NFC Tag Type 5 Specification

Document Type:	Software Technical Specification
Reference:	STS_NFC_0707-001 Version 1.8 (14516)
Release Date:	Nov. 18, 2011
File Name:	STS_NFC_0707-001 NFC Tag Type 5 Specification.pdf
Security Level:	General Business Use

Disclaimer

History

Version	Date	Comments
0.1	July 21, 2007	First Draft
1.0	Nov. 15, 2007	Public Release
1.1	Jan. 29, 2008	Public Release - Typo correction: the maximum length specified in the Capability Container is related to the NDEF file, and not the NDEF message (section 4.2.2). - Adding examples.
1.2	Feb. 7, 2008	Public Release - Typo correction: Rename Capacity Container by Capability Container. - Add reference to ISO 14443-3 standard and anti-collision procedure
1.3	May 15, 2008	Adding a precision for the ATTRIB command in chapter 5.1
1.4	May 26, 2008	Adding the compression format
1.5	April 8, 2009	Adding the configuration of the Picopass 2K.
1.6	Dec. 8, 2010	Changing document license.
1.7	May 20, 2011	Changing the document name from SPE_NFC_0707-001 into STS_NFC_0707-001.
1.8	Nov. 18, 2011	Removing the support of the tag compression.

Summary of Contents

1 Introduction

The Type 5 Tag is used to store NFC Forum NDEF messages in Picopass 2K/32K and Picotag 2K cards.

This document specifies the detailed format used to recognize and to program Type 5 Tags.

In this document, the devices reading the Type 5 Tag are named “Type 5 Device”.

2 References

[NDEF] NFC Forum™, “NFC Data Exchange Format (NDEF)” v1.0, July 2006.

[14443-3] ISO/IEC 14443-3:2001(E) - Identification cards - Contactless integrated circuit(s) cards - Proximity cards - Part 3: Initialization and anticollision

3 Technical Descriptions

The following information relates to the technical specifications of the Picopass and Picotag cards.

MEMORY

- Picopass/Picotag 2K: up to 32 x 8 bytes over one page (including 3 blocks reserved for personalization).
- Picopass 32K: up to 256 x 8 bytes over two pages (including 3 blocks reserved for personalization on each page).
- EEPROM read/write memory organized by 8-byte block: 1 block or 4 blocks at a time for a read action, and 1 block at a time for a write action.
- Power-Guard system **(anti-tearing)** for secure EEPROM writing.

CONTACTLESS COMMUNICATION

- Protocol: ISO14443B-3 (including anti-collision as specified in **[14443-3]**).
- Carrier frequency: 13.56 MHz +/- 7 KHz.
- Data rate: 106 or 424 Kbits/s (ISO14443B-3)

CHIP CARACTERISTICS

- EEPROM updating (erase and program) time: 5 ms per 8-byte block.
- EEPROM lifetime: >100 000 write/erase cycles.
- Data retention: minimum 10 years @ 85°C.
- Chip operating temperature range: from -40°C to +85°C.
- Power consumption independent of data read in memory.
- Internal personalization fuses.

4 Memory Structure

This chapter specifies the generic memory structure of the different Picopass cards, and the way the Type 5 Tag information is implemented from it.

4.1 Generic Mapping

4.1.1 2K Cards

The 2K Picopass and Picotag cards have the following memory mapping, with 3 8-byte blocks written at personalization, and 29 8-byte blocks left for the application.

Block	Size: 8 bytes
0	Serial Number
1	Configuration
2	Application Area Issuer
3	Application Area
4	
5	
6	
7	
8	
9	
10	
11	
12	
...	
31	

2K Card Mapping

4.1.2 32K Cards

The memory mapping of the 32 K Picopass cards is organized over two pages of 16Kbits. Each page contains 3 8-byte blocks written at personalization, and 253 8-byte blocks left for the application. If both pages are used to store a NDEF message, the Serial Number, Configuration and Application Issuer Area blocks SHOULD be exactly the same on both pages.

Block	Size: 8 bytes
0	Serial Number
1	Configuration
2	Application Area Issuer
3	Application Area
4	
5	
6	
7	
8	
9	
10	
11	
12	
...	
255	

Page 0

Block	Size: 8 bytes
0	Serial Number
1	Configuration
2	Application Area Issuer
3	Application Area
4	
5	
6	
7	
8	
9	
10	
11	
12	
...	
255	

Page 1

32K Card Mapping

4.1.3 Configuration

This block available on address 1, present on each page of the Picopass 32K cards, contains the relevant information to detect if the card is a correct Type 5 Tag.

Byte	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Designation	Not used for type 5 tag			Block Write Lock	Chip Config	Memory Config	Not used for type 5 tag	Fuses

Block Write Lock (read & write)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Designation	RO	Not used for type 5 tag **DO NOT CHANGE**						

This byte specifies if the page is read-only or not. As long as the RO value (bit 7) is set to 1, it is possible to write on the page. This flag can be modified accordingly to set the page to a read-only mode, by setting the RO bit to 0. Once changed, it is not possible to write on this page anymore, or to reverse the read-only flag. For Type 5 Tags based on the Picopass 32K cards, if both pages are used to store the NDEF message, this flag value should be identical on page 0 & 1. If this is not the case, the card should be discarded by the Type 5 Device.

Chip Config (read-only)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Designation	Not used for type 5 tag			0	Not used for type 5 tag			

If the bit 4 is not equal to 0, the card should be discarded by the Type 5 Device.

Memory Config (read only)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Designation	16K	RFU	Book	2K	RFU	ISOB-3	Reserved	
2K	0	X	0	1	X	0	X	X
32K	1	X	1	1	X	0	X	X

If this byte has not one of the specified values, the card should be discarded by the Type 5 Device.

Fuses (read only, apart from bit 3)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Designation	Not used for type 5 tag **DO NOT CHANGE**	Coding		Not used for type 5 tag **DO NOT CHANGE**				

The two Coding bits are used to get the protocol(s) supported by the Type 5 Tag:

- 00 or 01 for ISO14443B-3

If this byte has not one of the specified values, the card should be discarded by the Type 5 Device.

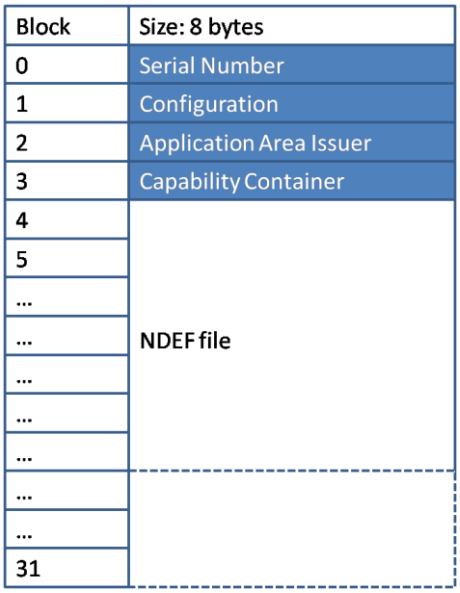
4.2 Type 5 Tag Mapping

This section describes the data mapping in the Type 5 Tags. All the multi-bytes integer values are encoded using the little-endian convention.

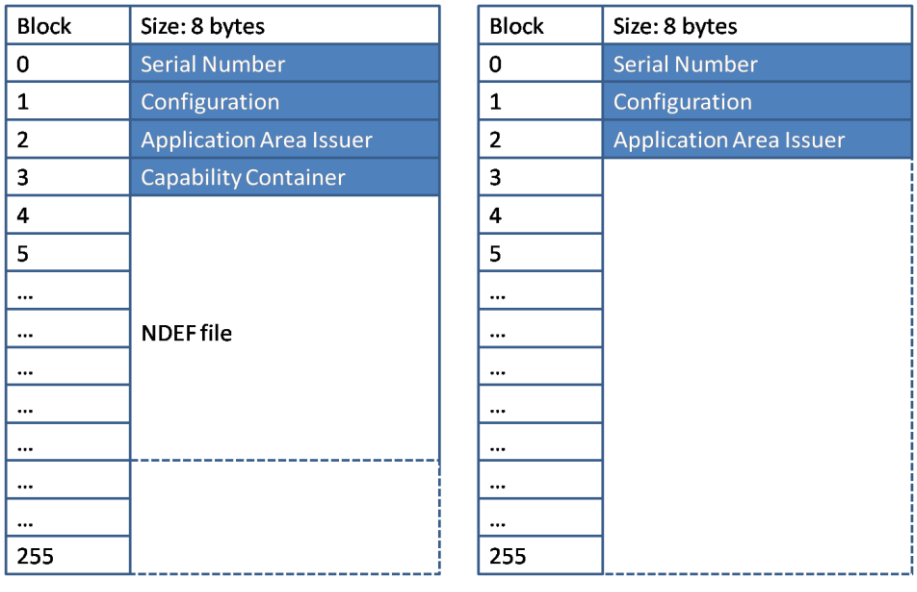
4.2.1 Diagram

The Type 5 Tag specification is based on **[NDEF]** for Message and Record usage. To retrieve the Message and Record information, two containers are specified:

- **Capability Container (CC):** contains the NDEF identification string "NDEF" (0x4E, 0x44, 0x45, 0x46 in hexadecimal) and the NDEF file information (mapping version and maximum NDEF file length) on block 3,
- **NDEF file:** contains, from block 4, the total NDEF messages length in bytes coded on the first two bytes, and one or more NDEF messages, from the second byte, on a maximum length depending on the Type 5 Tag capacity.
-



2K Card Mapping



32K Card Mapping

4.2.2 Capability Container

To detect and access NDEF messages, the Type 5 Device retrieves and uses the information contained in the Capability Container (CC) on block 3. The data structure of the CC file is described in Table 1.

Table 1: Data Structure of the Capability Container file

Offset (bytes)	Size (bytes)	Field	Remarks
00h	4	NDEF Identification string	Identifies a Type 5 Tag ("NDEF").
04h	1	Mapping Version	Indicates the mapping specification version it is compliant to. For the current specification, the version should be 10h.
05h	2	Maximum NDEF file length - 2	The maximum length of the NDEF file in bytes minus 2. Valid values are between 0000h to: - 2K: ((32 – 4) * 8) – 2 = 222 bytes. - 32K: ((512 – 4 – 3) * 8) – 2 = 4038 bytes.
07h	1	RFU	This value should be set to 0x00.

4.2.3 NDEF File

The format of the NDEF file is the length of the NDEF messages followed by the content of the NDEF messages. The data format of the NDEF message is defined in **[NDEF]**. The NDEF messages are stored consecutively from the second byte of block 4, in the NDEF file, using the data structure described in Table 2. The bytes of the NDEF message are stored in the block following the block order. In the 32K tags, if the message is larger than the capacity of the Book 0, the remaining part of the message is stored in the blocks of the Book 1, starting at the first byte of the block 3.

Table 2: Data Structure of the NDEF file

Offset	Size	Field	Remarks
0000h	2	NLEN [bytes]	The NDEF Length field (NLEN) indicates the total length in bytes of the NDEF messages stored in the NDEF file. Valid NLEN values are between 0000h and the maximum message size described in the Capability Container. The special value FFFFh is equivalent to 0000h.
0002h	NLEN	NDEF messages	NDEF message format is defined in [NDEF]. The messages may be compressed (see below).

5 Command set

This chapter describes the command set of the Type 5 Tags.

5.1 Activation of the Transmission Protocol

The activation of the transmission protocol is described in the ISO14443B-3 specification **[14443-3]**.

Note that the type 5 tags returns 4 bytes of application data in the ATQB response. The 4 bytes are the 4 most significant bytes of the serial number of the tag. The 4 less significant bytes of the serial number are in the PUPI.

The 4 bytes of the application data SHALL be used as Higher Layer INF bytes in the ATTRIB command used to select the tag.

5.2 High Level Command Set

The commands that SHALL be supported by the Type 5 Device are listed in Table 3. The format of the commands and the relative responses of Table 3 are described in section 5.3. To detect and access the NFC Forum data, the specific settings of the command and response fields are described in chapter 6.

Table 3: Command Set overview

Command/Response	*Description*
SELECTPAGE	Select page 0 or 1 (only relevant for 32K Type 5 Tags).
READ	Read data in 1 block (8 bytes).
READ4	Read data in 4 blocks (32 bytes).
UPDATE	Update (erase and write) data in one block (8 bytes).

NOTE This specification provides means of reading and writing the NDEF file. It does not cover the personalization of the Type 5 Tag and modifications of access rights. It is assumed that the Type 5 Tag has already been personalized as expected

5.3 Command/Response Format

The following commands are sent by a Type 5 Device to a Type 5 Tag.

5.3.1 SELECTPAGE

This command is only used on a 32K Type 5 Tag, to select either Page 0 (0x00), or Page 1 (0x10).

Command

	TYPE	**PAGE NUMBER**	**CRC**
LENGTH (byte)	1	1	2
DATA	0x84	32K only: 0x00 for page 0 0x10 for page 1	ISO14443B-3: CRC_B

Response

	DATA	CRC
LENGTH (byte)	8	2
DATA	BLOCK 1 (Configuration block)	ISO14443B-3: CRC_B

5.3.2 READ

This command is used to read 8 bytes in a designated block address.

Command

	TYPE	BLOCK NUMBER	CRC
LENGTH (byte)	1	1	2
DATA	0x0C	2K: 0 ≤ N ≤ 31 32K: 0 ≤ N ≤ 255	ISO14443B-3: CRC_B

Response

	DATA	CRC
LENGTH (byte)	8	2
DATA	BLOCK N	ISO14443B-3: CRC_B

5.3.3 READ4

This command is used to read 32 bytes from a designated block address.

Command

	TYPE	BLOCK NUMBER	CRC
LENGTH (byte)	1	1	2
DATA	0x06	2K: 0 ≤ N ≤ 31 32K: 0 ≤ N ≤ 255	ISO14443B-3: CRC_B

Response

	DATA	CRC
LENGTH (byte)	32	2
DATA	BLOCK N BLOCK N + 1 BLOCK N + 2 BLOCK N + 3	ISO14443B-3: CRC_B

5.3.4 UPDATE

This command is used to write 8 bytes in a designated block address. The response contains the 8 bytes available at the designated address. So it is possible to check if the write procedure was successful or not.

Command

	TYPE	BLOCK NUMBER	DATA	CRC
LENGTH (byte)	1	1	8	2
DATA	0x87	2K: N = 1 or 4 ≤ N ≤ 31 32K: N = 1 or 4 ≤ N ≤ 255	BLOCK N	ISO14443B-3: CRC_B

Response

	DATA	CRC
LENGTH (byte)	8	2
DATA	BLOCK N	ISO14443B-3: CRC_B

6 NDEF Detection and Access

This section describes how the NDEF messages are stored and accessed by the Type 5 Device. The format of the data stored and accessed is fully compliant with [NDEF].

6.1 Detection Procedure

The Type 5 Device SHALL use the following procedure to detect a Type 5 Tag:

1. Perform an antcollision procedure for type B as specified in **[14443-3].**
2. The tag answer must specify a protocol of level 3. If the answer specifies a protocol of level 4, the tag is not a Type 5 Tag and should be rejected.
3. The application data must be present with a length of 4 bytes. If the application data are not present or if the length of the application data is not 4 bytes, the tag is not a Type 5 Tag and should be rejected.
4. The Type 5 tags have a serial number of 8 bytes. The application data are the 4 most significant bytes of the serial number. The PUPI is the 4 less significant bytes of the serial number. The 2 most significant bytes of the application data must be 0xE0 0x12. If this is not the case, the tag is not a Type 5 Tag and should be rejected.

The Type 5 Device SHALL use the following procedure to detect a NDEF message inside a Type 5 Tag:

1. Perform a READ4 command on the blocks 1 to 4 to get the following data:
   - the Configuration information on block 1,
   - the application data (unused) on block 2,
   - the Capability Container (CC) information on block 3, and
   - the NDEF file length - 2 on block 4.
2. Check the Configuration information (see section 4.1.3).
   If all the values are not correct, the card is not a Type 5 Tag and should be rejected.
3. Check the Capability Container (CC) file information (see section 4.2.2):
   - The first four bytes of the block are equal to the Type 5 identification string “NDEF” (0x4E, 0x44, 0x45, 0x46 in hexadecimal),
   - The card mapping version is compliant to the mapping version implemented in the Type 5 Device (the Type 5 Device has a mapping version equal or higher than the mapping version of the card).

   If all the values are not correct, the card is not a Type 5 Tag and should be rejected.
4. Check the NDEF file length:
   - If NLEN>0000h and NLEN ≤ Maximum NDEF file length - 2, a NDEF message is detected inside the Type 5 Tag.
   - If NLEN is equal to 0000h or FFFFh, no NDEF message is detected in the Type 5 Tag.
   - If NLEN is not in these ranges, the Type 5 Tag is not in a valid state.

The following tables define the READ4 command used to get the Configuration information, the Capability Container information, and the NDEF file length.

Table 4: Read Procedure – Command

	TYPE	BLOCK NUMBER	CRC
DATA	0x06	0x01	ISO14443B-3: CRC_B

Table 5: Read Procedure – Response

	DATA		CRC
DATA	Configuration: Application: CC: NDEF File:	BLOCK 1 (8 bytes) BLOCK 2 (8 bytes) BLOCK 3 (8 bytes) BLOCK 4 (8 bytes)	ISO14443B-3: CRC_B

6.2 Read Procedure

The Type 5 Device SHALL execute the NDEF read procedure to read the NDEF file:

1. Detect successfully the NDEF file using the NDEF detection procedure in section 6.1, and
2. Read the NDEF messages starting at offset 0002h of the NDEF file using one or more READ4 commands.

The following tables define the READ4 command used to get the NDEF file information.

Table 6: Read4 Procedure – Command

	TYPE	BLOCK NUMBER	CRC
DATA	0x06	0x04	ISO14443B-3: CRC_B

Table 7: Read4 Procedure – Response

	DATA	CRC
DATA	BLOCK 4 (8 bytes) BLOCK 5 (8 bytes) BLOCK 6 (8 bytes) BLOCK 7 (8 bytes)	ISO14443B-3: CRC_B

6.3 Update Procedure

The Type 5 Device SHALL execute the NDEF update procedure to write or update an NDEF message inside the NDEF file:

1. Detect successfully the NDEF file using the NDEF detection procedure in section 6.1,
2. Check that the write access is granted on the current page, from the information provided in the Configuration block (section 4.1.3),
3. If the length of the NDEF message to be written, is bigger than the Maximum NDEF message length (see Capability Container information in section 4.2.2) the NDEF update procedure is aborted. Otherwise go to item 4.
4. Write the value 0000h in the NLEN field using the UPDATE command,
5. Write the NDEF message in the NDEF message field using one or more UPDATE commands, and
6. Write the length of the NDEF message in the NLEN field using the UPDATE command.

The following tables define the NDEF update command to write or to update the NDEF message inside the NDEF file.

Table 8: NDEF Update Procedure – Command

	TYPE	BLOCK NUMBER	DATA	CRC
DATA	0x87	2K: N = 1 or 4 ≤ N ≤ 31 32K: N = 1 or 4 ≤ N ≤ 255	BLOCK N	ISO14443B-3: CRC_B

Table 9: NDEF Update Procedure – Response

	DATA	CRC
DATA	BLOCK N (8 bytes)	ISO14443B-3: CRC_B

The item 4 to 6 MAY be processed using a single UPDATE command if the NLEN field and the NDEF message field fit inside the data field of the UPDATE command.

Furthermore, once the update procedure is finished, it is possible to set the current page to read-only, by setting the RO bit to 0, in the ***"Block Write Lock"*** byte (see section 4.1.3). To do so, the Type 5 Device MUST first read the Configuration block, set the RO bit to 0, and then update the Configuration block with the new information.

6.4 Select Page Procedure

Concerning 32K Type 5 Tags, if the NDEF file content is stored over two pages, a SELECTPAGE command should be sent to select Page 1, when reading/updating from block 255 of Page 0, to block 3 of Page 1 (the Serial Number, Configuration and Application Issuer Area blocks are present on both pages).

If a SELECTPAGE on Page 1 is processed, the Configuration block information (block 1) provided in the response should be checked again to verify that Page 1 is correctly configured for a Type 5 Tag, and if the write access is granted in case of an update command.

The following tables define the select page command to select Page 1.

Table 10: Select Page Procedure – Command

	TYPE	BLOCK NUMBER	CRC
DATA	0x84	0x10	ISO14443B-3: CRC_B

Table 11: Select Page Procedure – Response

	DATA	CRC
DATA	BLOCK 1 (8 bytes)	ISO14443B-3: CRC_B

7 Appendix A: Examples

7.1 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.

Table 12: Empty NDEF File Example

Block	Offset	Size	Value	Content
3	0h	4	4Eh 44h 45h 46h	“NDEF”
	4h	1	10h	Mapping version
	5h	2	LEN	LEN >= 0003h
	7h	1	00h	RFU
4	0h	2	0003h	NLEN, NDEF message length = 3 bytes
	2h	1	D0h	Empty NDEF message
	3h	1	00h	
	4h	1	00h	

7.2 URL NDEF Message

The URL “http://www.nfc-forum.org” is encoded in an NDEF message composed of one NDEF record.

Table 12: URL NDEF File Example

Block	Offset	Size	Value	Content
3	0h	4	4Eh 44h 45h 46h	“NDEF”
	4h	1	10h	Mapping version
	5h	2	LEN	LEN >= 0012h
	7h	1	00h	RFU
4	0h	2	0012h	NLEN, NDEF message length = 18 bytes
	2h	1	D1h	Header MB=1 ME=1 CF=0 SR=1 IL=0
	3h	1	01h	TYPE_LENGTH=01h
	4h	1	0Eh	PAYLOAD_LENGTH=0Eh
	5h	1	55h	‘U’ = URI
	6h	1	01h	“http://www.”
	7h	1	6Eh	“nfc-forum.org”
5	0h	1	66h	
	1h	1	63h	
	2h	1	2Dh	
	3h	1	66h	
	4h	1	6Fh	
	5h	1	72h	
	6h	1	75h	
	7h	1	6Dh	
6	0h	1	2Eh	
	1h	1	6Fh	
	2h	1	72h	
	3h	1	67h	

8 Appendix B: Configurations

8.1 Configuration of a Picopass 2K in Type 5 tag

The following array gives the configuration of a Picopass 2K tag including an empty NDEF record:

Block 0	Serial number
Block 1 (Config)	FFh FFh FFh FFh 0Fh 1Bh FFh 2Ch
Block 2 (AIA)	20h 00h 60h 00h 00h 00h 00h 00h
Block 3	4Eh 44h 45h 46h 10h 00h DEh 00h
Block 4	00h 03h D0h 00h 00h FFh FFh FFh
Block 5..32	FFh FFh FFh FFh FFh FFh FFh FFh