# About the NDEF Format



## NDEF (NFC Data Exchange Format)

The NFC Data Exchange Format (NDEF) is a standardised data format that can be used to exchange information between any compatible NFC device and another NFC device or tag. The data format consists of **NDEF Messages** and **NDEF Records**. The standard is maintained by the NFC Forum and is freely available for consultation but requires accepting a license agreement to download.

The NDEF format is used to store and exchange information like URIs, plain text, etc., using a commonly understood format. NFC tags like Mifare Classic cards can be configured as NDEF tags, and data written to them by one NFC device (NDEF Records) can be understood and accessed by any other NDEF compatible device. NDEF messages can also be used to exchange data between two active NFC devices in "peer-to-peer" mode. By adhering to the NDEF data exchange format during communication, devices that would otherwise have no meaningful knowledge of each other or common language are able to share data in an organised, mutually understandable manner.

Some helpful app notes and white papers relating to NDEF are listed below:

- NFC Data Exchange Format (NDEF) Technical Specification (requires accepting the license terms)
- NFC Record Type Definition (RTD) Specification (requires accepting the license terms)
- NXP White Paper - NFC Forum Type Tags

# NDEF Messages

NDEF Messages are the basic "transportation" mechanism for NDEF records, with each message containing one or more NDEF Records.

# NDEF Records

NDEF Records contain a specific payload, and have the following structure that identifies the contents and size of the record:

Download File
Copy Code

```
Bit 7     6       5       4       3       2       1       0
------  ------  ------  ------  ------  ------  ------  ------
[ MB ]  [ ME ]  [ CF ]  [ SR ]  [ IL ]  [        TNF          ]

[                         TYPE LENGTH                         ]

[                       PAYLOAD LENGTH                        ]

[                          ID LENGTH                          ]

[                         RECORD TYPE                         ]

[                              ID                             ]

[                           PAYLOAD                           ]
```

# Record Header (Byte 0)

The record header contains a number of important fields, including a 3-bit field that identifies the type of record that follows (the **Type Name Format** or TNF):

**TNF: Type Name Format Field**

The Type Name Format or **TNF Field** of an NDEF record is a 3-bit value that describes the record type, and sets the expectation for the structure and content of the rest of the record. Possible record type names include:

Download File
Copy Code

```
  TNF Value    Record Type
  ---------    -----------------------------------------
  0x00         Empty Record
```

```
                Indicates no type, id, or payload is associated with
this NDEF Record.
                This record type is useful on newly formatted cards
since every NDEF tag
                must have at least one NDEF Record.

  0x01          Well-Known Record
                Indicates the type field uses the RTD type name format.
This type name is used
                to stored any record defined by a Record Type
Definition (RTD), such as storing
                RTD Text, RTD URIs, etc., and is one of the mostly
frequently used and useful
                record types.

  0x02          MIME Media Record
                Indicates the payload is an intermediate or final chunk
of a chunked NDEF Record

  0x03          Absolute URI Record
                Indicates the type field contains a value that follows
the absolute-URI BNF
                construct defined by RFC 3986

  0x04          External Record
                Indicates the type field contains a value that follows
the RTD external
                name specification

  0x05          Unknown Record
                Indicates the payload type is unknown

  0x06          Unchanged Record
                Indicates the payload is an intermediate or final chunk
of a chunked NDEF Record
```

**IL: ID LENGTH Field**

The IL flag indicates if the ID Length Field is preent or not. If this is set to 0, then the ID Length Field is ommitted in the record.

**SR: Short Record Bit**

The SR flag is set to one if the PAYLOAD LENGTH field is 1 byte (8 bits/0-255) or less. This allows for more compact records.

**CF: Chunk Flag**

The CF flag indicates if this is the first record chunk or a

middle record chunk.

**ME: Message End**
The ME flag indicates if this is the last record in the message.MB: Message BeginThe MB flag indicates if this is the start of an NDEF message.

## Type Length

Indicates the length (in bytes) of the Record Type field. This value is always zero for certain types of records defined with the TNF Field described above.

## Payload Length

Indicates the length (in bytes) of the record payload. If the SR field (described above) is set to 1 in the record header, this value will be one byte long (for a payload length from 0-255 bytes). If the SR field is set to 0, this value will be a 32-bit value occupying 4 bytes.

## ID Length

Indicates the length in bytes of the ID field. This field is present only if the IL flag (described above) is set to 1 in the record header.

## Record Type

This value describes the 'type' of record that follows. The values of the type field must corresponse to the value entered in the TNF bits of the record header.

## Record ID

The value of the ID field if an ID is included (the IL bit in the record header is set to 1). If the IL bit is set to 0, this field is ommitted.

## Payload

The record payload, which will be exactly the number of bytes described in the Payload Length field earlier.

## Well-Known Records (TNF Record Type 0x01)

Probably the most useful record type is the **"NFC Forum Well-Known Type"** (TNF Type 0x01). Record types that

adhere to the "Well-Defined" type are each described by something called an RTD or **Record Type Definition**. Some of the current Well-Defined RTDs are:

# URI Records (0x55/'U')

The "Well Known Type" for a URI record is 0x55 ('U'), and this record type can be used to store a variety of useful information such as telephone numbers (tel:), website addresses, links to FTP file locations, etc.

URI Records are defined in the document "URI Record Type Definition" from the NFC Forum, and it has the following structure:

Download File
Copy Code

```
Name            Offset   Size      Description
----            ------   ----      -----------
Identifier Code 0        1 byte    See table below
URI Field       1        N bytes   The rest of the URI (depending on
byte 0 above)
```

The **URI Identifier Code** is use to shorten the URI length, and can have any of the following values:

Download File
Copy Code

```
Value    Protocol
-----    --------
0x00     No prepending is done ... the entire URI is contained in the
URI Field
0x01     http://www.
0x02     https://www.
0x03     http://
0x04     https://
0x05     tel:
0x06     mailto:
0x07     ftp://anonymous:anonymous@
0x08     ftp://ftp.
0x09     ftps://
0x0A     sftp://
0x0B     smb://
0x0C     nfs://
0x0D     ftp://
0x0E     dav://
0x0F     news:
0x10     telnet://
0x11     imap:
```

```
0x12    rtsp://
0x13    urn:
0x14    pop:
0x15    sip:
0x16    sips:
0x17    tftp:
0x18    btspp://
0x19    btl2cap://
0x1A    btgoep://
0x1B    tcpobex://
0x1C    irdaobex://
0x1D    file://
0x1E    urn:epc:id:
0x1F    urn:epc:tag:
0x20    urn:epc:pat:
0x21    urn:epc:raw:
0x22    urn:epc:
0x23    urn:nfc:
```

Following the URI Identifier Code is the **URI Field**. This field provides the URI as per RFC 3987 and contains the rest of the URI after the value corresponding to the URI Identifier is prepended (unless the URI ID is 0x00, in which case the complete URI will be contained in the URI Field).

# Test Records

To Do

# Smart Poster Records

To Do

# Example NDEF Records

| Well Known Records |
|---|

**URI Record**

An example of a URI record is shown in "Memory Dump of a Mifare Classic 1K Card with an NDEF Record" below.

**Text Record**

To Do

**Smartposter Record**

To Do

To Do

# Using Mifare Classic Cards as an NDEF Tag

Mifare Classic 1K and 4K cards can be configured as NFC Forum compatible NDEF tags, but they must be organised in a certain manner to do so. The requirements to make a Mifare Classic card "NFC Forum compliant" are described in the following App Note from NXP:

AN1304 - NFC Type MIFARE Classic Tag Operation

While the App Note above is the authoritative source on the matter, the following notes may also offer a quick overview of the key concepts involved in using Mifare Classic cards as NFC Forum compatible 'NDEF' tags:

## Mifare Application Directory (MAD)

In order to form a relationship between the sector-based memory of a Mifare Classic card and the individual NDEF records, the **Mifare Application Directory** (MAD) structure is used. The MAD indicates which sector(s) contains which NDEF record. The definitive source of information on the Mifare Application Directory is the following application note:

AN10787 - MIFARE Application Directory (MAD)

For reference sake, the two types of MADs (depending on the size of the card in question) are defined below:

## Mifare Application Directory 1 (MAD1)

**MAD1 can be used in any Mifare Classic card** regardless of the size of the EEPROM, although if it is used with cards larger than 1KB only the first 1KB of memory will be accessible for NDEF records.

The MAD1 is stored in the Manufacturer Sector (Sector 0x00) on the Mifare Classic card.

## Mifare Application Directory 2 (MAD2)

MAD2 can only be used on Mifare Classic cards with **more than 1KB of storage** (Mifare Classic 4K cards, etc.). It is **NOT** compatible with cards containing only 1KB of memory!

The MAD2 is stored in sectors 0x00 (the Manufacturer Sector) and 0x10.

## MAD Sector Access

The sectors containing the MAD1 (0x00) and MAD2 (0x00 and 0x10) are protected with a KEY A and KEY B (if you're not familiar with this concept, consult the Mifare Classic summary elsewhere in the PN532/NFC wiki). To ensure that these sectors can be read by any application, the following common KEY A should always be used:

Download File
Copy Code

```
Public KEY A of MAD Sectors
-------------------------------------------------
BYTE 0   BYTE 1   BYTE 2   BYTE 3   BYTE 4   BYTE 5
0xA0     0xA1     0xA2     0xA3     0xA4     0xA5
```

The MAD sector may optionally be write-protected using KEY B if you wish to limit the ability of customers to modify the card contents. The public KEY A will ensure that they can always read the data.

## Storing NDEF Messages in Mifare Sectors

NDEF messages/records may be stored in any sector of the Mifare card, other than the sector(s) use by the MAD or sectors beyond the 1K range if a MAD1 table is used.

When a sector is used to store NDEF records, it is referred to as an NFC Sector. As with the MAD Sector(s) described above, these sectors must always be accessible in at least read-only mode, and as such a common public KEY A also

exists for NFC Sectors, though it is not the same KEY A used in the MAD sector(s):

Download File
Copy Code

```
Public KEY A of NFC Sectors
-------------------------------------------------
BYTE 0   BYTE 1   BYTE 2   BYTE 3   BYTE 4   BYTE 5
0xD3     0xF7     0xD3     0xF7     0xD3     0xF7
```

In order to store an NDEF Message on the Mifare Classic card, the message needs to be wrapped inside something called a **TLV Block**. The basic structure of a TLV Block is described below.

# TLV Blocks

TLV is an abbreviation for three different fields: T for Tag Field, L for Length Field and V for Value Field. A TLV Block consist of one or more bytes, depending on which of these three fields is present. Note that the TLV Block will always be at least one byte since the T Field is mandatory in every case.

**Tag Field**

The Tag Field (or T Field) is the only mandatory field, and uses a single-byte to identify the type of TLV block accordingly to a pre-determined table of values:

Download File
Copy Code

```
TLV Block Types

  Block Type    Value  Description
  ------------ -----  --------------------------------------
  NULL          0x00   These blocks should be ignored
  NDEF Message  0x03   Block contains an NDEF message
  Proprietary   0xFD   Block contains proprietary information
  Terminator    0xFE   Last TLV block in the data area
```

**Length Field**

The Length Field (or L Field) contains the size (in bytes) of the value field. The Length Field can be organised in two different ways, using either one or three bytes.

The one byte format simple contains a single byte value

from 0x00..0xFF.

The three byte format consists of the following format:

Download File
Copy Code

```
  Byte 0:       Always 0xFF to indicate that we are using the three
byte format
  Byte 1..2:    Can be a value between 0x00FF and 0xFFFE
```

Both the one byte and three byte format must be supported for NFC Forum and NDEF compatability.

**Value Field**

The Value Field (or V Field) is only present if the Length Field (described above) is present and not equal to 0x00. If the Length Field is not equal to 0, the Value Fields will contain N bytes of data in the format indicated by the T Field above.

The value field is where the payload (an **NDEF Message**, for example) is stored.

**Terminator TLV**

The Terminator TLV is the last TLV block in the data area, and consist of a single byte: 0x0FE (see the TLV Block Type table above). This TLV Block in mandatory.

# Memory Dump of a Mifare Classic 1K Card with an NDEF Record

Download File
Copy Code

```
[                    Start of Memory Dump                          ]
------------------------Sector 0-------------------------
Block  0    3E  39  AB  7F  D3  88  04  00  47  41  16  57  4D  10  34  08
>9«▯Ó?..GA.WM.4.
Block  1    14  01  03  E1  03  E1  03  E1  03  E1  03  E1  03  E1  03  E1
...á.á.á.á.á.á.á
Block  2    03  E1  03  E1  03  E1  03  E1  03  E1  03  E1  03  E1  03  E1
.á.á.á.á.á.á.á.á
Block  3    00  00  00  00  00  00  78  77  88  C1  00  00  00  00  00  00
......xw?Á......
------------------------Sector 1-------------------------
```

```
Block  4    00  00  03  11  D1  01  0D  55  01  61  64  61  66  72  75  69
....Ñ..U.adafrui
Block  5    74  2E  63  6F  6D  FE  00  00  00  00  00  00  00  00  00  00
t.comþ..........
Block  6    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  7    00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 2-------------------------
Block  8    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  9    00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  10   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  11   00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 3-------------------------
Block  12   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  13   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  14   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  15   00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 4-------------------------
Block  16   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  17   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  18   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  19   00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 5-------------------------
Block  20   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  21   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  22   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  23   00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 6-------------------------
Block  24   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  25   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  26   00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
```

```
Block  27  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 7-------------------------
Block  28  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  29  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  30  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  31  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 8-------------------------
Block  32  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  33  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  34  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  35  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 9-------------------------
Block  36  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  37  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  38  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  39  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 10------------------------
Block  40  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  41  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  42  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  43  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 11------------------------
Block  44  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  45  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  46  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  47  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......□.?@......
------------------------Sector 12------------------------
Block  48  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  49  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
```

```
Block  50  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  51  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......?.?@......
------------------------Sector 13-------------------------
Block  52  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  53  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  54  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  55  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......?.?@......
------------------------Sector 14-------------------------
Block  56  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  57  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  58  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  59  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......?.?@......
------------------------Sector 15-------------------------
Block  60  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  61  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  62  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
................
Block  63  00  00  00  00  00  00  7F  07  88  40  00  00  00  00  00  00
......?.?@......
[                    End of Memory Dump                            ]
```

# NDEF Records

The above example contains two records, both located in sector 1 (sector 0 contains the MAD).

**Record 1**

The first record on the card can be identified by looking at the first byte of block 4 in sector 1.

Download File

Copy Code

```
Block  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15  Char Value
-----  -----------------------------------------------  ------------
04     00 00                                            ..
```

Every record on the Mifare card starts with the **TLV Block** (described above), and the first byte of the TLV Block

(the Tag Field) indicates that this is a **NULL Block type (value 0x00)**. The second byte is the Length Field, and is 0. Since there is no payload for this record (Length = 0), the third byte of the TLV block is not present (the Value Field).

This record was likely inserted when the card was first formatted to ensure that at least one record is present.

**Record 2**

The second record on the card starts at byte 0x02 of block 4 and continues into block 5.

Download File
Copy Code

```
Block  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15  Char Value
-----  -----------------------------------------------  ------------
04           03 11 D1 01 0D 55 01 61 64 61 66 72 75 69  Ñ..U.adafrui
05     74 2E 63 6F 6D                                   t.com
```

Starting with the **TLV Block** data in the first two bytes, we can determine the following:

Download File
Copy Code

```
Byte(s)   Value   Description
-------   -----   -----------
04:02     0x03    Field Type (0x03 = NDEF Message)
04:03     0x11    Length Field (17 bytes)
```

This indicates to us that the record contains an **NDEF Message** (value 0x03), and that the message is 17 bytes long (0x11 in hexadecimal = 17 in decimal value). This means that our NDEF message is contained in the next 17 bytes (04:04..05:04). The NDEF record can then be analysed as follows:

Download File
Copy Code

```
Byte(s)       Value  Description
-------       -----  -----------
04:04          0xD1    This byte is the **NDEF Record Header**, and
indicates that this is
                      an NFC Forum Well Known Record (0x01 in the first
3 bits),
                      and that this is the first and last record (MB=1,
ME=1),
```

```
                    and that this is a short record (SR = 1) meaning
the payload
                    length is less than or equal to 255 chars
(len=one byte).
                    TNF = 0x01 (NFC Forum Well Known Type)
                    IL  = 0    (No ID present, meaning there is no
ID Length or ID Field either)
                    SR  = 1    (Short Record)
                    CF  = 0    (Record is not 'chunked')
                    ME  = 1    (End of message)
                    MB  = 1    (Beginning of message)
04:05         0x01   This byte is the **Type Length** for the Record
Type Indicator
                    (see above for more information), which is 1
byte (0x55/'U' below)
04:06         0x0D   This is the payload length (13 bytes)
04:07        0x55   Record Type Indicator (0x55 or 'U' = URI Record)
04:08          0x01   This is the **start of the record payload**,
which contains the
                    URI Identifier ("http://www.") since this is a
URI Well-Defined
                    Record Type (see Well-Defined Records above).
This will be
                    prepended to the rest of the URI that follows in
the rest of the
                    message payload
04:09..05:04  ...    The remainder of the URI ("adafruit.com"), which
combined with the
                    pre-pended value from byte 04:08 yields:
http://www.adafruit.com
```

## TLV Terminator

Download File

Copy Code

```
Block  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15  Char Value
-----  ----------------------------------------------  ------------
05                   FE                                þ
```