OMNIKEY 5422 and 5122

Software Developer Guide

PLT-03296, Rev A.3

February 2018

hidglobal.com

Copyright

Trademarks

HID GLOBAL, HID, the HID Brick logo, the Chain Design, ICLASS, ICLASS SE, SEOS and OMNIKEY are the trademarks or registered trademarks of HID Global, ASSA ABLOY AB, or its affiliate(s) in the US and other countries and may not be used without permission. All other trademarks, service marks, and product or service names are trademarks or registered trademarks of their respective owners.

MIFARE, MIFARE Classic, MIFARE DESFire, MIFARE DESFire EV1, MIFARE PLUS and MIFARE Ultralight are registered trademarks of NXP B.V. and are used under license.

Revision History

Date	Description	Version
February 2018	Updated Section 2.2 and Section 4.1.	A.3
January 2018	Added OMNIKEY 5122.	A.2
December 2017	Added Section 1.5.	A.1
May 2017	Initial release.	A.0

Contacts

For additional offices around the world, see www.hidglobal.com corporate offices.

Americas and Corporate

611 Center Ridge Drive
Austin, TX 78753
USA
Phone: 866 607 7339
Fax: 949 732 2120

Asia Pacific

19/F 625 King's Road
North Point, Island East
Hong Kong
Phone: 852 3160 9833
Fax: 852 3160 4809

Europe, Middle East and Africa (EMEA)

Haverhill Business Park Phoenix Road
Haverhill, Suffolk CB9 7AE
England
Phone: 44 (0) 1440 711 822
Fax: 44 (0) 1440 714 840

Brazil

Condomínio Business Center
Av. Ermano Marchetti, 1435
Galpão A2 - CEP 05038-001
Lapa - São Paulo / SP
Brazil
Phone: +55 11 5514-7100

HID Global Technical Support: www.hidglobal.com/support

Contents

Chapter 1

Introduction

1.1 Product Description

OMNIKEY® 5422 and 5122 Smart Card Readers open new market opportunities for system integrators seeking simple reader integration and development using standard interfaces, such as CCID (Circuit Card Interface Device). This reader works without needing to install or maintain drivers, eliminating complex software lifecycle management issues in the field and accelerating introduction into the market. Only an operating system driver, for example, Microsoft CCID driver is necessary.

The OMNIKEY 5422 and 5122 readers include support for the common high frequency card technologies, including ISO/IEC 14443 A/B, iCLASS®, MIFARE® and ISO/IEC 7816-3 card technology including synchronous cards support.

Low power mode makes it ideal solution for portable devices like tablets or mobile phones.

Thanks to ability to upgrade device firmware, it is possible to add support for new card technologies in the future.

1.2 Key Features

- **CCID Support –** Removes the requirement to install drivers on standard operating systems to fully support capabilities of the reader board
- **High frequency card technologies –** Supports common high frequency card technologies, including ISO/IEC 14443 A/B, iCLASS® 15693 and MIFARE
- **ISO/IEC 7816-3 -** Supports the common contact card technology including synchronous cards support
- **Rapid and Easy Integration –** No special driver installation is required
- **Advanced Power Management –** Supports Low Power modes specified by USB:
  - Allows the host device to turn off the reader to save power (while the reader is still able to detect cards, with reduced power)
  - Allows the reader to wake up the host device

1.3 Reference Documents

Document Number	Description
USB 2.0 Specification	Universal Serial Bus Revision 2.0 specification provides the technical details to understand USB requirements and design USB compatible products. Refer to http://www.usb.org/developers/docs/usb20_docs/
CCID Specification	Specification for Integrated Circuit(s) Cards Interface Devices. Revision 1.1
PC/SC	PC/SC Workgroup Specifications version 2.01.9 April 2010
PC/SC-3	PC/SC - Part 3 - Requirements for PC Connected Interface Devices V2.01.09, June 2007
ISO/IEC 7816	Information technology - Identification cards - Integrated circuit(s) cards with contacts
ISO/IEC 14443	Identification cards - Contactless integrated circuit cards - Proximity cards
5422 Reader Data Sheet	Provides a summary of the OMNIKEY 5422 Reader's features
5122 Reader Data Sheet	Provides a summary of the OMNIKEY 5122 Reader's features
NIST Special Publication 800-108	Recommendation for Key Derivation Using Pseudorandom Functions

Note: HID Global is not allowed to support proprietary card layer protocols that may be implemented in the host device/application. For example, FeliCa application developers must contact Sony and MIFARE branded products must contact NXP to obtain these card layer protocols. HID Global is constantly expanding credential support in the reader, so, some card technologies support only the chip UID.

Contact HID Global Technical Support for further information: https://www.hidglobal.com/support

1.4 Abbreviations and Definitions

Abbreviation	Description
APDU	Application Protocol Data Unit
ATR	Answer To Reset
ATS	Answer To Select
CCID	Integrated Circuit(s) Cards Interface Device
CE	Conformité Européenne
CSN	Card Serial Number
EMD	Electromagnetic Disturbance
FCC	Federal Communications Commission
ICAO	International Civil Aviation Organization
IDm	Manufacture ID (FeliCa UID)
IFD	Interface Device
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
OID	Unique Object Identifier
PC/SC	Personal Computer / Smart Card
PCD	Proximity Coupling Device
PICC	Proximity Integrated Circuit Card
RFU	Reserved for Future Use
UID	Universal ID
USB	Universal Serial Bus
VCD	Vicinity Coupling Device
VICC	Vicinity Integrated Circuit Card

1.5 Commands Responses

Status word SW1 SW2 returned for successful command execution is always equal to 0x90 00.

Description of responses for specific commands are listed in corresponding paragraphs with the function description.

Common description of possible status words can be found in ISO/IEC 7816-4, Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange, paragraph 5.1.3 Status bytes.

A public available list for such command responses can be found here:
http://www.wrankl.de/SCTables/SCTables.html

This page intentionally left blank.

Chapter 2

Getting Started

2.1 Driver Installation

As stated previously, no extra driver installation is necessary and every CCID compliant driver should work with the reader. However, Microsoft's CCID driver prevents the execution of CCID Escape commands.

If an application uses those commands, apply the procedure described in *Appendix A: Using PC_to_RDR_Escape Command*.

2.2 HID OMNIKEY Workbench

The OMNIKEY 5422 and 5122 are both supported by the HID OMNIKEY Workbench utility. The HID OMNIKEY Workbench is available at https://www.hidglobal.com/drivers/29454.

This page intentionally left blank.

Chapter 3

Host Interfaces

The OMNIKEY 5422 and 5122 readers support the following Host Interface.

- USB 2.0 Full Speed (12 Mbits/s) Device Port

3.1 USB

The device enumerates as a composite device. The OMNIKEY 5422 and 5122 USB protocol stack implements the following device class:

- CCID (Integrated Circuit Cards Interface Device, v1.1) - contact slot
- CCID (Integrated Circuit Cards Interface Device, v1.1) - contactless slot

The USB CCID interface can be used to send Application Protocol Data Unit (APDU) to the reader. The OMNIKEY 5422 and 5122 support the standard PC/SC API (for example, `SCardConnect`, `SCardDisconnect`, `SCardTransmit`). Consequently, any application software using the PC/SC API commands should be able to communicate with the reader.

3.1.1 Endpoint Assignments

The table below lists the USB protocol stack endpoints, their parameters and functional assignment:

Endpoint	Description	Type	Max Packet Size
EP0	Contact Endpoint	CONTROL	64-Bytes
EP6	Smart card - contact slot	BULK OUT	8-Bytes
EP1	Smart card - contact slot	BULK IN	64-Bytes
EP7	Smart card - contact slot	INTERRUPT	8-Bytes
EP2	Smart card - contactless slot	BULK OUT	64-Bytes
EP3	Smart card - contactless slot	BULK IN	64-Bytes
EP4	Smart card - contactless slot=	INTERRUPT	8-Bytes

This page intentionally left blank.

Chapter 4

Human Interface

4.1 LEDs

The OMNIKEY 5422 and 5122 Smart Card Readers are equipped with two color LEDs to indicate the current status of the reader:

State	LED Color		Description
	5422	5122	
Power-up	White	Green	The LED blinks on power-up (if attached to a host PC via USB).
Ready	White	Green	The LED is on. Typically, the PC operating system powers the smart card when the card is inserted into the slot. If no application on the PC makes use of the smart card, the LED switches off after a few seconds, indicating that the card power is off (USB suspend state).
Busy	Blue	Red	The LED blinks when the smart card reader transmits/receives any data to/from a connected PC.
USB suspend	None	None	The LED is off when the smart card is in USB suspend state.

This page intentionally left blank.

Chapter 5

Contact Card Communication

The OMNIKEY 5422 and 5122 Smart Card Readers are compliant with CCID specification and offers the following features:

- Configurable Voltage Selection
- Configurable Operation Mode: ISO or EMVCo
- Data Exchange Level set to short and extended APDU

5.1 Card Activation

Before a card is ready for data exchange, it must be properly activated:

- Insert the card into a contact slot. The reader selects the relevant supply voltage and powers up the card.
- If there is no response from the card, another class is selected and the power up sequence is repeated. If the card responds the response is evaluated.
- If the card returns a valid Answer to Reset (ATR), the PPS procedure follows (if applicable). If there is no ATR (i.e. some I2C cards), the state of lines is checked and a fake ATR composed. Finally, the reader notifies CCID of the card presence

5.2 Voltage Selection

The OMNIKEY 5422 and 5122 Smart Card Readers support all classes listed in ISO/IEC 7816-3.

It is possible to set the sequence during voltage selection process. This can be used for a card that supports more than one class or to accelerate activation time.

The sequence of voltage selection is configurable with the `voltageSelection` parameter. See *Section 13.4.1: voltageSequence*. Voltage selection sequence is encoded in one byte:

Sequence	**-**		**3**		**2**		**1**	
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Voltage	0	0	01 - 1.8V 10 - 3.0V 11 - 5.0V		01 - 1.8V 10 - 3.0V 11 - 5.0V		01 - 1.8V 10 - 3.0V 11 - 5.0V	

If all bits are 0, automatic voltage selection is set. That means device driver is responsible for voltage selection. Microsoft CCID driver voltage selection sequence is: 1.8V, 3V, 5V. Bits 7 and 6 are ignored and should be set to 0.

Examples:

1Bh (27dec) = 00 01 10 11: 5 V → 3 V → 1.8 V

39h (57dec) = 00 11 10 01: 1.8 V → 3 V → 5 V

5.3 Data Exchange Level

The OMNIKEY 5422 and 5122 Smart Card Readers support the following protocols, as defined in ISO/IEC 7816-3 [ISO/IEC 7816-3]:

- T = 0, T = 1,
- S = 8, S = 9, S = 10.

The reader supports extended APDU exchange level. This exchange level is supported for T = 1 cards only. For T = 0 cards the application should use the ENVELOPE command instead.

5.4 Operating Mode

The OMNIKEY 5422 and 5122 Smart Card Readers support the following operating modes:

- ISO mode
- EMVCo mode

The type of operating mode is configurable with `operatingMode` parameter. See *Section 13.4.2: operatingMode*.

5.4.1 ISO Mode

This is default mode of operation for smart card reader. It is designated to operate with [ISO 7816-4] compatible cards and synchronous cards.

5.4.2 EMVCo mode

This mode is suitable for cards that fulfill EMVCo specification.

Compared to ISO Mode:

- Only 5 V power supply
- Synchronous cards not supported

5.5 Slot Handling

Handling of the contact card slot can be disabled in the OMNIKEY 5422 and 5122 Smart Card Readers. It is configurable with `contactSlotEnable` parameter. See *Section 13.4.3: contactSlotEnable*.

When slot handling is disabled, notification about contact cart insertion is not presented.

This page intentionally left blank.

Chapter 6

Contactless Card Communication

The OMNIKEY 5422 and 5122 Smart Card Readers are compliant with CCID specification. Data exchange with a host is done via short and extended APDUs. Since the CCID specification does not define contactless protocols, T=1 protocol is emulated.

The reader supports sleep mode for low power applications. When in low power mode, the OMNIKEY 5422 and 5122 are able to detect a HF credential by frequently polling the field and then wake up the entire system.

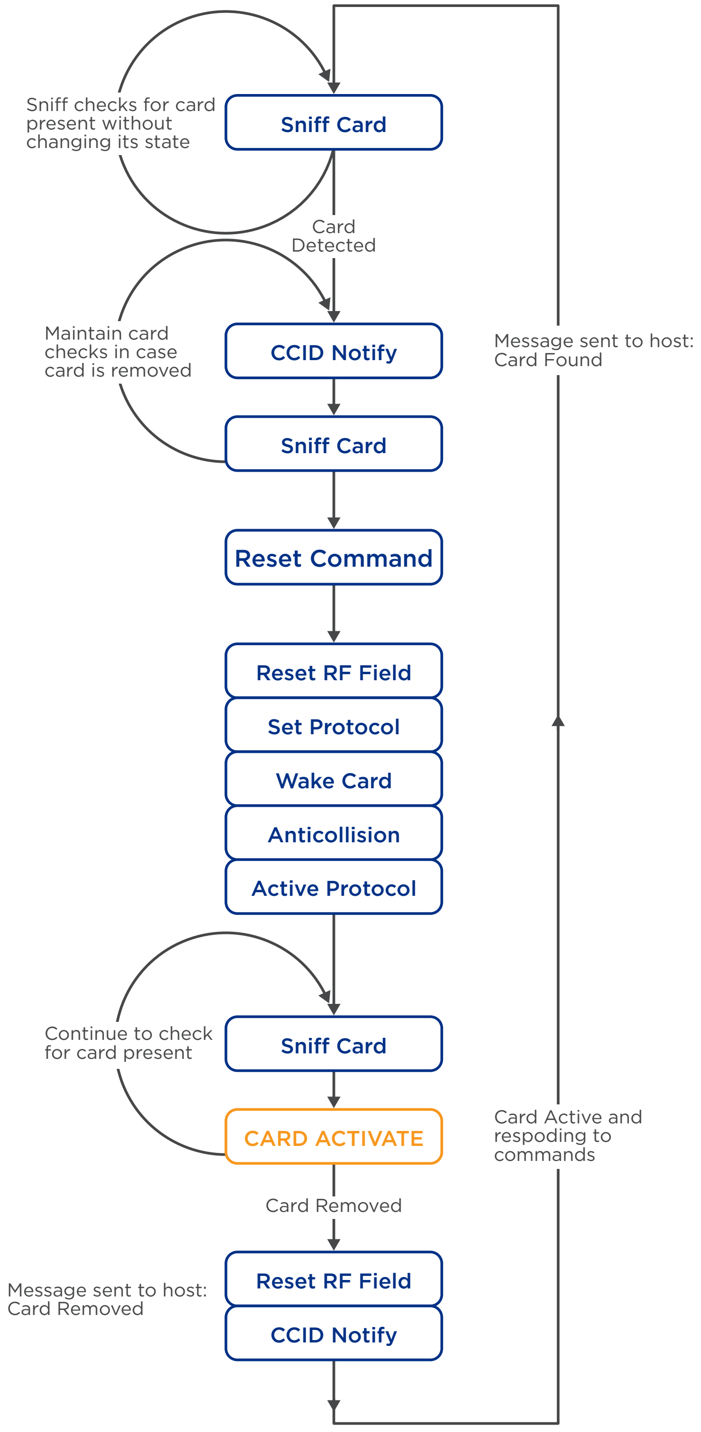
6.1 Polling Mode

The OMNIKEY 5422 and 5122 support a single polling mode. The reader polls for cards automatically using a set sequence of card protocols. It is possible to enable or disable each protocol individually but it is not possible to change the sequence. The factory default sequence is:

- ISO/IEC 14443 Type A
- ISO/IEC 14443 Type B
- iCLASS ISO/IEC 15693

When a card or cards are found the host application is notified through CCID. When the host powers up the card (CCID mode) the relevant anti-collision procedure is executed to achieve the selection of a single card. The reader to card airspeed is set to the highest value supported by both reader and card. Where applicable the card is put into the T=CL protocol state. The card details (that is ATR) are sent to the host. APDU layer communication is now possible through the CCID interface. The reader continues to poll for card removal, whereupon it sends an appropriate CCID message to the host application. On card removal, the cycle is repeated.

Polling Operation


Sniff checks for card present without changing its state
Sniff Card
Card Detected
Maintain card checks in case card is removed
CCID Notify
Message sent to host: Card Found
Sniff Card
Reset Command
Reset RF Field
Set Protocol
Wake Card
Anticollision
Active Protocol
Continue to check for card present
Sniff Card
CARD ACTIVATE
Card Active and respoding to commands
Card Removed
Message sent to host: Card Removed
Reset RF Field
CCID Notify

6.1.1 Polling and Power Consumption

When the reader is in low power mode it must periodically enable the RF field to detect new cards. The power consumption is directly related to the time the field is enabled. This time is longer when multiple protocols are enabled.

To limit the power consumption in sleep mode:

- Disable polling in low power mode
- Limit the number of protocols in polling
- Use low polling frequency

When the polling is disabled, the reader is unable to detect cards and wake up the host. In full power mode the number of enabled protocols does not matter because the field is always on.

Note: The reader compliance with USB 2.0 Low Power mode was evaluated with the default reader settings: ISO/IEC 14443 Type A and Type B enabled, iCLASS ISO/IEC 15693 enabled, polling period set to ~1.5s.

This page intentionally left blank.

Chapter 7

Personal Computer/Smart Card

OMNIKEY Smart Card Readers access contact cards through the framework defined in PC/SC. This makes card integration a straight forward process for any developer who is already familiar with this framework.

The Microsoft Developer Network (MSDN) Library contains valuable information and complete documentation of the SCard API within the MSDN Platform SDK.

See: http://msdn.microsoft.com/en-us/library/windows/desktop/aa380149(v=vs.85).aspx

7.1 How to Access Smart Card or Reader through PC/SC

The following steps provide a guideline to create your first smart card application using the industry standard, PC/SC compliant API function calls. The function definitions provided are taken verbatim from the MSDN Library [MSDNLIB]. For additional descriptions of these and other PC/SC functions provided by the Microsoft Windows PC/SC smart card components, refer to the MSDN Library.

See: http://msdn.microsoft.com/en-us/library/ms953432.aspx

1. Establish Context

   This step initializes the PC/SC API and allocates all resources necessary for a smart card session. The `SCardEstablishContext` function establishes the resource manager context (scope) within which database operations are performed.

   ```
   LONG SCardEstablishContext( IN DWORD dwScope,
                               IN LPCVOID pvReserved1,
                               IN LPCVOID pvReserved2,
                               OUT LPSCARDCONTEXT phContext);
   ```

2. Get Status Change

   Checks the status of the reader for card insertion, removal or availability.
   The `SCardGetStatusChange` function blocks execution until the current availability of the cards in a specific set of readers change. The caller supplies a list of monitored readers and the maximum wait time (in milliseconds) for an action to occur on one of the listed readers.

   ```
   LONG SCardGetStatusChange( IN SCARDCONTEXT hContext,
                              IN DWORD dwTimeout,
                              IN OUT LPSCARD_READERSTATE rgReaderStates,
                              IN DWORD cReaders);
   ```

3. List Readers

   To acquire a list of all PC/SC readers use the SCardListReaders function, look for HID Global OMNIKEY Smart Card Reader in the returned list. If multiple Contact Smart Card readers are connected to your system, they will be enumerated.

   Example: HID Global OMNIKEY 5422 Smartcard Reader 0, and HID Global OMNIKEY 5422CL Smartcard Reader 0.

```
LONG SCardListReaders( IN SCARDCONTEXT hContext,
                       IN LPCTSTR mszGroups,
                       OUT LPTSTR mszReaders,
                       IN OUT LPDWORD pcchReaders);
```

4. Connect

   Connect to the card. The SCardConnect function establishes a connection (using a specific resource manager context) between the calling application and a smart card contained by a specific reader. If no card exists in the specified reader, an error is returned.

```
LONG SCardConnect( IN SCARDCONTEXT hContext,
                   IN LPCTSTR szReader,
                   IN DWORD dwShareMode,
                   IN DWORD dwPreferredProtocols,
                   OUT LPSCARDHANDLE phCard,
                   OUT LPDWORD pdwActiveProtocol);
```

5. Exchange Data and Commands with the Card or the Reader

   Exchange command and data through APDUs. The SCardTransmit function sends a service request to the smart card, expecting to receive data back from the card.

```
LONG SCardTransmit( IN SCARDHANDLE hCard,
                    IN LPCSCARD_I0_REQUEST pioSendPci,
                    IN LPCBYTE pbSendBuffer,
                    IN DWORD cbSendLength,
                    IN OUT LPSCARD_IO_REQUEST pioRecvPci,
                    OUT LPBYTE pbRecvBuffer,
                    IN OUT LPDWORD pcbRecvLength);
```

Note: The application communicates through SCardControl()in environments where:

- SCardTransmit() is not allowed without an ICC
- SCardTransmit() is not allowed for any other reasons
- Developers prefer the application communicate through SCardControl()

The application retrieves the control code corresponding to FEATURE_CCID_ESC_COMMAND (see part 10, rev.2.02.07). In the case that this feature is not returned, the application may try SCARD_CTL_CODE (3500) as a control code to use.

```
LONG SCardControl( IN SCARDHANDLE hCard,
                   IN DWORD dwControlCode,
                   IN LPCVOID lpInBuffer,
                   IN DWORD nInBufferSize,
                   OUT LPVOID lpOutBuffer,
                   IN DWORD nOutBufferSize,
                   OUT LPDWORD lpBytesReturned);
```

6. Disconnect

It is not necessary to disconnect the card after the completion of all transactions, but it is recommended. The SCardDisconnect function terminates a connection previously opened between the calling application and a smart card in the target reader.

```
LONG SCardDisconnect( IN SCARDHANDLE hCard,
                      IN DWORD dwDisposition);
```

dwDisposition Values

Value	Implemented Function
SCARD_LEAVE_CARD	Do nothing special with card. Card is left in current state.
SCARD_RESET_CARD	Any contact card is reset. For contactless MIFARE card, the authenticated state is reset. Any contactless card working on layer 4 is left in current state.
SCARD_UNPOWER_CARD	Contact card is unpowered. The RF field is briefly disabled, resulting in the contactless card being unpowered. The field is then re-enabled, making it possible to connect to the card. In this scenario the reader does not send card removal or card present events.
SCARD_EJECT_CARD	Same as SCARD_UNPOWER_CARD. The reader does not support eject mechanism.

7. Release

This step ensures all system resources are released. The SCardReleaseContext function closes an established resource manager context, freeing any resources allocated under that context.

```
LONG SCardReleaseContext( IN SCARDCONTEXT hContext);
```

7.2 Vendor Specific Commands

Card readers support features outside the specified commands of PC/SC. To allow applications to control these features a generic command needs to be used. Use of a generic command prevents conflicts of reserved INS values used by certain card readers.

This command allows applications to control device specific features provided by the reader.

Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
0xFF	0x70	0x07	0x6B	xx	DER TLV coded PDU (Vendor Payload)	xx

The IFD supports the INS Byte 0x70 for vendor specific proprietary commands. P1 and P2 constitute the vendor ID. For OMNIKEY Smart Card Reader products the VID = 0x076B. The Data Field is constructed as ASN.1 objects/items, whereby every OMNIKEY Smart Card Reader object is identified by a unique Object Identifier (OID). OIDs are organized as a leaf tree under an invisible root node. The following table shows the first root nodes.

Tag Value	Vendor Payload Branch
0xA2 (constructed)	`readerInformationApi` see *Section 13.3: Reader Capabilities*.
0xA2 (constructed)	`contactSlotConfiguration` see *Section 13.4: Contact Slot Configuration*.
0xA4 (constructed)	`contactlessSlotConfiguration` see *Section 13.5: Contactless Slot Configuration*.
0xA6 (constructed)	`synchronousCardCommand` see *Section 9.1: Vendor Specific Synchronous Command Set*.
0xA7 (constructed)	`readerEEPROM` see *Section 13.6: Reader EEPROM*.
0x9D (primitive) 0xBD (constructed)	`resonse`
0x9E (primitive)	`errorResponse`

7.2.1 Response APDU

For all commands encapsulated in generic 0x70 APDU, the IFD returns the response frame constructed as follows.

Data Field	SW1 SW2
DER TLV coded Response PDU	See ISO 7816-4

The two last bytes of the response frame are always return code, SW1SW2.

In cases of an ISO 7816 violation, the return code is according to ISO 7816-4 and the data field may be empty.

In cases of positive processing or internal errors, the IFD returns SW1SW2 = 9000 and the data field is encapsulated in the response TAG (0x9D or 0xBD) or error response TAG (0x9E).

The response includes more than one leaf, depending on the request. Each leaf is encapsulated in the leaf tag.

7.2.2 Error Response

The error response tag caused by the firmware core is 0x9E (Class Context Specific) + (Primitive) + (0x1E). The length is two bytes. The first byte is the cycle in which the error occurred and the second byte is the exception type.

9E 02 xx yy 90 00	
Value	**Description**
0x9E	Tag = Error Response (0x0E) + (Class Context Specific) + (Primitive)
0x02	Len = 2
cycle	Value byte 1: Cycle in which the error occurred, see **Error Cycle**
error	Value byte 2: Error code, see **Error Code**
SW1	90
SW2	00

Error Cycle

First value byte	
Cycle	**Description**
0	HID Proprietary Command APDU
1	HID Proprietary Response APDU
2	HID Read or Write EEPROM Structure
3	RFU
4	RFU
5	RFU

Error Code

Second value byte		
Exception		**Description**
3	0x03	NOT_SUPPORTED
4	0x04	TLV_NOT_FOUND
5	0x05	TLV_MALFORMED
6	0x06	ISO_EXCEPTION
11	0x0B	PERSISTENT_TRANSACTION_ERROR
12	0x0C	PERSISTENT_WRITE_ERROR
13	0x0D	OUT_OF_PERSISTENT_MEMORY
15	0x0F	PERSISTENT_MEMORY_OBJECT_NOT_FOUND
17	0x11	INVALID_STORE_OPERATION
19	0x13	TLV_INVALID_STRENGTH
20	0x14	TLV_INSUFFICIENT_BUFFER
21	0x15	DATA_OBJECT_READONLY
31	0x1F	APPLICATION_EXCEPTION (Destination Node ID mismatch)
42	0x2A	MEDIA_TRANSMIT_EXCEPTION (Destination Node ID mismatch)
43	0x2B	SAM_INSUFFICIENT_MSGHEADER (Secure Channel ID not allowed)
47	0x2F	TLV_INVALID_INDEX
48	0x30	SECURITY_STATUS_NOT_SATISFIED
49	0x31	TLV_INVALID_VALUE
50	0x32	TLV_INVALID_TREE
64	0x40	RANDOM_INVALID
65	0x41	OBJECT_NOT_FOUND

Chapter 8

Asynchronous Cards Support

Asynchronous cards contain a CPU or are memory cards that are accessed through [ISO7816-4] compliant framed APDU commands. This type of card supports at least one of the asynchronous protocols T=0 or T=1. No additional libraries or third-party software components are necessary to integrate contactless CPU cards.

There is no standard list of APDUs (except those specified in [PCSC-3]). Typically a card has its own list of unique commands. Consult the specific card specification for full list of supported commands.

8.1 Standard APDU

Standard APDU is the application data unit that allows a maximum of 255 bytes of data to be sent to the card. It is supported by all [ISO7816-4] compatible cards.

8.1.1 Command APDU Definition

Command APDU is sent by the reader to the card. It contains a mandatory four byte header and from 0 to 255 bytes of data.

CLA	INS	P1	P2	Lc	Command Data	Le

CLA - Instruction class

INS - Instruction code

P1, P2 - Command parameters

Lc - Number of command data bytes of data

Command Data - Lc number bytes of data

Le - Maximum number of expected response bytes

If Le is omitted, any number of bytes in response is accepted. If Le=0, the reader expects max available data.

If the length of command data is 0, Lc must be omitted.

8.1.2 Response APDU Definition

Response APDU is sent by the card to the reader. It contains from 0 to 255 bytes of data and two mandatory status bytes SW1 and SW2.

Command Data	SW1	SW2

8.2 Extended APDU

Extended APDU is an extension to the standard APDU. It allows for more than 255 bytes of data to be transmitted in command and response. Extended APDU is backward compatible with Standard APDU. Command and response APDU look exactly the same for both types if the data length is equal or less than 255 bytes.

To use extended APDU the card must support it and the reader must operate in Extended APDU mode. The only difference between Standard APDU and Extended APDU is the length of the Lc and Le fields. In Extended APDU these may be omitted, 1 or 3 bytes depending of the length of data.

If the length of command data is less or equal to 255 the same rules apply to Lc as for Standard APDU. If the length of data is greater than 255, Lc must be 3 bytes in length with the first byte equal to 0 and the two following bytes encoding actual command data length.

Similar rules apply to Le. It may be omitted or 1 byte in the same way as for Standard APDU. If more than 255 bytes of data is expected, it may be 2 bytes long if Lc indicates extended length or 3 bytes long with first byte equal to 0 if there is no Lc field or Lc indicates length less than 256 bytes (Standard APDU).

For more detailed description of data length encoding rules, please see [ISO7816-4].

Chapter 9

Synchronous Cards Support

The device provides only Vendor Specific proprietary synchronous API to access synchronous cards. The PC/SC - like command set that uses standard APDU syntax and standard `SCardTransmit()` API is not supported.

9.1 Vendor Specific Synchronous Command Set

These commands allow applications to communicate with Synchronous Contact Cards using raw native card commands directly sent to the card by proper Bus Protocol.

All Synchronous Contact Card commands are identified by unique ASN.1 leaf. The root is defined as Synchronous Card Command and is encapsulated in vendor specific generic command. Under this root are specific commands for 2WBP (2 Wire Bus Protocol), 3WBP (3 Wire Bus Protocol) and I2C cards, organized as follows.

Synchronous Card Command Structure

Vendor Payload Branch	Command
synchronousCardCommand (0x06)	2WBPReadWrite (0x00) 3WBReadWrite (0x01) I2CReadWrite (0x02)

9.1.1 Two Wire Bus Protocol (2WBP) Read/Write

This command allows communication with synchronous contact smart card which supports two Wire Bus Protocol (2WBP) such as SLE 4432/42.

Each command consists of three bytes:

- Control
- Address
- Data

For SLE 4432 there are four commands available:

2WBP Common Commands

Byte 1 Control								Byte 2 Address	Byte 3 Data	Operation
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**	**A7-A0**	**D7-D0**	
0	0	1	1	0	0	0	0	address	no effect	Read Main Memory
0	0	1	1	1	0	0	0	address	input data	Update Main Memory
0	0	1	1	0	1	0	0	no effect	no effect	Read Protection Memory
0	0	1	1	1	1	0	0	address	input data	Write Protection Memory

There are three additional commands available for the SLE 4442:

Byte 1 Control								Byte 2 Address	Byte 3 Data	Operation
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**	**A7-A0**	**D7-D0**	
0	0	1	1	0	0	0	1	no effect	no effect	Read Security Memory
0	0	1	1	1	0	0	1	address	input data	Update Security Memory
0	0	1	1	0	0	1	1	address	input data	Compare Verification Data

More information can be found in Siemens' *ICs for Chip Cards Intelligent 256-Byte EEPROM SLE 4432/SLE 4442* datasheet.

Examples

See *Section 7.2: Vendor Specific Commands* for more information.

Update Main Memory

APDU: FF 70 07 6B 07 A6 05 A0 03 38 yy xx 00

Control byte = 0x38 (0b00111000), Address = yy, Data = xx (data to write)

A6 - `synchronousCardCommand`

A0 - `2WBPReadWrite`

Sample Response: 9D 00 90 00

Read Main Memory

APDU: FF 70 07 6B 07 A6 05 A0 03 30 yy 00 00

Control byte = 0x30 (0b00110000), Address = yy, Data = 0x00

A6 - `synchronousCardCommand`

A0 - `2WBPReadWrite`

Sample Response: 9D 01 xx 90 00

xx - received data

9.1.2 Three Wire Bus Protocol (3WBP) Read/Write

This command allows communication with synchronous contact smart card which supports three Wire Bus Protocol (3WBP) such as SLE 4418/28.

Each command consists of three bytes.

Byte 1								Byte 2	Byte 3	Operation
A9	A8	S5	S4	S3	S2	S1	S0	A7-A0	D7-D0	
address bit 8 and 9		1	1	0	0	0	1	address bits 7-0	input data	Write and erase with protect bit
		1	1	1	0	1	1		input data	Write and erase without protect bit
		1	1	0	0	0	0		comparison data	Write protect bit with data comparison (verification)
		0	0	1	1	0	0		no effect	Read 9 bits, data with protect bit
		0	0	1	1	1	0		no effect	Read 8 bits, data without protect bit
		1	1	0	0	0	1		input data	Write and erase with protect bit

3WBP Control Words for Command Entry, User Identification

Byte 1								Byte 2	Byte 3	Operation
A9	A8	S5	S4	S3	S2	S1	S0	A7-A0	D7-D0	
1	1	1	1	0	0	1	0	0xFD	bit mask	Write error counter
1	1	0	0	1	1	0	1	0xFE	PIN byte 1	Verify 1st PIN byte
1	1	0	0	1	1	0	1	0xFF	PIN byte 2	Verify 2nd PIN byte

More information can be found in Siemens' *ICs for Chip Cards SLE 4418/SLE 4428 Intelligent 8-Kbit EEPROM* datasheet.

Examples

See *Section 7.2: Vendor Specific Commands* for more information.

Write Memory with Protect Bit

APDU: FF 70 07 6B 07 A6 05 A1 03 33 yy xx 00

Byte 1 = 0x33 (0byy110011), Byte 2 = yy (part of address), Byte 3 = xx (data to write)

A6 - `synchronousCardCommand`

A1 - `3WBPReadWrite`

Sample Response: 9D 00 90 00

Read Memory with Protect Bit

APDU: FF 70 07 6B 07 A6 05 A1 03 0C yy 00 00

Byte 1 = 0x0C (0byy001100), Byte 2 = yy (part of address), Byte 3 = 0x00

A6 - `synchronousCardCommand`

A1 - `3WBPReadWrite`

Sample Response: 9D 01 xx yy 90 00

xx - received data

yy - protect bit, where 0x80 = bit not set, 0x00 = bit set

9.1.3 I2C Read/Write

This command allows communication with synchronous contact smart card which supports I2C Bus Protocol such as AT24C01/02/04/.../1024. Each command consists of 5+N bytes.

I2C Read/Write Command

Offset	Field	Size (byte)	Value	Description
0	Address length	1	1, 2, 3	Length of address: 1 = slave address only 2 = slave address + byte-subaddress 3 = slave address + word-subaddress
1	Bytes to read/write	1	N	Number of bytes to read/write
2	Address	1		Device address
3	Subaddress1	1		1st byte of subaddress
4	Subaddress2	1		2nd byte of subaddress
5	Data	N		Array of data bytes

Note: N = 252 (0xFC) bytes is the maximum value of **Bytes to read/write** field.

The device address word consists of a mandatory one, zero sequence for the first four most significant bits as shown. This is common to all I2C EEPROM devices.

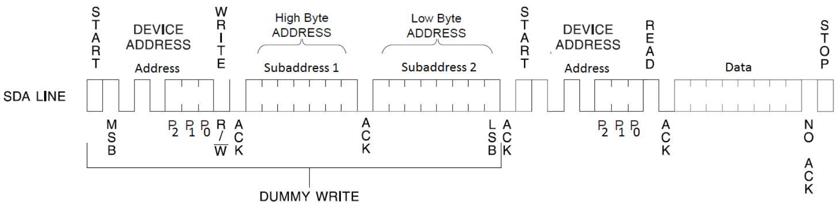
I2C Device Address

1	0	1	0	P_2	P_1	P_0	R/W
MSB							LSB

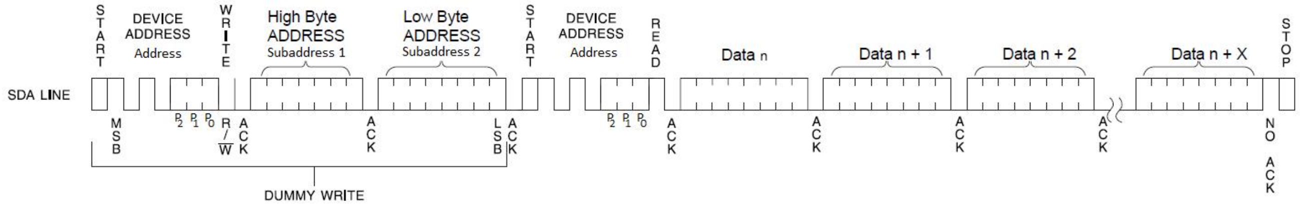
The next 3 bits are used for memory page addressing. These page addressing bits should be considered the most significant bits of the data word address.

More information about I2C EEPROM memory addressing, can be found in Atmel's *Two-wire serial EEPROM* datasheets, for example, AT24C01 or AT24C1024.

I2C Byte Read Command

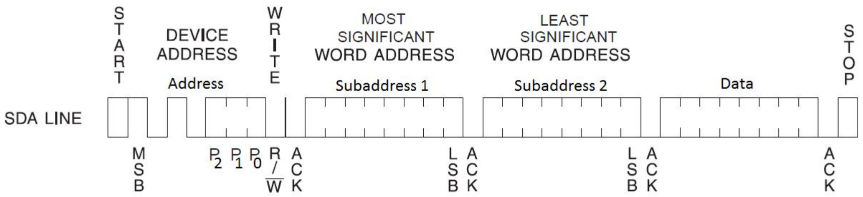


I2C N-Byte Read Command

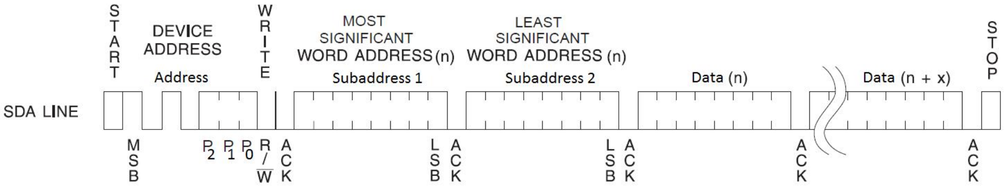


Synchronous contact smart card documentation must be consulted to see address length. For some Smart Cards, such as AT24C01, the 2 bytes sub-address is not necessary.

I2C Byte Write Command



I2C N-Byte Write Command



The N-Byte Write command can write a maximum of 32 bytes at once, however when the internally generated word address (inside the Smart Card EEPROM memory), reaches the page boundary, the following byte is placed at the beginning of the same page. The address "rollover" during write is from the last byte of the current page to the first byte of the same page.

The correct synchronous contact smart card documentation should be checked to determine how big the memory page is.

Example Parameters of I2C Cards

Card	PageSize	Number of Address Bytes	Memory Size
ST14C02C	8	1	256
ST14C04C	8	1	512
ST14E32	32	2	4096
M14C04	16	1	512
M14C16	16	1	2048
M14C32	32	2	4096
M14C64	32	2	8192
M14128	64	2	16384
M14256	64	2	32768
GFM2K	8	1	256
GFM4K	16	1	512
GFM32K	32	2	4096
AT24C01A	8	1	128
AT24C02	8	1	256
AT24C04	16	1	512
AT24C08	16	1	1024
AT24C16	16	1	2048
AT24C164	16	1	2048
AT24C32	32	2	4096
AT24C64	32	2	8192
AT24C128	64	2	16384
AT24C256	64	2	32768
AT24CS128	64	2	16384
AT24CS256	64	2	32768
AT24C512	128	2	65536
AT24C1024	256	2	131072
X24026	4	1	256

Examples

See *Section 7.2: Vendor Specific Commands* for more information.

Read 16 Bytes From Address 0x100 (AT24C1024, 3 bytes of address length)

APDU: FF 70 07 6B 09 A6 07 A2 05 03 10 A1 01 00 00

A6 - `synchronousCardCommand`

A2 - `I2CReadWrite`

Sample Response: 9D 01 xx...xx 90 00

xx...xx - received data (16 bytes)

Read 8 Bytes From Address 0x50 (AT24C16, 2 bytes of address length)

APDU: FF 70 07 6B 09 A6 07 A2 05 02 08 A1 50 00 00

A6 - `synchronousCardCommand`

A2 - `I2CReadWrite`

Sample Response: 9D 01 xx...xx 90 00

xx...xx - received data (8 bytes)

Read 1 Byte From Address (1 byte of address length)

APDU: FF 70 07 6B 09 A6 07 A2 05 01 01 A1 50 00 00

A6 - `synchronousCardCommand`

A2 - `I2CReadWrite`

Sample Response: 9D 01 xx 90 00

xx - received data

Write 8 Bytes to Address 0x50 (AT24C16, 2 byte of address length)

APDU: FF 70 07 6B 11 A6 0F A2 0D 02 08 A0 50 00 xx...xx 00

A6 - `synchronousCardCommand`

A2 - `I2CReadWrite`

xx...xx - data to send (8 bytes)

Sample Response: 9D 00 90 00

This page intentionally left blank.

Chapter 10

Contactless Protocol Support

10.1 ISO/IEC 14443 Type A

The OMNIKEY 5422 and 5122 support ISO/IEC 14443 Type A compliant cards. Anti-collision is as described in ISO/IEC 14443-3:2001(E) section 6.4. Protocol mode when supported is T=CL as described in ISO/IEC 14443-4.

The following ISO/IEC 14443A cards are supported by the reader:

- MIFARE Classic
- MIFARE Plus
- MIFARE DESFire EV1
- MIFARE Ultralight, Ultralight C

Note: The OMNIKEY 5422 and 5122 have no transparent communication and access to restricted area for MIFARE Ultralight C is not supported by the reader.

The OMNIKEY 5422 and 5122 allow accessing any T=CL card directly through PC/SC. MIFARE Classic, Ultralight, and MIFARE Plus in MIFARE Classic emulation mode are supported by the PC/SC commands described in *Section 11.1: PC/SC Commands*.

By default, the card will normally be switched to the highest possible speed supported by both the reader and the card. This is as described in the card ATS, but can be limited by the configuration options as described below. The maximum speed supported by the reader is 848Kbit/s. Protocol mode will then be enabled when supported by the card.

Configurable ISO14443A Parameters

Item	Description
iso14443aEnable	Enables or Disables support for ISO/IEC 14443 Type A
iso14443aRxTxBaudRate	Sets the maximum Baud Rate in the PCD to PICC/PICC to PCD direction
MIFAREKeyCache	Enable or Disable MIFARE key caching
MIFAREPreferred	Prefers MIFARE mode of a card

10.2 ISO/IEC 14443 Type B

The OMNIKEY 5422 and 5122 support all ISO/IEC 14443 Type B compliant cards. Anti-collision is as described in ISO/IEC 14443-3:2001(E) section 7. Protocol activation when supported is T=CL according to ISO/IEC 14443-3:2001(E) section 7.

By default, the card will normally be switched to the highest possible speed supported by both the reader and the card. This is as described in the card ATS, but can be limited by the configuration options as described below. The maximum speed supported by the reader is 848 kbps. Protocol mode will then be enabled when supported by the card.

Configurable ISO14443B Parameters

Item	Description
iso14443bEnable	Enables or Disables support for ISO/IEC 14443 Type B
iso14443bRxTxBaudRate	Sets the maximum Baud Rate in the PCD to PICC/PICC to PCD direction

10.3 iCLASS 15693

Access to iCLASS card data is through the proprietary set of pseudo APDUs. To comply with HID Global's security recommendations, iCLASS must be accessed through a secure session.

iCLASS Configurable Parameters

Item	Description
iCLASS15693Enable	Enables or Disables support for iCLASS 15693 card polling.
iCLASS15693Timeout	Sets time to wait for response to a command.
iCLASSActallTimeout	Sets time to wait for response to ACT/ACTALL.

Chapter 11

Contactless Card Communication

Before communicating with a contactless card it is necessary to select the card and, in some cases, authenticate with a known key. For a USB-connected host with an operating system the card selection is done automatically. To enhance user experience, the OMNIKEY 5422 and 5122 support so called “key caching” which reduces the number of authentication calls required to access certain areas of a card that use the same key. Key caching is disabled by default.

Communication with MIFARE Classic, MIFARE Plus, iCLASS, and Ultralight credentials is normally done using the PC/SC APDUs described in *Chapter 12: Secure Session* MIFARE DESFire cards on the other hand are only supported using T=CL pass through commands and the user must handle all of these details of encryption, authentication, reading writing etc., in their application code. This section includes the PC/SC commands required to communicate with a card. Examples of communication with certain specific card types are included in *Chapter 12: Secure Session*.

11.1 PC/SC Commands

11.1.1 Commands Sets

The PC/SC command set for contactless cards is defined in section 3.2 of *Interoperability Specification for ICCs and Personal Computer Systems - Part 3. Requirements for PC-Connected Interface Devices*, and is available from the PC/SC Workgroup website http://www.pcscworkgroup.com. The commands use standard APDU syntax and standard SCardTransmit API, but use the reserved value of the CLA byte of 0xFF.

PC/SC Commands

Instruction	Description	Comments
0xCA	Get Data	Fully supported
0x82	Load Keys	Partially supported
0x86	General Authenticate	Supported for HID iCLASS and MIFARE cards
0x20	Verify	Not supported
0xD4	Increment	Supported for MIFARE cards
0xD8	Decrement	Supported for MIFARE cards

Common SW1SW2 Return Codes

SW1SW2	Definition
0x9000	Operation successful
0x6700	Wrong length (Lc or Le)
0x6A81	Function not supported
0x6B00	Wrong parameter (P1 or P2)
0x6F00	Operation failed

11.1.2 0xCA - Get Data

This command is used to retrieve certain specific information relating to the card itself such as card serial number, rather than data on the card itself. The items which can be retrieved are listed in the following table.

CLA	INS	P1	P2	Lc	Data In	Le
0xFF	0xCA	0x00 0x01	0x00	-	-	xx

General: Works with any type of card, unless P1 = 0x01 (see below)

Get Data Command Response

P1	Card Type	Data Out	SW1SW2	
0x00	ISO/IEC 14443 Type A	4, 7 or 10-byte UID	0x9000	Operation successful
	ISO/IEC 14443 Type B	4-byte PUPI		
	iCLASS 15693	8-byte CSN		
0x01	ISO/IEC 14443 Type A	n Historical bytes		
	Other	-	0x6A81	Function not supported

Note: For the ISO/IEC 14443 Type A Innovision Jewel card, the Data field is 7 bytes of 0x00.

Note: The number of historical bytes returned is limited to 15.

11.1.3 0x82 - Load Keys

This command allows the application to load keys onto the reader, including MIFARE keys, iCLASS keys, and secures session keys. All keys with the exception of MIFARE keys must be loaded during a secure session. MIFARE keys can be loaded whether a secure session is established or not.

Load Keys Command APDU

CLA	INS	P1	P2	Lc	Data In	Le
0xFF	0x82	Key Structure	Key Number	Key Length	Key	-

General: Works with any type of card or can be sent using `SCardControl()`

Load Keys P1 Coding (Key Structure)

b7	b6	b5	b4	b3	b2	b1	b0	Description
0	--		RFU	----				Card Key
1	--			----				Reader Key
--	0	--		----				Fixed to 0. Plain transmission
--		0		----				Stored in volatile memory
--		1		----				Stored in non-volatile memory
--				0000				Fixed value 0000

Key Number: See *Section 11.2: Key Locations*

Key Length: 6 or 8 or 16 bytes

Key: Key in plain text

Load Keys Response

Data Out	SW1SW2	
-	0x9000	Operation successful
	0x6982	Card key not supported
	0x6983	Reader key not supported
	0x6985	Secured transmission not supported
	0x6986	Volatile memory is not available
	0x6987	Non-volatile memory is not available
	0x6988	Key number not valid
	0x6989	Key length is not correct
	0x6990	Security error

11.1.4 0x86 – General Authenticate

This command allows the user to authenticate a credential. Before using this command the correct keys must have been loaded to the relevant key slot. For iCLASS keys these keys are preloaded onto the reader, so the application must just select the correct key number for the area they are attempting to access.

General Authenticate Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
0xFF	0x86	0x00	0x00	0x05	Data, see below	–

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Version = 0x01	Address MSB	Address LSB	Key Type	Key Number

Key Types: 0x00 = PicoPass Debit Key (KD)
0x01 = PicoPass Credit Key (KC)
0x60 =MIFARE KeyA
0x61 = MIFARE KeyB

For MIFARE cards:

Address MSB = 0, Address LSB = the block number counted from 0 to [19 (MINI), 63 (1K), 127(2K) or 255(4K)].

For iCLASS the following scheme is used:

Address LSB: Page number 0 - 7

Address MSB: Book number 0 or 1, bit 0 - book number, bit 1 select flag.

Select flag 0 - authenticate without implicit select

Select flag 1 - authenticate with implicit select book page according LSB bit3:0

General Authenticate Supported Card Addressing

Supported Card	Memory Addressing
iCLASS	MSB = Book / LSB = Page
MIFARE	Any block number in the requested sector

Response APDU:

General Authenticate Response

Data Field	SW1SW2
empty	See following table

General Authenticate Return Codes

Type	SW1SW2	Description
Normal	0x9000	Successful
Warning		
Execution Error	0x6400	No Response from media (Time Out)
	0x6581	Illegal block number (out of memory space)
Checking Error	0x6700	Wrong APDU length
	0x6982	Security status not satisfied (not authenticated)
	0x6986	Wrong key type
	0x6988	Wrong key number

11.1.5 0xB0 – Read Binary

The Read Binary command returns the data on a credential. For MIFARE Classic and Plus cards this requires a prior general authenticate command to succeed. For iCLASS all blocks except blocks 0-5 require the relevant page to be authenticated beforehand, but the correct book and page must be selected to avoid reading the wrong data. See *Section 8.1.1: Command APDU Definition* for an APDU command. MIFARE Ultralight cards do not require authentication.

Read Binary Command APDU

CLA	INS	P1	P2	Lc	Data In	Le
0xFF	0xB0	Address MSB	Address LSB	–	–	xx

Read Binary Supported Cards

Supported Cards	Memory Addressing	Size
iCLASS	Block number	Any multiple of a block (8 bytes) less than the page size
MIFARE 1K/4K	Block number	Any multiple of a block (16 bytes) less than the sector size
Ultralight	Block number	Any multiple of a block (4 bytes) less than the total card size
NXP iCode	Block number	1 block of 4 bytes

Read Binary P1 Coding for iCLASS

b7	b6	b5:0				Description
0	0	b5	b4	b3	b2:0	
		RFU	0	0	0 0 0	read block number (P2) without SELECT
			1	0	x x x	read block number (P2) with SELECT book 0, page xxx
			1	1	x x x	read block number (P2) with SELECT book 1, page xxx
0	1	0 0 0 0 0 0				read with DES decrypted
1	0					RFU
1	1					read with 3-DES decrypted

Using P1 to indicate the targeted book and page allows reading the addressed block numbers without a dedicated prior authentication command. This is only applicable for free accessible blocks e.g. block 0-2 and 5. The most significant bits 6 and 7 of P1 indicate whether the IFD is forced to either read the data in plain or to decrypt the data using DES or 3DES.

Read Binary Response

Data Field	SW1SW2
Media Data according Le, dependent on supported block size	See the following table

Note: If the media is readable then the IFD returns always the number of data bytes according to the Le value. If Le is less than block size the data field is cut off the Le position and the return code is 6Cxx, where xx is the real block size. If Le is greater than the available block size the IFD returns the number of available bytes and the return code 6282 (warning end of data reached before Le bytes). If the application requests a multiple of media block size in the Le field than the IFD returns all requested bytes and the return code is 9000. This ensures a high performance particular for medias with "Read Multiple Blocks" support..

Read Binary SW1SW2 Values

Type	SW1SW2	Description
Normal	0x9000	Successful
Warning	0x6282	End of data reached before Le bytes (Le is greater than data length).
Execution Error	0x6400	No Response from media (Time Out)
Checking Error	0x6700	Wrong APDU length
	0x6982	Block not authenticated (Security status not satisfied)
	0x6A82	Illegal block number (File not found)
	0x6Cxx	Wrong Le; SW2 encodes the exact number of available data bytes

11.1.6 0xD6 – Update Binary

This command allows data to be written to a credential. For MIFARE Classic, Plus and iCLASS the relevant block must have been authenticated by a prior general authenticate command.

Update Binary Command APDU

CLA	INS	P1	P2	Lc	Data In	Le
0xFF	0xD6	Address MSB	Address LSB	xx	Data	–

Update Binary Supported Cards

Supported Cards	Memory Addressing	Size
iCLASS	Block number	1 block of 8 bytes
MIFARE 1K/4K	Block number	1 block of 16 bytes
Ultralight	Block number	1 to 4 blocks of 4 bytes
NXP iCode	Block number	1 block of 4 bytes

Note: iCLASS update binary - selecting the book and page is not necessary because the write operation requires a prior authentication command. The most significant bits 6 and 7 of P1 indicate whether the IFD is forced to either write data in plain or to encryption the data using DES or 3DES.

Update Binary Response

Data Field	SW1SW2
empty	See the following table

Read Binary SW1SW2 Values

Type	SW1SW2	Description
Normal	0x9000	Successful
Warning		
Execution Error	0x6400	No Response from media (Time Out)
	0x6581	Not usable block number in the memory area (Memory failure)
Checking Error	0x6700	Wrong APDU length
	0x6982	Block not authenticated (Security status not satisfied)
	0x6A82	Illegal block number (File not found)
	0x6Cxx	Wrong Lc; SW2 encodes the exact number of expected data bytes

11.1.7 0xD4 – Increment

This command increments the value of a designated block. This command is currently supported for MIFARE cards only.

Increment Command APDU

CLA	INS	P1	P2	Lc	Data In	Le
0xFF	0xD4	0x00	Block number	0x04	Data	–

Increment Supported Cards

Supported Cards	Memory Addressing	Size
MIFARE 1K/4K	Block number	-

Increment Response

Data Field	SW1SW2
empty	See the following table

Increment Response SW1SW2 Bytes

Type	SW1SW2	Description
Normal	0x9000	Successful
Warning		
Execution Error	0x6400	No Response from media (Time Out)
	0x6581	Memory failure (unsuccessful increment / decrement)
Checking Error	0x6700	Wrong APDU length
	0x6981	Incompatible command
	0x6982	Block not authenticated (Security status not satisfied)
	0x6986	Command not allowed
	0x6A81	Function not supported
	0x6A82	Invalid block number

11.1.8 0xD8 – Decrement

This command decrements the value of a designated block. This command is currently supported for MIFARE cards only.

Decrement Command APDU

CLA	INS	P1	P2	Lc	Data In	Le
0xFF	0xD8	0x00	Block number	0x04	Data	–

Decrement Supported Cards

Supported Cards	Memory Addressing	Size
MIFARE 1K/4K	Block number	-

Decrement Response

Data Field	SW1SW2
empty	See the following table

Decrement Response SW1SW2 Bytes

Type	SW1SW2	Description
Normal	0x9000	Successful
Warning		
Execution Error	0x6400	No Response from media (Time Out)
	0x6581	Memory failure (unsuccessful increment / decrement)
Checking Error	0x6700	Wrong APDU length
	0x6981	Incompatible command
	0x6982	Block not authenticated (Security status not satisfied)
	0x6986	Command not allowed
	0x6A81	Function not supported
	0x6A82	Invalid block number

11.2 Key Locations

OMNIKEY 5422 and 5122 support 74 key slots. Each slot has fixed size and function which cannot be changed. The content of volatile slots is lost when power is off. MIFARE, iCLASS and secure session slots are initialized to default values in the factory.

Count	First Slot	Last Slot	Key Slot	Volatile	Non-volatile	Reader Key	Card Key	Description
32	0	31	6		✓		✓	MIFARE
32	32	63	8		✓		✓	iCLASS Non-volatile
1	64	64	8		✓	✓		iCLASS DES card content
2	65	66	8	✓			✓	iCLASS volatile
1	67	67	16		✓	✓		iCLASS 3DES card content
6	68	73	16		✓	✓		Secure session AES128

11.2.1 Key Loading

New key values can be loaded to a slot using PC/SC command Load Key. The only exception is slot 32 (iCLASS KD key) which cannot be changed.

For more information how to use Load Key command see *Section 11.1.3: 0x82 - Load Keys*.

11.2.2 Key Types

There are six types of keys supported by OMNIKEY 5422 and 5122.

- MIFARE keys
- iCLASS non-volatile keys
- iCLASS volatile keys
- iCLASS DES keys
- iCLASS 3DES keys
- Secure session keys

11.2.2.1 MIFARE Keys

6 bytes keys designated for MIFARE cards usage. All key slots for MIFARE cards are initialized by 0xFFFFFFFFFFFF value. MIFARE keys can be loaded without or during secure session.

11.2.2.2 iCLASS Non-volatile Keys

8 bytes keys designated for iCLASS usage. There are 32 8 bytes key slots 32-53. iCLASS keys can be loaded only in secure session. Default values are:

Slot	Role	Slot	Role
32	Book 0/page 0 KD	48	Book 1/page 0 KD
33	Book 0/page 0 KC	49	Book 1/page 0 KC
34	Book 0/page 1 KD	50	Book 1/page 1 KD
35	Book 0/page 1 KC	51	Book 1/page 1 KC
36	Book 0/page 2 KD	52	Book 1/page 2 KD
37	Book 0/page 2 KC	53	Book 1/page 2 KC
38	Book 0/page 3 KD	54	Book 1/page 3 KD
39	Book 0/page 3 KC	55	Book 1/page 3 KC
40	Book 0/page 4 KD	56	Book 1/page 4 KD
41	Book 0/page 4 KC	57	Book 1/page 4 KC
42	Book 0/page 5 KD	48	Book 1/page 5 KD
43	Book 0/page 5 KC	49	Book 1/page 5 KC
47	Book 0/page 6 KD	50	Book 1/page 6 KD
45	Book 0/page 6 KC	61	Book 1/page 6 KC
46	Book 0/page 7 KD	62	Book 1/page 7 KD
47	Book 0/page 7 KC	63	Book 1/page 7 KC

11.2.2.3 iCLASS Volatile Keys

The same as iCLASS non-volatile keys but stored in RAM. Content of these 2 keys is lost when the power is switched off.

11.2.2.4 iCLASS DES Keys

This key is used to encrypt or decrypt data when update binary or read binary command specifies DES algorithm.

11.2.2.5 iCLASS 3DES Keys

This key is used to encrypt or decrypt data when update binary or read binary command specifies 3DES algorithm.

11.2.2.6 Secure Session Keys

16 bytes AES keys used for opening secure session. Keys are grouped in pairs. The first key in pair is used as cipher key, the other one as MAC key. For detailed key roles see *Section 12.2: Secure Session Access Rights*.

11.3 OMNIKEY Specific Commands

Card reader supports features outside the specified commands of PC/SC-3; see *Section 1.3: Reference Documents*. Vendor specific proprietary command allows applications to control device specific features provided by the reader; see *Section: 7.2 Vendor Specific Commands*. Use of such a generic command prevents conflicts of reserved INS values but used by certain card reader.

11.3.1 Reader Information API

This command group is reserved for GET and SET of reader specific information. See *Section 13.3: Reader Capabilities*.

11.4 Communication Examples

In the examples below, the following color-coding is used for APDUs:

Color Coding of Examples

CLA	INS	P1, P2	Lc	Data	Le
Red	Orange	Green	Blue	Black	Purple

11.4.1 MIFARE Classic 1K/4K Example

To read and write to a MIFARE card, first authenticate the card with the correct key, pre-loaded into the reader. The PC/SC Load Keys, General Authenticate, Read Binary and Update Binary APDUs can be used for these functions. An example APDU sequence is as follows:

Load a 6-byte MIFARE key of all FFs to Key Number 1:

FF 82 00 01 06 FF FF FF FF FF FF

Authenticate block 1 with Key Number 1:

FF 86 00 00 05 01 00 01 60 01

Read block1 (16 bytes):

FF B0 00 01 10

Write 16 bytes of data to block 2:

FF D6 00 02 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

11.4.2 MIFARE DESFire Example

The example APDU sequence below shows how to read a standard data file, which is not protected by a key from a DESFire or DESFire EV1 (All values are LSB first):

Select Application with AID = xx xx xx (that is the application which contains the file to be read):

Command: 90 5A 00 00 03 xx xx xx 00

Response: 91 00

Read 10 bytes of file xx (the file to be read), starting at byte 0:

Command: 90 BD 00 00 07 xx 00 00 00 0A 00 00 00

Response: xx xx xx xx xx xx xx xx xx xx 91 00

The xx bytes in the response are the data from the file.

For full details of all DESFire commands, refer to the NXP data sheets.

This page intentionally left blank.

Chapter 12

Secure Session

The secure session model provides a secure way of communicating with the iCLASS cards. As the commands are encrypted this prevents any snooping of messages between the host application and the reader. The secure session model allows for write or read access to iCLASS card to be conditional based on the key used to establish secure session.

12.1 Using a Secure Session

To establish a secure session the user must use OMNIKEY specific functions to initialize and continue authentication described in detail in the following sections. Once the secure session has been established then data is exchanged with the device using the OMNIKEY specific APDU for data exchange described below. This will allow access to the data stored on a card and loading custom encryption keys. The secure session will terminate following an error in the data exchange or encryption or if the user sends the terminate session commands described below.

For secure channel communication two AES 128 bit keys are required. One key is used for encryption and the other to compute MAC. The keys are stored in slots, for secure session only keys from slot 68 to 73 can be used. Every time secure channel is opened new session keys are generated based on data exchanged between the reader and the client and keys stored in slots. Key derivation algorithm is based on NIST Special Publication 800-108.

12.1.1 Commands Available in Secure Session

The following commands are available in secure session.

- Load Key
- iCLASS Read binary
- iCLASS Update binary

Please note that the iCLASS update and loading the keys operation may be limited by the session key access rights, see *Section 12.2: Secure Session Access Rights* for more details.

12.1.2 Establish and Manage a Secure Session

For a secure channel transmission the `SCardConnect` should be used with a `ShareMode` of `SCARD_SHARE_EXCLUSIVE`. The Client (host application) must ensure the correct termination of the secure cannel after the last transaction.

There are three phases of secure channel communication:

1. Authentication, to establish secured channel
2. Exchange of encrypted packets to securely transmit data to and from a card
3. Termination to close secure channel

12.1.3 Authentication

The authentication process must be successfully performed before any data can be sent securely. To start secure session two specific commands must be sent: `Get Challenge` and `Mutual Authentication`.

To initialize the secured channel the client must send Get Challenge packet to the reader.

CLA	INS	P1	P2	Lc	Data Field	Le
0xFF	0x72	0x00	Key	0x00		

P2 parameter (key) indicates key slots used for cryptography operations in Mutual Authentication phase. Only secure session keys from slots 68 to 73 can be used for that purpose. The key indicated in Get Challenge frame must be used for all following frames. As two keys are required for secure transmission, the reader automatically use slot indicated in command for cipher and key from slot+1 for MAC.

The reader responds with 16 bytes of random data plus 2 bytes of SW1SW2. RDRnonce is used later to compute SSC counter.

Data	SW1SW2
16 bytes RDRnonce	0x9000

In the next step the client sends Mutual Authentication frame. The client must prepare 2 random 16 bytes data blocks: PCKey and PCnonce and send them to the reader with RDRnonce received in response to Get Challenge.

CLA	INS	P1	P2	Lc	Data Field	Le
0xFF	0x72	0x01	0x00	0x40	64 crypto data	

Data filed consists of 4 subfields:

RDRnonce	PCRnonce	PCKey	MAC
Encrypted 16 bytes	Encrypted 16 bytes	Encrypted 16 bytes	16 bytes

PCnonce - 16 bytes random number from the client

RDRnonce - 16 bytes sent in Get Challenge response packet

PCKey - 16 bytes key randomized by the client

MAC - digital signature in plain text

In response, the reader sends 64 bytes plus 2 SW1SW2 bytes:

RDRnonce	PCRnonce	PCKey	MAC	SW1SW2
Encrypted 16 bytes	Encrypted 16 bytes	Encrypted 16 bytes	16 bytes	0x90000

Plain values of PCnonce, RDRnonce are the same values as in the request. RDRKey is 16 bytes key randomized by the reader.

The client must verify if the values sent in the request are the same as in the response. If the values are correct, secure channel is successfully established.

12.1.4 Data Exchange in Secure Session

To exchange data during a secure session the complete PC/SC command should be encrypted. The secure message is wrapped in an APDU of the form FF 72 00 00 + Lc + message.

Note: The APDU should not use Le value.

12.1.4.1 SSC

SSC is a 16 bytes counter which must be incremented after every packet (request and response).

The initial value of SSC counter is made of the first 8 bytes of PCnonce and the first 8 bytes of RDRnonce.

SSC	
First 8 bytes of PCnonce	First 8 bytes of RDRnonce

12.1.4.2 Session key

After authentication phase the reader and the client compute two session keys based on RDRKey and PCKey values. As mentioned earlier the first key is used to encrypt data and the second one for MAC computation.

The key derivation algorithm is based on NIST Special Publication 800-108.

12.1.4.3 Data frame

Data frame is used to safely communicate between a card and the client.

The card command (complete APDU) must be encrypted using cipher session key. This data and MAC computed using MAC session key can be send to the reader with fixed header FF 72 02 00 Lc.

Encryption algorithm uses AES in SIV mode. There is no need to pad plain text message.

After sending command SSC counter must be increased by 1.

CLA	INS	P1	P2	Lc	Data Field	Le
0xFF	0x72	0x02	0x00	length	Encrypted data + 16 bytes MAC	

The response from a card is encrypted and together with MAC and 2 byte SW1SW2 send back to the client.

Data	SW1SW2
Encrypted card response + 16 bytes MAC	0x9000

When the client receives the response SSC counter must be increased by 1.

12.1.5 Terminate session

To finish secure channel transmission the client must send terminate session command. In addition if any other command with wrong APDU header is received or MAC is wrong, the session is terminated automatically.

CLA	INS	P1	P2	Lc	Data Field	Le
0xFF	0x72	0x03	0x00	0x00		

The response contains only 2 SW1SW2 bytes.

SW1SW2
0x9000

12.2 Secure Session Access Rights

The OMNIKEY 5422 and 5122 support six secure session keys which occupies slots from 68 to 73. Every key pair has assigned access rights which cannot be changed. The following table summarizes the Access Rights.

Slot	Name	Access Rights
68	User admin Cipher key	Can load all keys and read/write iCLASS
69	User admin MAC key	Can load all keys and read/write iCLASS
70	iCLASS Read/Write Cipher key	Can read/write iCLASS
71	iCLASS Read/Write MAC key	Can read/write iCLASS
72	iCLASS Read only Cipher key	Can read iCLASS only
73	iCLASS Read only MAC key	Can read iCLASS only

For obtaining the pre-configured secure session key, contact HID Tech Support https://www.hidglobal.com/support.

12.3 Changing the Secure Session Keys

You should change the secure session keys default values. The default values are the same for every customer and are therefore unsecure.

Note: It is the responsibility of the customer to maintain the new keys. If the key values are lost, they cannot be recovered by HID Global. To change the keys use Load Key PC/SC command.

This page intentionally left blank.

Chapter 13

Reader Configuration

All OMNIKEY 5422 and 5122 configurable items are identified by a unique ASN.1 leaf. A full description is given below, including default values and example APDU commands to `get` and `set`.

13.1 APDU Commands

If the attached host implements a PC/SC environment, the OMNIKEY 5422 and 5122 ASN.1 leafs are accessible using proprietary APDU commands sent through the CCID USB device class. The APDU commands are used to set and get the configuration items and to control the reader – to apply, store or reset the changes.

APDUs supported by the OMNIKEY 5422 and 5122 readers fall into the following groups:

- Standard inter-industry commands as defined in ISO/IEC 7816-4:2005(E) - these commands are passed transparently to the contactless card related to the CCID slot
- PC/SC commands as defined in *Interoperability Specification for ICCs and Personal Computer Systems - Part 3*
- ICAO (International Civil Aviation Organization) test commands as defined in Appendix C of *RF Protocol and Application Test Standard for e-Passport - Part 4*
- OMNIKEY 5422 and 5122 commands - these include APDUs to manage the reader, to directly access the configuration items

13.2 Accessing Configuration

OMNIKEY Specific Commands include the Reader Information API command group that provides access to reader configuration and allows to control the reader.

Configuration Structure

Root	Request	Branch
readerInformationApi (0x02)	Get (0x00)	readerCapabilities (0x00) tlvVersion (0x00) deviceID (0x01) productName (0x02) productPlatform (0x03) enabledCLFeatures (0x04) firmwareVersion (0x05) hfControllerVersion (0x08) hardwareVersion (0x09) hostInterfaces (0x0A) numberOfContactSlots (0x0B) numberOfContactlessSlots (0x0C) numberOfAntennas (0x0D) humanInterfaces (0x0E) vendorName (0x0F) exchangeLevel (0x11) serialNumber (0x12) hfControllerType (0x13) sizeOfUserEEProm (0x14) firmwareLabel (0x16)
	Get (0x00) Set (0x01)	contactSlotConfiguration (0x03) voltageSequence (0x02) operatingMode (0x03) contactSlotEnable (0x05)
	Get (0x00) Set (0x01)	contactlessSlotConfiguration (0x04) contactlessCommon (0x00) sleepModePollingFrequency(0x0D) sleepModeCardDetectionEnable (0x0E) pollingSearchOrder (0x09) emdSuppresionEnable (0x07) iso14443aConfig (0x02) iso14443aEnable (0x00) iso14443aRxTxBaudRate (0x01) MIFAREKeyCache (0x03) MIFAREPreferred (0x04) iso14443bConfig (0x03) iso14443bEnable(0x00) iso14443bRxTxBaudRate (0x01) iCLASSConfig (0x06) iCLASS15693Enable (0x03) iCLASS15693Timeout (0x05) iCLASSActallTimeout (0x06)
	Get (0x00) Set (0x01)	readerEEPROM (0x07) eepromOffset (0x01) eepromRdLength (0x02) eepromWrData (0x03)
	Set (0x01)	readerConfigurationControl (0x09) applySettings (0x00) restoreFactoryDefaults (0x01) rebootDevice (0x03)

Note: After SET requests, apply settings to apply the changes.

13.2.1 Example: Get Reader Information

With Get Reader Information the host application get specific information's about the reader. The following example shows how to read a single item:

DER TLV PDU for retrieve single IFD information (productName):

```
A2 06                           // CHOICE ReaderInformationAPI
   A0 04                        // CHOICE GetReaderInformation
      A0 02                     // CHOICE ReaderCapabilites
      82 00                     // SEQUENCE productName
```

The reply of single information is TLV coded:

BD 0F 82 0D **4F 4D 4E 49 4B 45 59 20 35 30 32 32 00** 90 00 // 'OMNIKEY 5422' + return code

For a Reader Information GET Request the Response Tag (1D) is always CONSTRUCTED. The response can include more than one leaf, depending on the request.

DER TLV PDU for retrieve single IFD information (productPlatform):

```
A2 06                           // CHOICE ReaderInformationAPI
   A0 04                        // CHOICE GetReaderInformation
      A0 02                     // CHOICE ReaderCapabilites
      83 00                     // SEQUENCE productplatform
```

The reply of single information is TLV coded:

BD 0A 83 08 **41 56 69 61 74 6F 52 00** 90 00   // 'AViatoR' + return code success

13.3 Reader Capabilities

Reader Capabilities Structure

Root	Branch
readerCapabilities (0x02)	tlvVersion (0x00) deviceID (0x01) productName (0x02) productPlatform (0x03) enabledCLFeatures (0x04) firmwareVersion (0x05) hfControllerVersion (0x08) hardwareVersion (0x09) hostInterfaces (0x0A) numberOfContactSlots (0x0B) numberOfContactlessSlots (0x0C) numberOfAntennas (0x0D) humanInterfaces (0x0E) vendorName (0x0F) exchangeLevel (0x11) serialNumber (0x12) hfControllerType (0x13) sizeOfUserEEProm (0x14) firmwareLabel (0x16)

13.3.1 tlvVersion

Tag	0x00
Access	Read-only
Type	INTEGER
Length	1 byte
Value	0x00 - 0xFF
Description	The version of the TLV encoding used by APDUs
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 80 00 00
Sample Response	BD 03 80 01 **01** 90 00

13.3.2 deviceID

Tag	0x01
Access	Read-only
Type	OCTET STRING
Length	2 bytes
Value	0x0000 - 0xFFFF
Description	Product ID
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 81 00 00
Sample Response	BD 04 81 02 **00 07** 90 00

13.3.3 productName

Tag	0x02
Access	Read-only
Type	OCTET STRING
Length	Variable
Value	Null terminated string
Description	The name of the reader
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 82 00 00
Sample Response	BD 0F 82 0D **4F 4D 4E 49 4B 45 59 20 35 30 32 33 00** 90 00

13.3.4 productPlatform

Tag	0x03
Access	Read-only
Type	OCTET STRING
Length	Variable
Value	Null terminated string
Description	The name of the Platform the product is based upon.
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 83 00 00
Sample Response	BD 0A 83 08 **41 56 69 61 74 6F 52 00** 90 00

13.3.5 enabledCLFeatures

Tag	0x04
Access	Read-only
Type	OCTET STRING
Length	2 bytes
Value	Bit mask, see below
Description	Provides information about what contactless protocols are supported
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 84 00 00
Sample Response	BD 04 84 02 **0B 80** 90 00

CL Features:

0x0001 – FeliCa support

0x0002 – EMVCo support

0x0004 – Calypso support

0x0008 – NFC P2P support

0x0010 – SIO processor available

0x0020 – SDR (LF processor) available

0x0040 – Native FW Secure Engine

0x0080 – T=CL support

0x0100 – ISO 14443 A support

0x0200 – ISO 14443 B support

0x0400 - ISO 15693 support

0x0800 – PicoPass 15693-2 support

0x1000 – PicoPass 14443B-2 support

0x2000 – PicoPass 14443A-3 support

0x4000 – RFU

0x8000 – RFU

13.3.6 firmwareVersion

Tag	0x05
Access	Read-only
Type	OCTET STRING
Length	3 bytes
Value	Null terminated string
Description	The version number of the reader's firmware. 1st byte is Major, 2nd byte is Minor, 3rd byte is Revision number.
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 85 00 00
Sample Response	BD 05 85 03 **01 00 00** 90 00

13.3.7 hfControlerVersion

Tag	0x08
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Version number
Description	The version of the HF frontend used for controlling high frequency credentials
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 88 00 00
Sample Response	BD 03 88 01 **18** 90 00

13.3.8 hardwareVersion

Tag	0x09
Access	Read-only
Type	OCTET STRING
Length	variable
Value	Null terminated string
Description	The version of the reader hardware used
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 89 00 00
Sample Response	BD 11 89 0F **50 43 42 2D 30 30 31 37 35 20 52 45 56 32 00** 90 00

13.3.9 hostInterfaceFlags

Tag	0x0A
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Bit mask
Description	Provides information on the interfaces supported by the reader for communication with the host
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 8A 00 00
Sample Response	BD 03 8A 01 **02** 90 00

Host Interface Flags:

0x01 – Ethernet available

0x02 – USB available

0x04 – Serial RS232 available

0x08 – SPI available

0x10 – I^{2}C available

13.3.10 numberOfContactSlots

Tag	0x0B
Access	Read-only
Type	INTEGER
Length	1 byte
Value	Number of contact slots
Description	Number of contact slots supported by the reader
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 8B 00 00
Sample Response	BD 03 8B 01 **01** 90 00

13.3.11 numberOfContactlessSlots

Tag	0x0C
Access	Read-only
Type	INTEGER
Length	1 byte
Value	Number of contactless slots
Description	The number of contactless PCSC slots supported by the reader
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 8C 00 00
Sample Response	BD 03 8C 01 **01** 90 00

13.3.12 numberOfAntennas

Tag	0x0D
Access	Read-only
Type	INTEGER
Length	1 byte
Value	Number of antennas
Description	The number of antennas the reader has
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 8D 00 00
Sample Response	BD 03 8D 01 **01** 90 00

13.3.13 humanInterfaces

Tag	0x0E
Access	Read-only
Type	INTEGER
Length	1 byte
Value	List of tags describing human interfaces
Description	Provides information on the human interfaces supported by the reader
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 8E 00 00
Sample Response	BD 05 8E 03 80 01 **00** 90 00

13.3.14 vendorName

Tag	0x0F
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Null terminated string
Description	The vendor of the reader
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 8F 00 00
Sample Response	BD 0D 8F 0B **48 49 44 20 47 6C 6F 62 61 6C 00** 90 00

13.3.15 exchangeLevel

Tag	0x11
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Bit mask, see below
Description	Provides information about the different APDU levels supported by the reader
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 91 00 00
Sample Response	BD 03 91 01 **04** 90 00

Host Interface Flags:

0x01 – TPDU

0x02 – APDU

0x04 – Extended APDU

13.3.16 serialNumber

Tag	0x12
Access	Read-only
Type	OCTET STRING
Length	Variable, max 32 bytes
Value	Serial number
Description	The serial number of the reader.
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 92 00 00
Sample Response	BD 19 92 17 **4B 54 2D 30 38 36 33 30 30 33 30 2D 31 36 31 30 2D 30 30 30 31 31 34** 90 00

13.3.17 hfControllerType

Tag	0x13
Access	Read-only
Type	OCTET STRING
Length	Variable, max 32 bytes
Value	Null terminated chip name
Description	The IC used for control of HF credentials
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 93 00 00
Sample Response	BD 08 93 06 **52 43 36 36 33 00** 90 00

13.3.18 sizeOfUserEEPROM

Tag	0x14
Access	Read-only
Type	OCTET STRING
Length	2 bytes
Value	Size in bytes
Description	The amount of user EEPROM memory available
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 94 00 00
Sample Response	BD 04 94 02 **04 00** 90 00

13.3.19 firmwareLabel

Tag	0x16
Access	Read-only
Type	OCTET STRING
Length	Variable
Value	Firmware unique ID as string
Description	Detailed information about the firmware version
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 96 00 00
Sample Response	BD 35 96 33 **4F 4B 35 34 32 32 2D 31 2E 30 2E 30 2E 32 33 33 2D 32 30 31 37 30 31 32 37 54 31 32 34 30 31 35 2D 43 34 31 38 45 32 36 30 36 32 45 37 2D 46 4C 41 53 48** 90 00

13.4 Contact Slot Configuration

Contact Slot Configuration Structure

Root	Branch
contactSlotConfiguration (0x03)	voltageSequence (0x02) operatingMode (0x03) contactSlotEnable (0x05)

13.4.1 voltageSequence

Tag	0x02
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	Voltage sequence, see *Section 5.2: Voltage Selection*.
Description	The voltage sequence for contact card power supply
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A3 05 A0 03 82 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A3 04 A0 02 82 00 00
Sample Response	BD 03 82 01 **xx** 90 00

13.4.2 operatingMode

Tag	0x03
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 EMVCo mode, 0x00 ISO/IEC 7816 mode
Description	The operating mode of contact card
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A3 05 A0 03 83 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A3 04 A0 02 83 00 00
Sample Response	BD 03 83 01 **xx** 90 00

13.4.3 contactSlotEnable

Tag	0x05
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Handling of contact card slot
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A3 05 A0 03 85 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A3 04 A0 02 85 00 00
Sample Response	BD 03 85 01 **xx** 90 00

13.5 Contactless Slot Configuration

Contactless Slot Configuration Structure

Root	Branch	
contactlessSlotConfiguration (0x04)	contactlessCommon (0x00)	emdSuppresionEnable (0x07) pollingRFmoduleEnable (0x0A) sleepModePollingFrequency(0x0D) sleepModeCardDetectionEnable (0x0E)
	iso14443aConfig (0x02)	iso14443aEnable (0x00) iso14443aRxTxBaudRate (0x01) MIFAREKeyCache (0x03) MIFAREPreferred (0x04)
	iso14443bConfig (0x03)	iso14443bEnable(0x00) iso14443bRxTxBaudRate (0x01)
	iCLASSConfig (0x06)	iCLASS15693Enable (0x03) iCLASS15693Timeout (0x05) iCLASSActallTimeout (0x06)

13.5.1 Baud Rates

OMNIKEY 5422 and 5122 allow setting maximum baud rate to and from a card for ISO/IEC 14443 Type A and ISO/IEC 14443 Type B protocols.

Commands: iso14443aRxTxBaudRate and iso14443bRxTxBaudRate use the same format. One byte argument defines separately baud rate for receiving (Rx) and transmitting (Tx) data.

The first 4 bits are used to set Rx baud rate, the other for Tx baud rate. The resulting value is combination of bits:

Bit 0 (0x01) – 212 kbps
Bit 1 (0x02) – 424 kbps
Bit 2 (0x04) – 848 kbps

The reader always supports 106 kbps regardless of bit settings. If a card does not support specific transmission speed the reader would use the other value.

For example 0x77 means the reader supports 106, 212, 424, 848 kbps for Rx and Tx. If a card supports only 106 kbps and 424 kbps the reader would use 424 kbps or 106 kbps (in case card activation at 424 kbps fails).

Note: Doubling baud rate does not double transmission speed. In extreme example changing baud rate from 424 kbps to 848 kbps increases transmission speed less than 10%. The number may vary depending on the amount of data transmitted. The worst ratio is for short packets.

Increasing maximum baud rate may cause transmission problems and shorten maximum effective distance between a card and the reader.

13.5.1.1 Examples

0x00 – 106 kbps for Rx and Tx

0x23 – 106 and 424 kbps for Rx and 106, 212, 424 kbps for Tx

0x71 – 106, 212, 424, 848 kbps for Rx and 106, 212 kbps for Tx

13.5.1.2 Default values

ISO/IEC 14443 Type A: 0x33 – 106, 212, 424 kbps for Rx and Tx

ISO/IEC 14443 Type B: 0x33 – 106, 212, 424 kbps for Rx and Tx

13.5.2 Common Parameters

13.5.2.1 emdSupressionEnabled

Tag	0x07
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables EMD suppression
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A0 03 87 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A0 02 87 00 00
Sample Response	BD 03 87 01 **xx** 90 00

13.5.2.2 pollingRFmoduleEnable

Tag	0x0A
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables RF field polling
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A0 03 8A 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A0 02 8A 00 00
Sample Response	BD 03 8A 01 **xx** 90 00

13.5.2.3 sleepModeCardDetectionEnable

Tag	0x0E
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables card detection in sleep mode
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A0 03 8E 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A0 02 8E 00 00
Sample Response	BD 03 8E 01 **xx** 90 00

13.5.2.4 sleepModePollingFrequency

Tag	0x0D
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	Frequency index, see below
Description	The frequency of card insertion check in sleep mode
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A0 03 8D 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A0 02 8D 00 00
Sample Response	BD 03 8D 01 **xx** 90 00

Frequency index:

0x00 - 41Hz (24ms)

0x01 - 20Hz (48ms)

0x02 - 10Hz (96ms)

0x03 - 5Hz (0.2s)

0x04 - 2.5Hz (0.4s)

0x05 - 1.3Hz (0.8s)

0x06 - 0.7Hz (1.4s)

0x07 - 0.3Hz (3.1s)

0x08 - 0.15Hz (6.2s)

0x09 - 0.08Hz (12.3s)

13.5.3 ISO/IEC 14443 Type A

13.5.3.1 iso14443aEnable

Tag	0x00
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables support for ISO/IEC 14443 Type A cards
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A2 03 80 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A2 02 80 00 00
Sample Response	BD 03 80 01 **xx** 90 00

13.5.3.2 iso14443aRxTxBaudRate

Tag	0x01
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	See *Section 13.5.1: Baud Rates*
Description	Sets supported baud rates for ISO/IEC 14443 Type A cards.
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A2 03 81 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A2 02 81 00 00
Sample Response	BD 03 81 01 **xx** 90 00

13.5.3.3 MIFAREKeyCache

Tag	0x03
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables key cache for MIFARE cards
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A2 03 83 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A2 02 83 00 00
Sample Response	BD 03 83 01 **xx** 90 00

13.5.3.4 MIFAREPreferred

Tag	0x04
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables MIFARE preferred mode
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A2 03 84 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A2 02 84 00 00
Sample Response	BD 03 84 01 **xx** 90 00

13.5.4 ISO/IEC 14443 Type B

13.5.4.1 iso14443bEnable

Tag	0x00
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables support for ISO/IEC 14443 Type B cards
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A3 03 80 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A3 02 80 00 00
Sample Response	BD 03 80 01 **xx** 90 00

13.5.4.2iso14443bRxTxBaudRate

Tag	0x01
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	See *Section 13.5.1: Baud Rates*
Description	Sets supported baud rates for ISO/IEC 14443 Type B cards.
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A3 03 81 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A3 02 81 00 00
Sample Response	BD 03 81 01 **xx** 90 00

13.5.5 iCLASS

13.5.5.1 iCLASS15693Enable

Tag	0x03
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables support for iCLASS ISO15693 cards.
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A6 03 83 01 **xx** 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A6 02 83 00 00
Sample Response	BD 03 83 01 **xx** 90 00

13.6 Reader EEPROM

OMNIKEY 5422 and 5122 provide a user available area (1024 bytes) in internal EEPROM memory. The content of this memory is preserved even when the power is off.

When specifying a command to read or write, data offset must be specified (Tag 0x01; 2 bytes).

Root	Branch
readerEEPROM (0x07)	eepromOffset (0x01) eepromRdLength (0x02) eepromWrData (0x03)

13.6.1 Read

Tag	0x02
Access	Read-only
Type	OCTET STRING
Length	Variable
Value	yy yy - address (0x0000-0x03FF) ss - number of bytes to read, xx - read out data
Description	Reads data from user EEPROM area
Get APDU	FF 70 07 6B 0D A2 0B A0 09 A7 07 81 02 **yy yy** 82 01 **ss** 00
Sample Response	9D **ss xx** ... 90 00

13.6.2 Write

Tag	0x03
Access	Write-only
Type	OCTET STRING
Length	Variable
Value	yy yy - address (0x0000-0x03FF) ss - number of bytes to write (0x01-0xEF) xx - data to write
Description	Writes data to user EEPROM area
Sample Set APDU	FF 70 07 6B **0C+ss** A2 **0A+ss** A1 **08+ss** A7 06+ss 81 02 **yy yy** 83 **ss xx** ... 00 Write 1 byte: FF 70 07 6B **0D** A2 **0B** A1 **09** A7 **07** 81 02 **yy yy** 83 01 **xx** 00 Write 16 bytes: FF 70 07 6B **1C** A2 **1A** A1 **18** A7 **16** 81 02 **yy yy** 83 10 **xx** ... 00 Write 64 bytes: FF 70 07 6B **4C** A2 **4A** A1 48 A7 **46** 81 02 **yy yy** 83 10 **xx** ... 00 Write 115 bytes: FF 70 07 6B **7F** A2 7D A1 **7B** A7 **79** 81 02 **yy yy** 83 10 **xx** ... 00 For number of bytes to write (ss value) bigger than 115 bytes, the tag's length is coded with two bytes according to DER TLV coding FF 70 07 6B 10+ss A2 81 0D+ss A1 81 0A+ss A7 07+ss 81 02 yy yy 83 81 ss xx ... 00 Write 116 bytes: FF 70 07 6B **84** A2 **81 81** A1 **81 7E** A7 **81 7B** 81 02 **yy yy** 83 **81 74 xx** ... 00 Write 239 bytes: FF 70 07 6B **FF** A2 **81 FC** A1 **81 F9** A7 **81 F6** 81 02 **yy yy** 83 **81 EF xx** ... 00
Sample Response	9D 00 90 00

13.7 Reader Configuration Control

13.7.1 applysettings

Tag	0x00
Access	Write-only
Type	
Length	
Value	None
Description	Apply settings. This command must be used to accept changes in the reader configuration. The only settings that takes changes immediately are iso14443aRxTxBaudRate, iso14443bRxTxBaudRate. The commands resets device.
Set APDU	FF 70 07 6B 08 A2 06 A1 04 A9 02 80 00 00
Sample Response	9D 00 90 00

13.7.2 restoreFactoryDefaults

Tag	0x01
Access	Write-only
Type	
Length	
Value	None
Description	Sets reader configuration to factory defaults. The commands resets device.
Set APDU	FF 70 07 6B 08 A2 06 A1 04 A9 02 81 00 00
Sample Response	9D 00 90 00

13.7.3 rebootDevice

Tag	0x03
Access	Write-only
Type	
Length	
Value	None
Description	Reboots the reader.
Set APDU	FF 70 07 6B 08 A2 06 A1 04 A9 02 83 00 00
Sample Response	9D 00 90 00

Chapter 14

ICAO Test Commands

14.1 Command Set

The International Civil Aviation Organization (ICAO) has defined a set APDUs for testing e-Passport readers. These are defined in Annex C of the technical report “RF Protocol and Application Test Standard for e-Passport - Part 4”, available from the ICAO website www.icao.int. The standard APDU syntax and standard SCardTransmit API are used with the reserved value of the CLA byte of “FF” and the values of the INS byte are also reserved (in the range of 0x9x).

The commands supported by this reader are as follows:

14.1.1 ICAO Commands

Instruction	Description	Comments
0x92	ISO/IEC 14443-2	Partially supported
0x94	Transmit Pattern	Partially supported
0x96	ISO/IEC 14443-3	Partially supported
0x98	ISO/IEC 14443-4	Not supported
0x9A	Miscellaneous	Partially supported

All of the ICAO test commands are attempted regardless of card presence or type.

14.1.2 0x92 - ISO/IEC 14443-2: ISO/IEC 14443-2 Command APDU

CLA	INS	P1	P2	Pc	Data In	Le
0xFF	0x92	XX	RFU	XX	Lc bytes	–

General: Any data received back from the card is ignored in this test.

14.1.3 ISO/IEC 14443-2 P1 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
0	0	------						Turn off RF Field FF 92 00 00	Yes
0	1	------						Turn on RF Field with no sub-carrier	Yes
1	0	------						Turn on RF Field and transmit Lc bytes	No
1	1	------						RFU	No
--		0	0	------				ISO/IEC 14443 Type A transmission	No
--		0	1	------				ISO/IEC 14443 Type B transmission	No
--		1	0	------				ISO/IEC 15693 transmission (proprietary)	No
--		1	1	------				iCLASS 15693 transmission (proprietary)	No
----				RFU		0	0	106 kbps	No
----						0	1	212 kbps	No
----						1	0	424 kbps	No
----						1	1	848 kbps	No

14.1.4 ISO/IEC 14443-2 Response

Data Out	SW1SW2	
-	0x9000	Operation successful
	0x6700	Wrong length (e.g. Lc absent when P1 b7 =1)
	0x6401	Internal error (e.g. protocol setup failed)

14.1.5 0x94 - Transmit Pattern Command APDU

CLA	INS	P1	P2	Pc	Data In	Le
0xFF	0x94	XX	XX	XX	Lc bytes	XX

General: This test can be used to transmit and/or receive data to/from the card. No parity bit or CRC bytes are added, but framing (that is, start/stop bits, SOF/EOF) WILL be added. This is NOT fully compliant with the ICAO test standard.

14.1.6 ICAO Transmit Pattern P1 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
RFU				---			0	ISO/IEC 14443 Type A transmission	Yes
				---			1	ISO/IEC 14443 Type B transmission	Yes
				xxx			-	Number of bits in last byte to be transmitted	Yes

14.1.7 ICAO Transmit Pattern P2 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
RFU		---			RFU	0	0	Tx - 106 kbps	Yes
		---				0	1	Tx - 212 kbps	Yes
		---				1	0	Tx - 424 kbps	Yes
		---				1	1	Tx - 848 kbps	Yes
		RFU	0	0	---			Rx - 106 kbps	Yes
			0	1	---			Rx - 212 kbps	Yes
			1	0	---			Rx - 242 kbps	Yes
			1	1	---			Rx - 848 kbps	Yes

14.1.8 ICAO Transmit Pattern SW1SW2 Response Bytes

Data Out	SW1SW2	
XX bytes (if Le present)	0x9000	Operation successful
–	0x6700	Wrong length (e.g. Lc and Le are both absent)
	0x6A8A	Modulation index not supported (P1 b7:b4)
	0x6401	Internal error (e.g. protocol setup failed or transceive failed)

14.1.9 0x96 - ISO/IEC 14443-3 Command APDU

CLA	INS	P1	P2	Pc	Data In	Le
0xFF	0x96	XX	XX	XX	Lc bytes	XX

14.1.10 ISO/IEC 14443-3 P1 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
–		--	ISO/IEC 14443 Type A commands						
			0	0	0	0	1	REQA	Yes
			0	0	0	1	0	WUPA	Yes
			0	0	0	1	1	HLTA	No
			0	0	1	0	0	Full ISO-14443A Part 3 (that is, REQA + ANTI-COLLISION + SELECT)	No
			0	1	0	0	1	ANTI-COLLISION CL1	No
			0	1	0	1	0	ANTI-COLLISION CL2	No
			0	1	0	1	1	ANTI-COLLISION CL3	No
			0	1	1	0	0	SELECT (0x70 + UID + BCC in Data In)	No
			ISO/IEC 14443 Type B commands						No
			1	0	0	0	1	REQB (Number of slots in P2)	Yes
			1	0	0	1	0	WUPB (Number of slots in P2)	Yes
			1	0	0	1	1	HLTB (PUPI may be in Data In)	No
			1	0	1	0	0	Slot-MARKER (Slot no in P2)	No
			1	0	1	0	1	ATTRIB (Bit rate in P2 or PUPI + PARAM in Data In)	No
–		xx	-----					No of repetitions of the command	No
0		--	-----					P2 has other parameters (all others are defaults)	No
1		--	-----					Data In contains command data, P2 not used	No

14.1.11 ISO/IEC 14443-3 P2 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
Number of slots for REQB/WUPB command									
RFU					xxx			N = $2^{(b2b1b0)}$ (that is, for b2b1b0 = 0, N = 1)	Yes
Slot no for Slot-MARKER command									
RFU				xxxx				Slot number (0001 = 2, 1111 = 16)	No
Bit rate for ATTRIB command									
--		---			RFU	0	0	Tx - 106 kbps	No
		---				0	1	Tx - 212 kbps	No
		---				1	0	Tx - 424 kbps	No
		---				1	1	Tx - 848 kbps	No
---		RFU	0	0	---			Rx - 106 kbps	No
			0	1	---			Rx - 212 kbps	No
			1	0	---			Rx - 424 kbps	No
			1	1	---			Rx - 848 kbps	No
CID		------						Card Identifier	No

14.1.12 ISO/IEC 14443-3 SW1SW2 Response Bytes

Data Out	SW1SW2	
XX bytes (if Le present)	0x9000	Operation successful
–	0x6700	Wrong length (e.g. Lc absent when P1 b7 is set)
	0x6400	Execution error (e.g. command timeout)
	0x6401	Internal error (e.g. protocol setup failed or transceive failed)
	0x6A88	Requested buffer size too big

14.1.13 Cases for which Data Out is Command Dependent

Command	Data Out
REQA	ATQA (2 bytes)
WUPA	ATQA (2 bytes)
HLTA	–
REQA + ANTI-COLLISION + SELECT	UID (4,7 or 10 bytes) + SAK (1 byte)
ANTI-COLLISION CL1	Cascade UID (4 bytes) + BCC (1 byte)
ANTI-COLLISION CL2	Cascade UID (4 bytes) + BCC (1 byte)
ANTI-COLLISION CL3	Cascade UID (4 bytes) + BCC (1 byte)
SELECT	SAK (1 byte)
REQB	ATQB (14 bytes)
WUPB	ATQB (14 bytes)
HLTB	0x00 + CRCB (2 bytes)
Slot-MARKER	ATQB (14 bytes)
ATTRIB	MBLI+CID (1 byte) + CRCB (2 bytes)

Note: ATQB comprises: 0x50 + PUPI (4 bytes) + APP (4 bytes) + PROTO (3 bytes) +CRCB (2 bytes).

14.1.14 0x98 - ISO/IEC 14443-4 Command APDU

CLA	INS	P1	P2	Pc	Data In	Le
0xFF	0x98	XX	XX	XX	Lc bytes	XX

14.1.15 ISO/IEC 14443-4 P1 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
RFU						0	0	ISO/IEC 14443 Type A RATS (FSDI+CID in P2)	No
						0	1	ISO/IEC 14443 Type A PPS (Bit rate in P2)	No
						1	0	T=CL transmit - Data In contains data to be sent, including the PCB and CID bytes	No
						1	1	T=CL transmit - Data In contains data to be sent (I-Block only), PCB and CID bytes handled by reader	No

14.1.16 ISO/IEC 14443-4 P2 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
FSDI+CID for RATS command									
FSDI				CID				FSDI codes FSD as in ISO/IEC 14443-4	No
Bit rate for PPS command									
RFU		---			RFU	0	0	Tx - 106 kbps	No
		---				0	1	Tx - 212 kbps	No
		---				1	0	Tx - 424 kbps	No
		---				1	1	Tx - 848 kbps	No
		RFU	0	0	---			Rx - 106 kbps	No
			0	1	---			Rx - 212 kbps	No
			1	0	---			Rx - 424 kbps	No
			1	1	---			Rx - 848 kbps	No

14.1.17 ISO/IEC 14443-4 Response Bytes

Data Out	SW1SW2	
XX bytes (if Le present)	0x9000	Operation successful
–	0x6700	Wrong length (e.g. Lc absent for transmit commands)
	0x6400	Execution error (e.g. command timeout)
	0x6A88	Requested buffer size too big

Note: Data Out may also contain an SW1SW2 from the card.

14.1.18 0x9A: ICAO Miscellaneous Command APDU

CLA	INS	P1	P2	Pc	Data In	Le
0xFF	0x9A	XX	XX	XX	Lc bytes	XX

14.1.19 ICAO Miscellaneous P1 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
0	0	0	0	0	0	0	1	Reader information (coded in P2)	No
0	0	0	0	0	0	1	0	ISO/IEC 14443 Type A trigger signal - NOT SUPPORTED	No
0	0	0	0	0	0	1	1	ISO/IEC 14443 Type B trigger signal - NOT SUPPORTED	No
0	0	0	0	0	1	0	0	Reader control (coded in P2)	Yes

Note: All other values of P1 are RFU.

14.1.20 ICAO Miscellaneous P2 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Supported
Coding for Reader information (Lc absent, Le present)									
0	0	0	0	0	0	0	1	Vendor name	No
0	0	0	0	0	0	1	0	Vendor ID	No
0	0	0	0	0	0	1	1	Product name	No
0	0	0	0	0	1	0	0	Product ID	No
0	0	0	0	0	1	0	1	Product serial number	No
0	0	0	0	0	1	1	0	Product firmware version	No
Coding for Reader control (Lc and Le both absent)									
0	0	0	0	0	0	0	0	Turn off polling for card (enter test mode)	Yes
0	0	0	0	0	0	0	1	Turn on polling for card (exit text mode)	Yes

Note: All other values of P2 are either RFU or not supported.

14.1.21 ICAO Miscellaneous Response

Data Out	SW1SW2	
XX bytes (if Le present)	0x9000	Operation successful
–	0x6700	Wrong length (e.g. Le absent for Reader information)
	0x6A82	Function not supported
	0x6A89	Information not available
	0x6A90	Trigger signal not available

Appendix A

Using PC_to_RDR_Escape Command

The PC/SC layer does not allow the use of the SCardTransmit API unless the reader has previously signalled the presence and activation of a card. This prevents the use of commands such as the ICAO test commands or the HID commands without being able to properly recognize and activate a card. In order to be able to use these commands even without a previous card activation, the same functionality of pseudo-APDUs (CLA = 0xFF) is provided through the `PC_to_RDR_Escape` command.

To use the `PC_to_RDR_Escape` command with the default Microsoft CCID driver, the functionality must be first enabled in the Windows registry.

To issue the `PC_to_RDR_Escape` command without a card being present, the reader must be first opened with the `SCardConnect` function with the following settings:

```
dwShareMode = SCARD_SHARE_DIRECT
dwPreferredProtocols = 0
```

Then the vendor IOCTL for the Escape command is defined as follows:

#define IOCTL_CCID_ESCAPE SCARD_CTL_CODE(3500)

The following is an example of the call:

SCardControl(hCard, IOCTL_CCID_ESCAPE, …)

or:

SCardControl(hCard, SCARD_CTL_CODE(3500), …)

The data in the *lpInBuffer* parameter of the length given in *nInBufferSize* are copied to the abData field of the `PC_to_RDR_Escape` command and all the data in the response in `RDR_to_PC_Escape` *abData* field are copied back to the *lpOutBuffer*.

The abData field of the `PC_to_RDR_Escape` must contain the pseudo-APDU to be executed (typically, an ICAO test command or reader configuration). The maximum allowed size of *abData* in `PC_to_RDR_Escape` is currently 262 bytes and the maximum size of the response data in the abData field in the `RDR_to_PC_Escape` response is 464 bytes. The `PC_to_RDR_Escape` and `RDR_to_PC_Escape` do not support any form of chaining to extend the lengths of the supported parameters.

This page intentionally left blank.

This page intentionally left blank.

hidglobal.com