Android NDEF Push Protocol Specification

Version 1
2011-02-22

Table of Contents

1. Revision History

Version	Date	Description
1	2011-02-22	Initial revision.

2. Overview

The NDEF Push Protocol (NPP) is a simple protocol built on top of LLCP which is designed to push an NDEF message from one device to another. The procedure itself is one way and defines pushing NDEF messages from a client to a server. A device that supports NPP MUST always run an NPP server, and MAY run the NPP client procedure when it has an NDEF message available to push. This allows for bi-directional NDEF exchange between NPP devices.

3. Data Format

3.1. Protocol Versions

The protocol has major and minor versions. A change in the major version is not forwards compatible with older clients where a change in the minor version must be compatible with all clients using the same major version. A change in the minor version will signal new action codes. Servers must ignore unknown action codes they receive.

3.2. Header

The data sent from the client to the server MUST start with the following header:

Byte	Description	Expected Value
0	Protocol version	Most significant 4 bits are the major version. Least significant 4 bits are the minor version. This specification defines major version 0x0, minor version 0x1, for a value of 0x01.

Byte	Description	Expected Value
1...4	Number of NDEF entries	Big-endian unsigned integer. In version 0x01 of the specification, this value will always be 0x00, 0x00, 0x00, 0x01, because only 1 action code is supported.
5...n	NDEF entries	A sequence of valid NDEF Entries. See NDEF Entry for a description of the format.

3.3. NDEF Entry

Immediately following the header comes the sequence of messages with each message entry formatted as follows:

Byte	Description	Expected Value
0	Action code	See Action Codes. In version 0x01 of the specification, this value will always be 0x01, because only one action code is supported.
1...4	NDEF length	Big-endian unsigned integer. The number of bytes in the following NDEF message.
5...n	NDEF message	The byte stream for the NDEF message.

3.4. Action Codes

Currently the only supported action code is 0x01, which means the NDEF message MUST be processed as if it had been read from a passive tag. Only one NDEF entry with the action code of 0x01 is allowed. So for this minor version, only one NDEF entry is allowed.

Future minor versions of the specification may define additional action codes. Only the first NDEF entry with action code 0x01 MAY be processed and any further NDEF entries with action code 0x01 MUST be ignored.

All other action codes are reserved for future use and MUST be ignored by version 0x01 compliant NPP servers, and MUST not be sent by version 0x01 compliant NPP clients.

Action Code Value	Description	Minor version introduced
0x01	NDEF message MUST be processed as if it had been read from a passive tag.	1
All other values	reserved	N/A

3.5. Server

The server MUST register itself with the service name com.android.npp and be discoverable using the Service Discovery Protocol. Upon receiving a push from another device the server MUST process the received NDEF entries according to their action codes in the order they were received. The server knows the client has completed the transaction when the last complete NDEF entry has been received and the connection is closed.

An NPP 1.0 server MUST ignore the data if the major protocol version pushed by the client is not 0x0. An NPP server MUST accept all connections with the major protocol version 0x0 and MUST ignore all NDEF entries with unknown action codes.

3.6. Client

Upon the establishment of an NFC-DEP connection, the NPP client MUST immediately attempt the NPP push procedure if it has valid NDEF message(s) available. The procedure is:

1. Connect to LLCP socket with service name com.android.npp
2. Send the NPP Header following by NDEF Entries as defined in the Data Format section
3. Disconnect the LLCP socket

The client MUST consider a failure at any stage to indicate the NPP push failed. There is no provision for error acknowledgement or retry in the current protocol.

The client MUST only send action codes that are defined in the protocol version used by the client.