# Simple NDEF Exchange Protocol

Technical Specification

NFC Forum™

SNEP 1.0

NFCForum-TS-SNEP_1.0

2011-08-31

## RESTRICTIONS ON USE

# Contents



# Figures

## Tables

# 1. Introduction

The Simple NDEF Exchange Protocol (SNEP) is an application-level protocol suitable for sending or retrieving of application data units, in the form of NFC Data Exchange Format (NDEF) messages, between two NFC Forum Devices operating in NFC Forum Peer Mode.

## 1.1. Applicable Documents or References

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, S. Bradner, March 1997, Internet Engineering Task Force

[NDEF] NFC Data Exchange Format, Version 1.0, NFC Forum

[LLCP] Logical Link Control Protocol, Version 1.0, NFC Forum

## 1.2. Administration

The Simple NDEF Exchange Protocol Specification is an open specification supported by the Near Field Communication Forum, Inc., located at:

401 Edgewater Place, Suite 600
Wakefield, MA, USA 01880

Tel.: +1 781-876-8955
Fax: +1 781-610-9864

http://www.nfc-forum.org/

The NFC Forum Reference Applications Framework Working Group maintains this specification. Comments, errors, and other feedback can be submitted at
http://www.nfc-forum.org/apps/group_public/document.php?document_id=10115&wg_abbrev=chairs

## 1.3. Name and Logo Usage

The Near Field Communication Forum's policy regarding the use of the trademarks *NFC Forum* and the NFC Forum logo is as follows:

- Any company MAY claim compatibility with NFC Forum specifications, whether a member of the NFC Forum or not.
- Permission to use the NFC Forum logos is automatically granted to designated members only as stipulated on the most recent Membership Privileges document, during the period of time for which their membership dues are paid.
- Member's distributors and sales representatives MAY use the NFC Forum logo in promoting member's products sold under the name of the member.
- The logo SHALL be printed in black or in color as illustrated on the Logo Page that is available from the NFC Forum at the address above. The aspect ratio of the logo SHALL be maintained, but the size MAY be varied. Nothing MAY be added to or deleted from the logos.

- Since the NFC Forum name is a trademark of the Near Field Communication Forum, the following statement SHALL be included in all published literature and advertising material in which the name or logo appears:

  ***NFC Forum and the NFC Forum logo are trademarks of the Near Field Communication Forum.***

## 1.4. Intellectual Property

The Simple NDEF Exchange Protocol Specification conforms to the Intellectual Property guidelines specified in the NFC Forum's *Intellectual Property Rights Policy*, as outlined in the NFC Forum *Rules of Procedures*. These documents are available on the NFC Forum website.

## 1.5. Special Word Usage

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 1.6. Abbreviations

Table 1: Abbreviations

| Abbreviation | Description |
|---|---|
| LLCP | Logical Link Control Protocol |
| NDEF | NFC Data Exchange Format |
| NFC | Near Field Communication |
| SNEP | Simple NDEF Exchange Protocol |

## 1.7. Glossary

*NDEF message*

The basic message construct defined by the NFC Data Exchange Format Specification. An NDEF message contains one or more NDEF records.

*NDEF payload*

The application data carried within an NDEF record.

*NDEF record*

Contains a payload described by a type, a length, and an optional identifier.

*NFC-enabled device*

Device that is capable of performing near field communication.

*NFC Forum Device*

Device that implements at least the mandatory parts of the NFC Forum Protocol Stack and complies with the NFC Forum interoperability requirements. An NFC Forum Device is capable of operating either in NFC Forum Peer Mode, in NFC Forum Reader/Writer Mode, or in NFC Forum Card Emulation Mode.

# 2. Overview

## 2.1. SNEP Communication Protocol

The Simple NDEF Exchange Protocol (SNEP) is a request/response protocol. A SNEP client sends a request to a SNEP server in the form of a protocol version, a request method, the length of an information field in octets, and an information field. The SNEP server performs the action indicated by the request method using the information provided, and then responds with a message containing a protocol version, a status code indicating success or failure of the method invocation, the length of an information field in octets, and an information field.

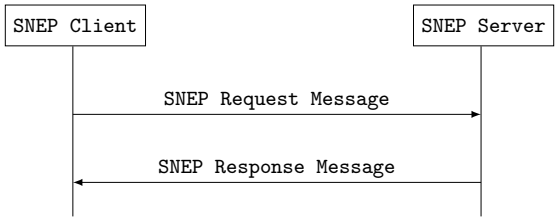


Figure 1: SNEP Communication Model

The purpose of the Simple NDEF Exchange Protocol is to exchange NDEF messages. NDEF messages are transmitted in the information field of SNEP request or response messages. The maximum possible length of any NDEF message to be transmitted is $2^{32} - 1$ (the largest integer that can be encoded in the SNEP request or response header length field). Figure 2 illustrates a SNEP Put request message and the enclosure of an NDEF message in the information field. The format of the SNEP request and response message is defined in Section 3.

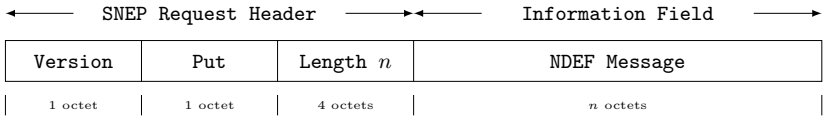


Figure 2: Illustration of a Put Request Carrying an NDEF Message

If the total length of a SNEP request or response message exceeds the capacity of a single service data unit that is acceptable to the underlying transport protocol, the SNEP message SHALL be transmitted in fragments. Fragments SHALL be built by repeatedly removing and transmitting a number of subsequent octets of the complete SNEP message. In order for the receiver of a fragmented SNEP message to determine the number of octets that are to be received with subsequent fragments, the first fragment SHALL include at least the entire SNEP message header. Fragmentation is illustrated in Figure 3 with a SNEP message split into three fragments.

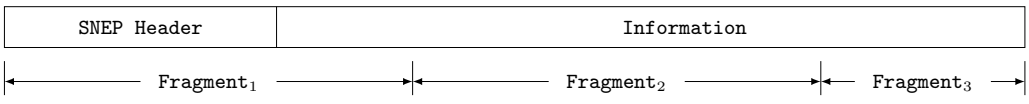


Figure 3: SNEP Message Fragmentation

The receiver of a fragmented SNEP message, after receipt of the first fragment, SHALL indicate to the sender its ability to receive the remaining fragments. The remaining fragments SHALL be transmitted only if the sender receives an indication to continue. The remaining fragments SHALL NOT be acknowledged by the receiver.

It is permissable to transmit a fragmented SNEP message even if the underlying transport protocol would be able to transmit the complete SNEP message within a single service data unit.

If the receiver of a SNEP message does not support the protocol version of the received message, it SHALL determine that the message is complete. Otherwise, the receiver of a SNEP message SHALL determine whether the message is fragmented by evaluating the length of the received data against the length of the information field specified in the SNEP message header. A SNEP message is fragmented if not all octets in the Information field are received with the first service data unit.

Note that for a SNEP server that does not understand the protocol version of a received SNEP request message, the appropriate response is Unsupported Version. If a SNEP client does not understand the protocol version of a received SNEP response message, such response implies that further communication with the server is not possible.

Figure 4 illustrates a SNEP message exchange where both the client's request and the server's response are fragmented.

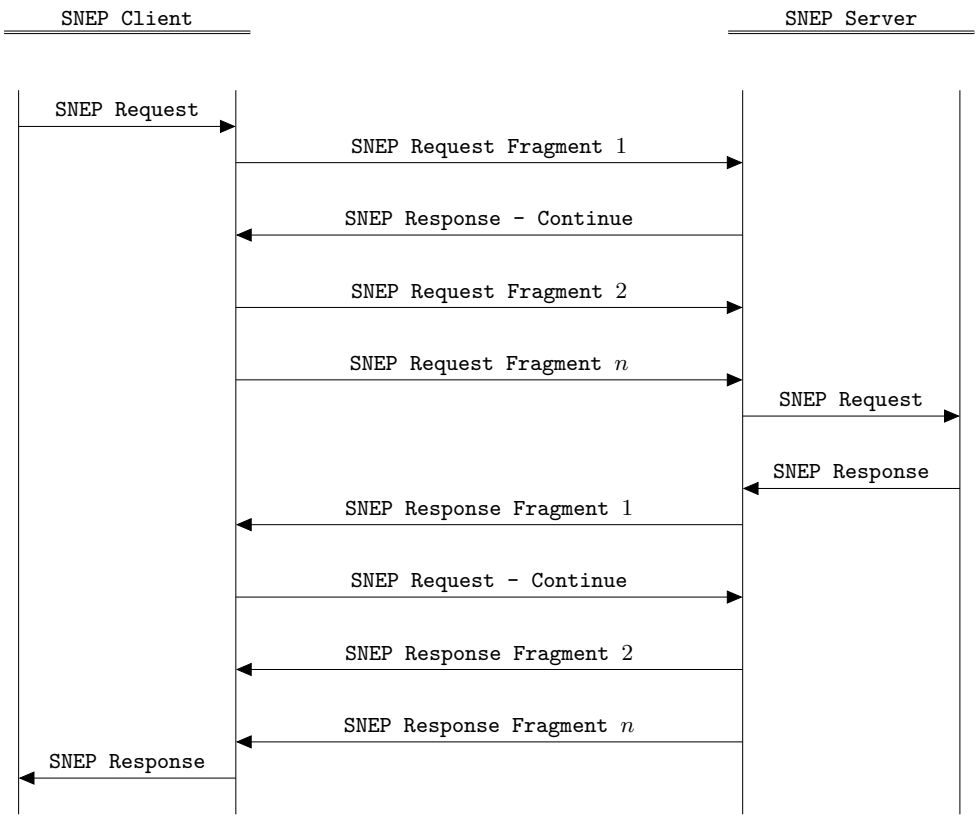


Figure 4: Fragmented Message Exchange

If a SNEP request message is completely transmitted with a single service data unit of the underlying transport, the receiving SNEP server SHALL NOT send an indication to continue but the final

response. Figure 5 illustrates this with the example of a GET request sent by the client to retrieve an NDEF message from the server. Here the request is transmitted within a single service data unit, but the server's response requires fragmentation.

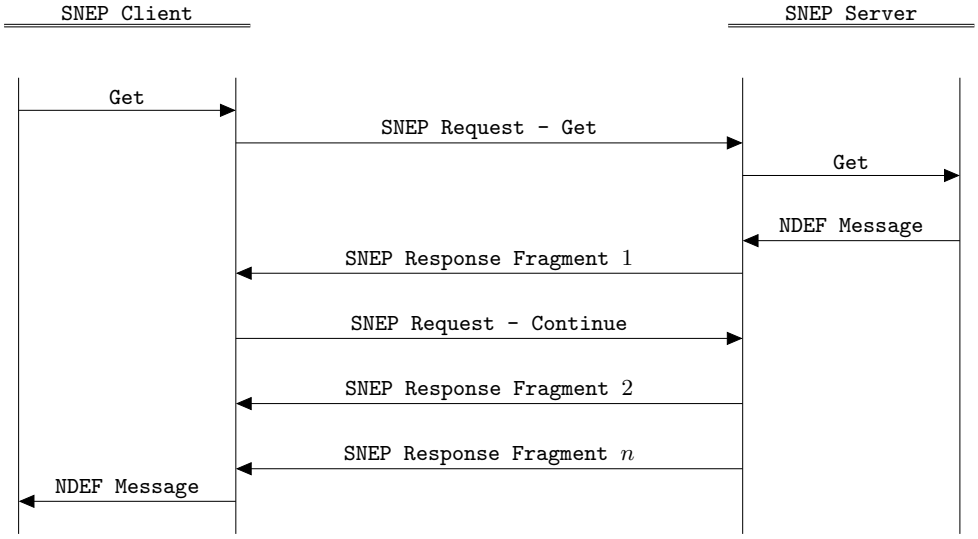


Figure 5: Example Get Request

If a SNEP response is completely transmitted with a single service data unit, the SNEP server SHALL NOT wait for an indication to continue, but will regard the transaction as completed. Figure 6 illustrates this with the example of a Put request sent by the client to convey an NDEF message to the server. Here the request requires fragmentation, but the server's response is sent within a single service data unit.

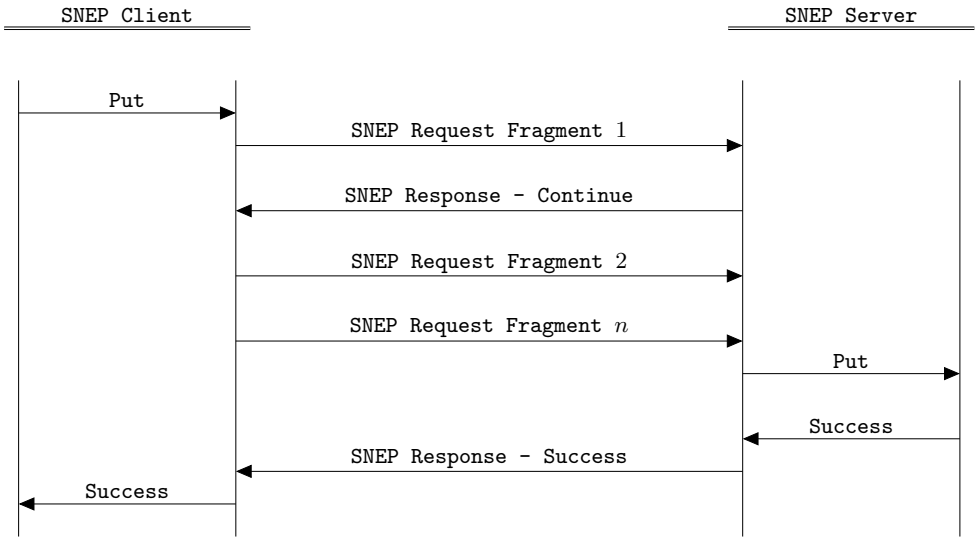


Figure 6: Example Put Request

## 2.2. SNEP Message Transport

Exchanging SNEP messages requires a reliable transport protocol capable of accepting service data units of 6 octets or more. Multiple simultaneous client–server conversations require that the transport protocol supports logically separated, simultaneous connections.

In the NFC Forum architecture, SNEP is a protocol layer on top of the Logical Link Control Protocol (LLCP). LLCP provides a connection-oriented transport service with sequenced and guaranteed delivery of service data units where each service data unit can accomodate at least 128 octets. An LLCP connection-oriented transport is established between a local and a remote LLC service access point and termed *data link connection*. Multiple, simultaneous data link connections are supported by LLCP.

SNEP messages SHALL be transmitted over LLCP data link connections using LLCP connection-oriented transport service. An active SNEP server SHALL await connect requests on an LLC service access point. The address of that service access point is specified in the connect request originating from the client (note that LLCP also allows a data link connection to be established by specifying a service name in a connect request sent to the service discovery component; this is logically equivalent). Once a data link connection is established, the client SHALL only send SNEP request messages and the server SHALL only return SNEP response messages on that data link connection. Finally, the client closes the data link connection when the conversation is finished.

The NFC Forum reserves LLC service access point address 4 and the service name “urn:nfc:sn:snep” for a default SNEP server defined in Section 6.

## 2.3. SNEP Versioning Policy

SNEP uses a major/minor numbering scheme to indicate different versions of the protocol. The version is indicated by a Version field in the first octet of each SNEP message and allows the sender to indicate its capacity for understanding further SNEP communication. The minor number is incremented when the changes made to the protocol add features that do not change the general message parsing algorithm, but may add to the message semantics and imply additional capabilities of the sender. The major number is incremented when the format of a message within the protocol is changed.

A SNEP server receiving a request message with a version number that differs from its own version number in the major part MAY return an *Unsupported Version* response. A SNEP server receiving a version number that differs from its own version number only in the minor part SHALL agree on the lower of the two minor version numbers and return the appropriate response.

# 3. Protocol Messages

SNEP messages consist of requests from client to server and responses from server to client.

## 3.1. SNEP Request Message

A SNEP request message is sent by a SNEP client to a SNEP server to invoke a specific method on the server.

The format of the SNEP request message SHALL be as shown in Figure 7.

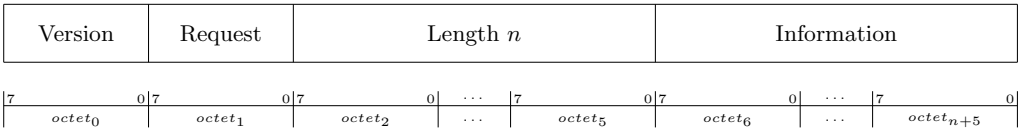


Figure 7: Request Message Format

### 3.1.1. Version Field

The Version field indicates the format of the SNEP protocol message and the sender's capacity for understanding further SNEP communication. The Version field is a single octet representing a structure of two 4-bit unsigned integers. The most significant 4 bits SHALL denote the major protocol version. The least significant 4 bits SHALL denote the minor protocol version. The major and minor protocol version SHALL be set as the major and minor release level of the SNEP specification. The Version field format is illustrated in Figure 8.

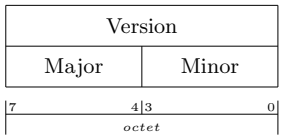


Figure 8: Version Field Format

### 3.1.2. Request Field

The Request field indicates the action to be performed by the SNEP server. The Request field is a single octet representing an 8-bit unsigned integer. Permissable values of the Request field are specified in Table 2 and further explained in Section 4.

Table 2: Request Field Values

| Code | Name | Description | Section |
|---|---|---|---|
| 00h | Continue | Send remaining fragments | 4.1 |
| 01h | Get | Return an NDEF message | 4.2 |
| 02h | Put | Accept an NDEF message | 4.3 |
| 03h - 7Eh | | Reserved for future use | |
| 7Fh | Reject | Do not send remaining fragments | 4.4 |
| 80h - FFh | | Reserved for response field values | |

### 3.1.3. Length Field

The Length field specifies the total length in octets of the Information field. The Length field is four octets representing a 32-bit unsigned integer. Transmission order SHALL be most significant octet first.

### 3.1.4. Information Field

The content of the Information field depends on the value of the Request field. The Information field is omitted if the value of the Length field is zero.

## 3.2. SNEP Response Message

A SNEP response message is sent by a SNEP server to a SNEP client to transmit the result of a method invocation that had been requested by the client.

The format of the SNEP response message SHALL be as shown in Figure 9.

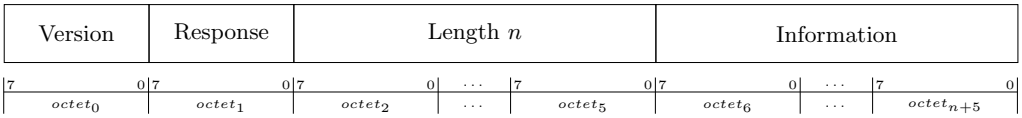


Figure 9: Response Message Format

### 3.2.1. Version Field

The Version field indicates the format of the SNEP protocol message and the sender's capacity for understanding further SNEP communication. The Version field is a single octet representing a structure of two 4-bit unsigned integers. The most significant 4 bits SHALL denote the major protocol version. The least significant 4 bits SHALL denote the minor protocol version. The major and minor protocol version SHALL be set as the major and minor release level of the SNEP specification. The Version field format is illustrated in Figure 8.

### 3.2.2. Response Field

The Response field indicates the result of the server's attempt to understand and satisfy the clients request. The Response field is a single octet representing an 8-bit unsigned integer. Possible values of the Response field are specified in Table 3 and further explained in Section 5.

Table 3: Response Field Values

| Code | Name | Description | Section |
|---|---|---|---|
| 00h - 7Fh | | Reserved for request field values | |
| 80h | Continue | Continue send remaining fragments | 5.1 |
| 81h | Success | Operation succeeded | 5.2 |
| 82h - BFh | | Reserved for future use | |
| C0h | Not Found | Resource not found | 5.3 |
| C1h | Excess Data | Resource exceeds data size limit | 5.4 |
| C2h | Bad Request | Malformed request not understood | 5.5 |
| C3h - DFh | | Reserved for future use | |
| E0h | Not Implemented | Unsupported functionality requested | 5.6 |
| E1h | Unsupported Version | Unsupported protocol version | 5.7 |
| E2h - FEh | | Reserved for future use | |
| FFh | Reject | Do not send remaining fragments | 5.8 |

### 3.2.3. Length Field

The Length field specifies the total length in octets of the Information field. The Length field is four octets representing a 32-bit unsigned integer. Transmission order SHALL be most significant octet first.

### 3.2.4. Information Field

The content of the Information field depends on the value of the Response field. The Information field is omitted if the value of the Length field is zero.

# 4. Request Codes

## 4.1. Continue

The client requests that the server send the remaining fragments of a fragmented SNEP response message. The client indicates its ability to receive the remaining fragments and successfully reassemble the complete SNEP response message. This request code SHALL only be sent after receipt of the first fragment of a fragmented SNEP response message. An information field SHALL NOT be transmitted with this request.

## 4.2. Get

The client requests that the server return an NDEF message designated by the NDEF message transmitted with the request. The information field of the request SHALL contain a 32-bit unsigned integer set to the maximum length of the NDEF message that the client will be able to accept in the response, followed by a single NDEF message that identifies the resource to be retrieved.

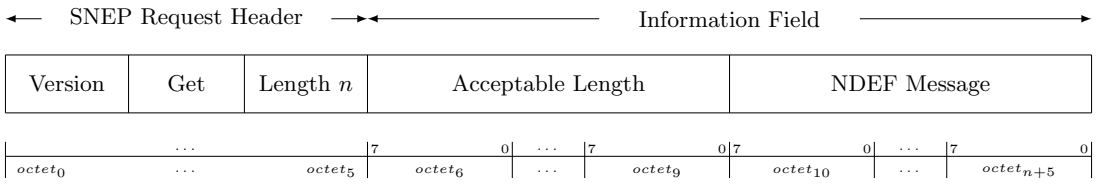


Figure 10: Contents of the Get Request Message

### 4.2.1. Acceptable Length Field

The Acceptable Length field specifies the maximum length in octets of the Information field in the server's response message. The Acceptable Length field is four octets representing a 32-bit unsigned integer. Transmission order SHALL be most significant octet first.

## 4.3. Put

The client requests that the server accept the NDEF message transmitted with the request. The information field of the request SHALL contain a single NDEF message. Acceptance of the NDEF message does not imply any specific processing.

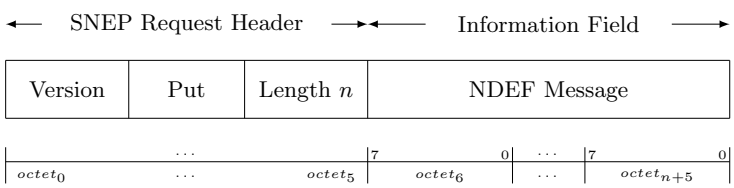


Figure 11: Contents of the Put Request Message

### 4.4. Reject

The client is unable to receive remaining fragments of a fragmented SNEP response message. The client is not expecting or willing to handle further fragments of this message, and receipt of further fragments might force the client to close the data link connection. This request code SHALL only be sent after receipt of the first fragment of a fragmented SNEP response message. An information field SHALL NOT be transmitted with this request.

# 5. Response Codes

## 5.1. Continue

The server received the first fragment of a fragmented SNEP request message and is able to receive the remaining fragments. The server indicates its ability to receive the remaining fragments and successfully reassemble the complete SNEP request message. This response code SHALL only be sent after receipt of the first fragment of a fragmented SNEP request message. An information field SHALL NOT be transmitted with this response.

## 5.2. Success

The request has succeeded. The information returned with the response is dependent on the request code used in the SNEP request message:

Get — Information field SHALL contain an NDEF message
Put — Information field SHALL NOT be present

## 5.3. Not Found

The server has not found anything matching the request. No indication is given of whether the condition is temporary or permanent. An information field SHALL NOT be transmitted with this response.

## 5.4. Excess Data

The server has found a resource matching the request, but returning the result would exceed the maximimum acceptable length the client has specified within the request message. An information field SHALL NOT be transmitted with this response.

## 5.5. Bad Request

The request could not be understood by the server due to malformed syntax. An information field SHALL NOT be transmitted with this response.

## 5.6. Not Implemented

The server does not support the functionality required to fulfill the request. This response is appropriate when the server does not recognize the request code. An information field SHALL NOT be transmitted with this response.

## 5.7. Unsupported Version

The server does not support, or refuses to support, the SNEP protocol version that was used in the request message. An information field SHALL NOT be transmitted with this response.

## 5.8. Reject

The server is unable to receive remaining fragments of a fragmented SNEP request message. The server is not expecting or willing to handle further fragments of this message, and receipt of further fragments might force the server to close the data link connection. This response code SHALL only be sent after receipt of the first fragment of a fragmented SNEP request message. An information field SHALL NOT be transmitted with this response.

# 6. NFC Forum Default SNEP Server

This section defines the mandatory capabilities and behavior of the NFC Forum default SNEP server. The service access point address of the default SNEP server is 4. The service name of the default SNEP server is “urn:nfc:sn:snep”.

## 6.1. Functional Description

The default SNEP server provides a logical inbox. A client connected to the default server can place NDEF messages into the inbox using Put request messages. The default server MUST accept Put request messages with an information field of up to 1024 octets. Acceptance of larger information fields is an implementation choice.

No specific processing of an accepted NDEF message is defined. Implementations might base further processing decisions on the received NDEF message type and might, for some NDEF message types, have means to process received NDEF messages in a local context.

The default server SHALL NOT accept Get requests. The appropriate response for a Get request message is Not Implemented.

# A. Revision History

The following table outlines the revision history of the Simple NDEF Exchange Protocol Specification.

Table 4: Revision History

| Document Name | Revision and Release Date | Status | Change Notice | Supersedes |
|---|---|---|---|---|
| Simple NDEF Exchange Protocol | Version 1.0, August 2011 | Final | | |