AN2023

ICODE as NFC Type ICODE Tag

Rev. 0.7 — 20 January 2011 Application note

Document information

Info	Content
Keywords	NFC Forum, NFC Forum data mapping, ICODE SLI, ICODE SLI-S, ICODE SLI-L, ICODE SLIX, ICODE SLIX-S, ICODE SLIX-L, ISO-15693
Abstract	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. 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. The “NXP Type ICODE Tag Operation” Application Note has been developed to describe how the reader/writer device (called NFC device) can store/retrieve NDEF data on an ICODE SLI family product. This document extends the information and the functionalities about how an NFC device can manage the ICODE SLI family as an NFC Type ICODE Tag.

Revision history

Rev	Date	Description
0.1	2011-01-04	First draft version
0.3	2011-01-10	Updated Formatting and Card Identification Procedure and Annex: Examples
0.4	2011-01-18	Updated Life Cycle chapter
0.6	2011-01-18	General Review
0.7	2011-01-19	Added review from Christian Weidinger

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 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 (also called NDEF data or NFC Forum data) inside current contactless card products a mapping model is required. The application note [NXPTICODET] describes this mapping model and how the NFC device manages an NXP Type ICODE Tag platform to store NDEF formatted data.

ICODE tag products (see [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L]) are ISO/IEC 15693 compliant contactless cards and compliant to the application note [NXPTICODET] and it can be used as ICODE platform.

This document specifies from the NFC device perspective in Reader/Writer mode:

- how to identify a specific ICODE tag product IC (e.g. ICODE SLI or ICODE SLI-S),
- how to format an ICODE tag product as NXP Type ICODE Tag,
- how to manage an ICODE tag product IC as NXP Type ICODE Tag, and
- how to make use of the additional features of the ICODE tag product IC when operating as NXP Type ICODE Tag.

1.1 Implementation Guidelines

Implementers MAY decide to NOT implement all the possible features (procedures, states…) that this document specifies but only the recommended ones that are needed to support [NXPTICODET] using ICODE Family tag, and the ones required by implementers themselves or customer requirements.

It is RECOMMENDED to implement at least the features listed below to support [NXPTICODET] using ICODE Family tag:

- the memory layout and the relative card identification procedure, see chapter 2,
- the basic states: INITIALISED, READ/WRITE and READ-ONLY, see chapter 6, and
- the formatting procedures, see section 6.5.

Note that the [NXPTICODET] mandates only the support of the mandatory NDEF Message TLV i.e. the 1st NDEF Message TLV even though this application note describes additional features.

1.2 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, Revision 3.1, February 2005.

[ICODE SLI-L] “ICODE SLI-L/ ICODE SLI-L HC SL2 ICS50/ SL2 ICS51”, Data Sheet, Revision 3.0, 14 March 2007, Document Identifier 136430.

[ICODE SLI-S] “ICODE SLI-S SL2 ICS53/ SL2 ICS54”, Data Sheet, Revision 3.0, 14 March 2007, Document Identifier 113730.

[ICODE SLIX] “ICODE SLIX SL2S2002”, Data Sheet, Revision 3.0, 28 May 2010, Document Identifier 178030.

[ICODE SLIX-L] “ICODE SLIX-L SL2S5002”, Data Sheet, Revision 3.0, 28 May 2010, Document Identifier 193130.

[ICODE SLIX-S] “ICODE SLIX-S SL2S5302”, Data Sheet, Revision 3.0, 28 May 2010, Document Identifier 192130.

[NXPTICODET] “NXP Type ICODE Tag Operation”, Application Note, January 2011.

1.3 Convention and notations

1.3.1 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 is 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

1.3.2 Terms and Definition

According to the [NDEF] technical specification the data is represented in Network Byte Order (i.e. big endian). This means Most Significant Byte first and Most Significant Bit first (i.e. MSB first, and msb first)

1.4 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].

1.5 Acronyms or Definitions or Glossary

Table 1. Terms and definitions

Term	Definition
Blank card	An ICODE tag after production phase with its default setting is called blank card.
card	An ICODE tag
CC	Capability Container, the CC stores control data for managing the NFC Forum defined data inside the tag. The CC bytes and the OTP bytes are the same.
lsb	least significant bit
LSB	least significant byte
ICODE Family	In this document the term ICODE Family indicates the family of IC products covering ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L as well as possible future versions.
ICODE	Contactless IC product as described in [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L].
msb	most significant bit
MSB	most significant byte
NDEF	NFC Data Exchange Protocol, 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 http://www.nfc-forum.org/home
NULL TLV	Single byte TLV block mainly used for padding.
Proprietary TLV	TLV block that contains proprietary data
RF	Radio Frequency
RFU	Reserved for Future Use
Tag	An ICODE tag
Terminator TLV	Last TLV block of the tag
TLV	Type Length Value block, data structure element composed of Tag, Length and Value field, see [NXPTICODET]
UID	Unique Identifier

2. Memory Layout

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 EEPROM	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.

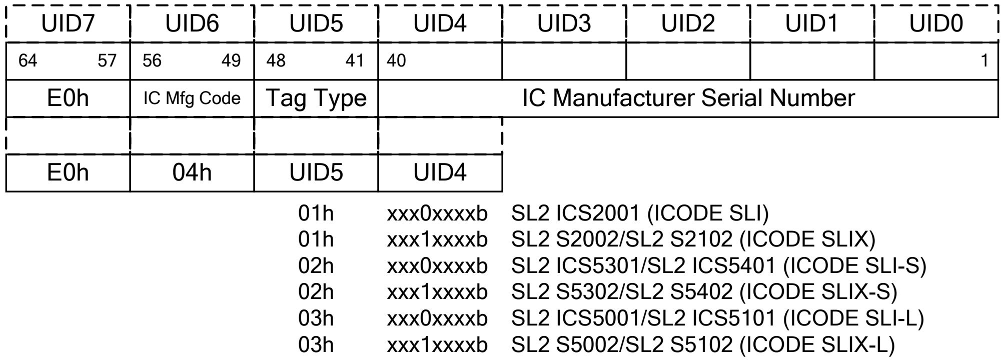
In this document the bit and byte order when defining packets and messages follows the big-endian byte order.

The next two sections describe in detail the memory structures (also called layouts) of ICODE family products.

2.1 UID Coding of ICODE types

The present section describes the Unique IDentifier (UID) coding of ICODE types. The main fields of the UID are (see also Fig 1):

- UID 7 equal to E0h
- UID 6 (also called IC Manufacturer code) equal to 04h indicating NXP Semiconductors
- UID 5 indicating the Tag Type
- UID 4 to UID 0 indicating the IC manufacturer serial number except bit 37.

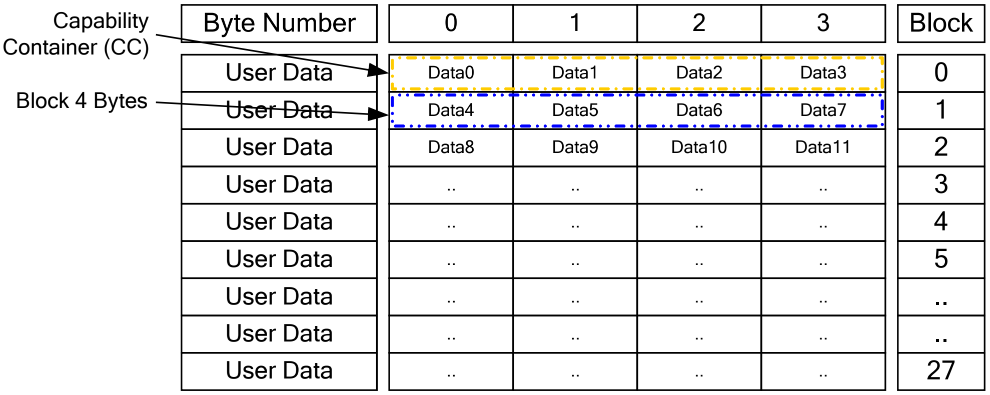


UID5	UID4	
01h	xxx0xxxxb	SL2 ICS2001 (ICODE SLI)
01h	xxx1xxxxb	SL2 S2002/SL2 S2102 (ICODE SLIX)
02h	xxx0xxxxb	SL2 ICS5301/SL2 ICS5401 (ICODE SLI-S)
02h	xxx1xxxxb	SL2 S5302/SL2 S5402 (ICODE SLIX-S)
03h	xxx0xxxxb	SL2 ICS5001/SL2 ICS5101 (ICODE SLI-L)
03h	xxx1xxxxb	SL2 S5002/SL2 S5102 (ICODE SLIX-L)

Fig 1. UID coding of ICODE types. Value indicates with “x” belongs to the IC Manufacturer Serial Number

2.2 ICODE SLI / ICODE SLIX Memory Layout

Fig 2 outlines a simplified memory layout of the ICODE SLI / ICODE SLIX product. The memory area of the ICODE SLI / ICODE SLIX is organized in 28 blocks (block 0 to 27). For more information about the ICODE SLI / ICODE SLIX see [ICODE SLI, ICODE SLIX].

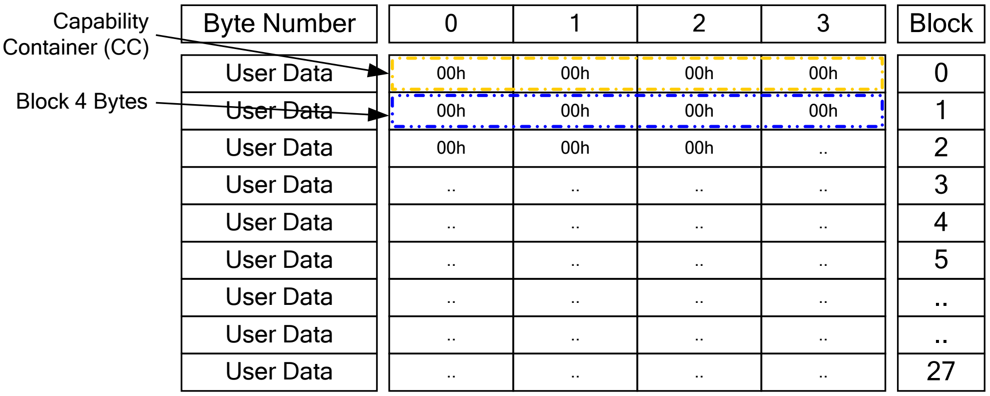


Byte Number	0	1	2	3	Block
User Data	Data0	Data1	Data2	Data3	0
User Data	Data4	Data5	Data6	Data7	1
User Data	Data8	Data9	Data10	Data11	2
User Data	..	..	..	..	3
User Data	..	..	..	..	4
User Data	..	..	..	..	5
User Data	..	..	..	..	..
User Data	..	..	..	..	..
User Data	..	..	..	..	27

Fig 2. ICODE SLI / ICODE SLIX Memory Layout

2.2.1 ICODE SLI / ICODE SLIX Blank Card Settings

The Fig 3 shows the settings (i.e. byte values) for a Blank ICODE SLI / ICODE SLIX Card.

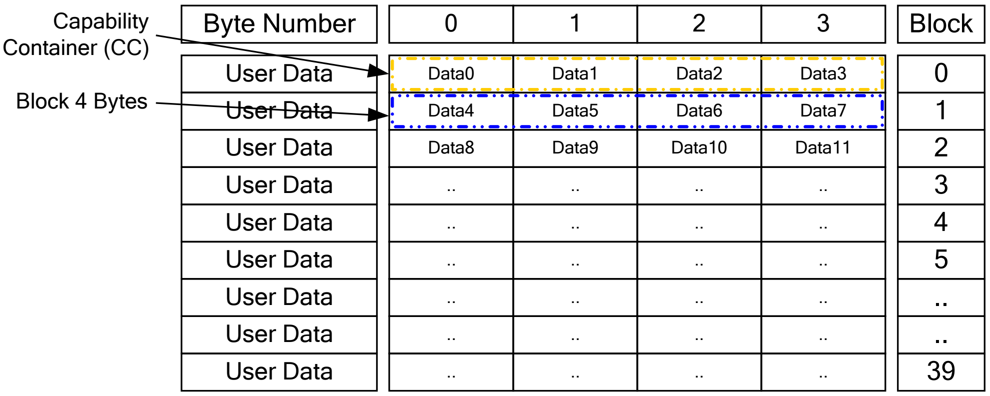


Byte Number	0	1	2	3	Block
User Data	00h	00h	00h	00h	0
User Data	00h	00h	00h	00h	1
User Data	00h	00h	00h	..	2
User Data	..	..	..	..	3
User Data	..	..	..	..	4
User Data	..	..	..	..	5
User Data	..	..	..	..	..
User Data	..	..	..	..	..
User Data	..	..	..	..	27

Fig 3. ICODE SLI / ICODE SLIX Blank Memory Layout

2.3 ICODE SLI-S, ICODE SLIX-S Memory Layout

Fig 4 outlines a simplified memory layout of the ICODE SLI-S, ICODE SLIX-S product. The memory area is organized in 40 blocks (block 0 to 39). For more information about the ICODE SLI-S, ICODE SLIX-S see [ICODE SLI-S, ICODE SLIX-S].

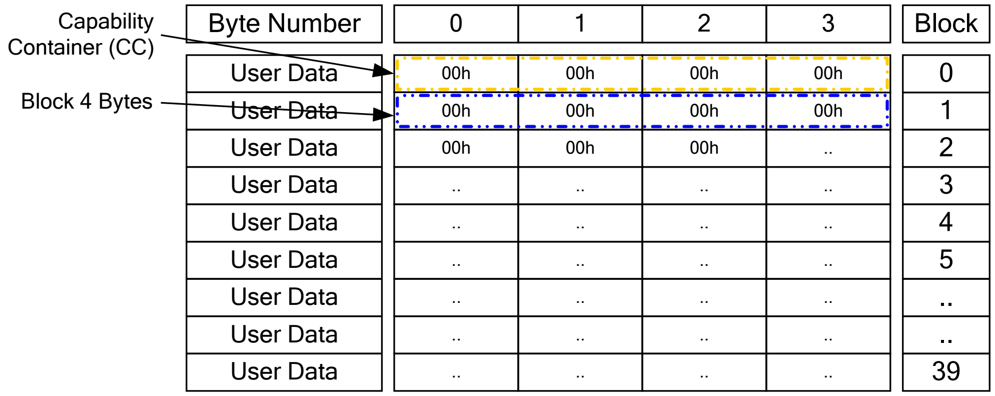


Byte Number	0	1	2	3	Block
User Data	Data0	Data1	Data2	Data3	0
User Data	Data4	Data5	Data6	Data7	1
User Data	Data8	Data9	Data10	Data11	2
User Data	..	..	..	..	3
User Data	..	..	..	..	4
User Data	..	..	..	..	5
User Data	..	..	..	..	..
User Data	..	..	..	..	..
User Data	..	..	..	..	39

Fig 4. ICODE SLI-S, ICODE SLIX-S Memory Layout

2.3.1 ICODE SLI-S, ICODE SLIX-S Blank Card Settings

The Fig 5 shows the settings (i.e. byte values) for a Blank ICODE SLI-S / ICODE SLIX-S Card.

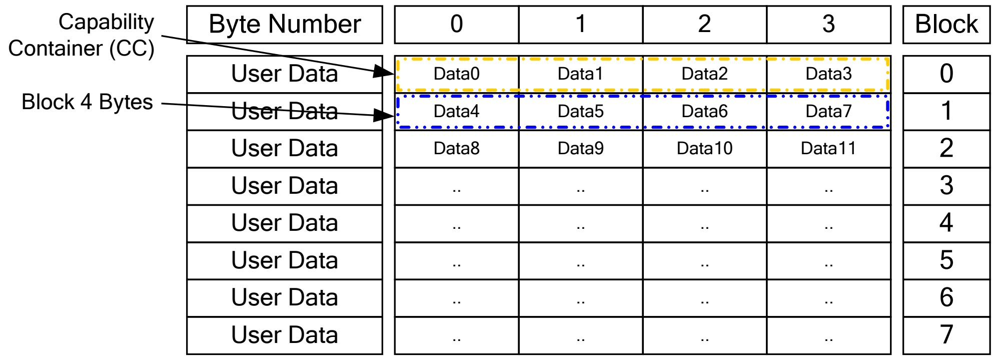


Byte Number	0	1	2	3	Block
User Data	00h	00h	00h	00h	0
User Data	00h	00h	00h	00h	1
User Data	00h	00h	00h	..	2
User Data	..	..	..	..	3
User Data	..	..	..	..	4
User Data	..	..	..	..	5
User Data	..	..	..	..	..
User Data	..	..	..	..	..
User Data	..	..	..	..	39

Fig 5. ICODE SLI-S, ICODE SLIX-S Blank Memory Layout

2.4 ICODE SLI-L, ICODE SLIX-L Memory Layout

Fig 6 outlines a simplified memory layout of the ICODE SLI-L, ICODE SLIX-L product. The memory area is organized in 8 blocks (block 0 to 7). For more information about the ICODE SLI-L, ICODE SLIX-L see [ICODE SLI-L, ICODE SLIX-L].

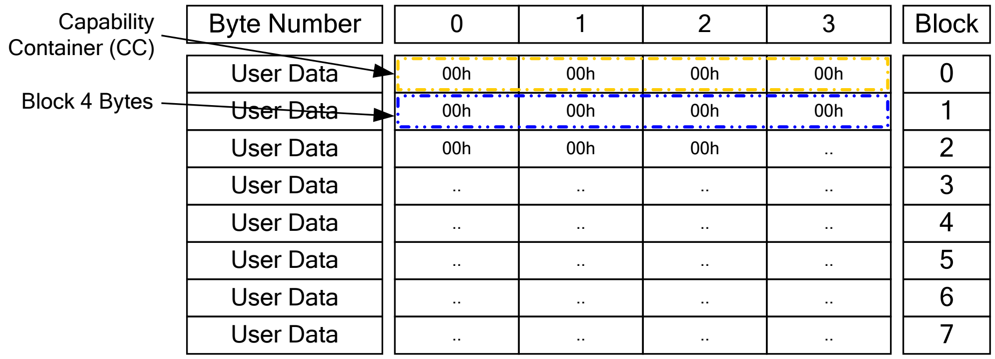


Byte Number	0	1	2	3	Block
User Data	Data0	Data1	Data2	Data3	0
User Data	Data4	Data5	Data6	Data7	1
User Data	Data8	Data9	Data10	Data11	2
User Data	..	..	..	..	3
User Data	..	..	..	..	4
User Data	..	..	..	..	5
User Data	..	..	..	..	6
User Data	..	..	..	..	7

Fig 6. ICODE SLI-L, ICODE SLIX-L Memory Layout

2.4.1 ICODE SLI-L, ICODE SLIX-L Blank Card Settings

The Fig 7shows the settings (i.e. byte values) for a Blank ICODE SLI-L / ICODE SLIX-L Card.



Byte Number	0	1	2	3	Block
User Data	00h	00h	00h	00h	0
User Data	00h	00h	00h	00h	1
User Data	00h	00h	00h	..	2
User Data	..	..	..	..	3
User Data	..	..	..	..	4
User Data	..	..	..	..	5
User Data	..	..	..	..	6
User Data	..	..	..	..	7

Fig 7. ICODE SLI-L, ICODE SLIX-L Blank Memory Layout

2.5 Mapping of NFC Forum data

The NFC device SHALL map NFC Forum data inside ICODE Family Products using the TLV blocks defined in [NXPTICODET]. The TLV blocks are contained inside the data area according to [NXPTICODET]. The ICODE Family Products s contains the following TLV blocks:

1. one or more NDEF Message TLV,
2. zero, one or more Proprietary TLV,
3. zero, one or more NULL TLV, and
4. zero or one Terminator TLV.

To set a blank ICODE Family card into a valid NXP Type ICODE Tag platform the NFC device SHALL use the formatting procedure (see section 6.5). Only then it is possible to store the above named TLV blocks.

2.6 Card Identification Procedure

The NFC device SHALL use the card identification procedures to be able to identify a ICODE Family. Two card identification procedures have been specified:

5. card identification procedure for blank card (i.e. cards with settings defined after the production phase, see section 2.6.1), and
6. card identification procedure when the card is in a valid state according to chapter 6 (see section 2.6.2).

The card identification procedures retrieve the following information:

- the card is a ICODE Family Products,
- which type of ICODE Family Products the card is,
- the memory layout, and
- additional information depending of the setting of the card e.g. settings of the ICODE Family Products …

To identify a ICODE tag the NFC device SHALL perform the following common steps to the two card identification procedure (see Fig 8):

7. Use the Inventory command to get the UID and check:

a. the byte UID7 to be equal to E0h

b. the byte UID6 to be equal to 04h to indicate the NXP manufacturer.

8. If the Inventory command returns successfully and the UID7 and UID6 are according Step 1, the card is a ICODE Family Products
9. Use the Read Single Block command (see chapter 5) to read the Block 0 of the memory area. The Read Single Block command reads 4 bytes.
10. If the 4 bytes of block 0 are all equal to 00h. The ICODE Family card may be a blank card. The NFC device SHALL use the card identification procedure for blank cards, see section 2.6.1.
11. If any byte of the 4 bytes of block 0 is different from 00h, the ICODE Family card may contain NFC Forum data. The NFC device SHALL use the card identification procedure for cards in a valid state according to chapter 6, see section 2.6.2.

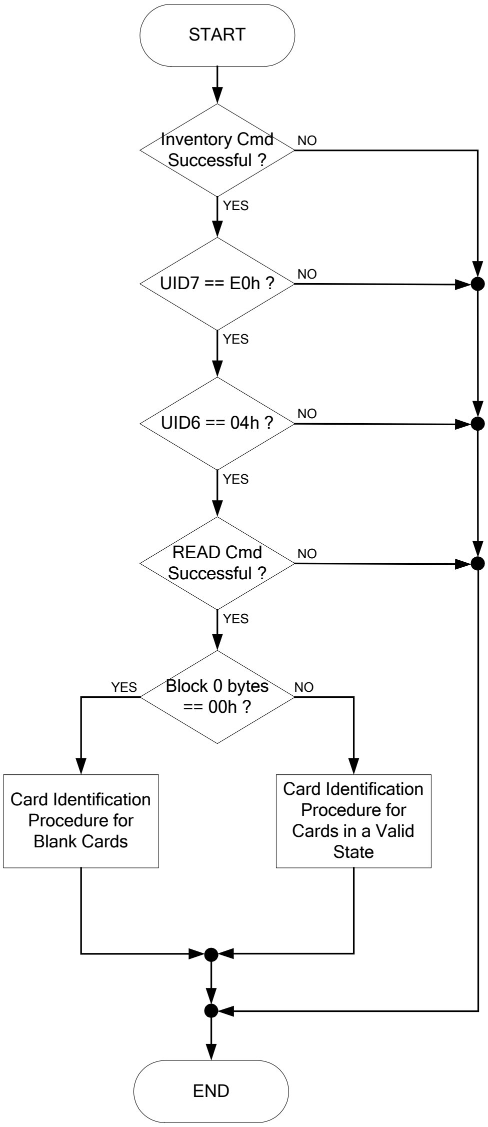

START
Inventory Cmd Successful ?
NO
YES
UID7 == E0h ?
NO
YES
UID6 == 04h ?
NO
YES
READ Cmd Successful ?
NO
YES
YES
Block 0 bytes == 00h ?
NO
Card Identification Procedure for Blank Cards
Card Identification Procedure for Cards in a Valid State
END

Fig 8. Card Identification Procedure

© NXP B.V. 2010. All rights reserved.

2.6.1 Card Identification Procedure for Blank Cards

After production phase the ICODE Family card is blank.

The memory layout information i.e. the memory size and the block size, are retrieved using the following procedure:

12. The NFC device issues the Get System Information command to retrieve the IC reference and the VICC memory size (see [ISOIEC 15693-3]).

    If the Get System Information command fails, the card is not an ICODE family card.

    If the Get System Information command is successful, the card is a blank ICODE family card.

    a. If IC reference is equal to 03h, it indicates that the card is an ICODE SLI-L / ICODE SLIX-L memory layout. This ICODE is supported by this application note, see section 2.4.1. The memory size of the ICODE SLI-L / ICODE SLIX-L is 32 bytes with Block of 4 bytes each.

    b. If IC reference is different from 03h, the memory size and the block size is indicated by the VICC memory size field.

13. The NFC device issues:

    a. in case the Get System Information command to retrieve the IC reference bit 2 equal to 0b, the command Get multiple block security status. If all blocks are not locked the card is blank and can be formatted (see section 6.5)

    b. in case the Get System Information command to retrieve the IC reference bit 2 equal to 1b, the command Get multiple block protection status. If all blocks are not locked and all passwords are disabled the card is blank and can be formatted (see section 6.5)

The identification of the memory layout is needed by the Formatting Procedures (see section 6.5).

2.6.2 Card Identification Procedure for Cards in a Valid State

In case any byte of the 4 bytes of block 0 is different from 00h, the ICODE Family card can contain NFC Forum data. In this case the NFC device SHALL apply the card identification procedure for cards in a valid state to check if the card is really in a valid state according to chapter 6, and to retrieve the memory layout.

To perform the card identification procedure when the card is in a valid state (see chapter 6), the NFC device SHALL apply the NDEF detection procedure (see [NXPTICODET]) to detect the presence of the 1st NDEF Message TLV (the mandatory one) and to check the ICODE Family card is in a valid state (see section 6).

3. Read/Write Access

Depending on the ICODE Family tag the memory blocks may have free or restricted read/write access.

All ICODE Family tags can set the memory block to read-only (no write access allowed).

A subset of the ICODE Family tags (ICODE SLI-S and ICODE SLIX-S) has a password-restricted both read and write access.

For more information, see [ICODE SLI, ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L].

4. Framing / Transmission Handling

The framing and the transmission handling for ICODE Family tag is specified in [NXPTICODET].

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 3 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 3. Command Set / Memory Operations

Memory Operations	
Command	**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
Get System Information	This command allows the retrieval of the system information of the ICODE label.
Reset to ready	After receiving a Reset to ready command, the ICODE label goes back to the Ready state.

Memory Operations	
Command	**Description**
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
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. Life Cycle

An ICODE Family Products can be classified into several states. The state is reflected by the content of the ICODE Family Products. A transaction is a set of operations to change the state on the tag starting from a specific state. Each state has its own valid transactions. In a specific moment in time the ICODE Family Products is in only one state. An entry is an operation to prepare the ICODE Family Products blank card (see chapter 2) into a specific state. The entries are also called formatting procedures (see section 6.5).

In this document two life cycles are presented:

- *The NFC Forum-like life cycle* (see section 6.1). This life cycle is the life cycle specified in [NXPTICODET]. The transitions and entries of this life cycle SHALL be implemented to have a reader device capable to act as NFC device for ICODE family platform, and to operate the ICODE Family product in order to switch between the different states. This application note describes in details the entries mentioned in [NXPTICODET].
- *The ICODE life cycle* (see section 6.2). This life cycle shows the NFC Forum-like life cycle together with additional states and transitions that make use of additional ICODE Family features. The additional transitions, and states MAY be partially implemented due to specific requirements and scenarios that are not covered by [NXPTICODET].

6.1 NFC Forum-like Life Cycle

The specification [NXPTICODET] describes the life cycle from the NFC device perspective as a combination of states, transitions and entries. Fig 9 describes the NFC Forum-like life cycle for the NXP Type ICODE Tag platform. The entries also called formatting procedures are just mentioned in the [NXPTICODET] specification and they are described in details in section 6.5 of this application note.

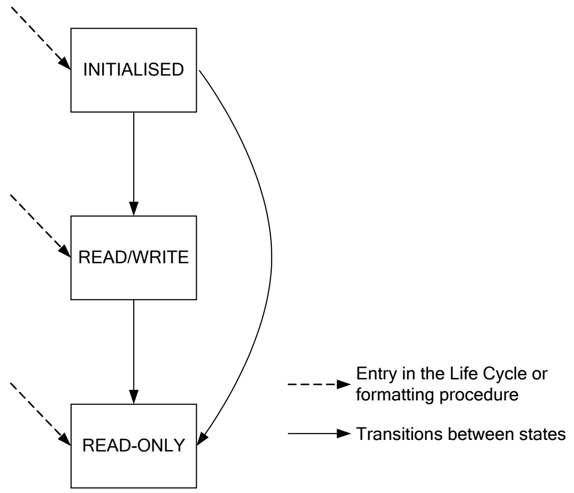

INITIALISED
READ/WRITE
READ-ONLY
Entry in the Life Cycle or formatting procedure
Transitions between states

Fig 9. NFC Forum-like Life Cycle

6.2 ICODE Life Cycle

The ICODE Family tag MAY have additional states than the ones specified in the [NXPTICODET]. These additional states together with additional transitions create the ICODE life cycle (see Fig 10).

The additional states are named in Fig 10 using the prefix "ICODE" being ICODE Family specific. A rounded square indicates parts of the ICODE life cycle that corresponds to the NFC Forum-like life cycle (see Fig 9).

To guarantee compatibility with the [NXPTICODET] specification, the ICODE READ/WRITE state, and ICODEREAD-ONLY state are seeing from a NFC device that implements only the [NXPTICODET] specification as READ/WRITE state and READ-ONLY state. Table 4 shows the relations between the states defined in [NXPTICODET] (see Fig 9) and the states defined in the ICODE life cycle (see Fig 10).

Table 4. Relation between the [NXPTICODET] states and the states defined by this application note

States detected by a NFC device implementing only the [NXPTICODET] technical specification	States detected by a NFC device implementing this application note
INITIALISED	INITIALISED
READ/WRITE	READ/WRITE
	ICODE READ/WRITE
READ-ONLY	READ-ONLY
	ICODE READ-ONLY

The states ICODE READ/WRITE and ICODE READ-ONLY have been introduced to be able:

- to block the overall configuration of the tag,
- to manage the block access in a more flexible way e.g. being able to make specific blocks read/write or read-only, and
- to be compatible with the [NXPTICODET] specification as described in Table 4.

In Fig 10 the dotted arrows indicate the additional transitions that are not described in the [NXPTICODET]. The additional transitions are ICODE Family specific.

The entries or formatting procedures are the same ones described in the NFC Forum life cycle (see section 6.1).

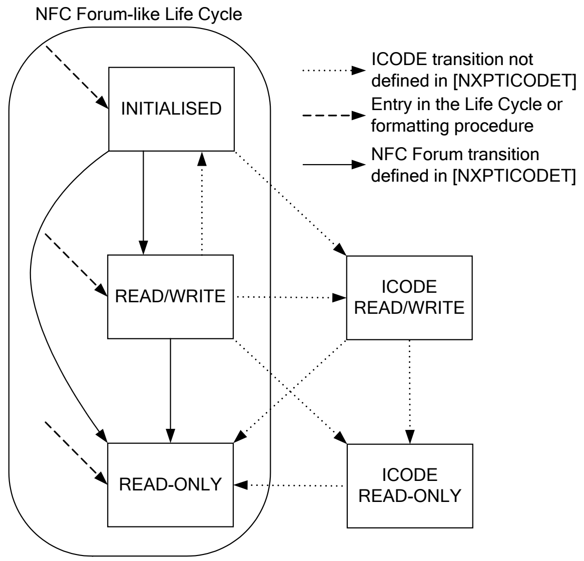

NFC Forum-like Life Cycle
INITIALISED
READ/WRITE
READ-ONLY
ICODE READ/WRITE
ICODE READ-ONLY
ICODE transition not defined in [NXPTICODET]
Entry in the Life Cycle or formatting procedure
NFC Forum transition defined in [NXPTICODET]

Fig 10. ICODE Life Cycle

Overall there are 7 states, 3 entries/formatting procedures and 14 transitions. It is not mandatory to support all states, entries and transitions in the NFC device. An implementation that supports the NXP Type ICODE Tag platform MAY be tailored to support a subset of states, entries and transitions of the ICODE Life Cycle.

During any formatting procedure or transition the ICODE Family Products needs to be hold close to the NFC device to be continuously powered. This can be communicated to the user e.g. showing a message on the user interface. It is up to the implementers to design the recovery mechanisms in case of interruptions or errors during a formatting procedure or a transition.

The identification of the states is based on: different settings of the block access, different settings of the read/write password protection, different content in the data area, and different setting of the capability container (i.e. CC bytes). Table 5 compares the ICODE Life Cycle states highlighting the differences between them. A more comprehensive description of the states is given in section 6.3.

The different settings of the block access, and (if it is the case) of the read/write password protection can be retrieved using:

- In case MBREAD is set to 1b, the Get multiple block security status command
- In case IPREAD is set to 1b, the Get multiple block protection status command

Table 5. Comparative table between the different states

STATE	Access of blocks with CC and 1st NDEF Message TLV blocks	Access of blocks without CC and 1st NDEF Message TLV	CC Byte 1 – Least Significant Nibble (R/W access)	Read Access Password[3]	Write Access Password[3]	Access to TLV blocks
INITIALISED[1]	Free read and write access	read and write access	0000b	Disabled	Disabled	
READ/WRITE[1]	Free read and write access	read and write access	0000b	Disabled	Disabled	
ICODE READ/WRITE	Free read and write access	At least one block has no free read or free write access	0000b	Enabled / Disabled	Enabled / Disabled	Blocks with CC and 1st NDEF Message TLV SHALL NOT be read or write password protected; Blocks with other NDEF Message TLVs or Proprietary TLV MAY be read or write password protected
READ-ONLY	read-only[2]	read-only[2]	0011b	Disabled	Disabled	-
ICODE READ-ONLY	either read-only[2] or write password protected or both	At least one block is not read-only[2]	00xxb, with xxb ≠ 00b	Enabled / Disabled	Enabled / Disabled	Blocks with CC and 1st NDEF Message TLV SHALL NOT be read password protected; Blocks with other NDEF Message TLVs or Proprietary TLV MAY be read or write password protected

Note:

[1] ICODE family tag in INITIaLISED state contains an empty 1st NDEF Message TLV. ICODE family tag in READ/WRITE state contains an non-empty 1st NDEF Message TLV.

[2] read-only means that the block(s) has been locked using the command Lock Block

[3] Read and Write Access by means of password may use either:

- the same Read password, or
- the Read plus the Write password, or
- the Read password for read access and the Write password for write access

In the [NXPTICODET] the states INITIALISED, READ/WRITE and READ/ONLY are identified using the CC Byte 1 – Least Significant Nibble and the 1st NDEF Message TLV (the mandatory one, see [NXPTICODET]). However to identify any of the additional 2 states (ICODE READ/WRITE and ICODE READ-ONLY) also the access block settings and the read/write password settings are needed.

The NFC device that implements one or more of the additional states SHALL identify the states using the settings of the block access and read/write password, and the 1st NDEF Message TLV.

The CC Byte 1 – Least Significant Nibble is overridden by these settings. The NFC device SHALL always set the CC Byte 1 – Least Significant Nibble following Table 5. In

ICODE READ-ONLY state the CC Byte 1 – Least Significant Nibble is coded as indicated in section 6.3.5.

6.2.1 Read Access Condition

The Read Access Condition values (bit 3-2 of CC Byte 1 – Least Significant Nibble) are described in the Table 6.

Table 6. Read Access Condition values

Read Access Condition value (bit_3 bit_2)	Description
00b	free read access
01b	RFU
10b	RFU
11b	RFU

6.2.2 Write Access Condition

The Write Access Condition values (bit 1-0 of CC Byte 1 – Least Significant Nibble) are described in the Table 7.

Table 7. Write Access Condition values

Write Access Condition value (bit_1 bit_0)	Description
00b	free write access
01b	RFU
10b	Write password access (data fields can be written after sending the Write password)
11b	No write access

6.3 States

This section describes how the NFC device SHALL identify the different states shown in section 6.2 and Table 5. If needed the description of the states already described in the NFC Forum specification [NXPTICODET] is explained more in detail.

It is supposed that the Capability Container (CC bytes) is set correctly as described in [NXPTICODET]. In particular the value of the Byte 1 – Least Significant Nibble of the CC is not use to identify the state however it SHALL be set correctly as define in the previous section 6.2 to be compatible with the [NXPTICODET] specification.

The sections below complete the settings for each ICODE state that are described in the previous section 6.2 and Table 5.

6.3.1 INITIALISED State

This state is also described in [NXPTICODET]. The formatting procedure to prepare the ICODE Family tag in this state is described in section 6.5.1.

The INITIALISED state SHALL be identified by:

- all blocks have free read and write access,
- Read/Write Access Passwords are disabled, and
- the ICODE Family tag contains an empty 1st NDEF Message TLV.

6.3.2 READ/WRITE State

This state is also described in [NXPTICODET]. The formatting procedure to prepare the ICODE Family tag in this state is described in section 6.5.2.

The READ/WRITE state SHALL be identified by:

- all blocks have free read and write access,
- Read/Write Access Passwords are disabled, and
- the ICODE Family tag contains an non-empty 1st NDEF Message TLV.

6.3.3 READ-ONLY State

This state is also described in [NXPTICODET]. The formatting procedure to prepare the ICODE Family tag in this state is described in section 6.5.3.

The READ/ONLY state SHALL be identified by:

- all blocks have read-only access,
- Read/Write Access Passwords are disabled, and
- the ICODE Family tag contains an non-empty 1st NDEF Message TLV.

6.3.4 ICODE READ/WRITE State

The ICODE READ/WRITE state is a special case of the READ/WRITE one. Particular settings of the block access and of the read/write passwords identify this state.

The ICODE READ/WRITE state SHALL be identified by:

- all blocks where the CC and the 1st NDEF Message TLV are memorized, have free read and write access,
- at least one of the following is true:
  - o At least one block where the CC and 1st NDEF Message TLV are not memorized, has no free read or write access,
  - o The Read Access Password is enabled, or
  - o The Write Access Password is enabled
- the ICODE Family tag contains an non-empty 1st NDEF Message TLV.

A ICODE Family tag in ICODE READ/WRITE state MAY have blocks that are read-only or readable after the read password is has been verified.

An NFC device that implements only the [NXPTICODET] specification detects in READ/WRITE state an ICODE Family tag in ICODE READ/WRITE state (see Table 4).

The ICODE READ/WRITE state MAY be used to protect an additional NDEF Message TLV or a Proprietary TLV.

6.3.5 ICODE READ-ONLY State

The ICODE READ-ONLY state is a special case of the READ-ONLY one. Particular settings of the block access and of the read/write passwords identify this state.

The ICODE READ-ONLY state SHALL be identified by:

- all blocks where the CC and the 1st NDEF Message TLV are memorized, are read-only,
- at least one of the following is true:
  - At least one block where the CC and 1st NDEF Message TLV are not memorized, is not read-only,
  - The Read Access Password is enabled, or
  - The Write Access Password is enabled
- the ICODE Family tag contains an non-empty 1st NDEF Message TLV.

A ICODE Family tag in ICODE READ-ONLY state MAY have memory blocks that are read/write.

An NFC device that implements only the [NXPTICODET] specification detects in READ-ONLY state a ICODE Family tag in ICODE READ-ONLY state (see Table 4).

The ICODE READ-ONLY state MAY be used to protect the 1st NDEF Message TLV making the relative blocks read-only, but keeping the remaining data area read/write to store additional NDEF Message TLVs, Proprietary TLVs, NULL TLVs, and Terminator TLV.

6.4 State changes/Transitions

This section describes the possible state changes of the ICODE Family tag. Fig 10 shows the states and the state changes (also called transitions) between them.

A transition SHALL be allowed when the ICODE Family tag is in a valid state as defined in section 6.3.

In the transitions below the CC Byte 1 – Least Significant Nibble SHALL be set by the NFC device according to the values defined in Table 5 of the final state.

6.4.1 Transition from READ/WRITE to INITIALSED

This transition SHOULD NOT be implemented. It is described in this document only for sake of completeness.

To perform this transition the NFC device SHALL:

14. Detect the 1st (mandatory) NDEF Message TLV stored in the data area of the ICODE Family tag, using the NDEF detection procedure (see [NXPTICODET]). If no NDEF Message TLV is detected the ICODE Family tag is not in READ/WRITE state and the transition operations SHALL be stopped.
15. Replace the 1st NDEF Message TLV with an empty NDEF Message TLV (i.e. L field equal to 00h, and no V field) using the Write single block command (see chapter 5 and [NXPTICODET] for details concerning the Write single block command).
16. Write the Terminator TLV in the first byte after the first NDEF Message TLV using the Write single block command.

The transition invalidates but does not clear the data written after the Terminator TLV that was present before. To clear the data after the Terminator TLV a sequence of Write single block commands.

6.4.2 Transitions from READ/WRITE to ICODE READ/WRITE

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

- set at least one of the following:
  - at least one block that do NOT contain the CC and the 1st NDEF Message TLV to either no free read access or no free write access or both, or
  - Read password to enabled
  - Write password to enabled
- keep the blocks that contain the CC and the 1st NDEF Message TLV with free read and write access

6.4.3 Transition from INITIALSED to ICODE READ/WRITE

To perform the transition from INITIALISED to ICODE READ/WRITE the NFC device SHALL combine in the following order:

- the transition from INITIALISED to READ/WRITE defined by the NFC Forum (see [NXPTICODET]), and
- the transition from READ/WRITE to ICODE READ/WRITE (see section 6.4.2).

6.4.4 Transitions from ICODE READ/WRITE to ICODE READ-ONLY

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

- set bit 1-0 of CC Byte 1 – Least Significant Nibble to a value different from 00b according to section 6.2.2.
- set the blocks that contain the CC and the 1st NDEF Message TLV to either read-only or write password protected or both, and

6.4.5 Transitions from ICODE READ/WRITE to READ-ONLY

To perform the transition from INITIALISED to ICODE READ/WRITE the NFC device SHALL:

- set bit 1-0 of CC Byte 1 – Least Significant Nibble to 11b
- set all blocks to read-only
- disable Read password and Write password.

6.4.6 Transitions from READ/WRITE to ICODE READ-ONLY

To perform the transition from READ/WRITE to ICODE READ-ONLY the NFC device SHALL combine in the following order:

- the transition from READ/WRITE to ICODE READ/WRITE (see section 6.4.2), and
- the transition from ICODE READ/WRITE to ICODE READ-ONLY (see section 6.4.4).

6.4.7 Transitions from ICODE READ-ONLY to READ-ONLY

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

- set bit 1-0 of CC Byte 1 – Least Significant Nibble to 11b
- set all blocks to read-only
- disable Read password and Write password.

6.5 Formatting Procedures

The formatting procedures for ICODE Family tag are a sequence of commands.

In the formatting procedures the Block 0 of the ICODE Family tag (see chapter 2) is used to store the CC bytes. The mandatory (i.e. 1st) NDEF Message TLV and the Terminator TLV are written in the data area (see [NXPTICODET]).

Byte 0, byte 1 and byte 2 of the Block 0 (or CC area using the terminology in [NXPTICODET]) SHALL NOT be modified once they have been written by the formatting procedure. In particular being the version number (CC byte 1) fixed once written, it means that it is not possible to update a ICODE Family tag to a new future version of the NXP ICODE Family Tag.

6.5.1 INITIALISED Formatting Procedure

The NFC device SHOULD use the INITIALISED formatting procedure to prepare the tag to store NFC Forum defined data (e.g. NDEF message) in INITIALISED state (see section 6.3.1). It is assumed that the ICODE Family tag is configured to allow the INITIALISED formatting procedure i.e. ICODE Family tag is a blank card (see chapter 2).

The INITIALISED formatting procedure writes inside the ICODE Family tag the empty NDEF Message TLV and the Terminator TLV.

In case of ICODE tag, the INITIALISED formatting procedure does not write any Lock Control TLV, or Memory Control TLV.

In case of in case of ICODE SLIX, ICODE SLI-S, ICODE SLIX-S, ICODE SLI-L, ICODE SLIX-L, it is STRONGLY RECOMMENDED to diversify the password.

Before performing the INITIALISED formatting procedure, the NFC device SHALL use the card identification procedure described in section 2.6.1 to detect that the tag is an ICODE blank card that is needed by the INITIALISED formatting procedure.

Below the INITIALISED formatting procedure is shown (see chapter 5 for command details):

17. Send Write Single Block command to set the CC – Block 0 in the following way:
    a. byte 0 is equal to E1h i.e. the magic number,
    b. byte 1 is equal to 40h:
        - The Most Significant Nibble is equal to the version of the NXP Type ICODE Tag specification (e.g. in case of the byte is equal to 0100h)
        - The Least Significant Nibble is equal to 0000b indicating read and write access granted without any security
    c. byte 2 equal to the data area size divided by 8 including the Capability Container, the data area size is calculated during the card identification procedure (see section 2.6.1):
        - in case of ICODE SLI-L / ICODE SLIX-L (IC reference is equal to 03h) byte 2 is equal to 04h,

- in case of other ICODE Family Products the Byte 2 value is calculated from the VICC memory size field in particular, $Byte2 = (BlockSize * NumberOfBlocks)/8$

d. byte 3 is set in the following way:

- in case the IC reference bit 2 is equal to 0b, byte 3 is equal to 01h i.e. MBREAD is set to 1b,

- in case the IC reference bit 2 is equal to 1b, byte 3 is equal to 02h i.e. IPREAD is set to 1b

18. Send one Write Single Block command to write an empty NDEF Message TLV, the Terminator TLV, and in case NULL TLVs in the data area with:

a. The empty NDEF Message TLV equal to 0300h, and

b. The Terminator TLV equal to FEh,

6.5.2 READ/WRITE Formatting Procedure

The READ/WRITE formatting procedure is a combination of two procedures listed below:

19. the INITIALISED formatting procedure (see section 6.5.1), and
20. the transition from INITIALISED to READ/WRITE (see [NXPTICODET]).

The previous list also indicates in which order the NFC device SHALL execute the procedures.

6.5.3 READ-ONLY Formatting Procedure

The READ/WRITE formatting procedure is a combination of three procedures:

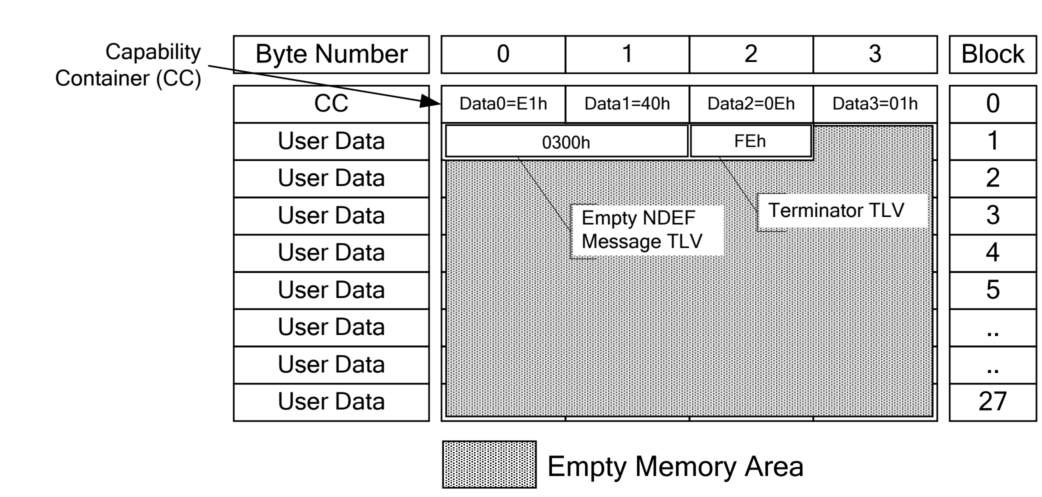
21. the INITIALISED formatting procedure (see section 6.5.1),
22. the transition from INITIALISED to READ/WRITE (see [NXPTICODET]), and
23. the transition from READ/WRITE to READ-ONLY (see [NXPTICODET]).

The previous list also indicates in which order the NFC device SHALL execute the procedures.

7. Example of a ICODE SLI / ICODE SLIX in INITIALISED state

The example below in Fig 11 shows an ICODE SLI / ICODE SLIX formatted as Type 2 Tag in INITIALISED state (see [NXPTICODET]). In particular:

- the bytes in Block 0 are set as the Capability Container in [NXPTICODET]:
  - Byte 0 is equal to E1h (magic number),
  - Byte 1 is equal to 40h indicating version number 1.0 and read and write access capability without any security,
  - Byte 2 is set to 0Eh i.e. 112 bytes of data area including the CC, and
  - Byte 3 is set to 01h i.e. MBREAD equal to 1b, the ICODE label supports Multiple Block Read command.
- NDEF Message TLV on block 1: *Tag* equal to 03h and *Length* equal to 00h.
- Terminator TLV on block 1: *Tag* equal to FEh.



Byte Number	0	1	2	3	Block
CC	Data0=E1h	Data1=40h	Data2=0Eh	Data3=01h	0
User Data	0300h		FEh	▨	1
User Data	▨	▨	▨	▨	2
User Data	▨	▨	▨	▨	3
User Data	▨	▨	▨	▨	4
User Data	▨	▨	▨	▨	5
User Data	▨	▨	▨	▨	..
User Data	▨	▨	▨	▨	..
User Data	▨	▨	▨	▨	27

Fig 11. ICODE SLI / ICODE SLIX in INITIALISED state

8. ANNEX: Examples

In the examples below each command and response are written in hexadecimal format. The top-left byte of each command and response is sent first. CRC0 and CRC1 indicate the CRC bytes.

The example ICODE SLI tag has UID: 97 F1 95 0C 00 01 04 E0. The commands below use ISO15693 addressed mode: the (first) Flags byte of each command equals 20h; the (second) Command byte is followed by the tag's UID, which is then followed by the command's payload. DSFID and AFI are reserved bytes.

For more information about command and response formatting see [ISOIEC15693].

8.1 Example of INITIALISED Formatting Procedure for ICODE SLI and ICODE SLIX

This example shows how the INITIALISED Formatting Procedure (see section 6.5.1) may be implemented for an ICODE SLI and ICODE SLIX tag.

The example of INITIALISED Formatting Procedure is described below

24. READ command with block address = 00h to check if the first block is equal to 0h.
    a. Command: 20 20 97 F1 95 0C 00 01 04 E0 00 CRC0 CRC1
    b. Expected Response: 00 00 00 00 00 CRC0 CRC1 (first Flags byte: no error; followed by 4-byte block contents)
25. Get System Information command to retrieve the IC reference and the VICC memory size
    a. Command: 20 2B 97 F1 95 0C 00 01 04 E0 CRC0 CRC1
    b. Expected Response: 00 0F UID0 ... UID7 DSFID AFI 1B 03 01 (no error, info flags byte, 1Bh = no. blocks – 1, 03h = bytes per block – 1, IC reference = 01h)
    c. Calculated VICC memory size: (1Bh + 1) * (03h + 1) = 112 bytes
26. WRITE command with block address = 00h to set the CC to byte 0 = E1h, byte 1 = 40h, byte 2 = 0Eh, byte 3 = 01h.
    a. Command: 20 21 97 F1 95 0C 00 01 04 E0 00 E1 40 0E 01 CRC0 CRC1
    b. Expected Response: 00h (no error)
27. WRITE command with block address = 01h to write an empty NDEF Message TLV and the Terminator TLV.
    a. Command: 20 21 97 F1 95 0C 00 01 04 E0 01 03 00 FE 00 CRC0 CRC1
    b. Expected Response: 00h (no error)

8.2 Example of Writing an NDEF Message setting the ICODE SLI / ICODE SLIX tag in ICODE BLOCKED READ/WRITE

This example shows how to set a ICODE tag from the INITIALISED state to the ICODE BLOCKED READ/WRITE state. It is a combination of:

- Transition from INITIALISED to READ/WRITE (see [NXPTICODET]), and
- transition from READ/WRITE to ICODE READ/WRITE (see section 6.4.2)

© NXP B.V. 2010. All rights reserved.

The NDEF Message TLV occupies 2 blocks (block 01h and 02h). The block 02h is filled with NULL TLVs to have only the NDEF Message TLV in block 01h and 02h. In this way the Terminator TLV is located on byte 0 of block 03h. The blocks 01h and 02h are blocked to have read-only access to the NDEF Message, but not to the rest of the data area i.e. from block 03h where the Terminator TLV is stored.

As precondition the ICODE tag is in INITILIAZED state. The NDEF message inside the NDEF Message TLV is an empty NDEF message (see APPENDIX A of [NXPTICODET]).

The example is described below:

28. WRITE command with block address 01h to set byte 0 = 03h, byte 1 = 03h, byte 2 = D0h, byte 3 = 00h to write the first part of the NDEF Message TLV.
    a. Command: 20 21 97 F1 95 0C 00 01 04 E0 01 03 03 D0 00 CRC0 CRC1
    b. Expected Response: 00h (no error)
29. WRITE command with block address 02h to set byte 0 = 00h, byte 1 = 00h, byte 2 = 00h, byte 3 = 00h to write the second part of the NDEF Message TLV, and three NULL TLVs.
    a. Command: 20 21 97 F1 95 0C 00 01 04 E0 02 00 00 00 00 CRC0 CRC1
    b. Expected Response: 00h (no error)
30. WRITE command with block address 03h to set byte 0 = FEh, byte 1 = 00h, byte 2 = 00h, byte 3 = 00h to write the Terminator TLV.
    a. Command: 20 21 97 F1 95 0C 00 01 04 E0 03 FE 00 00 00 CRC0 CRC1
    b. Expected Response: 00h (no error)
31. LOCK BLOCK command with block address 01h.
    a. Command: 20 22 97 F1 95 0C 00 01 04 E0 01 CRC0 CRC1
    b. Expected Response: 00h (no error)
32. LOCK BLOCK command with block address 02h.
    a. Command: 20 22 97 F1 95 0C 00 01 04 E0 02 CRC0 CRC1
    b. Expected Response: 00 (no error)

9. Legal information

9.1 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.

9.2 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 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.

9.3 Licenses

Purchase of NXP <xxx> components
<License statement text>

9.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

<Patent ID> — owned by <Company name>

9.5 Trademarks

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

<Name> — is a trademark of NXP B.V.

© NXP B.V. 2010. All rights reserved.

10. List of figures

11. List of tables

© NXP B.V. 2010. All rights reserved.

12. Contents

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

© NXP B.V. 2010. All rights reserved.

For more information, visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 20 January 2011

Document identifier: AN2023