



# NFC Tag Type 6 Specification

Document Type:	Software Technical Specification
Reference:	STS_NFC_0707-002 Version 1.4 (12114)
Release Date:	Nov. 29, 2011
File Name:	STS_NFC_0707-002 NFC Tag Type 6 Specification v1.4.pdf
Security Level:	General Business Use

## Disclaimer

This document is licensed under the Creative Commons Attribution 3.0 license (<http://creativecommons.org/licenses/by/3.0/>). (You may use the content of this document in any way that is consistent with this license and if you give proper attribution (<http://www.open-nfc.org/license.html#attribution>)).

Copyright © 2007-2011 Inside Secure

Open NFC and the Open NFC logo are trademarks or registered trademarks of Inside Secure.

Other brand, product and company names mentioned herein may be trademarks, registered trademarks or trade names of their respective owners.

## History

Version	Date	Comments
0.1	July 21, 2007	First Draft
1.0	Jan. 31, 2008	Public Release
1.1	March 7, 2008	Rename Capacity Container by Capability Container.
1.2	Dec. 8, 2010	Changing document license.
1.3	May 20, 2011	Changing the document name from SPE_NFC_0707-002 into STS_NFC_0707-002.
1.4	Nov. 29, 2011	Fix Block Size and Number in Appendix A: Examples

## Summary of Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
<b>2</b>	<b>References .....</b>	<b>6</b>
<b>3</b>	<b>Technical Descriptions .....</b>	<b>7</b>
<b>4</b>	<b>Memory Structure .....</b>	<b>8</b>
4.1	Generic Mapping .....	8
4.1.1	Diagram.....	8
4.1.2	Block number and size .....	8
4.1.3	Security status .....	8
4.1.4	Request flag .....	9
4.1.5	Response flag & Error code.....	9
4.2	Type 6 Tag Mapping.....	10
4.2.1	Diagram.....	10
4.2.2	Capability Container .....	10
4.2.3	NDEF File.....	11
<b>5</b>	<b>Command set.....</b>	<b>12</b>
5.1	Activation of the Transmission Protocol.....	12
5.2	High Level Command Set.....	12
5.3	Command/Response Format .....	12
5.3.1	READ_SINGLE / READ_MULTIPLE.....	12
5.3.2	WRITE_SINGLE / WRITE_MULTIPLE .....	13
5.3.3	GET_SYSTEM_INFO .....	14
5.3.4	GET_SECURITY_STATUS .....	14
5.3.5	LOCK_BLOCK.....	15
<b>6</b>	<b>NDEF Detection and Access .....</b>	<b>16</b>
6.1	Detection Procedure.....	16
6.2	Read Procedure .....	17
6.3	Update Procedure .....	17
6.4	Lock Procedure .....	18
<b>7</b>	<b>Appendix A: Examples .....</b>	<b>19</b>
7.1	Empty NDEF Message .....	19
7.2	URL NDEF Message .....	20

# 1 Introduction

The Type 6 Tag is used to store NFC Forum NDEF messages in ISO 15693 cards.

This document specifies the detailed format used to recognize and to program Type 6 Tags.

In this document, the devices reading the Type 6 Tag are named “Type 6 Device”.

## 2 References

- [ISO15693-3] ISO/IEC, "Identification cards — Contactless, integrated circuit(s) cards — Vicinity, cards — Part 3: Anticollision and transmission protocol"
- [NDEF] NFC Forum™, "NFC Data Exchange Format (NDEF)" v1.0, July 2006.

### **3 Technical Descriptions**

The following information relates to the technical specifications of the ISO 15693 cards.

#### **MEMORY**

- Up to 256 blocks can be addressed.
- Block size can be of up to 256 bits.
- This leads to a maximum memory capacity of up to 8 kBytes (64 kBits).
- EEPROM read/write memory organized by block(s), by specifying the first block and the number of blocks addressed.

#### **CONTACTLESS COMMUNICATION**

- Protocol: ISO15693-3.
- Carrier frequency: 13.56 MHz +/- 7 Khz.
- Data rate: 26.48 kbits/s ( $f_c/512$ ).
- Data coding: both data coding modes, described in the ISO15693-3, are supported.

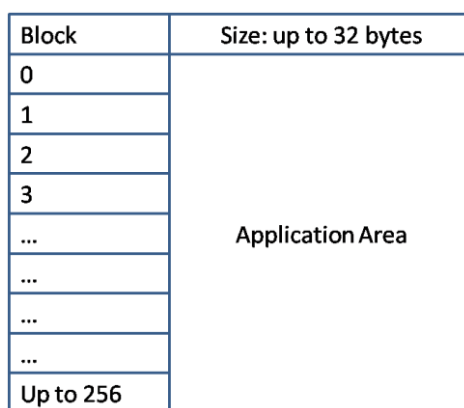
## 4 Memory Structure

This chapter specifies the generic memory structure of the 15693-3 cards, and the way the Type 6 Tag information is implemented from it.

### 4.1 Generic Mapping

#### 4.1.1 Diagram

The ISO 15693 cards have the generic mapping below, with a specific number of blocks and bytes per block, depending on the card/manufacturer.



#### 4.1.2 Block number and size

The number of block and the size of a block may be obtained by sending a Get System Information command to the Type 6 Tag. This command being specified as optional in **[ISO15693-3]**, as well as the content of the response itself, it may not be supported by all the Type 6 Tags.

#### 4.1.3 Security status

The security status of a Type 6 Tag is obtained by sending a Get Security Status command to the Type 6 Tag, on all the blocks, from 00h to the number of blocks - 1.

**Table 1: Security status byte**

Bit	Flag name	Value	Description
b1	Lock_flag	0	Not locked
		1	Locked
b2 to b8	RFU	0	

Depending on the value of the bit1 in all the block security status bytes received in the Type 6 Tag answer, two configurations are possible:

- The bit1s of all the block security status are set to 0b, the CC area and the data area of the Type 6 Tag can be read and written.
- The bit1s of all the block security status are set to 1b, the CC area and the data area of the Type 6 Tag can only be read.

The blocks are lockable via a Lock command sent for each block. This process is irreversible. Once locked, the blocks cannot be unlocked.

**NOTE** This command being specified as optional in **[ISO15693-3]**, it may not be supported by all the Type 6 Tags.

#### 4.1.4 Request flag

The request flag of all the commands is set to 04h. The meaning of this value is described below.

Flags	Flag name	Value	Description
<b>b1</b>	Sub-carrier	0	Single sub-carrier
<b>b2</b>	Data rate	1	High data rate
<b>b3</b>	Inventory	0	Not an inventory command
<b>b4</b>	Protocol extension	0	No protocol format extension
<b>b5</b>	Select	0	Request shall be executed by any VICC
<b>b6</b>	Address	0	Non-addressed request
<b>b7 to b8</b>	RFU	0	

**Request flag**

#### 4.1.5 Response flag & Error code

The response format in case of error is as below. The response flag is the first byte of the answer. The error code is the second byte of the answer.

Flags	Flag name	Value	Description
<b>b1</b>	Error_flag	0	No Error
		1	Error detected
<b>b2 to b8</b>	RFU	0	

**Response flag**

Error code	Description
<b>01h</b>	The command is not supported
<b>02h</b>	The command is not recognized
<b>03h</b>	The command option is not supported
<b>0Fh</b>	Error with no information
<b>10h</b>	The specified block is not available
<b>11h</b>	The specified block is already locked
<b>12h</b>	The specified block is locked and its content cannot be changed
<b>13h</b>	The specified block was not successfully programmed
<b>14h</b>	The specified block was not successfully locked
<b>A0h – DFh</b>	Custom command error codes
<b>All others</b>	RFU

**Response error code**

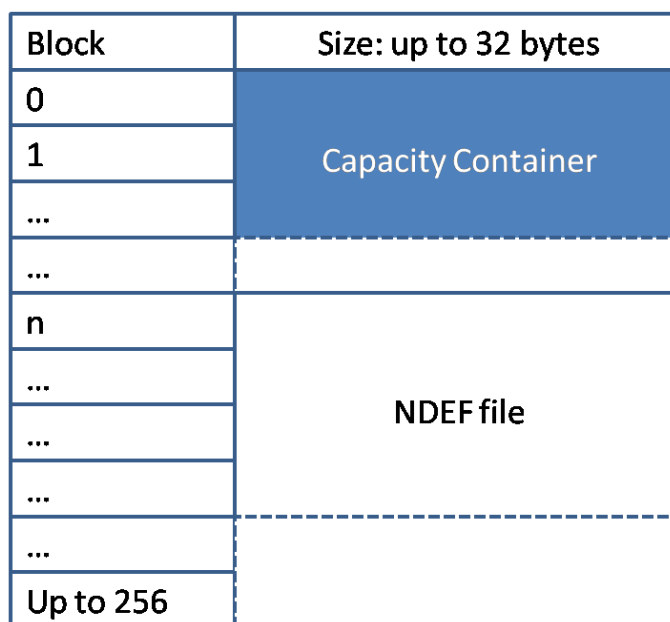
## 4.2 Type 6 Tag Mapping

This section describes the data mapping in the Type 6 Tags. All the multi-bytes integer values are encoded using the little-endian convention.

### 4.2.1 Diagram

The Type 6 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):** starting from block 0, it contains the NDEF identification string "NDE" (0x4E, 0x44, 0x45 in hexadecimal), the mapping version, the number of blocks in the tag and the size of each block, the Tag write access, and the maximum size of the NDEF file,
- **NDEF file:** contains, from the block address specified in the Capability Container:
  - o the total length in bytes of the NDEF messages coded on the first two bytes, and
  - o the content of one or more NDEF messages.



### 4.2.2 Capability Container

To detect and access NDEF messages, the Type 6 Device retrieves and uses the information contained in the Capability Container (CC) from block 0. The data structure of the CC file is described in Table 2.

The Get System Information and the Get Block Security Status commands being optional in [ISO15693-3], the Capability Container contains 2 bytes to code the block number and size, and one byte to tell if the Type 6 Tag is writable or not.

**Table 2: Data Structure of the Capability Container file**

Offset (bytes)	Size (bytes)	Field	Remarks
00h	3	NDEF Identification string	Identifies a Type 6 Tag ("NDE").
03h	1	Mapping Version	Indicates the mapping specification version it is compliant to. For the current specification, the version should be 10h.
04h	2	Block number and size	The number of blocks and the block size available in the Type 6 Tag. The information is coded according to the VICC memory size information in [ISO15693-3]. Refer to Table 3.
06h	1	Write access condition	The value 00h indicates that the write access granted without any security. The value FFh indicates that no write access is granted at all (read-only). Other values are RFU.
07h	1	MAX_LEN	The maximum size in blocks of the NDEF File. Valid values are between 0000h to (total block number – CC file size in blocks).
08h	-	Padding set to zero	To reach the boundary of the next block.

**Table 3: VICC memory size information [ISO15693-3]**

MSB				LSB			
15	13	12		8	7		0
RFU	Block size (1 to 32 bytes)			Number of blocks (1 to 256)			

### 4.2.3 NDEF File

The NDEF file is stored from the first block following the Capability Container. 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 the NDEF file, using the data structure described in Table 4.

**Table 4: 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 MAX_LEN specified in the CC file.
0002h	x	NDEF messages	NDEF messages, see [NDEF].

## 5 Command set

This chapter describes the command set of the Type 6 Tags.

### 5.1 Activation of the Transmission Protocol

The activation of the transmission protocol is described in [ISO15693-3]. This process provides the UID and the DSFID of the Type 6 Tag, using the Inventory command.

### 5.2 High Level Command Set

The commands that MAY be supported by the Type 6 Device are listed in Table 5, these commands being optional in [ISO15693-3]. The format of the commands and the relative responses of Table 5 are described in section 5.3. Furthermore, all these commands being optional, their use is dependent on the addressed Type 6 Tag. To detect and access the NFC Forum data, the specific settings of the command and response fields are described in chapter 6.

**Table 5: Command Set overview**

<i>Command/Response</i>	<i>Description</i>
READ_SINGLE / READ_MULTIPLE	Read data in 1 or several blocks.
WRITE_SINGLE / WRITE_MULTIPLE	Write data in 1 or several blocks.
GET_SYSTEM_INFO	Get the system information.
GET_SECURITY_STATUS	Get the security status of a number of blocks.
LOCK_BLOCK	Lock a block.

**NOTE** This specification provides means of reading and writing the NDEF file, getting information on the Type 6 Tag, and locking blocks. It does not cover the personalization of the Type 6 Tag. It is assumed that the Type 6 Tag has already been personalized as expected.

### 5.3 Command/Response Format

The command is sent by a Type 6 Device according to [ISO15693-3].

#### 5.3.1 READ SINGLE / READ MULTIPLE

These commands being optional, the use of a Read Single Block or a Read Multiple Block command is dependent on the addressed Type 6 Tag. You MAY first try to use the Read Multiple Block command, and if the Type 6 Tag returns the error code 01h, 02h or 03h on such a command, you MUST use a Read Single Block instead.

#### **Read Single Command**

	<b>FLAGS</b>	<b>TYPE</b>	<b>BLOCK NUMBER</b>	<b>CRC</b>
<b>LENGTH (byte)</b>	1	1	1	2
<b>DATA</b>	02h	20h	00h < N < Block number – 1	CRC16

### Read Multiple Command

	FLAGS	TYPE	FIRST BLOCK NUMBER	NUMBER OF BLOCKS	CRC
LENGTH (byte)	1	1	1	1	2
DATA	02h	23h	00h < N < Block number - 1	00h < M < Block number - 1	CRC16

### Response

	FLAGS	DATA	CRC
LENGTH (byte)	1	Block length * Number of blocks requested	2
DATA	00h	Read Single: Block size Read Multiple: ( M + 1 ) * Block size	CRC16

### 5.3.2 WRITE SINGLE / WRITE MULTIPLE

These commands being optional, the use of a Write Single Block or a Write Multiple Block command is dependent on the addressed Type 6 Tag. You MAY first try to use the Write Multiple Block command, and if the Type 6 Tag returns the error code 01h, 02h or 03h on such a command, you MUST use a Write Single Block instead.

#### Write Single Command

	FLAGS	TYPE	BLOCK NUMBER	DATA	CRC
LENGTH (byte)	1	1	1	Block Length	2
DATA	02h	21h	00h < N < Block number - 1	Data	CRC16

#### Write Multiple Command

	FLAGS	TYPE	FIRST BLOCK NUMBER	NUMBER OF BLOCKS	DATA	CRC
LENGTH (byte)	1	1	1	1	Block length * number of addressed blocks	2
DATA	02h	24h	00h < N < Block number - 1	00h < M < Block number - 1	Data	CRC16

### Response

	FLAGS	CRC
LENGTH (byte)	1	2
DATA	00h	CRC16

**NOTE** For the Tag It tags from Texas Instrument, the option flag should be set (FLAGS value equals to 0x42) in the Write Single command (the Write Multiple command being not supported by this tag), and an EOF command should be sent afterwards to get the answer from the tag.

### 5.3.3 GET SYSTEM INFO

This command being optional, it MAY be used to get the system information of the Type 6 Tag. If the addressed Type 6 Tag does not support this command, the block number and size can be found in the CC file (see section 4.1.2). If the information in the response to a Get System Information command, and the CC file are different, the card should not be considered as a valid Type 6 Tag. The UID and DSFID values are obtained with the Inventory command.

#### Get System Info Command

	FLAGS	TYPE		CRC
<b>LENGTH (byte)</b>	1	1		2
<b>DATA</b>	02h	2Bh		CRC16

#### Response

	FLAGS	INFO FLAG	UID	DSFID	AFI	Other fields	CRC
<b>LENGTH (byte)</b>	1	1	8	1	1	N	2
<b>DATA</b>	00h	See below	The UID value	The DSFID value	The AFI value		CRC16

The presence of the DSFID, AFI and other fields depends on the Info Flags value coded as below.

Flags	Flag name	Value	Description
<b>b1</b>	DSFID	0	Not supported / not present
		1	Supported / present
<b>b2</b>	AFI	0	Not supported / not present
		1	Supported / present
<b>b3</b>	VICC memory size	0	Memory size not present
		1	Memory size present
<b>b4 to b8</b>	RFU	0	No protocol format extension

#### *Info flags*

Refer to Table 3 to see how the VICC memory size is coded over two bytes.

### 5.3.4 GET SECURITY STATUS

This command being optional, it MAY be used to get the security status of the Type 6 Tag. If the addressed Type 6 Tag does not support this command, the write access can be found in the CC file. If the information in the response to a Get Security Status command, and the CC file, are different, the card should not be considered as a valid Type 6 Tag.

#### Get Security Status Command

	FLAGS	TYPE	FIRST BLOCK NUMBER	NUMBER OF BLOCKS	CRC
<b>LENGTH (byte)</b>	1	1	1	1	2
<b>DATA</b>	02h	2Ch	00h < N < Block number - 1	00h < M < Block number - 1	CRC16

**Response**

	FLAGS	DATA	CRC
<b>LENGTH (byte)</b>	1	Security status on the number of requested blocks	2
<b>DATA</b>	00h	M bytes coded as in section 4.1.3.	CRC16

**5.3.5 LOCK BLOCK**

This command being optional, it MAY be used to lock one block of the Type 6 Tag. When a request is received to lock the Type 6 Tag, this command MAY be used to lock all the blocks of the Type 6 Tag.

**Lock block Command**

	FLAGS	TYPE	BLOCK NUMBER	CRC
<b>LENGTH (byte)</b>	1	1	1	2
<b>DATA</b>	02h	22h	00h < N < Block number – 1	CRC16

**Response**

	FLAGS	CRC
<b>LENGTH (byte)</b>	1	2
<b>DATA</b>	00h	CRC16

**NOTE** For the Tag It tags from Texas Instrument, the option flag should be set (FLAGS value equals to 0x42) in the LOCK BLOCK command, and an EOF command should be sent afterwards to get the answer from the tag.

## 6 NDEF Detection and Access

This section describes how the NDEF messages are stored and accessed by the Type 6 Device. The format of the data stored and accessed is fully compliant with [NDEF].

### 6.1 Detection Procedure

The Type 6 Device SHALL use the following procedure to detect a NDEF message inside a Type 6 Tag:

1. Get the Capability Container (CC) information from block 0, using one or more READ\_SINGLE commands or a READ\_MULTIPLE, over 8 bytes.
2. Check the CC file information (see section 4.2.2):
  - The first three bytes of the block are equal to the Type 6 identification string "NDEF" (0x4E, 0x44, 0x45 in hexadecimal),
  - The tag mapping version is compliant to the mapping version implemented in the Type 6 Device (the Type 6 Device has a mapping version equal or higher than the mapping version of the tag).
  - The block number and size are coherent with [ISO15693-3]. The block number should be lower or equal to 256 bytes, and the block size should be lower or equal to 32 bytes.
  - The maximum NDEF file size MAX\_LEN is less or equal to (block number – size in block of the CC file).

If all the values are correct, go to item 3. Otherwise the Type 6 Tag is not in a valid state.

3. Get the NDEF file length from the NDEF file block address, on two bytes, using a READ\_SINGLE command.
4. Check the NDEF file length:
  - If NLEN > 0000h and NLEN ≤ (MAX\_LEN \* block size) - 2, the NDEF message is detected inside the Type 6 Tag.
  - If NLEN is equal to 0000h, no NDEF message is detected in the Type 6 Tag.
  - If NLEN is superior to MAX\_LEN, the Type 6 Tag is not in a valid state.

The following tables define the READ\_SINGLE / READ\_MULTIPLE command used to get the 8-byt Capability Container information, and the READ\_SINGLE command used to get the 2-byte NDEF file length.

**Table 6: CC Read Procedure – Command**

	FLAGS	TYPE	FIRST BLOCK NUMBER	NUMBER OF BLOCKS	CRC
DATA	02h	23h 20h	00h 00h	01h -	CRC16

**Table 7: CC Read Procedure – Response**

	FLAGS	DATA	CRC
DATA	00h	Read Multiple: Block length * 2 Read Single: Block length	CRC16

**Table 8: NDEF File Read Procedure – Command**

	FLAGS	TYPE	FIRST BLOCK NUMBER	NUMBER OF BLOCKS	CRC
DATA	02h	20h	NDEF File block address	-	CRC16

**Table 9: NDEF File Read Procedure – Response**

	FLAGS	DATA	CRC
DATA	00h	NDEF File block data	CRC16

## 6.2 Read Procedure

The Type 6 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 message that starts at offset 0002h of the NDEF file block address, using one or more READ\_SINGLE / READ\_MULTIPLE commands.

Refer to the Tables in section 6.1 for further information on how to code the commands.

## 6.3 Update Procedure

The Type 6 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 Type 6 Tag, from the information provided in the CC file (see section 4.2.2) and the Get System Info command (see section 4.1.2),
3. If the length of the NDEF message to be written is superior to MAX\_LEN (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 WRITE\_SINGLE command,
5. Write the NDEF message in the NDEF message field using one or more WRITE\_SINGLE / WRITE\_MULTIPLE commands, and
6. Write the length of the NDEF message in the NLEN field using the WRITE\_SINGLE command.

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

**Table 10: NDEF Update Procedure – Command**

	FLAGS	TYPE	FIRST BLOCK NUMBER	NUMBER OF BLOCKS	DATA	CRC
DATA	02h	21h 24h	NDEF File block address	- 00h < M < Block number	Data	CRC16

**Table 11: NDEF Update Procedure – Response**

	FLAGS	CRC
DATA	00h	CRC16

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

**NOTE** For the Tag It tags from Texas Instrument, the option flag should be set (FLAGS value equals to 0x42) in the Write Single command (the Write Multiple command

being not supported by this tag), and an EOF command should be sent afterwards to get the answer from the tag.

## 6.4 Lock Procedure

Furthermore, it is possible to set the Type 6 Tag to read-only, by setting the write status in the CC file. To do so, the Type 6 Device MUST first read the CC file (refer to Tables 6 & 7), set the write access to FFh (see section 4.2.2), and then update the CC file with the new information, using the command below:

**Table 11: CC Update Procedure – Command**

	FLAGS	TYPE	FIRST BLOCK NUMBER	NUMBER OF BLOCKS	DATA	CRC
DATA	02h	21h	NDEF File block address	-	Data	CRC16

**Table 12: CC Update Procedure – Response**

	FLAGS	CRC
DATA	00h	CRC16

**NOTE** For the Tag It tags from Texas Instrument, the option flag should be set (FLAGS value equals to 0x42) in the Write Single command (the Write Multiple command being not supported by this tag), and an EOF command should be sent afterwards to get the answer from the tag.

Furthermore, if the LOCK\_BLOCK command is supported by the tag, the Type 6 Device MUST as well set the block security status accordingly for all the blocks (see section 5.3.5).

## 7 Appendix A: Examples

The following examples are based on tags of 32 4-bytes long sectors.

### 7.1 Empty NDEF Message

An empty NDEF message (see [NDEF]) is defined as an NDEF message composed of one NDEF record.

**Table 12: Empty NDEF File Example**

Block	Offset	Size	Value	Content
0	0h	3	4Eh 44h 45h	"NDE"
	3h	1	10h	Mapping version
1	0h	2	031Fh	Tag with 32 block of 4 bytes
	2h	1	00h	Write access granted
	3h	1	1Eh	MAX_LEN = 30 blocks = 120 bytes
2	0h	2	0003h	NLEN, NDEF message length 3 bytes
	2h	1	D0h	Header MB=1 ME=1 CF=0 SR=1 IL=0
	3h	1	00h	TYPE_LENGTH=00h
3	0h	1	00h	PAYLOAD_LENGTH=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
0	0h	3	4Eh 44h 45h	“NDE”
	3h	1	10h	Mapping version
1	0h	2	031Fh	Tag with 32 block of 4 bytes
	2h	1	00h	Write access granted
	3h	1	1Eh	MAX_LEN = 30 blocks = 120 bytes
2	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
3	0h	1	0Eh	PAYLOAD_LENGTH=0Eh
	1h	1	55h	‘U’ = URI
	2h	1	01h	“http://www.”
	3h	1	6Eh	“nfc-forum.org”
4	0h	1	66h	
	1h	1	63h	
	2h	1	2Dh	
	3h	1	66h	
5	0h	1	6Fh	
	1h	1	72h	
	2h	1	75h	
	3h	1	6Dh	
6	0h	1	2Eh	
	1h	1	6Fh	
	2h	1	72h	
	3h	1	67h	