



NFCStack+Eva/Windows

Software Evaluation Kit User Guide

Release r06

Note

This product was developed for the purpose of communication. It is intended solely for our clients for integration into their own technical products after careful examination by experienced technical personnel for its suitability for the intended purpose. The product was not developed for or intended for use in any specific customer application. It may have to be adapted to the specific intended modalities of use or even replaced by other components in order to ensure flawless function in the respective areas of application. Performance data (range, power requirements, footprint, etc.) may depend on the operating environment, the area of application, the configuration, and method of control, as well as on other conditions of use; these may deviate from the technical specifications, the Design Guide specifications, or other product documentation. The actual performance characteristics can be determined only by measurements subsequent to integration in the target environment. Variations in the performance data of mass-produced devices may occur due to individual differences between such devices. Product samples were tested in a reference environment for compliance with the legal requirements applicable to the reference environment. No representation is made regarding the compliance with legal, regulatory, or other requirements in other environments. No representation can be made and no warranty can be assumed regarding the suitability of the product for a specific purpose. Stollmann reserves the right to make changes to the product without prior notice or to replace the product with a successor model. Of course, any changes to the product for which we have entered into a supply agreement with our customers will be made only if, and only to the extent that, such changes can reasonably be expected to be acceptable to our customers. No general commitment will be made regarding periods of availability; these must be subject to individual agreement. All agreements are subject to our Terms and Conditions for Deliveries and Payments, a copy of which is available from Stollmann.

Trademarks

The NFC word mark and logos are owned by the NFC Forum. Any use of such marks by Stollmann E+V GmbH is under license. Other trademarks and trade names are those of their respective owners.

Table of contents

1	Introduction	5
1.1	Supported environment	5
1.1.1	Operating system	5
1.1.2	NFC controller	5
1.1.3	NFC modes	6
1.1.4	NFC tags	6
1.2	System requirements	6
1.2.1	Windows	7
2	Installation & Setup	8
2.1	Windows	8
2.1.1	Arygon NFC/ Mifare Desktop Reader	8
2.1.2	Cardabra Stick Pro	8
2.1.3	NXP PN532/ PN544 development boards	9
2.1.4	STM ST21NFCA development board	9
2.1.5	TI TRF7970 development boards	9
2.1.6	Toshiba E2 development boards	10
2.1.7	NCI PN512 Reference development boards	10
3	NFaCe+SPA Evaluation Application (NFCPlayer)	11
3.1	Trace Flags Configuration	11
3.2	Controller Configuration	12
3.3	Stack Startup	13
3.4	Device Detection	13
3.5	Log	14

3.6	NFC Operation Modes	14
3.6.1	Read/ write NFC tags (R/W)	15
3.6.1.1	NDEF	15
3.6.1.2	Legacy	17
3.6.1.3	MIFARE Standard	19
3.6.2	Peer-to-peer (P2P)	20
3.6.2.1	Stollmann P2P API	21
3.6.2.2	SNEP	22
3.6.3	Card Emulation (CE)	23
4	History	26

1 Introduction

Stollmann NFCStack+Eva/Windows will introduce you to the world of the Stollmann NFC Software Stack and Programming Interface – the ultimate solution for developing NFC enabled software and modules. This package contains:

- NFC Player
- User Guide

1.1 Supported environment

1.1.1 Operating system

This package was designed to be cross-platform compatible. Therefore, various Windows and Linux distributions are supported:

- Windows XP SP3 (32-bit)
- Windows Vista (32-bit)
- Windows 7 (32/ 64-bit)
- Ubuntu 12.04 LTS (32/ 64-bit)

Note: This package has been tested with the listed operating systems but may also work on related systems. However, functionality with other than the listed systems is neither guaranteed nor supported by Stollmann.

1.1.2 NFC controller

This package supports the following NFC controllers:

- Arygon NFC/ Mifare Desktop Reader
- Cardabra Stick Pro
- NXP PN532/ PN544 development boards
- ST Micro ST21NFCA development boards

1.1.3 NFC modes

This package supports the following NFC modes:

- Read/ write NFC tag (R/W)
- Peer-to-Peer including SNEP (P2P)
- Host-based card emulation (CE)
- Secure elements (SE) (not supported for all releases)

1.1.4 NFC tags

This package supports the following NFC tags:

- ISO 14443A
- ISO 14443B
- ISO 15693 (not supported by all NFC controllers)
- NXP Mifare Standard 1K/ 4K/ S20 (not supported by all NFC controllers)
- NXP Mifare Ultralight
- NXP Mifare DESFire
- Innovision Jewel (not supported by all NFC controllers)
- Sony FeliCa/ FeliCa light

1.2 System requirements

- One of the supported operating systems, listed in section [1.1.1](#)
- One of the NFC devices, listed in section [1.1.2](#)

Note: This package requires one of the previously listed operating systems but may also work on related systems. However, functionality with other than the listed systems is neither guaranteed nor supported by Stollmann.

1.2.1 Windows

The Windows version of this package requires the installation of an NFC controller as described in section [2.1](#).

2 Installation & Setup

2.1 Windows

Execute "Setup.exe" from the root of the installation package. Please refer to "QuickInstallGuide.htm" located in the same folder. At the beginning of the installation you will be able to install drivers for some NFC reader (see section 2.1.1 and section 2.1.2 for details).

Prior to running NFaCe+SPA Evaluation Application, please ensure one of the compatible NFC controller has been correctly installed and connected.

2.1.1 Arygon NFC/ Mifare Desktop Reader

- Mark the checkbox 'Silabs driver' at the beginning of the installation if the corresponding driver should be installed.
- Follow the instructions of the Arygon NFC reader driver installation routine.
- Connect the Arygon NFC reader to a free USB port on the PC.
- Run NFaCe+SPA Evaluation Application from "Start" → "Programs" → "Stollmann NFCStack+Eva/Windows R[x] → NFCPlayer".
- The configuration parameters for this device are:
 - Controller: "PN532"
 - Port: "COMX" where X is the virtual COM port that the device is connected to

2.1.2 Cardabra Stick Pro

- Mark the checkbox 'Cardolution driver' at the beginning of the installation if the corresponding driver should be installed.
- Follow the instructions of the Cardolution driver installation routine.
- Connect the Cardabra Stick Pro NFC reader to a free USB port on the PC.
- Run NFaCe+SPA Evaluation Application from "Start" → "Programs" → "Stollmann NFCStack+Eva/Windows R[x] → NFCPlayer".
- The configuration parameters for this device are:

- Controller: "PN532"
- Port: "COMX" where X is the virtual COM port that the device is connected to

2.1.3 NXP PN532/ PN544 development boards

- Connect the development board to a free COM (RS232) or USB port on the PC, depending on your NFC controller board.
- Run NFaCe+SPA Evaluation Application from "Start" → "Programs" → "Stollmann NFCStack+Eva/Windows R[x] → NFCPlayer".
- The configuration parameters for this device are:
 - Controller: "PN532" or "ETSIHCI (PN544)"
 - Bus: "SERIAL" (in case of the PN532, this parameter is optional)
 - Port: "COMX" where X is the virtual COM port that the device is connected to

2.1.4 STM ST21NFCA development board

This device requires the installation of the Total Phase Aardvark I2C/ USB adapter.

- Install the Total Phase Aardvark I2C/ USB adapter drivers.
- Connect the development board to the Aardvark adapter and connect the Aardvark adapter to a free USB port on the PC.
- Run NFaCe+SPA Evaluation Application from "Start" → "Programs" → "Stollmann NFCStack+Eva/Windows R[x] → NFCPlayer".
- The configuration parameters for this device are:
 - Controller: "ETSIHCI (ST21NFCA)"
 - Bus: "I2C"

2.1.5 TI TRF7970 development boards

- Connect the development board to a free USB port on the PC.
- Run NFaCe+SPA Evaluation Application from "Start" → "Programs" → "Stollmann NFCStack+Eva/Windows R[x] → NFCPlayer".

- The configuration parameters for this device are:
 - Controller: "NCI (TRF7970)"
 - Bus: "SERIAL"
 - Port: "COMX" where X is the virtual COM port that the device is connected to

2.1.6 Toshiba E2 development boards

- Connect the development board to a free USB port on the PC.
- Run NFaCe+SPA Evaluation Application from "Start" → "Programs" → "Stollmann NFCStack+Eva/Windows R[x] → NFCPlayer".
- The configuration parameters for this device are:
 - Controller: "NCI (Toshiba E2)"
 - Bus: "SERIAL"
 - Port: "COMX" where X is the virtual COM port that the device is connected to

2.1.7 NCI PN512 Reference development boards

- Connect the development board to a free USB port on the PC.
- Run NFaCe+SPA Evaluation Application from "Start" → "Programs" → "Stollmann NFCStack+Eva/Windows R[x] → NFCPlayer".
- The configuration parameters for this device are:
 - Controller: "NCI (GENERIC)"
 - Bus: "SERIAL"
 - Port: "COMX" where X is the virtual COM port that the device is connected to

3 NFaCe+SPA Evaluation Application (NFCPlayer)

The Stollmann NFaCe+SPA Evaluation Application (NFCPlayer, figure 1) demonstrates the functionality of the Stollmann NFCStack+ (NFaCe+SPA) on an application user interface. It enables the user to perform reading, writing and formatting operations on NFC tags (R/W), host-based card emulation (CE), peer-to-peer data communication (P2P), and secure elements (SE). For further details, see section 3.6.

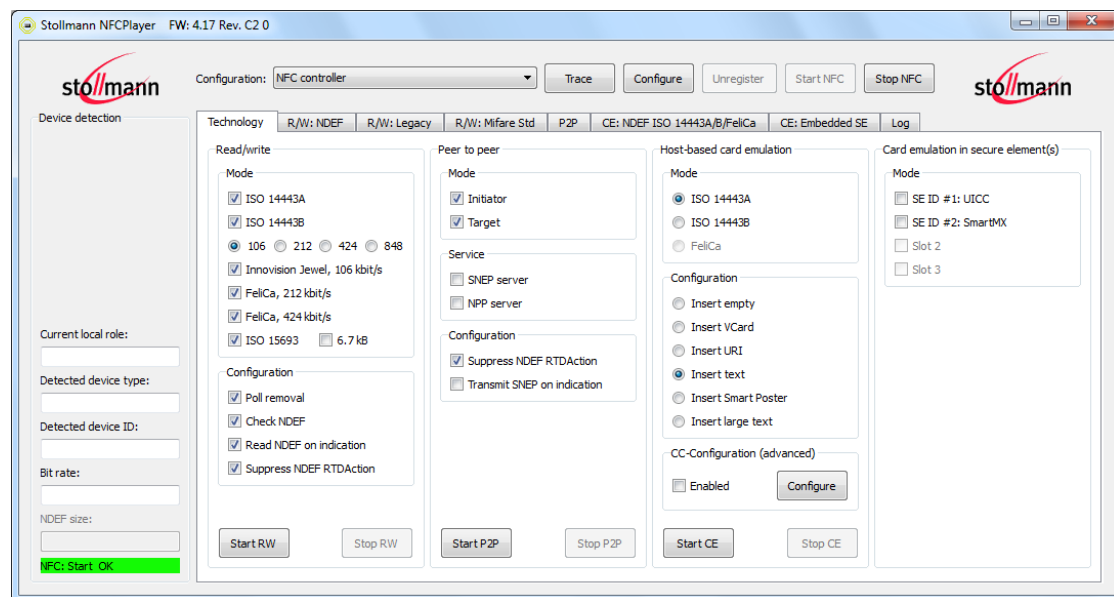


Figure 1: NFaCe+SPA Evaluation Application: Main dialog

3.1 Trace Flags Configuration

The "Trace" button opens a configuration dialog (figure 2) for trace output messages that can be logged with debugging tools like **DebugView**. In this context, the user configures the global setting for the trace flags. By default, the recommended setup for the trace output is preconfigured. This setting will only take effect when no trace output has been configured in controller context (i.e. (controller-configuration-specific) local settings). Local settings will take priority over global settings. It is possible to change the trace output on runtime.

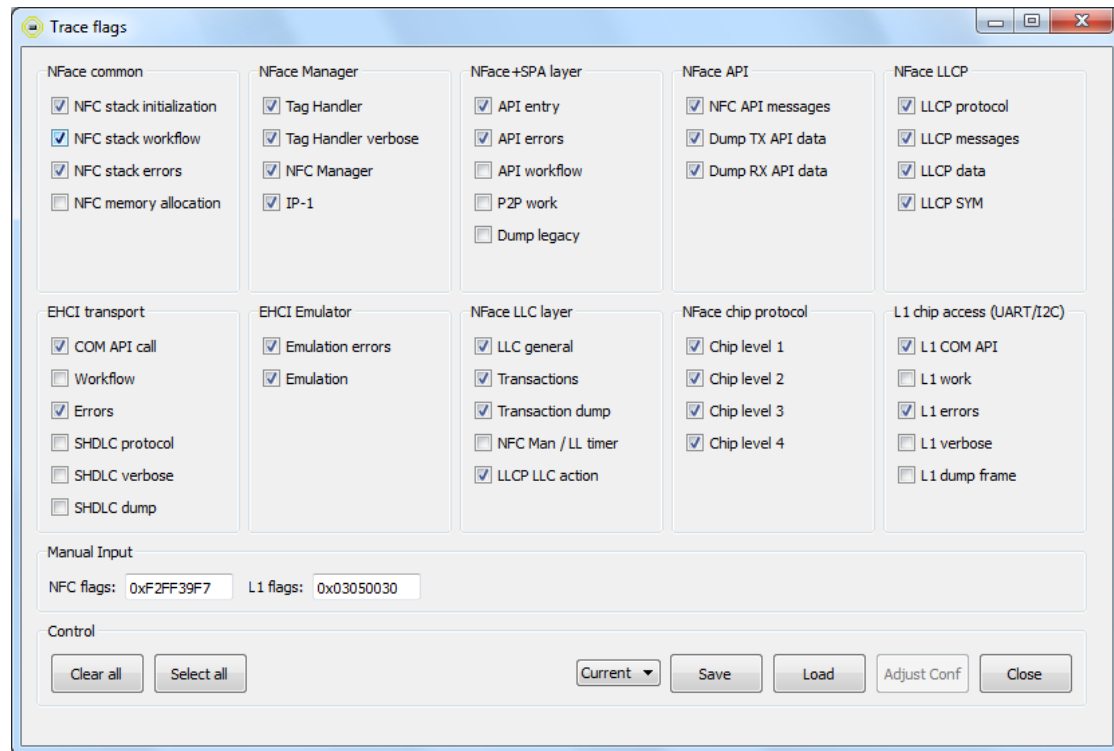


Figure 2: NFaCe+SPA Evaluation Application: Trace dialog

In this dialog, the user may change any flag of the trace output manually by either (un-)checking boxes or editing the text boxes "NFC flags" and "L1 flags". After manipulation, the user needs to save the settings to "Current" in order to take effect. For convenience, the user may choose to save other configurations to slots in the drop-down list and load them whenever they are needed.

3.2 Controller Configuration

The "Configure" button opens a dialog for configuration of the NFC controller (figure 3). Here, the user selects the "Controller", "Bus", and "Port" from the corresponding drop-down list. The field "Configuration" will be automatically adjusted accordingly. Its content represents the initialization string that is passed to the Stollmann NFC API. It should only be altered manually with a good understanding of the API. Similarly to the main dialog, the "Trace flags" button opens a configuration dialog for the trace output (see 3.1). In this context though, the user configures (controller-configuration-specific) local settings for the trace output. These will take priority over global settings. To apply changes of the trace output settings to the configuration string the user must click the "Adjust Conf" button.

The GUI will highlight all connected COM ports.

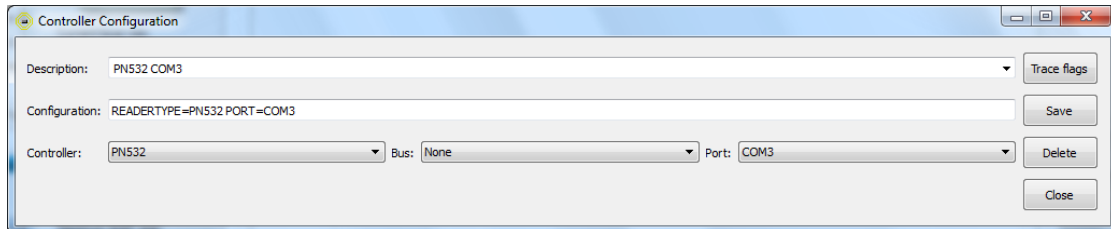


Figure 3: NFaCe+SPA Evaluation Application: Controller dialog

3.3 Stack Startup

After configuration of the trace output and the NFC controller (see section 3.1 and 3.2 for details), the NFC stack may be started:

- "Start NFC" starts the Stollmann NFC stack.
- "Stop NFC" stops the NFC stack.
- "Unregister" stops a current listener. Dependent on the currently active tab it will unregister the appropriate listener.

3.4 Device Detection

Information extracted from the last detected device is shown on the left of the panel. This information will be updated with every NFC discovery and removal event. The provided information comprises:

- "Current local role" represents the engaged listener modes of the local device.
- "Detected device type" shows the detected device type.
- "Detected device ID" shows the detected device ID.
- "Bit rate" shows the data rate which is currently used for communication.
- "NDEF size" shows the size of the current NDEF content and the capacity of detected tag. This field is only valid when an NDEF-formatted tag is present.
- "Status" shows the status of the last user-triggered NFC operation. Such operations are triggered by using UI-buttons or discovery/removal events.

3.5 Log

The "Log"-tab shows the log output of the NFCPlayer (figure 4). This is a very basic trace output and is supposed to give a brief overview. It does not substitute the use of a debugging tool such as **DebugView**.

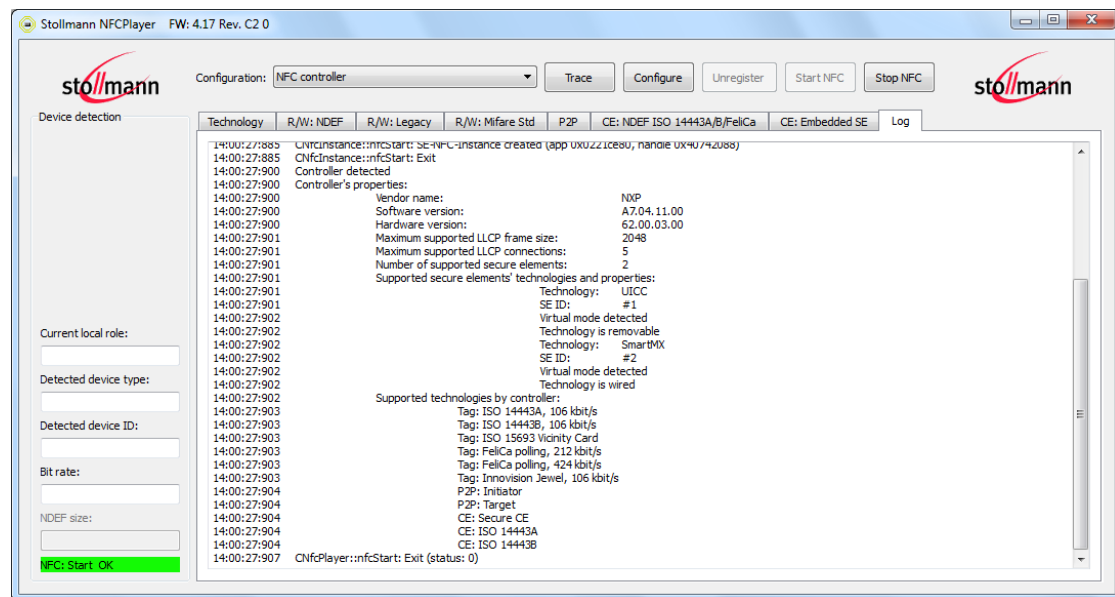


Figure 4: NFaCe+SPA Evaluation Application: Log tab

3.6 NFC Operation Modes

The NFCPlayer supports four NFC operation modes:

- Read/ write NFC tags (R/W)
- Peer-to-peer (P2P)
- Host-based card emulation (CE)
- Secure elements (SE)

3.6.1 Read/ write NFC tags (R/W)

The NFCPlayer supports three R/W modes that operate on different abstraction levels on the NFC tags:

- NDEF
- Legacy
- MIFARE Standard

NDEF is a standard defined by the NFC Forum that specifies a commonly used data format. In contrast, legacy "R/W" uses specific protocols that are only supported by some tags. The default and recommended settings for the RW-mode are preconfigured but the user may choose to adjust these settings:

- Configure tag technologies the NFCPlayer should react to by (un-)checking the corresponding options.
- Refine the behavior of the NFCPlayer:
 - "Poll removal" enables recognition of tag removal. If this option is deselected, only the first tag entering the field will be detected. For further detections, the listener must be restarted.
 - "Check NDEF" enables NDEF check on tag indication.
 - "Read NDEF on indication" enables automatic reading of NDEF content after tag indication.
 - "Suppress NDEF RTDAction" disables automatic execution of content triggered by an NDEF RTDAction record.

After finishing the configuration, the tag listener may be started:

- "Start RW" registers a RW-listener to the stack.
- "Stop RW" unregisters the RW-listener.

3.6.1.1 NDEF

Whenever an NFC tag with NDEF content is discovered, the "R/W: NDEF"-tab is enabled and automatically selected (figure 5).

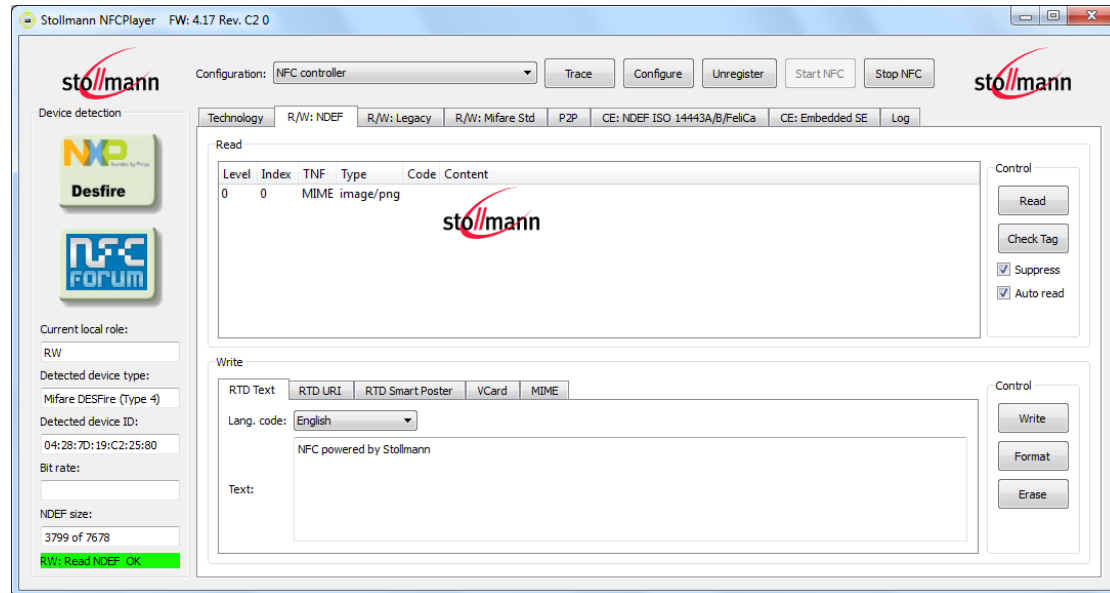


Figure 5: NFAcE+SPA Evaluation Application: Read/ write NDEF

In this tab, the reading section in the upper part provides the following elements:

- "Read" starts a read transaction requesting the device's first NFC forum compliant message. Most tags contain only one message. Once a message has been received, the payload is passed to the Stollmann NDEF library in order to parse its contents for valid NDEF formatted records. All successfully decoded records are displayed in the NDEF records list. For further information about the Stollmann NDEF library, please refer to the corresponding documentation.
- "Check tag" will query if the last detected tag is still in proximity.
- "Suppress" disables automatic execution of content triggered by an NDEF RTDAction record. This flag directly corresponds to "Suppress NDEF RTDAction" on the "Technology"-tab.
- "Auto read" enables automatic reading of the tag's first NDEF message on device indication. This flag directly corresponds to "Read on device indication" on the "Technology"-tab.

The writing section provides facilities to write four different types of NDEF-formatted messages to the connected device:

- "Write" writes user-defined data to the first storage location on the connected tag. Depending on the currently selected sub-tab the user may choose which of the following message types will be written to the tag. If the message length

exceeds the tag's memory capacity, the write transaction will fail and NFaCe+SPA will report an error code. In this case the tag will contain no data.

- RTD UTF-8-encoded text message with the selected language and text.
 - RTD URI message with the specified abbreviation of the leading standardized URI type string.
 - RTD SmartPoster message containing the following nested records:
 - * RTD UTF-8-encoded text message with the selected language code and text.
 - * RTD URI message with the specified abbreviation of the leading standardized URI type string.
 - * RTD Action message of type "Do" causing the receiving device to start its browser with the transmitted URI.
 - VCard message constructed with user-defined data taken from this tab.
 - RTD MIME image from a user-defined path.
- "Erase" erases the entire tag. If successful, the connected tag will be re-detected in legacy mode.
- "Format" creates a new empty NDEF container. If successful, the connected tag will be re-detected in NDEF mode.

3.6.1.2 Legacy

Whenever a legacy tag is discovered, the "R/W: Legacy"-tab is enabled and automatically selected (figure 6).

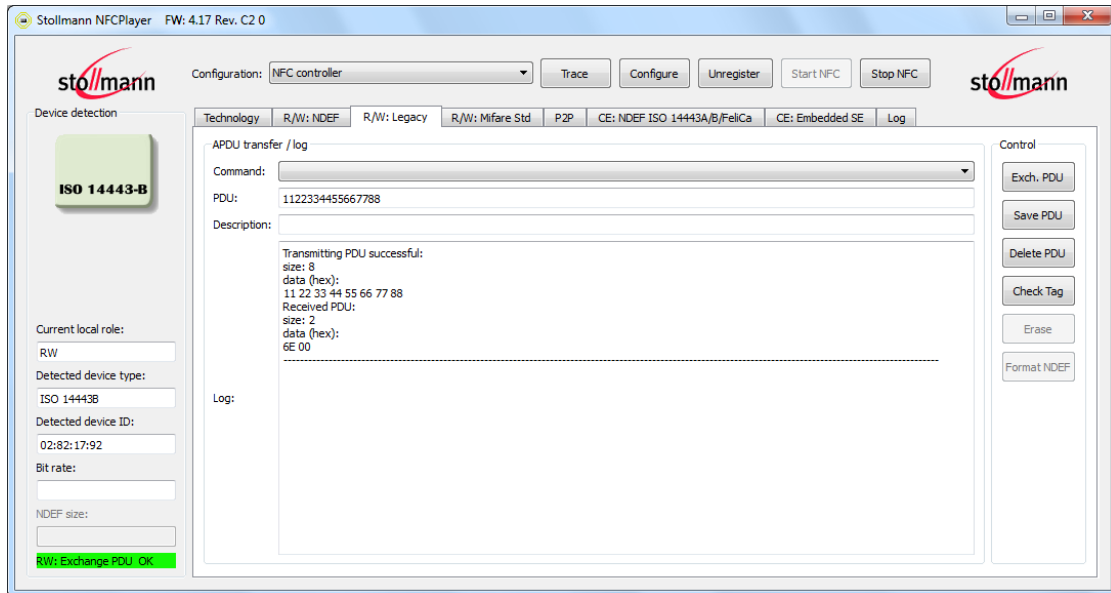


Figure 6: NFaCe+SPA Evaluation Application: Read/ write Legacy

Using the drop-down list the user may choose to select a predefined PDU or create a PDU by filling the fields "PDU" and "Description". The "PDU"-field holds the data to be sent to the tag and must be in hexadecimal format (two hex digits per data byte). Any "0x" and empty spaces in the input data are optional and omitted by the application. Other than being unique in this context the "Description" is arbitrary. Internally, this field is used as a key. Using the same key again will cause the first entry to be overwritten. On this tab the following controls are available:

- "Save PDU" saves a user-defined entry (the combination of "PDU" and "Description"). The item is added to the drop down list.
- "Delete PDU" removes an entry from the drop down list.
- "Exchange PDU" transmits the current "PDU" message to the connected tag. Exchanged data is displayed in the "Log"-field.
- "Check tag" will query if the last detected tag is still in proximity.
- "Erase" erases the entire tag completely. If successful, the connected tag will be re-detected in legacy mode.
- "Format" creates a new empty NDEF container. If successful, the connected tag will be re-detected in NDEF mode.

3.6.1.3 MIFARE Standard

Whenever an NFC tag is discovered which is a not NDEF-formatted MIFARE Standard, the "R/W: Mifare Std"-tab is enabled and automatically selected (figure 7).

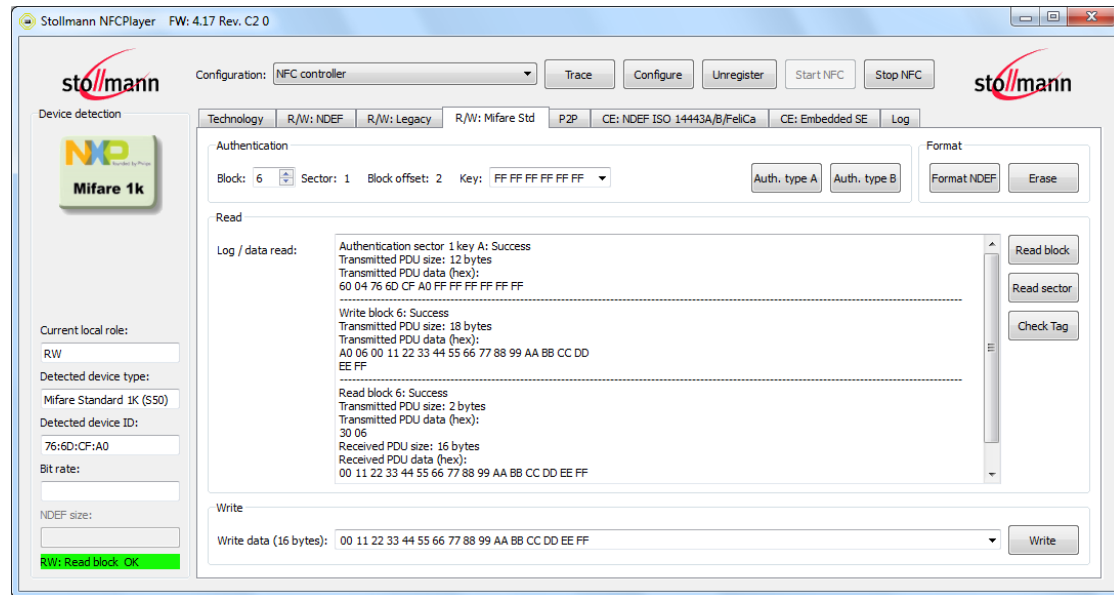


Figure 7: NFaCe+SPA Evaluation Application: Read/ write MIFARE Standard

- "Authenticate A/B" performs the corresponding MIFARE authentication. Internally an authentication command PDU is applying the user-configurable "Block index" and "Key". The fields "Sector" and "Block offset" are automatically updated according to "Block index".
- "Read block" reads the block from the tag corresponding to the selected "Block index".
- "Read sector" reads the entire "Sector" from the tag to which the "Block index" belongs to.
- "Check tag" will query if the last detected tag is still in proximity.
- "Write" writes 16 bytes to the block on the tag corresponding to the selected "Block index". For input, hexadecimal format is required (two hex digits per data byte). Any "0x" and empty spaces in the input data are optional and omitted by the application. To successfully write data to the tag, it is necessary to send 16 data bytes (one block).
- "Erase" erases the entire tag completely. If successful, the connected tag will be re-detected in legacy mode.

- "Format" creates a new empty NDEF container. If successful, the connected tag will be re-detected in NDEF mode.

Any received or sent data will be displayed in "Log / data read".

3.6.2 Peer-to-peer (P2P)

The NFCPlayer supports four different P2P protocols/technologies:

- Stollmann P2P API
- SNEP
- NPP*

* Note: NFCPlayer only provides a basic NPP server which will create log output that indicates that a discovered device used the NPP protocol.

The default and recommended settings for the P2P-mode are preconfigured but the user may choose to adjust these settings:

- Configure the P2P mode ("Initiator", "Target", or both) by (un-)checking the corresponding options.
- Start a SNEP or NPP server. Starting or stopping these services is also valid when an P2P-listener is already registered.
- Refine the behavior of the NFCPlayer:
 - "Suppress NDEF RTDAction" disables automatic execution of content triggered by an NDEF RTDAction record.
 - "Transmit SNEP on indication" enables automatic transmission of SNEP-content triggered by a P2P device indication.

After finishing the configuration the tag listener may be started:

- "Start P2P" registers a P2P-listener to the stack and the application will be able to detect other P2P devices.
- "Stop P2P" unregisters the P2P-listener.

Whenever another P2P-node is discovered, the "P2P"-tab is enabled and automatically selected. That tab comprises two sub-tabs:

- "Stollmann API"
- "SNEP"

3.6.2.1 Stollmann P2P API

Operating P2P communication based on LLCP requires two instances running NFCPlayer. It is recommended to use two separate systems. At least one of the applications must be configured as "Target" and the other as "Initiator". The NFCPlayer allows to use two LLCP connections (SAP 15 and 14) simultaneously.

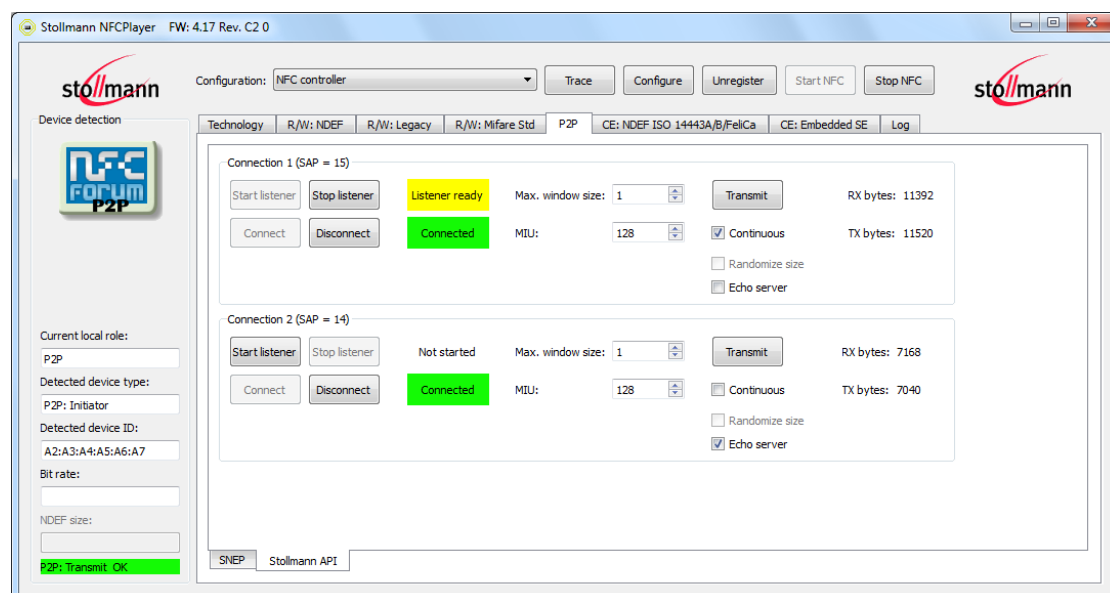


Figure 8: NFaCe+SPA Evaluation Application: Stollmann API

The "Stollmann API"-tab (figure 8) provides various controls for establishing and closing connection and transmitting data to another P2P-node:

- "Start listener" opens a LLCP server listening for incoming LLCP connections. On success a listener is started that waits for incoming connect requests.
- "Stop listener" closes the instance from listening to connect requests. If a connection is already established it will not be disconnected.
- "Connect" sends a connect request to a LLCP server. The server on the other system must have been started prior to this operation. Both systems are required to use the same "Max. window size" and "MIU", see NFC Forum LLCP specification for further details. On success, a connection between the two LLCP instances is established. Once a connection has been set up, both LLCP

instances have the same privileges on that connection. This means both instances are allowed to transmit data and emit disconnect requests.

- "Disconnect" sends a disconnect request to the other communication node.
- "Transmit" sends predefined data via LLCP to the other communication node. The transmitted and received data can be monitored on both applications. It is possible to modify the transmission of data to "Continuous", resulting in a continuous data flow from one side to the other without further user interaction. The receiver may also be modified to an "Echo server" that echoes every PDU back to the sender.

3.6.2.2 SNEP

Operating SNEP communication based on LLCP requires two systems running either two instances of NFCPlayer or one NFCPlayer and a SNEP-compliant smartphone. When running two NFCPlayer it is recommended to use two separate systems. At least one of the applications must be configured as "Target" and the other as "Initiator". In order to receive data via SNEP, it is required to start a SNEP server.

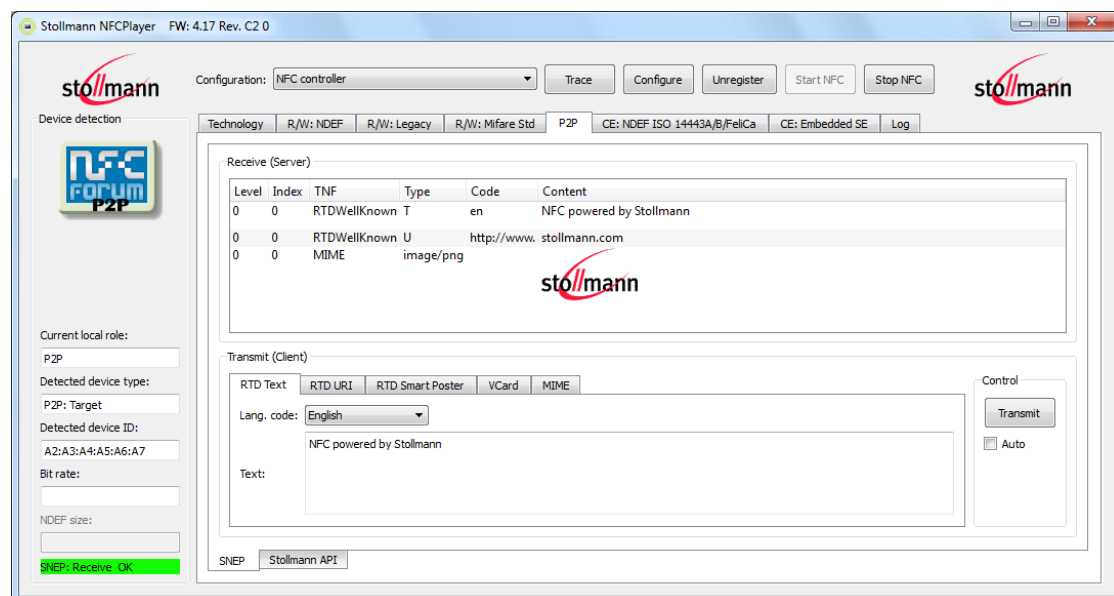


Figure 9: NFaCe+SPA Evaluation Application: SNEP

The "SNEP"-tab (figure 9) provides facilities to manage and monitor SNEP-traffic. The transmitting section offers to send four different types of NDEF RTD-well-known formatted messages to the connected device, depending on the selected sub-tab:

- "Transmit" sends user-defined NDEF-data via SNEP to the other communication node.
- "Auto" enables automatic transmission of SNEP-content triggered by a P2P device indication. The user may choose the following content NDEF-types:
 - RTD UTF-8-encoded text message with the selected language and text.
 - RTD URI message with the specified abbreviation of the leading standardized URI type string.
 - RTD SmartPoster message containing the following nested records:
 - * RTD UTF-8-encoded text message with the selected language code and text.
 - * RTD URI message with the specified abbreviation of the leading standardized URI type string.
 - * RTD Action message of type "Do" causing the receiving device to start its browser with the transmitted URI.
 - VCard message constructed with user-defined data taken from this tab.
 - RTD MIME image from a user-defined path.

3.6.3 Card Emulation (CE)

In this mode, one of the supported tag types is emulated. Operating this mode requires two systems running either two instances of NFCPlayer or one NFCPlayer and a device capable of reading NFC tags. The NFCPlayer supports three different CE technologies:

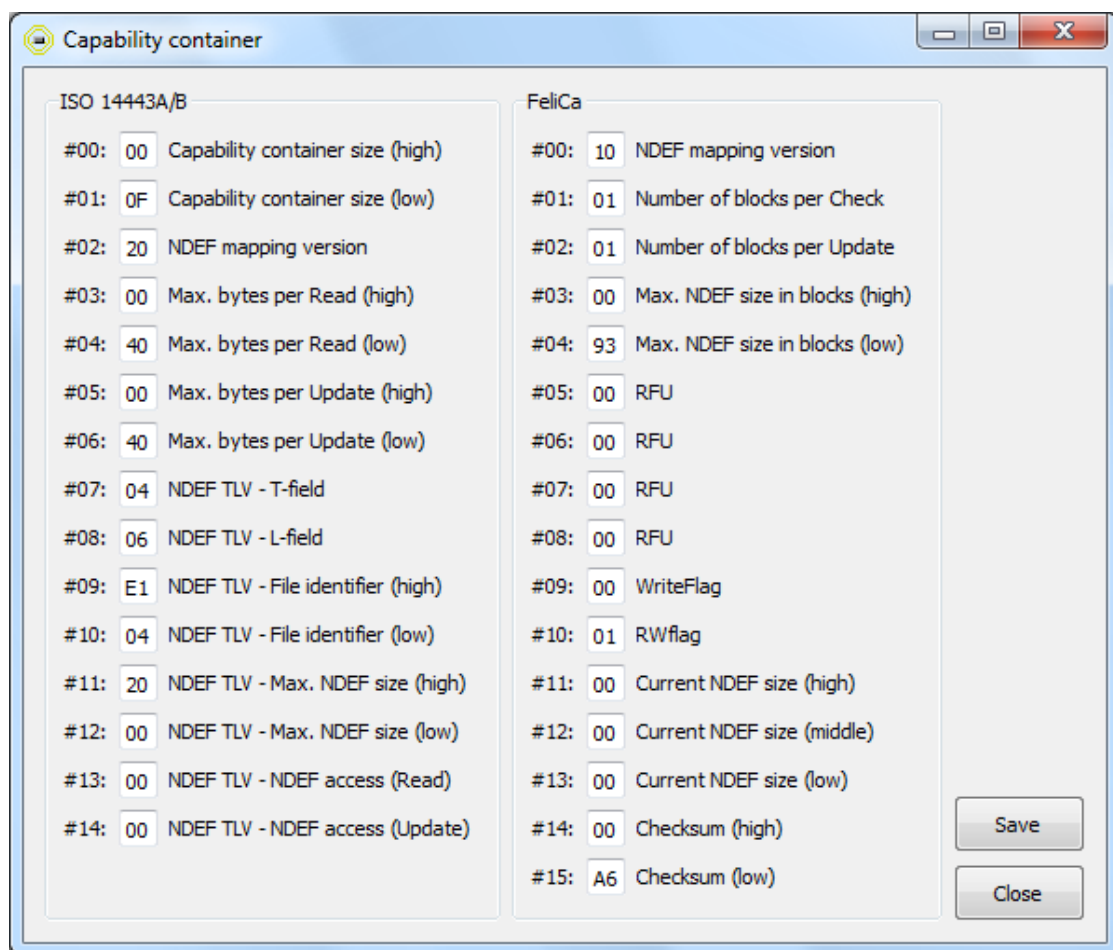
- ISO 14443A
- ISO 14443B
- FeliCa

The configuration section provides six different options to initialize the emulated tag:

- "Insert empty" creates an empty NDEF container.
- "Insert VCard" creates a generic VCard message.
- "Insert URI" creates a generic RTD URI message.
- "Insert text" creates a generic RTD UTF-8-encoded text message.

- "Insert Smart poster" creates a generic RTD Smart Poster message.
- "Insert large text" creates a generic large text.

As an advanced option the user may choose to configure the capability container of the card emulation by clicking "Configure". This will open a new dialog where every byte of the container may be changed (figure 10). Fields such as "Checksum" or "Max. NDEF size" will be changed automatically by the program when the CE is set up.



ISO 14443A/B	FeliCa
#00: 00 Capability container size (high)	#00: 10 NDEF mapping version
#01: 0F Capability container size (low)	#01: 01 Number of blocks per Check
#02: 20 NDEF mapping version	#02: 01 Number of blocks per Update
#03: 00 Max. bytes per Read (high)	#03: 00 Max. NDEF size in blocks (high)
#04: 40 Max. bytes per Read (low)	#04: 93 Max. NDEF size in blocks (low)
#05: 00 Max. bytes per Update (high)	#05: 00 RFU
#06: 40 Max. bytes per Update (low)	#06: 00 RFU
#07: 04 NDEF TLV - T-field	#07: 00 RFU
#08: 06 NDEF TLV - L-field	#08: 00 RFU
#09: E1 NDEF TLV - File identifier (high)	#09: 00 WriteFlag
#10: 04 NDEF TLV - File identifier (low)	#10: 01 RWflag
#11: 20 NDEF TLV - Max. NDEF size (high)	#11: 00 Current NDEF size (high)
#12: 00 NDEF TLV - Max. NDEF size (low)	#12: 00 Current NDEF size (middle)
#13: 00 NDEF TLV - NDEF access (Read)	#13: 00 Current NDEF size (low)
#14: 00 NDEF TLV - NDEF access (Update)	#14: 00 Checksum (high)
	#15: A6 Checksum (low)

Figure 10: NFaCe+SPA Evaluation Application: Capability container dialog

After finishing the configuration, the card emulation may be started:

- "Start CE" registers a CE-listener.
- "Stop CE" unregisters the CE-listener.

After setup of an CE-instance, an external NFC controller placed within the local controller's field range will be able to detect the emulated tag. When that happens the "CE: NDEF ISO 14443A/B/FeliCa"-tab is enabled and automatically selected (figure 11). Through this tab the communication between the emulated tag and the external controller is logged and can be monitored. Additionally, this tab provides functionality to load and modify the current NDEF content of the card emulation:

- "Load" extracts the current NDEF content and displays it in the GUI depending on type.
- "Save" reads the data from the form and replaces the current content with the new data.

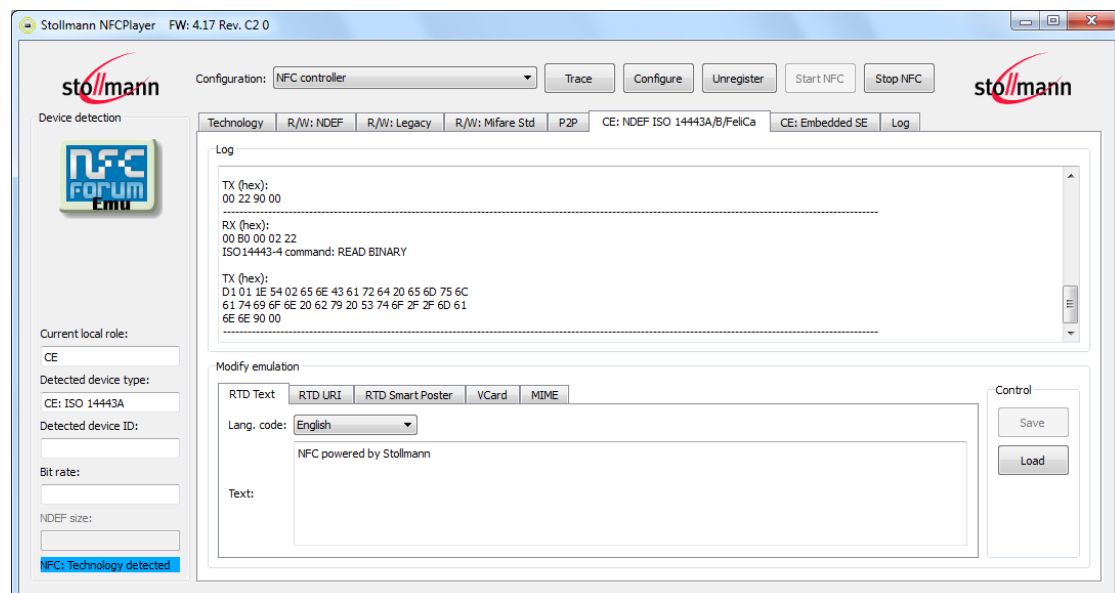


Figure 11: NFaCe+SPA Evaluation Application: Card emulation

4 History

Version	Release Date	By	Change description
R04d01	16.11.2012	DS	First release
R04d02	28.03.2013	DS	Some updates
R05d01	11.06.2013	DS	Added new chips and updated images
R05d02	27.07.2013	DS	Added new chips
R05d03	22.08.2013	CK	Added Cardabra Stick Pro
R05d04	04.09.2013	DS	Updated CE-Tab
R06	14.05.2014	CK	Changed productnames

Stollmann Entwicklungs- und Vertriebs-GmbH
Mendelssohnstraße 15 D
22761 Hamburg
Germany

Phone: +49 (0)40 890 88-0
Fax: +49 (0)40 890 88-444
E-mail: info@stollmann.de
www.stollmann.de