

AN11032

NXP Type ICODE Tag Operation

Rev. 1.1 — 1 July 2011
202211

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	NFC Forum, NFC data mapping, NFC Type ICODE Tag, NFC-enabled Tag, NDEF, ICODE SLI, ICODE SLI-S, ICODE SLI-L, ICODE SLIX, ICODE SLIX-S, ICODE SLIX-L, ISO-15693
Abstract	<p>The NFC Forum is a standardization consortium that was formed to advance the use of Near Field Communication technology by developing specifications, ensuring interoperability among devices and services, and educating the market about NFC technology.</p> <p>The NFC Forum has defined a data format called NDEF to store different kind of application data. NDEF structured data may be stored inside a contactless tag. The NFC Forum has also defined four different tag types that are able to stored NDEF data.</p> <p>This document extends the tag types of the NFC Forum describing how the Reader device (called NFC device) can store NDEF data inside ICODE family products.</p>



Revision history

Rev	Date	Description
1.1	20110701	Editorial updates
1.0	20110301	Graphics updated to latest standard, no content change
0.12	2011-01-21	Document identifier changed
0.11	2011-01-19	Editorial clarification "...to optimize the access time..." regarding clarification sentence about Multiple Page Read and Inventory Page Read in section 6.4.1 and 6.4.2
0.10	2011-01-19	Corrected Table 5, bits 5 and 4 to minor version number.
0.9	2011-01-19	Added internal review
0.8	2011-01-19	Added clarification sentence about Multiple Page Read and Inventory Page Read in section 6.4.1 and 6.4.2
0.7	2011-01-18	General Review
0.6	2011-01-18	Updated section 6.3.3 READ-ONLY State
0.5	2011-01-07	General updates
0.1	2010-07-29	First draft version

Contact information

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

The NFC technology (e.g. NFC IC PN544) allows to access standard ISO 15693 card products such as the ICODE family. A specification to store data for any kind of service or application is currently specified in the NFC Forum and it is called NFC Data Exchange Format (NDEF, see [NDEF]). To store NDEF formatted data (or also called NDEF data or NFC-defined data) inside a contactless tag product a mapping model is required.

The ICODE tag product (see [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L]) is a contactless card.

This application note describes:

- the mapping model to store one NDEF Message (or NDEF formatted data) inside ICODE platform,
- the command set and the life cycle of the ICODE platform to manage the NDEF Message, and
- how the Reader device (also called NFC device) can detect, read and write the NDEF Message in the ICODE platform

1.1 Applicable Documents

[ISOIEC 15693-2]	ISO/IEC 15693-2 Identification Cards - Contactless Integrated circuit(s) cards- Vicinity Cards- Part 2: Air interface and initialization
[ISOIEC 15693-3]	ISO/IEC15693-3 Identification Cards- Contactless Integrated circuit(s) cards- Vicinity Cards- Part 3: Anticollision and transmission protocol
[NDEF]	“NFC Data Exchange Format (NDEF)”, NFC Forum, Technical Specification, May 2006.
[RFC2119]	RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels.
[ICODE SLI]	“ICODE SLI SL2 ICS20”, Data Sheet
[ICODE SLI-L]	“ICODE SLI-L/ ICODE SLI-L HC SL2 ICS50/ SL2 ICS51”, Data Sheet, Document Identifier 1364**).
[ICODE SLI-S]	“ICODE SLI-S/ ICODE SLI-S HC SL2 ICS53/ SL2 ICS54”, Data Sheet, Document Identifier 1137**.
[ICODE SLIX]	“ICODE SLIX SL2S2002”, Data Sheet, Document Identifier 1780**.
[ICODE SLIX-L]	“ICODE SLIX-L SL2S5002”, Data Sheet, Document Identifier 1931**.
[ICODE SLIX-S]	“ICODE SLIX-S SL2S5302”, Data Sheet, Document Identifier 1921**.

**) ... document version number

1.2 Convention and notations

1.2.1 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, the Least Significant Byte (LSB) on the right.

Example: F5h, 1DEFh

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

Example: 245

1.3 Special Word Usage

The key words "SHALL", "SHALL NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are used to signify the requirements in this document.

SHALL, and REQUIRED have the same meaning. SHOULD and RECOMMENDED have the same meaning. MAY and OPTIONAL mean also the same. The key words are interpreted as described in [RFC2119].

1.4 Acronyms or Definitions or Glossary

Table 1. Terms and definitions

Term	Definition
label	ICODE family contactless label, see [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L]
CC	Capability Container, the CC stores control data for managing the NFC defined data inside the label
Label	A ICODE platform label
lsb	least significant bit
LSB	least significant byte
msb	most significant bit
MSB	most significant byte
NDEF	NFC Data Exchange Format, see [NDEF]
NDEF Message	Data packet structured as specified by the [NDEF] specification.
NDEF Message TLV	TLV block that contains an NDEF Message
NFC	Near Field Communication
NFC Forum	Standardization body, see: www.nxp.com/redirect/nfc-forum

Term	Definition
NFC-defined data	Data contained inside a ICODE family product defined by the NFC Forum e.g. NDEF Message
NFC device	NFC Reader device
NULL TLV	Single byte TLV block mainly used for padding.
PCD	Proximity Coupling Device according the ISO/IEC 15693. The term PCD describes a reader/writer for contactless cards
VICC	Proximity Card according to the ISO/IEC 15693. The ICODE platform
Proprietary TLV	TLV block that contains proprietary data
Reader device / Reader	Reader/writer for contactless cards. It may be a NFC device or a PCD device.
RF	Radio Frequency
RFU	Reserved for Future Use
Tag	A ICODE platform label
Terminator TLV	Last TLV block of the tag
TLV	Type Length Value block, data structure element to store different kind of data.
UID	Unique Identifier

2. Memory Structure and Management

ICODE family products are based on particular memory chips with a certain memory size and space for data. The following sections briefly describe the details of such memory chips and in particular their memory structure and management (for more details see [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L]).

[Table 2](#) gives an overview of the ICODE family products.

Table 2. Overview of ICODE products

	ICODE SLI / ICODE SLIX	ICODE SLI-S / ICODE SLIX-S	ICODE SLI-L / ICODE SLIX-L
Name	SL2 ICS2001 / SL2 S2002, SL2 S2102	SL2 ICS5301, SL2 ICS5401 / SL2 S5302, SL2 S5402	SL2 ICS5001, SL2 ICS5101 / SL2 S5002, SL2 S5102
User Data memory	112 bytes	160 bytes	32 bytes

The memory structure (or memory layout) is defined for each of the ICODE family products. The memory structures are divided into blocks. Each block is numbered starting from 0. The number associated to a block is called block number. Each block contains 4 bytes numbered from 0 to 3. For each block byte 0 is the MSB and byte 3 is the LSB. Byte 0 of block 0 indicates the MSB. Byte 3 of the last block indicates the LSB.

In this document the bit and byte order when defining packets and messages follows the big-endian byte order.

[Fig 1](#) outlines a generic memory layout of the NFC Type ICODE Tag based on the ICODE family products. The variable n depends on the User Data Memory. When the ICODE family products is configured as NFC Type ICODE Tag: the Block 0 contains the Capability Container (CC), and the Read/Write Data area from Block 1 to Block n is used to store the NFC-defined data i.e. NDEF message.

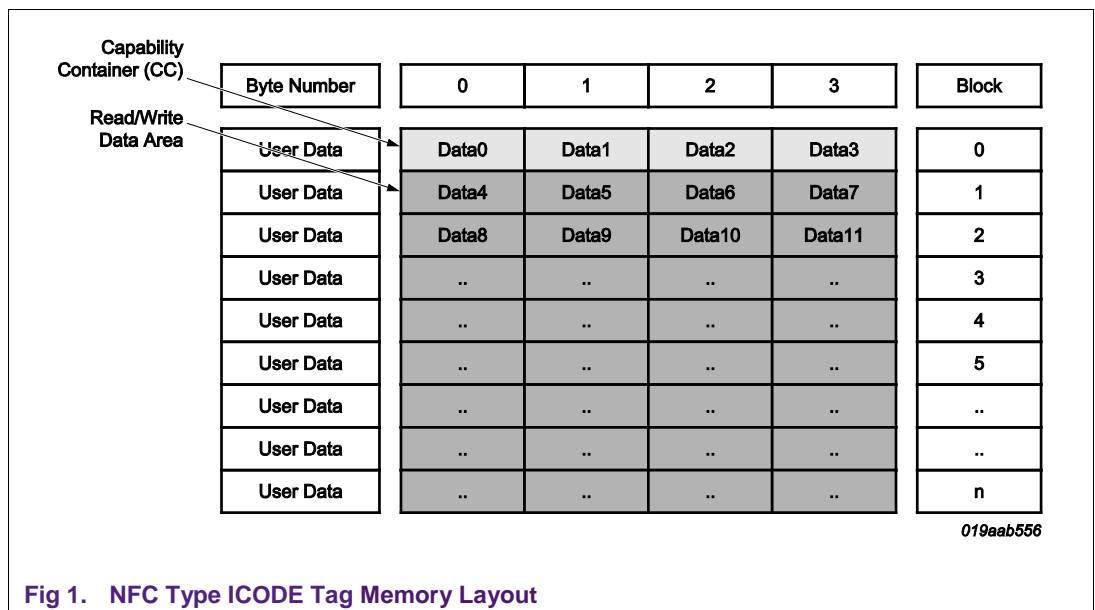


Fig 1. NFC Type ICODE Tag Memory Layout

2.1 TLV blocks

A TLV block consists of one to three fields:

- T** (tag field, or T field) SHALL identify the type of the TLV block (see [Table 3](#)) and SHALL consist of a single byte encoding a number from 00h to FFh. The tag values 01h, 02h, 04h to FCh and FFh are reserved for future use.
- L** (length field, or L field) SHALL provide the size in bytes of the value field. It has two different formats composed of one, or three bytes. The Reader device SHALL understand all two length field formats. [Fig 2](#) 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 one byte format SHALL code the length of the value field between 00h and FEh bytes. This byte SHALL be interpreted as a cardinal if the value is between 00h and FEh. If it contains FFh the value SHALL be interpreted as flag that specifies that the length field is composed of more than one byte.
 - Three consecutive bytes format: This format SHALL 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 length SHALL be interpreted as word. This word SHALL be interpreted as a cardinal if the value is between 00FFh and FFFEh. The value FFFFh is reserved for future use (RFU).

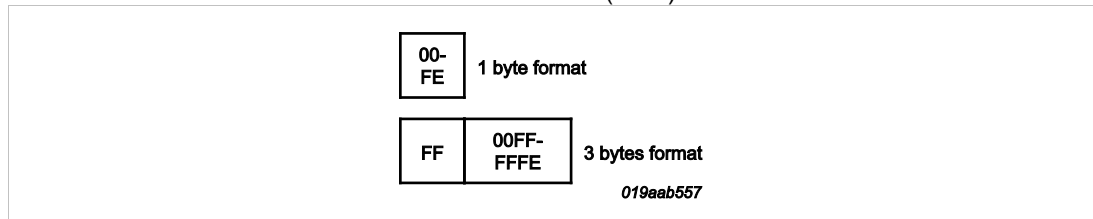


Fig 2. Length Field Formats

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

[Table 3](#) lists the TLV blocks specified by this document that are described in the following sections.

Table 3. Defined TLV blocks

TLV Block Name	Tag Field Value	Short Description
NULL TLV	00h	It might be used for padding of memory areas and the Reader device SHALL ignore this
NDEF Message TLV	03h	It contains the NDEF Message, see [NDEF]
Proprietary TLV	FDh	Tag proprietary information
Terminator TLV	FEh	Last TLV block in the data area

The Reader device SHALL write the TLV blocks in a specific order inside the data area following the rules below:

- The TLV blocks SHALL be written in order starting from byte 0 of block 0.
- If present the Terminator TLV is the last TLV block on the ICODE label.

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

Reader devices SHALL ignore and jump over those TLV blocks that make use of reserved tag field values (see above). To jump over a TLV block with reserved tag field values, the Reader device SHALL read the length field to understand the length of the value field.

Future definitions of TLV blocks composed of only the tag field are not backward compatible with this application note.

2.1.1 NDEF Message TLV

At least one NDEF Message TLV SHALL always be present inside the ICODE label. The NDEF Message TLV stores the NDEF Message inside the Value field (see [NDEF]). The Reader device SHALL be able to read and process the NDEF Message TLV found by the NDEF Detection Procedure (also called mandatory NDEF Message TLV or first NDEF Message TLV, see [section 6.4.1](#)); anyhow further NDEF Message TLV blocks MAY be present. Below the encoding of the 3 TLV fields of NDEF Message TLV is shown:

T SHALL be equal to 03h (see [Table 3](#)).

L SHALL be equal to the size in bytes of the stored NDEF Message.

V SHALL store the NDEF Message (see [NDEF]).

An empty NDEF Message TLV SHALL be 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 MAY contain either empty or non-empty NDEF Messages. The definition of empty NDEF Message is given in [chapter 7](#).

2.1.2 Proprietary TLV

The Proprietary TLV contains proprietary information. An ICODE label SHALL contain zero, one or more Proprietary TLV. The Reader device might ignore the data contained in this TLV block. Below the encoding of the 3 TLV fields of Proprietary TLV are shown:

T SHALL be equal to FDh (see [Table 3](#)).

L SHALL be equal to the size in bytes of the proprietary data in the Value field.

V SHALL contain any proprietary data.

2.1.3 NULL TLV

The NULL TLV MAY be used for padding of the data area. An ICODE label MAY contain zero, one or more NULL TLV. The Reader device SHALL ignore this TLV block. NULL TLV SHALL be composed of 1 byte tag field. Below the encoding of the tag field of the NULL TLV are shown:

T SHALL be equal to 00h (see [Table 3](#)).

L SHALL NOT be present.

V SHALL NOT be present.

2.1.4 Terminator TLV

The Terminator TLV MAY be present inside the ICODE label, and an Reader device SHALL be able to read/process it. The Terminator TLV is the last TLV block in the data memory area. Terminator TLV SHALL be composed of 1 byte tag field. Below the encoding of the tag field of the Terminator TLV are shown:

- T** SHALL be equal to FEh (see [Table 3](#)).
- L** SHALL NOT be present.
- V** SHALL NOT be present.

3. RF Interface

The ICODE family products comply with the RF interface as defined in the [ISO/IEC 15693-2].

The ISO/IEC 15693-2 terminology uses the term PCD for Proximity Coupling Device and VICC for Proximity Integrated Circuit(s) Card. In this application note the PCD is called reader device or NFC device, and the VICC is called ICODE label.

4. Framing/Transmission Handling

The framing and transmission handling of the ICODE label SHALL follow [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L].

5. Command Set

This chapter describes the command set as well as the overall state diagram of the ICODE label. It provides the basis to: detect and activate the ICODE label, detect the NFC data, get read and write access to the NFC data, and deactivate the ICODE label.

5.1 Tag Commands and Responses Set

The ICODE label accepts the following command set, sent by the Reader device. [Table 4](#) shows the command set (here called memory operations) of the ICODE label (see [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L, ISOIEC 15693-3]).

Table 4. Command Set / Memory Operations

Memory Operations	
Operation	Description
Inventory	After receiving the Inventory request, the ICODE label executes the anticollision sequence.
Stay quiet	After receiving the Stay quiet command, the ICODE label goes into quiet state.
Read single block	When receiving the Read single block command, the ICODE label responds with the requested block value.
Write single block	After receiving the Write single block command, the ICODE label writes the requested block with the data contained in the request and sends back the success of the operation in the response.
Lock block	After receiving the Lock block command, the ICODE label locks the addressed block.
Read multiple blocks	After receiving the Read multiple block command, the ICODE label responds with the blocks values. The support of this command is indicated in the Bit 0 of Byte 3 of the CC called MBREAD.
Inventory page read	When receiving the Inventory Page Read request the ICODE label performs the same as in the anti-collision sequence, with the difference that instead of the UID and the DSFID the requested content is re-transmitted from the ICODE label. The support of this command is indicated in the Bit 1 of Byte 3 of the CC called IPREAD. Inventory Page Read Command SHALL be set as following: - flag byte shall be set for 1 timeslot and option flag 0 - mask length is 0x40 hex - mask value are the 8 bytes of the UID
Reset to ready	After receiving a Reset to ready command, the ICODE label goes back to the Ready state.
Get multiple block security status	After receiving the Get multiple block security status command, the ICODE label responses back the block security status, i.e. the block lock information

Memory Operations	
Operation	Description
Get multiple block protection status	After receiving the Get multiple block protection status command, the ICODE label responses back the block protection status, i.e. the block lock information and the password protection information.

6. NDEF Detection and Access

This chapter describes how NFC-defined data (e.g. NDEF Message) SHALL be stored and accessed in the ICODE label.

The NDEF Message that this application note manages inside an ICODE label, is stored inside a NDEF Message TLV that is called mandatory NDEF Message TLV or first NDEF Message TLV. The mandatory NDEF Message TLV is also the NDEF Message TLV found by the NDEF Detection Procedure (see [section 6.4.1](#)).

6.1 NDEF Management

The NFC device reads the Capability Container (CC) to detect and access the NFC defined data inside the ICODE label. The CC contains NFC management data.

The CC is stored in the block 0 of the memory structure (see [chapter 2](#)). The CC bytes can be written using the Write single block or Write multiple blocks (see [section 5.1](#)).

The NFC device SHALL NOT use the CC to store any application related data.

Table 5. Block 0 – Capability Container Structure

MSB Byte 0	Byte 1								Byte 2	LSB Byte 3							
	Most Significant Nibble				Least Significant Nibble					Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Magic Number	Mapping Version Number				Access Conditions				Memory Size	RFU	RFU	RFU	RFU	RFU	RFU	IPR EAD	MBR EAD
E1h	Major Version Number		Minor Version Number		Read Access Condition		Write Access Condition		0b	0b	0b	0b	0b	0b			

The NFC device SHALL code the CC bytes of block 0 in the following way (see also [Table 5](#)):

- Byte 0 is equal to E1h (magic number) to indicate that NFC defined data is stored in the data area (see [chapter 2](#)).
- Byte 1 - Most Significant Nibble is the version number of this document supported by the ICODE label (see also [section 6.1.1](#)).
 - a. Bit 7-6 indicates the major version number, and
 - b. Bit 5-4 indicates the minor version number.

The version number of this specification has major version number equal to 01b and minor version number equal to 00b i.e. the version is 1.0 and Byte 1 - Most Significant Nibble is equal to 0100h.
- Byte 1 – Least Significant Nibble indicates the read and write access capability of the data area and CC area of the ICODE label.
 - a. Bit 3-2 indicate the read access condition:
 - i. The value 00b indicates read access granted without any security.

- ii. The values 01b, 10b and 11b are reserved for future use.
 - b. Bit 1-0 indicate the write access condition:
 - i. The value 00b indicates write access granted without any security.
 - ii. The value 11b indicates no write access granted at all.
 - iii. The value 01b is reserved for future use
 - iv. The value 10b is proprietary.
- Byte 2 indicates the memory size of the data area of the ICODE label including the Capability Container (Block 0). The value of byte 2 multiplied by 8 is equal to the data area size measured in bytes. The value 0 is RFU and SHALL NOT be used. For examples:
 - a. 32 bytes are indicated by byte 2 value equal to 04h
 - b. 112 bytes are indicated by byte 2 value equal to 0Eh
 - c. 160 bytes are indicated by byte 2 value equal to 14h
 - d. 2040 bytes are indicated by byte 2 value equal to FFh.
- Byte 3 indicates the following:
 - a. Bit 0, MBREAD equal to 1b indicates that the ICODE label supports the Read multiple blocks (see [section 5.1](#))
 - b. Bit 1, IPREAD equal to 1b indicates that the ICODE label supports the Inventory page read (see [section 5.1](#))
 - c. Bit 2 to Bit 7 are RFU and SHALL be set to 0b

6.1.1 Version Treating

The CC contains the mapping version number of the applied mapping model of the ICODE label. The mapping version number is indicated with two numbers: major version number and minor version number.

The handling of the different major and minor version numbers of the ICODE label (called MSVNo) and the one implemented in the Reader device (called NFCDevVNo) is explained in the 4 cases of [Table 6](#).

Table 6. Handling of the mapping document version numbers

No	Version Number Case	Handling
1	Major NFCDevVNo is equal to major MSVNo, and minor NFCDevVNo is bigger than or equal to minor MSVNo	The Reader device SHALL access the ICODE label and SHALL use all features of the applied mapping document to this ICODE label.
2	If major NFCDevVNo is equal to major MSVNo, and minor NFCDevVNo is lower than minor MSVNo	Possibly not all features of the ICODE label can be accessed. The Reader device SHALL use all its features and SHALL access this ICODE label.
3	If major NFCDevVNo is smaller than major MSVNo	Incompatible data format. The Reader device cannot understand the ICODE label data. The Reader device SHALL reject this ICODE label.

No	Version Number Case	Handling
4	If major NFCDevVNo is bigger than major MSVNo	The Reader device might implement the support for previous versions of this specification in addition to its main version. In case the Reader device has the support from previous version, it SHALL access the ICODE label. On the contrary, in case the Reader device has not the support from previous version, it SHALL reject the ICODE label.

6.1.2 ICODE SLI / ICODE SLIX Capability Container Coding

Table 7 shows an example of coding of the CC bytes. The example is related to an ICODE label:

- With NFC-defined data (byte 0 = E1h),
- Supporting the version 1.0 (major number 01b, minor number 00b) of the mapping document and with read and write access granted without any security (0000b) (byte 1 = 40h),
- With 112 bytes of the data area size (byte 2 = 0Eh), and
- With the support of the Read multiple blocks (byte 3 = 01h).

Table 7. Example of coding of the ICODE SLI / ICODE SLIX CC bytes of block 0

MSB Byte 0	Byte 1	Byte 2	LSB Byte 3
E1h	40h	0Eh	01h

6.1.3 ICODE SLI-S and ICODE SLIX-S Capability Container Coding

Table 8 shows an example of coding of the CC bytes in case of ICODE SLI-S and ICODE SLIX-S label:

- With NFC defined data (byte 0 = E1h),
- Supporting the version 1.0 (major number 01b, minor number 00b) of the mapping document and with read and write access granted without any security (0000b) (byte 1 = 40h),
- With 160 bytes of the data area size (byte 2 = 14h), and
- With the support of the Inventory page read (byte 3 = 02h).

Table 8. Example of coding of the ICODE SLI-S and ICODE SLIX-S CC bytes of block 0

MSB Byte 0	Byte 1	Byte 2	LSB Byte 3
E1h	40h	14h	02h

6.1.4 ICODE SLI-L and ICODE SLIX-L Capability Container Coding

[Table 9](#) shows an example of coding of the CC bytes. The example is related to an ICODE label:

- With NFC-defined data (byte 0 = E1h),
- Supporting the version 1.0 (major number 01b, minor number 00b) of the mapping document and with read and write access granted without any security (0000b) (byte 1 = 40h),
- With 32 bytes of the data area size (byte 2 = 04h), and
- With the support of the Inventory page read (byte 3 = 02h).

Table 9. Example of coding of the ICODE SLI-L and ICODE SLIX-L CC bytes of block 0

MSB Byte 0	Byte 1	Byte 2	LSB Byte 3
E1h	40h	04h	02h

6.2 NDEF Storage

The data format of the NDEF Message is defined in [NDEF]. The NDEF Message SHALL be stored inside the value field of the NDEF Message TLV (see [section 2.1.1](#)).

6.3 Life Cycle

A Reader device MAY detect an ICODE label in different states. The state is reflected by the content of the ICODE label. Every state has its own valid operations.

The state transitions are only relevant for Reader devices, which are capable of writing ICODE label.

The ICODE label MAY be in one of the following states: INITIALISED, READ/WRITE or READ-ONLY.

If the ICODE label is not in a valid state according to this Application Note, the Reader device SHALL ignore the ICODE label and its data. The reasons might be:

- Wrongly configured CC area.
- Not allowed NDEF Read Procedure (see [section 6.4.2](#)), if ICODE label is in READ/WRITE or READ-ONLY state.
- Not allowed NDEF Write Procedure (see [section 6.4.3](#)), if ICODE label is in INITIALISED or READ/WRITE state.
- Mismatch between overall TLV blocks length and actual length of the data area.
- Invalid TLV block.

6.3.1 INITIALISED State

An ICODE label SHALL be detected in INITIALISED state when:

- the CC is set as described in [section 6.1](#), in particular with bit 0-1 of Byte 1 equal to 00b and bit 2-3 of Byte 1 equal to 00b (read and write access granted),
- the User Data Memory contains one NDEF Message TLV (the mandatory one), and

- the length field of the mandatory NDEF Message TLV is equal to 00h.

In INITIALISED state the NFC device MAY modify the content of the mandatory NDEF Message TLV writing an NDEF Message in it. The [Annex B in chapter 8](#), the [Annex C in chapter 9](#) and the [Annex D in chapter 10](#) show 3 examples of respectively ICODE SLI / ICODE SLIX, ICODE SLI-S / ICODE SLIX-S and ICODE SLI-L / ICODE SLIX-L in INITIALISED state.

6.3.2 READ/WRITE State

An ICODE label SHALL be detected in READ/WRITE state when:

- the CC is set as described in [section 6.1](#), in particular with bit 0-1 of Byte 1 equal to 00b and bit 2-3 of Byte 1 equal to 00b (read and write access granted),
- the mandatory NDEF Message TLV is present in the User Data Memory, and
- the length field of the mandatory NDEF Message TLV is different from zero.

The READ/WRITE state SHALL be reached via the INITIALISED state. In this state the NFC device MAY modify the content of the mandatory NDEF Message TLV writing an NDEF Message in it.

6.3.3 READ-ONLY State

An ICODE label SHALL be detected in READ-ONLY state when:

- the CC is set as described in [section 6.1](#), in particular with bit 0-1 of Byte 1 is different from 00b and bit 2-3 of Byte 1 equal to 00b (no write access is granted, only read access is granted),
- the mandatory NDEF Message TLV is present, and
- the length field of the mandatory NDEF Message TLV SHALL be different from zero.

6.4 Command Sequence Description

In this section several procedures are described to manage NFC-defined data e.g. the mandatory NDEF Message TLV. The different state changes or transitions between the states of the ICODE label are shown in detail as well.

6.4.1 NDEF Detection Procedure

The NFC device SHALL use the NDEF detection procedure to detect the presence of an NDEF message (see [NDEF]) inside an ICODE label. The NDEF Message TLV that is found by the NDEF detection procedure is also called mandatory NDEF Message TLV or first NDEF Message TLV.

The detection procedure is based on the control of byte 0 and 1 of the CC, and the presence of an NDEF Message TLV that MAY contain an NDEF message.

The NFC device SHALL use the Read Multiple Block if MBREAD is equal to 1b, or the Inventory Page Read if IPREAD is equal to 1b. It is STRONGLY RECOMMENDED to use the Read Multiple Block or the Inventory Page Read to optimize the access time of the ICODE label.

The NDEF detection procedure is the following:

1. Read the CC (block 0) using the Read Single Block or Read Multiple Block or Inventory Page Read command specified in [section 5.1](#).
2. If byte 0 is equal to E1h and Byte 1 - most significant nibble describes the right version number (see [section 6.1.1](#))) and Bit 2-3 of Byte 1 in block 0 is equal to 00b then go to item 3. Otherwise no NDEF data is detected in the ICODE label.
3. Read sequentially the data area using the Read Single Block or Read Multiple Block or Inventory Page Read command specified in [section 5.1](#) starting from block 1 and search for NDEF Message TLV. Stop the searching as soon as a first NDEF Message TLV is found. If no NDEF message TLV is detected in the ICODE label, the tag is not in a valid state.
4. If the NDEF Message TLV is found:
 - a. If the length field is different from zero, the NDEF message (see [NDEF]) is detected in the ICODE label, or
 - b. If the length field is equal to zero no NDEF Message is detected. The label might be in INITIALISED state.

The NDEF detection procedure does not relate to a valid NDEF message (see [NDEF]). It reads the length of the store NDEF data and does not parse the NDEF data itself.

6.4.2 NDEF Read Procedure

The NDEF Read Procedure is used by the Reader device to read the NDEF Message from the mandatory NDEF Message TLV. Before reading the NDEF Message the NDEF Detection Procedure SHALL be executed (see [section 6.4.1](#)), and the ICODE label SHALL be in a valid state.

The NFC device SHALL use the Read Multiple Block if MBREAD is equal to 1b, or the Inventory Page Read if IPREAD is equal to 1b. It is STRONGLY RECOMMENDED to use the Read Multiple Block or the Inventory Page Read to optimize the access time of the ICODE label.

Using the NDEF Read Procedure the Reader device SHALL read the whole NDEF Message from the mandatory NDEF Message TLV using one or more Read Single Block or Read Multiple Block or Inventory Page Read command (see [section 5.1](#)). The length of the NDEF Message to be read is provided from the length field of the mandatory NDEF Message TLV (see [section 2.1.1](#)).

Implementations that read the NDEF data with less commands is preferred. So reading the whole tag memory in a single command after applying the NDEF Detection Procedure is the most optimal implementation. This means there is no requirement to do a dedicated read of the NDEF message length before reading the NDEF message, there is no requirement to not read beyond of the current NDEF message length but it is not allowed to read beyond the available memory of the tag as stored in the CC.

6.4.3 NDEF Write Procedure

The NFC device SHALL use the NDEF write procedure to write NFC-defined data i.e. the NDEF message inside the first NDEF Message TLV of the ICODE label.

The NDEF write procedure uses the Read Single Block, Read Multiple Block and Write Single Block commands (see [section 5.1](#)), and the NDEF detection procedure (see [section 6.4.1](#)).

The NFC device SHALL only write the NDEF message into ICODE label in INITIALISED or READ/WRITE state.

The NDEF write procedure is the following:

1. Check if the ICODE label is in INITIALISED or READ/WRITE state, and use the NDEF detection procedure.
2. If the ICODE label is in INITIALISED or READ/WRITE state, the first NDEF Message TLV is found (with or without NDEF message in it, see item a and b in [section 6.4.1](#)), and the available memory size for the NDEF Message TLV is big enough to contain the NDEF message, the operations below are allowed to be done in the following order using one or more Write Single Block commands (see [section 5.1](#)):
 - a. The length field of the found NDEF Message TLV is set to one byte long and the value of the length field is set to 00h,
 - b. The new NDEF message is written in the memory area starting from: the 2nd byte after the tag field of the found NDEF Message TLV if the new NDEF Message length is less than 255 bytes, or 4th bytes after the tag field of the found NDEF Message TLV if the new NDEF Message length is bigger than 254 bytes, and
 - c. The 1 or 3 bytes of the length field of the found NDEF Message TLV is updated with the length of the new NDEF message.
3. Otherwise if no NDEF Message TLV is found or the ICODE label is not in INITIALISED or READ/WRITE state or not enough memory space is available in the ICODE label, the NDEF message is not written in the ICODE label.
4. The Terminator TLV is written in the next byte after the first NDEF Message TLV using the Write Single Block command (see [section 5.1](#)), if the NDEF Message TLV block does not end at the last byte of the available data area. The Terminator TLV is not written, if the NDEF Message TLV block ends at the last byte of the available data area.

Concerning the operation item b, the writing of the value field of the found NDEF Message TLV leaves 1 or 3 bytes for the length field (see [section 2.1](#)) that are needed by the next operation item c to store the length of the NDEF message.

The NDEF write procedure over-writes the first NDEF Message TLV, and makes unusable the TLV blocks that were stored after it.

6.4.4 State Changes

This section describes the possible state changes (also called transitions) performed by the NFC device. [Fig 3](#) shows the states and the transitions between them. Below the possible transitions used by the NFC device for the ICODE label are listed:

- Transition from INITIALISED to READ/WRITE,
- Transition from INITIALISED to READ-ONLY, and
- Transition from READ/WRITE to READ-ONLY.

The NFC device SHALL be able to perform the 3 transitions for static and dynamic memory structure.

An ICODE label might be issued in any valid state. So, an ICODE label might be issued in INITIALISED state, READ/WRITE state or even in READ-ONLY state having a predefined NDEF message stored on it.

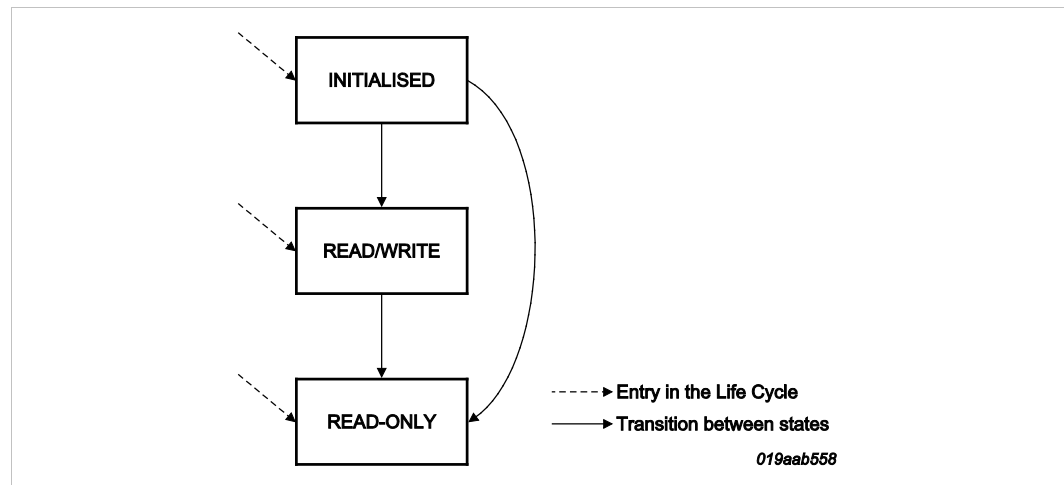


Fig 3. Life Cycle with State Change (transition)

6.4.4.1 Transition from INITIALISED to READ/WRITE

To perform the transition from INITIALISED to READ/WRITE, the NFC device SHALL use the NDEF write procedure (see [section 6.4.3](#)) to replace the empty NDEF Message TLV with a non empty NDEF Message TLV (length field different from zero).

The NFC device MAY perform the transition only if the ICODE label is in INITIALISED state, otherwise it SHALL NOT perform it.

The transition from READ/WRITE to INITIALISED is not defined. To delete or remove the NDEF message in the NDEF Message TLV, the NFC device MAY write an empty NDEF message (see [Annex A in Chapter 7](#)).

6.4.4.2 Transitions from READ/WRITE to READ-ONLY

To perform the transition from READ/WRITE to READ-ONLY the NFC device SHALL:

1. set the Bit 1-0 of Byte 1 of Block 0 (Capability Container Block) to 11h
2. use the Lock Block command to lock all available blocks of the memory layout (see [Chapter 2](#)).

The NFC device MAY perform the transition only if the ICODE label is in READ/WRITE state, otherwise it SHALL NOT perform it.

6.4.4.3 Transition from INITIALISED to READ-ONLY

To perform the transition from INITIALISED to READ-ONLY, the NFC device SHALL execute in the following order:

- The transition from INITIALISED to READ/WRITE (see [section 6.4.4.1](#)), and
- The transition from READ/WRITE to READ-ONLY (see [section 6.4.4.2](#)).

The NFC device MAY perform the transition only if the ICODE label is in INITIALISED state, otherwise it SHALL NOT perform it.

7. ANNEX A: Empty NDEF Message

An empty NDEF Message (see [NDEF]) is defined as an NDEF Message composed of one NDEF record. The NDEF record uses the NDEF short-record layout (SR=1b) with: Type Name Format (TNF) field value equal to 00h (empty, TYPE_LENGTH=00h, PAYLOAD_LENGTH=00h), no ID_LENGTH field (IL=0b), MB=1b, ME=1b, CF=0b. The empty NDEF record (i.e. the empty NDEF Message) is composed of 3 bytes and it is equal to D00000h.

8. ANNEX B: Example of ICODE SLI / ICODE SLIX in INITIALISED State

In this ANNEX an example of ICODE SLI / ICODE SLIX in INITIALISED state is given (see [Fig 4](#)).

Being in INITIALISED state the ICODE SLI / ICODE SLIX contains an empty NDEF message TLV. At the end of the empty NDEF Message TLV a Terminator TLV is present.

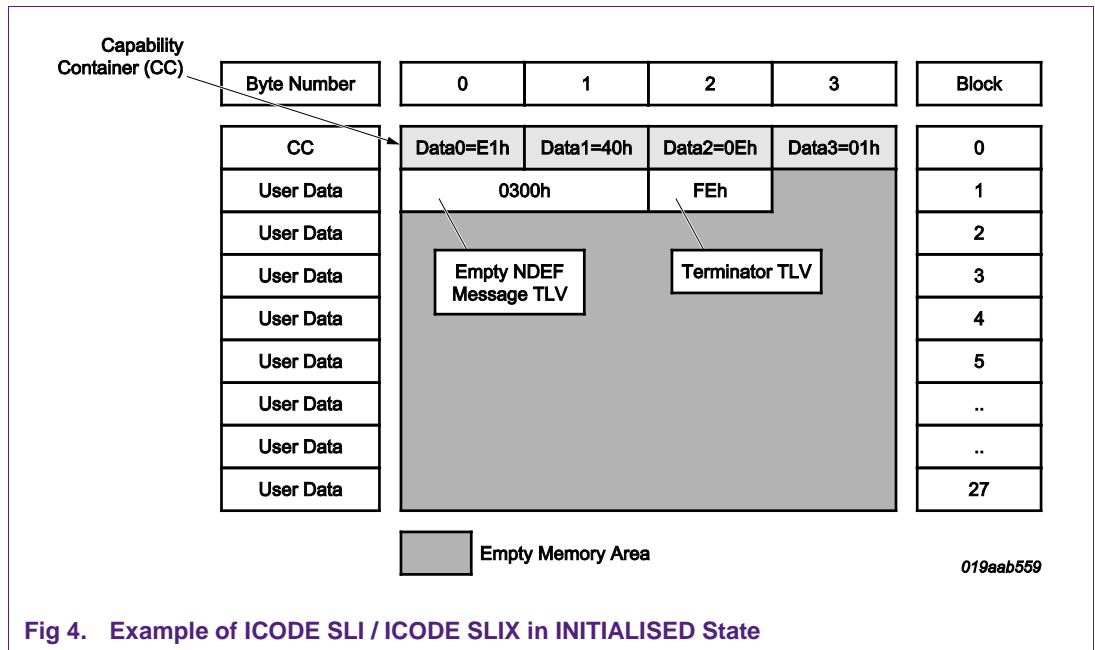
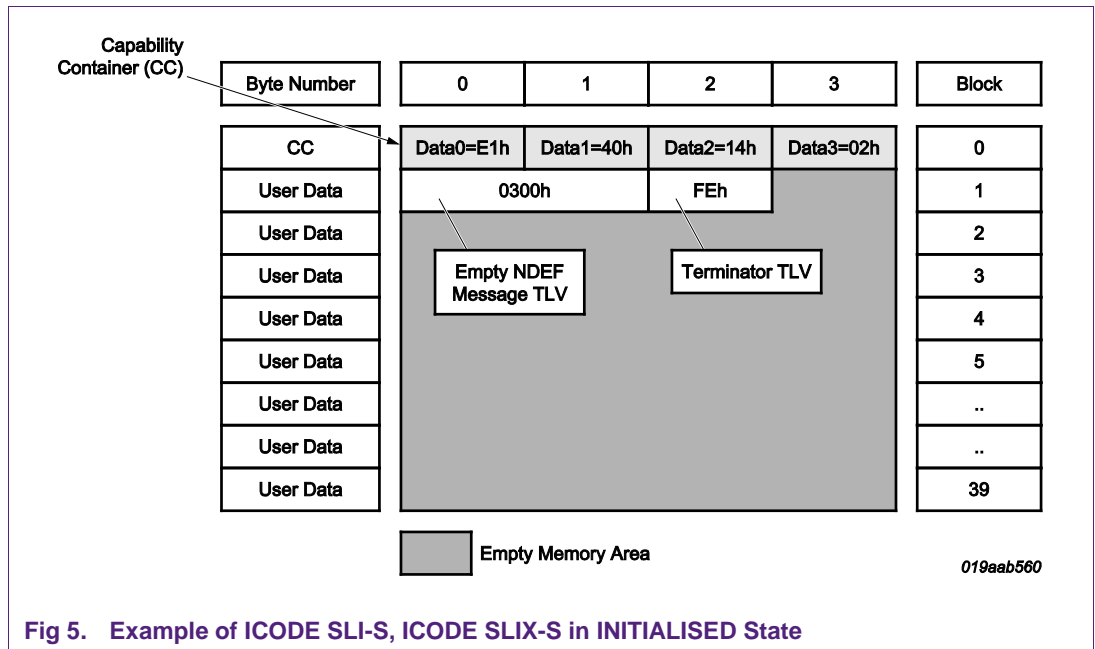


Fig 4. Example of ICODE SLI / ICODE SLIX in INITIALISED State

9. ANNEX C: Example of ICODE SLI-S and ICODE SLIX-S in INITIALISED State

In this ANNEX an example of ICODE SLI-S and ICODE SLIX-S in INITIALISED state is given (see Fig 5).

Being in INITIALISED state the ICODE SLI-S and ICODE SLIX-S contain an empty NDEF message TLV. At the end of the empty NDEF Message TLV a Terminator TLV is present.



10.ANNEX D: Example of ICODE SLI-L and ICODE SLIX-L in INITIALISED State

In this ANNEX an example of ICODE SLI-L and ICODE SLIX-L in INITIALISED state is given (see [Fig 6](#)).

Being in INITIALISED state the ICODE SLI-L and ICODE SLIX-L contains an empty NDEF message TLV. At the end of the empty NDEF Message TLV a Terminator TLV is present.

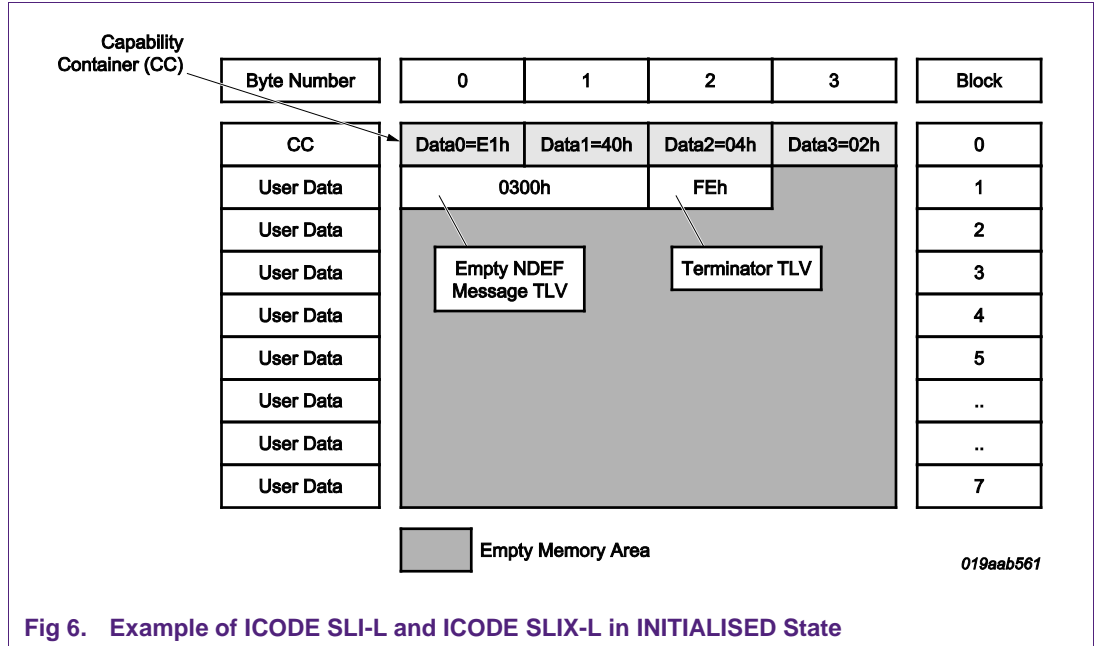


Fig 6. Example of ICODE SLI-L and ICODE SLIX-L in INITIALISED State

11. Legal information

11.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

11.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the

customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out of the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US\$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

11.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

ICODE and I-CODE — are trademarks of NXP B.V.

MIFARE — is a trademarks of NXP B.V.

12. List of figures

Fig 1.	NFC Type ICODE Tag Memory Layout	6
Fig 2.	Length Field Formats	7
Fig 3.	Life Cycle with State Change (transition)	21
Fig 4.	Example of ICODE SLI / ICODE SLIX in INITIALISED State	24
Fig 5.	Example of ICODE SLI-S, ICODE SLIX-S in INITIALISED State	25
Fig 6.	Example of ICODE SLI-L and ICODE SLIX-L in INITIALISED State	26

13. List of tables

Table 1.	Terms and definitions.....	4
Table 2.	Overview of ICODE products.....	6
Table 3.	Defined TLV blocks.....	7
Table 4.	Command Set / Memory Operations.....	12
Table 5.	Block 0 – Capability Container Structure	14
Table 6.	Handling of the mapping document version numbers.....	15
Table 7.	Example of coding of the ICODE SLI / ICODE SLIX CC bytes of block 0	16
Table 8.	Example of coding of the ICODE SLI-S and ICODE SLIX-S CC bytes of block 0	16
Table 9.	Example of coding of the ICODE SLI-L and ICODE SLIX-L CC bytes of block 0.....	17

14. Contents

1.	Introduction	3			
1.1	Applicable Documents	3			
1.2	Convention and notations	4			
1.2.1	Representation of numbers	4			
1.3	Special Word Usage	4			
1.4	Acronyms or Definitions or Glossary	4			
2.	Memory Structure and Management	6			
2.1	TLV blocks	7			
2.1.1	NDEF Message TLV	8			
2.1.2	Proprietary TLV	8			
2.1.3	NULL TLV	8			
2.1.4	Terminator TLV	9			
3.	RF Interface.....	10			
4.	Framing/Transmission Handling.....	11			
5.	Command Set	12			
5.1	Tag Commands and Responses Set	12			
6.	NDEF Detection and Access	14			
6.1	NDEF Management	14			
6.1.1	Version Treating.....	15			
6.1.2	ICODE SLI / ICODE SLIX Capability Container Coding.....	16			
6.1.3	ICODE SLI-S and ICODE SLIX-S Capability Container Coding	16			
6.1.4	ICODE SLI-L and ICODE SLIX-L Capability Container Coding	17			
6.2	NDEF Storage.....	17			
6.3	Life Cycle	17			
6.3.1	INITIALISED State	17			
6.3.2	READ/WRITE State	18			
6.3.3	READ-ONLY State	18			
6.4	Command Sequence Description.....	19			
6.4.1	NDEF Detection Procedure.....	19			
6.4.2	NDEF Read Procedure	19			
6.4.3	NDEF Write Procedure	20			
6.4.4	State Changes	21			
6.4.4.1	Transition from INITIALISED to READ/WRITE	21			
6.4.4.2	Transitions from READ/WRITE to READ-ONLY	21			
6.4.4.3	Transition from INITIALISED to READ-ONLY ..	22			
7.	ANNEX A: Empty NDEF Message	23			
8.	ANNEX B: Example of ICODE SLI / ICODE SLIX in INITIALISED State	24			
9.	ANNEX C: Example of ICODE SLI-S and ICODE SLIX-S in INITIALISED State	25			
10.	ANNEX D: Example of ICODE SLI-L and ICODE SLIX-L in INITIALISED State	26			
11.	Legal information	27			
11.1	Definitions.....	27			
11.2	Disclaimers.....	27			
11.3	Trademarks	27			
12.	List of figures.....	28			
13.	List of tables	29			
14.	Contents	30			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.
