These commands are defined in [DIGITAL].

5.6 Read Identification (RID)
This command is defined in [DIGITAL].
5.7 Read All Blocks 0-Eh (RALL)

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.

5.8 Read Byte (READ)

[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.
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.

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.

5.9 Write-Erase Byte (WRITE-E)

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.

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.

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

Figure 8: 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.

5.11 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.

5.12 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.
5.13 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.

5.14 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.

5.15 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.
6 NDEF Detection and NDEF Access

6.1 NDEF Management

6.1.1 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].

6.1.2 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.

6.1.3 Confirmation of Presence of NDEF Message in Type 1 Tag

[RQ_T1T_NDA_005] Although the qualification of the HR0 value will have identified the tag encountered as a Type 1 Tag and therefore capable of carrying an NDEF message, there may or may not be an actual NDEF message present.

To further 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.

6.1.4 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_011] The VNo may change during the life time of an NFC Forum Type 1 Tag.

[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 indicates no write access granted at all.
• 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 1.0 (major number 1h, minor number 0h) of the mapping document (byte 1 = 10h)
• With 120 bytes of memory size (byte 2 = 0Eh)
• With read and write access granted without any security (byte 3 = 00h)

<table>
<thead>
<tr>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
<th>Byte 5</th>
<th>Byte 6</th>
<th>Byte 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMN</td>
<td>VNo</td>
<td>TMS</td>
<td>RWA</td>
<td>Octet 1</td>
<td>Octet 2</td>
<td>Octet 3</td>
<td>Octet 4</td>
</tr>
<tr>
<td>E1h</td>
<td>10h</td>
<td>0Eh</td>
<td>00h</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

6.2 Version Treatment

[RQ_T1T_NDA_017] Byte 1 of the CC contains 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 rules for the handling of the different document version numbers applied to the Type 1 Tag (called T1VNo) and the one implemented in the NFC Forum device (called NFCDevVNo) are explained in the cases shown in Table 15.

**Table 15: Rules for Handling of the Version Number**

<table>
<thead>
<tr>
<th>No</th>
<th>Version Number Case</th>
<th>Handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Major NFCDevVNo is equal to major T1VNo, and minor NFCDevVNo is bigger than or equal to minor T1VNo</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>If major NFCDevVNo is equal to major T1VNo, and minor NFCDevVNo is lower than minor T1VNo</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>If major NFCDevVNo is smaller than major T1VNo</td>
<td>Incompatible data format. The NFC Forum device cannot understand the Type 1 Tag data. The NFC Forum device SHALL reject this Type 1 Tag.</td>
</tr>
<tr>
<td>4</td>
<td>If major NFCDevVNo is bigger than major T1VNo</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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).
6.3 NDEF Storage

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

[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 9.

<table>
<thead>
<tr>
<th>HR0</th>
<th>HR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>11h</td>
<td>xxh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory Block</th>
<th>Byte-0 (LSB)</th>
<th>Byte-1</th>
<th>Byte-2</th>
<th>Byte-3</th>
<th>Byte-4</th>
<th>Byte-5</th>
<th>Byte-6</th>
<th>Byte-7 (MSB)</th>
<th>Lockable</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>UID-0</td>
<td>UID-1</td>
<td>UID-2</td>
<td>UID-3</td>
<td>UID-4</td>
<td>UID-5</td>
<td>25h</td>
<td></td>
<td>Locked</td>
</tr>
<tr>
<td>1</td>
<td>CC0 (N)</td>
<td>CC1 (V)</td>
<td>CC2 (T)</td>
<td>CC3 (R)</td>
<td>NDEF</td>
<td>NDEF</td>
<td>Octet1</td>
<td>Octet2</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>=E1h</td>
<td>=10h</td>
<td>=0Eh</td>
<td>=00h</td>
<td>Message</td>
<td>Message</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Octet3</td>
<td>Octet4</td>
<td>Octet5</td>
<td>Octet6</td>
<td>Octet7</td>
<td>Octet8</td>
<td>Octet9</td>
<td>Octet10</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Octet11</td>
<td>Octet12</td>
<td>Octet13</td>
<td>Octet14</td>
<td>Octet15</td>
<td>Octet16</td>
<td>Octet17</td>
<td>Octet18</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Octet19</td>
<td>Octet20</td>
<td>Octet21</td>
<td>Octet22</td>
<td>Octet23</td>
<td>Octet24</td>
<td>Octet25</td>
<td>Octet26</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Octet27</td>
<td>Octet28</td>
<td>Octet29</td>
<td>Octet30</td>
<td>Octet31</td>
<td>Octet32</td>
<td>Octet33</td>
<td>Octet34</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Octet35</td>
<td>Octet36</td>
<td>Octet37</td>
<td>Octet38</td>
<td>Octet39</td>
<td>Octet40</td>
<td>Octet41</td>
<td>Octet42</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>Octet43</td>
<td>Octet44</td>
<td>Octet45</td>
<td>Octet46</td>
<td>Octet47</td>
<td>Octet48</td>
<td>Octet49</td>
<td>Octet50</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Octet51</td>
<td>Octet52</td>
<td>Octet53</td>
<td>Octet54</td>
<td>Octet55</td>
<td>Octet56</td>
<td>Octet57</td>
<td>Octet58</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>Octet59</td>
<td>Octet60</td>
<td>Octet61</td>
<td>Octet62</td>
<td>Octet63</td>
<td>Octet64</td>
<td>Octet65</td>
<td>Octet66</td>
<td>Yes</td>
</tr>
<tr>
<td>A</td>
<td>Octet67</td>
<td>Octet68</td>
<td>Octet69</td>
<td>Octet70</td>
<td>Octet71</td>
<td>Octet72</td>
<td>Octet73</td>
<td>Octet74</td>
<td>Yes</td>
</tr>
<tr>
<td>B</td>
<td>Octet75</td>
<td>Octet76</td>
<td>Octet77</td>
<td>Octet78</td>
<td>Octet79</td>
<td>Octet80</td>
<td>Octet81</td>
<td>Octet82</td>
<td>Yes</td>
</tr>
<tr>
<td>C</td>
<td>Octet83</td>
<td>Octet84</td>
<td>Octet85</td>
<td>Octet86</td>
<td>Octet87</td>
<td>Octet88</td>
<td>Octet89</td>
<td>Octet90</td>
<td>Yes</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Locked</td>
</tr>
<tr>
<td>E</td>
<td>LOCK-0</td>
<td>LOCK-1</td>
<td>OTP-0</td>
<td>OTP-1</td>
<td>OTP-2</td>
<td>OTP-3</td>
<td>OTP-4</td>
<td>OTP-5</td>
<td>OTP</td>
</tr>
</tbody>
</table>

**Figure 9: 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.


6.4 Life Cycle

6.4.1 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.

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.

6.4.2 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.

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].

6.4.3 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.

The Capability container as defined in Section 6.1.4 shall be present in the Initialized state.

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 A.2.

6.4.4 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.
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 6.2, 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

### 6.4.5 READ ONLY State

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.

The tag is considered to contain a valid NDEF message and be available for read-only access. It cannot be deleted or overwritten with a new NDEF message.

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 6.2, 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.

### 6.4.6 Determination of Life Cycle State

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
6.5 Rules for Life Cycle Operation

6.5.1 Detect NDEF on tag
Having determined the Life Cycle State of the tag as described in Section 6.4, 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 6.2) 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.
2. Parse the static data area contents already in memory and read the dynamic memory data areas if relevant.

6.5.2 Read NDEF Message
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
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
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.

Write NDEF Message
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. Appendix A

A.1 Example NDEF Mapping (Static Memory Model)

The contents of A.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>
<thead>
<tr>
<th>Offset</th>
<th>Content</th>
<th>Length</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0xD1</td>
<td>1</td>
<td>NDEF header. TNF = 0x01 (Well Known Type). SR=1, MB=1, ME=1</td>
</tr>
<tr>
<td>1</td>
<td>0x02</td>
<td>1</td>
<td>Record name length (2 bytes)</td>
</tr>
<tr>
<td>2</td>
<td>0x12</td>
<td>1</td>
<td>Length of the Smart Poster data (18 bytes)</td>
</tr>
<tr>
<td>3</td>
<td>“Sp”</td>
<td>2</td>
<td>The record name</td>
</tr>
<tr>
<td>5</td>
<td>0xD1</td>
<td>1</td>
<td>NDEF header. TNF = 0x01, SR=1, MB=1, ME=1</td>
</tr>
<tr>
<td>6</td>
<td>0x01</td>
<td>1</td>
<td>Record name length (1 byte)</td>
</tr>
<tr>
<td>7</td>
<td>0x0E</td>
<td>1</td>
<td>The length of the URI payload (14 bytes)</td>
</tr>
<tr>
<td>8</td>
<td>“U”</td>
<td>1</td>
<td>Record type: “U”</td>
</tr>
<tr>
<td>9</td>
<td>0x01</td>
<td>1</td>
<td>Abbreviation: “<a href="http://www.%E2%80%9D">http://www.”</a></td>
</tr>
<tr>
<td>10</td>
<td>“nfc-forum.org”</td>
<td>13</td>
<td>The URI itself.</td>
</tr>
</tbody>
</table>

After mapping onto the NFC Forum Type 1 Tag, the example NDEF message of Table 16 would look like the memory map of Figure 10 below.
<table>
<thead>
<tr>
<th>Block No.</th>
<th>Byte-0 (LSB)</th>
<th>Byte-1</th>
<th>Byte-2</th>
<th>Byte-3</th>
<th>Byte-4</th>
<th>Byte-5</th>
<th>Byte-6</th>
<th>Byte-7 (MSB)</th>
<th>Lockable</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>UID-0</td>
<td>UID-1</td>
<td>UID-2</td>
<td>UID-3</td>
<td>UID-4</td>
<td>UID-5</td>
<td>UID-6</td>
<td></td>
<td>Locked</td>
</tr>
<tr>
<td>1</td>
<td>CC0 (NMN) =E1h</td>
<td>CC1 (Vno) =10h</td>
<td>CC2 (TMS) =0Eh</td>
<td>CC3 (RWA) =00h</td>
<td>NDEF Message TLV T=03h</td>
<td>NDEF Message TLV L=17h</td>
<td>D1</td>
<td>02</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>‘S’</td>
<td>‘p’</td>
<td>D1</td>
<td>01</td>
<td>0E</td>
<td>‘U’</td>
<td>01</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>‘n’</td>
<td>‘f’</td>
<td>‘c’</td>
<td>‘.’</td>
<td>‘f’</td>
<td>‘o’</td>
<td>‘r’</td>
<td>‘u’</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>‘m’</td>
<td>‘.’</td>
<td>‘o’</td>
<td>‘r’</td>
<td>‘g’</td>
<td>Terminator TLV T=FEh</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Locked</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>LOCK-0</td>
<td>LOCK-1</td>
<td>OTP-0</td>
<td>OTP-1</td>
<td>OTP-2</td>
<td>OTP-3</td>
<td>OTP-4</td>
<td>OTP-5</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10: Memory Map of Example Smartposter NDEF Message
A.2 Example NDEF Mapping (Dynamic Memory Model)

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

Figure 11 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.

<table>
<thead>
<tr>
<th>HR0</th>
<th>HR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>12h</td>
<td>xxh</td>
</tr>
</tbody>
</table>

### EEPROM Memory Map

<table>
<thead>
<tr>
<th>Type</th>
<th>Block No.</th>
<th>Byte-0 (LSB)</th>
<th>Byte-1</th>
<th>Byte-2</th>
<th>Byte-3</th>
<th>Byte-4</th>
<th>Byte-5</th>
<th>Byte-6</th>
<th>Byte-7 (MSB)</th>
<th>Lockable</th>
</tr>
</thead>
<tbody>
<tr>
<td>UID</td>
<td>0</td>
<td>UID-0</td>
<td>UID-1</td>
<td>UID-2</td>
<td>UID-3</td>
<td>UID-4</td>
<td>UID-5</td>
<td>UID-6</td>
<td></td>
<td>Locked</td>
</tr>
<tr>
<td>Data</td>
<td>1</td>
<td>CC0 (NMN) =00h</td>
<td>CC1 (VNo) =10h</td>
<td>CC2 (TMS) =1Fh</td>
<td>CC3 (RWA) =00h</td>
<td>Lock Control TLV T=01h</td>
<td>Lock Control TLV L=03h</td>
<td>Lock Control TLV V0=F0h</td>
<td>Lock Control TLV V1=10h</td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>2</td>
<td>Lock Control TLV V2=33h</td>
<td>Memory Control TLV T=02h</td>
<td>Memory Control TLV L=03h</td>
<td>Memory Control TLV V0=F2h</td>
<td>Memory Control TLV V1=06h</td>
<td>Memory Control TLV V2=03h</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Reserved</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lock/Reserved</td>
<td>E</td>
<td>LOCK-0</td>
<td>LOCK-1</td>
<td>OTP-0</td>
<td>OTP-1</td>
<td>OTP-2</td>
<td>OTP-3</td>
<td>OTP-4</td>
<td>OTP-5</td>
<td></td>
</tr>
<tr>
<td>Lock/Reserved</td>
<td>F</td>
<td>LOCK-2</td>
<td>LOCK-3</td>
<td>Reserved -0</td>
<td>Reserved -1</td>
<td>Reserved -2</td>
<td>Reserved -3</td>
<td>Reserved -4</td>
<td>Reserved -5</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Data</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
Figure 11: 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 = 10h to indicate support of the version 1.0 (major number 1h, minor number 0h) of this operation document
  - CC2 = 1Fh to indicate 32 blocks or 256 bytes of memory size
  - CC3 = 00h to indicate 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;
      \[
      \text{ByteAddr} = \text{PageAddr} \times 2^{\text{BytesPerPage}} + \text{ByteOffset} = 15 \times 2^3 + 0 = 120
      \]
      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;

  \[
  \text{ByteAddr} = \text{PageAddr} \times 2^{\text{BytesPerPage}} + \text{ByteOffset} = 15 \times 2^3 + 2 = 122
  \]

  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
B. Revision History

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

**Table 17: Revision History**

<table>
<thead>
<tr>
<th>Document Name</th>
<th>Revision and Release Date</th>
<th>Status</th>
<th>Change Notice</th>
<th>Supersedes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1 Tag Operation Specification</td>
<td>Version 1.0, July 2007</td>
<td>Final</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Type 1 Tag Operation Specification</td>
<td>Version 1.1, April 2011</td>
<td>Final</td>
<td>Added CR.</td>
<td>Version 1.1, January 2011</td>
</tr>
</tbody>
</table>