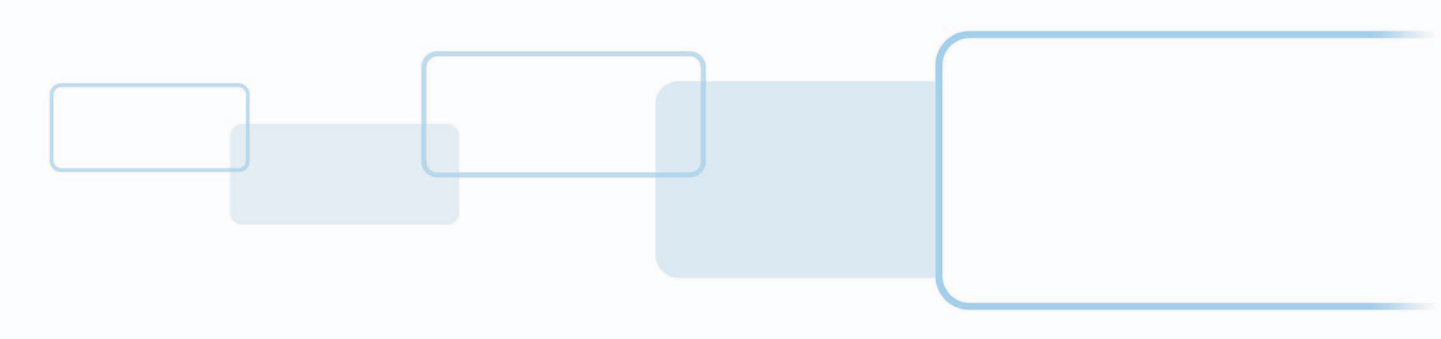
OMNIKEY 5022
Software Developer Guide

PLT-03092

Version: A.0

October 2016



hidglobal.com

Copyright

Trademarks

Contacts

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

Americas and Corporate	Asia Pacific
611 Center Ridge Drive Austin, TX 78753 USA Phone: 866 607 7339 Fax: 949 732 2120	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)	**Brazil**
Haverhill Business Park Phoenix Road Haverhill, Suffolk CB9 7AE England Phone: 44 (0) 1440 711 822 Fax: 44 (0) 1440 714 840	Condomínio Business Center Av. Ermano Marchetti, 1435 Galpão A2 - CEP 05038-001 Lapa - São Paulo / SP Brazil Phone: +55 11 5514-7100
HID Global Customer Support: www.hidglobal.com/customer-service	

Contents

Chapter 1

Introduction

1.1 Product Description

OMNIKEY® 5022 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 is necessary, for example Microsoft CCID driver.

The OMNIKEY 5022 reader features include supporting the common high frequency card technologies, including ISO/IEC 14443 A/B, ISO/IEC 15693, iCLASS®, MIFARE® and others.

Low power mode makes it an ideal solution for portable devices, such as tablets or mobile phones.

It is also possible to add support for new card technologies in the future through device firmware upgrades.

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, ISO/IEC 15693, iCLASS 15693, MIFARE, Topaz, Sony FeliCa® and others
- **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 7816-4	Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Inter-industry commands for interchange Rev. 2005
ISO/IEC 14443	Identification cards - Contactless integrated circuit cards - Proximity cards
ISO/IEC 15693	Identification cards - Contactless integrated circuit cards - Vicinity cards
5022 Reader Data Sheet	Provides a summary of the OMNIKEY 5022 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
ICC	Integrated Circuit Card
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

This page intentionally left blank.

Chapter 2

Host Interfaces

The OMNIKEY 5022 reader supports the following Host Interface.

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

2.1 USB

The device enumerates as a single device. The OMNIKEY 5022 USB protocol stack implements the following device class:

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

The USB CCID interface can be used to send Application Protocol Data Unit (APDU) to the reader. The OMNIKEY 5022 supports 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.

This page intentionally left blank.

Chapter 3

Contactless Card Interface

The OMNIKEY 5022 reader is compliant with CCID specification. Data exchange with a host is done via 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 5022 is able to detect a HF credential by frequently polling the field and to wake up the entire system.

3.1 Polling Mode

OMNIKEY 5022 supports a single polling mode.

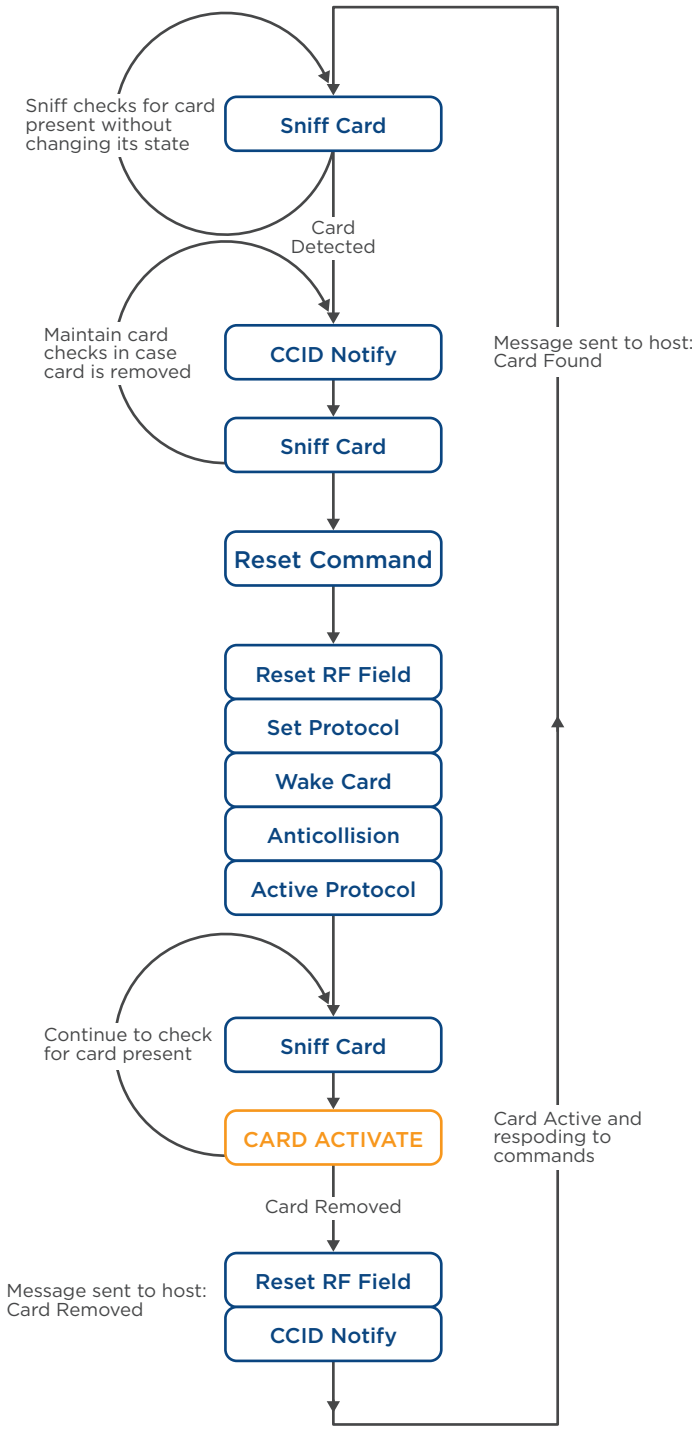
This polling mode operates as follows:

1. The reader polls for cards automatically using a set sequence of card protocols. It is possible to enable or disable each protocol individually and change the sequence.

   The factory default sequence is:

   - ISO/IEC 14443 Type A and Topaz
   - ISO/IEC 14443 Type B
   - iCLASS ISO/IEC 15693
   - FeliCa
   - ISO/IEC 15693
2. When a card or cards are found the host application is notified through CCID.
3. When the host powers up the card 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.
4. APDU layer communication is now possible through the CCID interface.
5. The reader continues to poll for card removal, whereupon it sends an appropriate CCID message to the host application.
6. On card removal, the cycle is repeated.

Polling Operation



3.1.1 Polling and Power Consumption

When the reader is in low power mode it must periodically enable RF field to detect new cards. The power consumption is directly related to 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 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 one protocol - ISO/IEC 14443 Type A enabled during polling sequence.

This page intentionally left blank.

Chapter 4

Contactless Protocol Support

4.1 ISO/IEC 14443 Type A

The OMNIKEY 5022 reader supports all 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. ISO/IEC 14443A cards supported by the reader include, but are not limited to the following:

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

The OMNIKEY 5022 reader allows accessing any T=CL card directly through PC/SC. MIFARE Classic, MIFARE Ultralight and MIFARE Plus in MIFARE Classic emulation mode are supported by the PC/SC commands described in *Section: 8.1 Command Set*.

By default, the card normally is 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

4.2 ISO/IEC 14443 Type B

The OMNIKEY 5022 reader supports all ISO/IEC 14443 Type B compliant cards. Protocol activation when supported is T=CL according to ISO/IEC 14443-3:2001(E) Section 7.

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

4.3 ISO/IEC 15693

The OMNIKEY 5022 reader supports most common ISO/IEC 15693 compliant cards. However, for some cards a low-level transparent mode must be used to make use of all card functions.

Configurable ISO15693 Parameters:

Item	Description
iso15693Enable	Enables or Disables support for ISO/IEC 15693.

4.4 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.

Configurable iCLASS Parameters:

Item	Description
iCLASS15693Enable	Enables or Disables support for iCLASS 15693 card polling.
iCLASS15693DelayTime	Sets minimum chip response to reader command delay.
iCLASS15693Timeout	Sets time to wait for response to a command.
iCLASSActallTimeout	Sets time to wait for response to ACT/ACTALL.

4.5 FeliCa

FeliCa support is limited to card selection with only one card present (no anti-collision) and IDm retrieval. Further card activity requires a Host application to use the low-level transparent mode with pseudo APDUs.

Configurable FeliCa Parameters:

Item	Description
felicaEnable	Enables FeliCa card polling
felicaRxTxBaudRate	Sets the maximum Baud Rate in both directions

This page intentionally left blank.

Chapter 5

Contactless Card Communication

Before communicating with a contactless card, it will be 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 OMNIKEY 5022 supports so called key caching that 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, MIFARE Ultralight and iCLASS credentials is normally done using the PC/SC APDUs described in the next section. 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 then encryption, authentication, reading writing etc., in their application code. The following sections include the PC/SC commands required to communicate with a card and examples of communication with certain specific card types are included in the next chapter.

5.1 PC/SC Commands

5.1.1 Command Set

The PC/SC command set for contactless cards is defined in Section 3.2 of the document "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 the standard APDU syntax and standard SCardTransmit API, but use the reserved value of the CLA byte of “FF”.

PC/SC Commands

Instruction	Description	Comments
0xCA	Get Data	Fully supported
0x82	Load Keys	Partially supported
0x86	General Authenticate	Supported for HID iCLASS cards, and MIFARE, UltraLight 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

5.1.2 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.

Get Data Command APDU

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		
	ISO/IEC 15693	8-byte UID		
	FeliCa	8-byte IDm		
	iCLASS 14443 Type B / 15693	8-byte CSN		
0x01	ISO/IEC 14443 Type A	n Historical bytes		
	Other	–	0x6A81	Function not supported

Notes:

- For the ISO/IEC 14443 Type A Innovision Jewel card, the Data field is 7 bytes of 0x00.
- The number of historical bytes returned is limited to 15.

5.1.3 0x82 – Load Keys

This command allows the application to load keys onto the reader. That includes MIFARE keys, iCLASS keys and secures session keys. All keys except MIFARE keys must be loaded during secure session. MIFARE keys can be loaded when 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: will work with any card type 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 000

KeyNumber: See *Section: 5.2 Key Locations*

KeyLength: 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 nor correct
	0x6990	Security error

5.1.4 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.

Note: The reader will not allow the user to authenticate the HID area of a card outside of a secure session.

General Authenticate Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
FF	86	00	00	05	Data, see below	–

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Version = 01	Address MSB	Address LSB	Key Type	Key Number

Key Types: 00h = PicoPass Debit Key (KD)
01h = PicoPass Credit Key (KC)
60h =MIFARE KeyA
61h = 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 PicoPass 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	SW1	SW2	Description
Normal	90	00	Successful
Warning			
Execution Error	64	00	No Response from media (Time Out)
	65	81	Illegal block number (out of memory space)
Checking Error	67	00	Wrong APDU length
	69	82	Security status not satisfied (not authenticated)
	69	86	Wrong key type
	69	88	Wrong key number

5.1.5 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: 7.1 APDU Commands* for an APDU command. MIFARE Ultralight cards do not require authentication.

Read Binary Command APDU

CLA	INS	P1	P2	Lc	Data In	Le
FF	BO	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
		b5	b4	b3	b2:0	
0	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	SW1	SW2	Description
Normal	90	00	Successful
Warning	62	82	End of data reached before Le bytes (Le is greater than data length).
Execution Error	64	00	No Response from media (Time Out)
Checking Error	67	00	Wrong APDU length
	69	82	Block not authenticated (Security status not satisfied)
	6A	82	Illegal block number (File not found)
	6C	xx	Wrong Le; SW2 encodes the exact number of available data bytes

5.1.6 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
FF	D6	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	SW1	SW2	Description
Normal	90	00	Successful
Warning			
Execution Error	64	00	No Response from media (Time Out)
	65	81	Not usable block number in the memory area (Memory failure)
Checking Error	67	00	Wrong APDU length
	69	82	Block not authenticated (Security status not satisfied)
	6A	82	Illegal block number (File not found)
	6C	xx	Wrong Lc; SW2 encodes the exact number of expected data bytes

5.1.7 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
FF	D4	Address MSB	Address LSB	04	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	SW1	SW2	Description
Normal	90	00	Successful
Warning			
Execution Error	64	00	No Response from media (Time Out)
	65	81	Memory failure (unsuccessful increment / decrement)
Checking Error	67	00	Wrong APDU length
	69	81	Incompatible command
	69	82	Block not authenticated (Security status not satisfied)
	69	86	Command not allowed
	6A	81	Function not supported
	6A	82	Invalid block number

5.1.8 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
FF	D4	Address MSB	Address LSB	04	Data	–

Decrement Supported Cards

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

Decrement Response

Data Field	SW1SW2
empty	See the following table

Increment/Decrement Response SW1SW2 Bytes

Type	SW1	SW2	Description
Normal	90	00	Successful
Warning			
Execution Error	64	00	No Response from media (Time Out)
	65	81	Memory failure (unsuccessful increment / decrement)
Checking Error	67	00	Wrong APDU length
	69	81	Incompatible command
	69	82	Block not authenticated (Security status not satisfied)
	69	86	Command not allowed
	6A	81	Function not supported
	6A	82	Invalid block number

5.2 Key Locations

OMNIKEY 5022 supports 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. iCLASS and secure session slots are initialized to default values in the factory.

Count	First Slot	Last Slot	Key Size	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	667	67	16		✓	✓		iCLASS 3DES card content
6	68	73	16		✓	✓		Secure session AES128

5.2.1 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: 5.1.3 0x82 – Load Keys*.

5.2.2 Key Types

There are five types of keys supported by OMNIKEY 5022:

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

5.2.2.1 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	38	Book 1/page 0 KD
33	Book 0/page 0 KC	39	Book 1/page 0 KC
34	Book 0/page 1 KD	40	Book 1/page 1 KD
35	Book 0/page 1 KC	41	Book 1/page 1 KC
36	Book 0/page 2 KD	42	Book 1/page 2 KD
37	Book 0/page 2 KC	43	Book 1/page 2 KC
36	Book 0/page 3 KD	44	Book 1/page 3 KD
37	Book 0/page 3 KC	45	Book 1/page 3 KC
36	Book 0/page 4 KD	46	Book 1/page 4 KD
37	Book 0/page 4 KC	47	Book 1/page 4 KC
36	Book 0/page 5 KD	48	Book 1/page 5 KD
37	Book 0/page 5 KC	49	Book 1/page 5 KC
36	Book 0/page 6 KD	50	Book 1/page 6 KD
37	Book 0/page 6 KC	51	Book 1/page 6 KC
36	Book 0/page 7 KD	52	Book 1/page 7 KD
37	Book 0/page 7 KC	53	Book 1/page 7 KC

5.2.2.2 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.

5.2.2.3 iCLASS DES keys

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

5.2.2.4 iCLASS 3DES Keys

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

5.2.2.5 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: 6.2 Secure Session Access Rights*.

5.3 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. Use of such a generic command prevents conflicts of reserved INS values but used by certain card reader.

OMNIKEY Specific Command APDU Format

CLA	INS	P1	P2	Lc	Data In	Le
FF	70	07	6B	xx	DER TLV coded PDU (Vendor Payload)	xx

The IFD supports the INS Byte 70 for vendor specific commands.

P1 and P2 constitute the vendor ID. For OMNIKEY products is the VID = 0x076B. The Data Field is constructed as ASN.1 objects/items.

Response for OMNIKEY Specific Commands

Data Field	SW1SW2
DER TLV Response PDU	See ISO 7816-4

OIDs are organized as a leaf tree under an invisible root node. The following table shows the first root nodes.

Vendor Payload Command Types

Vendor Payload	Tag Value (hex)	Description
readerInformationApi	0x02	Reader Information API
response	0x1D	Response
errorResponse	0x1E	Error Response

The following description explains the DER TLV coded data field.

Note: The L field uses the definite form. For the definite form the length octets shall consist of one or more octets, short form or long form. For the long form the IFD shall use the version with two subsequent octets.

5.3.1 Response APDU

For all commands encapsulated in generic 70h APDU, the IFD returns response frame constructed as follows:

Data Field	SW1SW2
DER TLV coded Response PDU	See ISO 7816-4

Two last bytes of response frame are always the 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 (9Dh or BDh) or error response TAG (9Eh).

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

5.3.2 Error Response

The error response TAG caused by the firmware core is 9Eh (Class Context Specific) + (Primitive) + (1Eh). Length is 2 byte. 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**
9Eh	Tag = Error Response (0Eh) + (Class Context Specific) + (Primitive)
02h	Len = 2
cycle	Value byte 1: Cycle in which the error is occurred. See *Error Cycle*, below
error	Value byte 2: Error code. See *Error Code*, below
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	03h	NOT_SUPPORTED
4	04h	TLV_NOT_FOUND
5	05h	TLV_MALFORMED
6	06h	ISO_EXCEPTION
11	0Bh	PERSISTENT_TRANSACTION_ERROR
12	0Ch	PERSISTENT_WRITE_ERROR
13	0Dh	OUT_OF_PERSISTENT_MEMORY
15	0Fh	PERSISTENT_MEMORY_OBJECT_NOT_FOUND
17	11h	INVALID_STORE_OPERATION
19	13h	TLV_INVALID_SETLENGTH
20	14h	TLV_INSUFFICIENT_BUFFER
21	15h	DATA_OBJECT_READONLY
31	1F	APPLICATION_EXCEPTION (Destination Node ID mismatch)
42	2Ah	MEDIA_TRANSMIT_EXCEPTION (Destination Node ID mismatch)
43	2Bh	SAM_INSUFFICIENT_MSGHEADER (Secure Channel ID not allowed)
47	2Fh	TLV_INVALID_INDEX
48	30h	SECURITY_STATUS_NOT_SATISFIED
49	31h	TLV_INVALID_VALUE
50	32h	TLV_INVALID_TREE
64	40h	RANDOM_INVALID
65	41h	OBJECT_NOT_FOUND

5.3.3 Reader Information API

This command group is reserved for GET and SET of reader specific information. See *Section: 7.2 Accessing Configuration*.

5.4 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

5.4.1 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

5.4.2 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.

5.4.3 MIFARE Ultralight C Example

To read and write to a MIFARE Ultralight 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 16-byte MIFARE key with transport key to key slots 0 and 1:

FF 82 00 00 08 49 45 4D 4B 41 45 52 42

FF 82 00 01 08 21 4E 41 43 55 4F 59 46

Authenticate with Key Number 0:

FF 86 00 00 05 01 00 01 60 00

Read block 1 (16 bytes):

FF B0 00 01 10

Write 4 bytes of data to block 4:

FF D6 00 04 04 00 11 22 33

Chapter 6

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.

6.1 Using 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 64 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.

6.1.1 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 chapter Secure Session Access Rights for more details.

6.1.2 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

6.1.3 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
FF	72	00	key	00		

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 SWSW2. `RDRnonce` is used later to compute SSC counter.

Data	SW1SW2
16 bytes `RDRnonce`	9000

In the next step the client sends Mutual Authentication frame. The client must prepare two 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
FF	72	01	00	40	64 crypto data	

Data filed consists of four subfields as described below:

PCnonce	RDRnonce	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 of plus 2 SW1SW2 bytes:

PCnonce	RDRnonce	RDRKey	MAC	SW1SW2
Encrypted 16 bytes	Encrypted 16 bytes	Encrypted 16 bytes	16 bytes	9000

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

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

6.1.4 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 that the APDU should not use **Le** value.

6.1.4.1 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.

SCC	
First 8 bytes of PCnonce	First 8 bytes of RDRnonce

6.1.4.2 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.

6.1.4.3 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
FF	72	02	00	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	9000

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

6.1.4.4 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
FF	72	03	00	00		

The response contains only 2 SW1SW2 bytes.

SW1SW2
9000

6.2 Secure Session Access Rights

The OMNIKEY 5022 supports 6 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.

6.3 Changing the Secure Session Keys

You should change the secure session keys default values. The default values are the same for every customer and therefore are 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 7

Reader Configuration

All OMNIKEY 5022 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`.

7.1 APDU Commands

If the attached host implements a PC/SC environment, the OMNIKEY 5022 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 5022 reader 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 5022 Specific Commands - these include APDUs to manage the reader, to directly access the configuration items

7.2 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 (0x02) tlvVersion (0x00) deviceID (0x01) productName (0x02) productPlatform (0x03) enabledCLFeatures (0x04) firmwareVersion (0x05) hfControllerVersion (0x08) hardwareVersion (0x09) hostInterfacesFlags (0x0A) numberOfContactSlots (0x0B) numberOfContactlessSlots (0x0C) numberOfAntennas (0x01) vendorName (0x01) exchangeLevel (0x01) serialNumber (0x01) hfControllerType (0x01) sizeOfUserEEProm (0x01) firmwareLabel (0x01)
	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) iso15693Config (0x04) iso15693Enable (0x00) felicaConfig (0x05) felicaEnable (0x00) felicaRxTxBaudRate (0x01) iClassConfig (0x06) iClass15693Enable (0x03) iClass15693DelayTime (0x04) iClass15693Timeout (0x05) iClassActallTimeout (0x06)
	Get (0x00) Set (0x01)	readerEEPROM (0x07) eepromOffset (0x01) eepromRead (0x02) eepromWrite (0x03)
	Set (0x01)	readerConfigurationControl (0x09) applySettings (0x00) restoreFactoryDefaults (0x01) rebootDevice (0x03)

7.2.1 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 5022' + 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

7.3 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)
	hostInterfacesFlags (0x0A)
	numberOfContactSlots (0x0B)
	numberOfContactlessSlots (0x0C)
	numberOfAntennas (0x01)
	vendorName (0x01)
	exchangeLevel (0x01)
	serialNumber (0x01)
	hfControllerType (0x01)
	sizeOfUserEEProm (0x01)
	firmwareLabel (0x01)

7.3.1 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

7.3.2 deviceID

Tag	0x01
Access	Read-only
Type	OCTET STRING
Length	2 bytes
Value	0x0000 - 0xFFFF
Description	Product ID, 0x0005 for OMNIKEY 5022
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 81 00 00
Sample Response	BD 04 81 02 00 05 90 00

7.3.3 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 32 00 90 00

7.3.4 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, “AViatoR” for OMNIKEY 5022
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

7.3.5 enabledCLFeatures

Tag	0x04
Access	Read-only
Type	OCTET STRING
Length	2 bytes
Value	Bit mask, 0x0F91 for OMNIKEY 5022
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 0F 81 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

7.3.6 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

7.3.7 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

7.3.8 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 30 34 34 20 52 45 56 32 00 90 00

7.3.9 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. Bit 1 (0x02) = USB interface.
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^2C available

7.3.10 numberOfContactSlots

Tag	0x0B
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Number of contact slots
Description	Number of contact slots supported by the reader. 0 for OMNIKEY 5022
Get APDU	FF 70 07 6B 08 A2 06 A0 04 A0 02 8B 00 00
Sample Response	BD 03 8B 01 00 90 00

7.3.11 numberOfContactlessSlots

Tag	0x0C
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Number of contactless slots
Description	The number of contactless PCSC slots supported by the reader. 1 for OMNIKEY 5022.
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

7.3.12 numberOfAntennas

Tag	0x0D
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Number of antennas
Description	The number of antennas the reader has. 1 for OMNIKEY 5022.
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

7.3.13 vendorName

Tag	0x0F
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Null terminated string.
Description	The vendor of the reader, “HID Global” for OMNIKEY 5022
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

7.3.14 exchangeLevel

Tag	0x11
Access	Read-only
Type	OCTET STRING
Length	1 byte
Value	Bit mask. 0x01 – TPDU, 0x02 – APDU, 0x04 – Extended APDU
Description	Provides information about the different APDU levels supported by the reader. 0x04 for OMNIKEY 5022 (Extended APDU).
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

7.3.15 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

7.3.16 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. “RC663” for OMNIKEY 5022
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

7.3.17 sizeOfUserEEPROM

Tag	0x14
Access	Read-only
Type	OCTET STRING
Length	2 bytes
Value	Size in bytes
Description	The amount of user EEPROM memory available. For OMNIKEY 5022 1024 bytes.
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

7.3.18 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 30 32 32 2D 31 2E 30 2E 30 2E 32 37 30 2D 32 30 31 36 30 31 32 36 54 31 35 32 30 32 34 2D 30 36 43 41 35 34 31 46 30 38 32 41 2D 46 4C 41 53 48 90 00

7.4 Contactless Configuration

Contactless Slot Configuration Structure

Root	Branch	
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)
		iso14bRxTxBaudRate (0x01)
	iso15693Config (0x04)	iso15693Enable (0x00)
	felicaConfig (0x05)	felicaEnable (0x00)
		felicaRxTxBaudRate (0x01)
	iClassConfig (0x06)	iClass15693Enable (0x03)
		iClass15693DelayTime (0x04)
		iClass15693Timeout (0x05)
		iClassActallTimeout (0x06)

7.4.1 Baud Rates

OMNIKEY 5022 allows setting maximum baud rate to and from a card for ISO/IEC 14443 Type A, ISO/IEC 14443 Type B and FeliCa protocols.

Commands: `iso14443aRxTxBaudRate`, `iso14443bRxTxBaudRate` and `felicaRxTxBaudRate` 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.

7.4.1.1 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

7.4.1.2 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

FeliCa: 0x11 – 106, 212 kbps for Rx and Tx

7.4.2 Common Parameters

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)

sleepModeCardDetectionEnable

Tag	0x0E
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enable or Disable 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

pollingSearchOrder

Tag	0x09
Access	Read/Write
Type	OCTET STRING
Length	5 bytes
Value	See description below
Description	Sets polling order
Set APDU	FF 70 07 6B 0F A2 0D A1 0B A4 09 A0 07 89 05 xx xx xx xx xx 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A0 02 89 00 00
Sample Response	BD 07 89 05 xx xx xx xx xx 90 00

The command expects 5 bytes which indicate polling order. Byte position sets priority of the protocol. Protocol on the first byte is checked first and protocol on 5th byte as the last one. Values assigned to protocols:

0x00 – none
0x01 – ISO/IEC 15693
0x02 – ISO/IEC 14443 Type A
0x03 – ISO/IEC 14443 Type B
0x04 – iCLASS ISO/IEC 15693
0x06 – FeliCa

For example 02 03 04 06 01 means order: ISO/IEC 14443 Type A, ISO/IEC 14443 Type B, iCLASS ISO/IEC 15693, FeliCa, ISO/IEC 15693. To support only ISO/IEC 14443 Type A protocol use: 02 00 00 00 00.

Note: If protocol is not included in search order table the card will not be recognized even if the specific protocol is enabled.

emdSupressionEnable

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

7.4.3 ISO/IEC 14443 Type A

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

iso14443aRxTxBaudRate

Tag	0x01
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	See *Section: 7.4.1 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

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

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

7.4.4 ISO/IEC 14443 Type B

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

iso14443bRxTxBaudRate

Tag	0x01
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	See *Section: 7.4 Contactless Configuration*
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

7.4.5 ISO/IEC 15693

iso15693Enable

Tag	0x00
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables support for ISO/IEC 15693 cards.
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A4 03 80 01 xx 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A4 02 80 00 00
Sample Response	BD 03 80 01 xx 90 00

7.4.6 FeliCa

felicaEnable

Tag	0x00
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables support for FeliCa cards.
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A5 03 80 01 xx 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A5 02 80 00 00
Sample Response	BD 03 80 01 xx 90 00

felicaRxTxBaudRate

Tag	0x01
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	See *Section: 7.4 Contactless Configuration*
Description	Sets supported baud rates for FeliCa cards
Set APDU	FF 70 07 6B 0B A2 09 A1 07 A4 05 A5 03 81 01 xx 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A5 02 81 00 00
Sample Response	BD 03 81 01 xx 90 00

7.4.7 iCLASS

iCLASS15693Enable

Tag	0x03
Access	Read/Write
Type	INTEGER
Length	1 byte
Value	0x01 enable, 0x00 disable
Description	Enables or Disables support for iCLASS 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

iCLASS15693DelayTime

Tag	0x04
Access	Read/Write
Type	INTEGER
Length	4 bytes
Value	Timeout
Description	Sets or Gets minimum chip response to reader command delay
Set APDU	70 07 6B 0E A2 0C A1 0A A4 08 A6 06 84 04 xx xx xx xx 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A6 02 84 00 00
Sample Response	BD 06 84 04 xx xx xx xx 90 00

iCLASS15693Timeout

Tag	0x05
Access	Read/Write
Type	INTEGER
Length	4 bytes
Value	Timeout
Description	Sets or Gets time to wait for response to a command
Set APDU	70 07 6B 0E A2 0C A1 0A A4 08 A6 06 85 04 xx xx xx xx 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A6 02 85 00 00
Sample Response	BD 06 85 04 xx xx xx xx 90 00

iCLASSActallTimeout

Tag	0x06
Access	Read/Write
Type	INTEGER
Length	4 bytes
Value	Timeout
Description	Sets or Gets time to wait for response to ACT/ACTALL
Set APDU	70 07 6B 0E A2 0C A1 0A A4 08 A6 06 86 04 xx xx xx xx 00
Sample Response	BD 00 90 00
Get APDU	FF 70 07 6B 0A A2 08 A0 06 A4 04 A6 02 86 00 00
Sample Response	BD 06 86 04 xx xx xx xx 90 00

7.5 Reader EEPROM

OMNIKEY 5022 provides user available area (1024 bytes) in internal EEPROM memory. The content of this memory is preserved even when the power is off.

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

Root	Branch
readerEEPROM (0x07)	eepromOffset (0x01)
	eepromRead (0x02)
	eepromWrite (0x03)

7.5.1 EEPROM Read

Tag	0x02
Access	Read-only
Type	OCTET STRING
Length	variable
Value	yyyy – address (0x0000–0x03FF), nn – number of bytes to read
Description	The version of the TLV encoding used by APDUs
Get APDU	FF 70 07 6B 0D A2 0B A0 09 A7 07 81 02 yy yy 82 01 nn 00
Sample Response	9D ss xx xx xx xx xx 90 00

7.5.2 EEPROM Write

Tag	0x03
Access	Write-only
Type	OCTET STRING
Length	variable
Value	yyyy – address (0x0000–0x03FF), ss – number of bytes to write, xx – data to write
Description	Writes data to user EEPROM area.
Get APDU	FF 70 07 6B 11 A2 0F A1 0D A7 0B 81 02 yy yy 83 ss xx xx xx xx xx 00
Sample Response	9D 00 90 00

7.6 Reader Configuration Control

Commands to apply, reset and store configuration changes.

Reader Configuration Control Structure

Root	Branch
readerConfigurationControl (0x09)	applySettings (0x00)
	restoreFactoryDefaults (0x01)
	rebootDevice (0x03)

7.6.1 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, felicaRxTxBaudRate. The commands resets device.
Get APDU	FF 70 07 6B 08 A2 06 A1 04 A9 02 80 00 00
Sample Response	9D 00 90 00

7.6.2 restoreFactoryDefaults

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

7.6.3 rebootDevice

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

Chapter 8

Transparent Session

Transparent Session provides a mechanism to transparently pass data from an application to a contactless ICC and return the received data transparently back to the application. In addition it provides a user with convenient way of switching the air protocols.

The Transparent Session is often used for using specific functionality which is not available through other standard APIs.

8.1 Command Set

Transparent Session commands use standard APDU syntax, but use the reserved value of the CLA byte of '0xFF', the INS byte of "0x68" and the P1 byte of "0x0E". The commands can be sent via `SCardTransmitl()` or `SCardControl()`.

CLA	INS	P1	P2	Lc	Data Field	Le
FF	68	0E	Command	xx	Data depending on a command in P2	xx

P2	Command	Description
00	Set Protocol	Switching to the specific protocol and different layers of the standard
01	Set Speed	Set communication speed (baud rate)
02	Set Field	Turn on/off the RF field
03	Transceive	Transparent data exchange – transmits and receives any bit or bytes from ICC; IFD is just a transparent channel
06	Enter Transparent Session	Must be called to enter transparent session prior to any other action taken in transparent session
07	Exit Transparent Session	Must be called to exit transparent session and consequently access cards via HID PC/SC commands
08	Get Version	Get version number of the Transparent Session API supported by the IFD
xx	Other values are RFU	

8.1.1 Get Version

Get version number of the Transparent Session API supported by the IFD. The IFD shall return the highest version supported by the IFD. The application will decide whether to proceed or abandon session.

Get Version Command APDU

CLA	INS	P1	P2	Lc	Major Requested Version	Minor Requested Version	Le
FF	68	0E	08	02	XX 1 byte BCD	XX 1 byte BCD	–

Get Version Command Response

Major Requested Version	Minor Requested Version	SW1SW2
XX 1 byte BCD	XX 1 byte BCD	9000

8.1.2 Enter Transparent Session

Stop card tracking by the IFD and initialize the Transparent Session. Divert control card to the application.

Enter Transparent Session Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
FF	68	0E	06	01	00	–

Enter Transparent Session Command Response

Data Field	SW1SW2
empty	See following table

Enter Transparent Session Return Codes

Type	SW1	SW2	Description
Normal	90	00	Successful
Warning			
Execution Error	64	00	No Response from media (Time Out)
	65	81	Illegal block number (out of memory space)
Checking Error	67	00	Wrong APDU length

8.1.3 Exit Transparent Session

Polling and tracking of cards will be enabled after exiting transparent session.

The command is mandatory, if transparent session was entered before, otherwise reader will not restart tracking and polling activity.

As a user is responsible for card handling during transparent session - it is mandatory to have valid state of the card (no error state) prior to exiting transparent session.

If communication to card was lost, user can try to Disconnect() from PC/SC connection with disposition mode set to card resetting.

Exit Transparent Session Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
FF	68	0E	07	01	00	-

Exit Transparent Session Command Response

Data Field	SW1SW2
empty	See following table

Exit Transparent Session Return Codes

Type	SW1	SW2	Description
Normal	90	00	Successful

8.1.4 Set Protocol

Set air protocol type.

Set Protocol Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
FF	68	0E	00	01	Protocol Type	-

Protocol Type

Value	Description
01	ISO/IEC 15693
02	ISO/IEC 14443 Type A
03	ISO/IEC 14443 Type B
04	iCLASS 15693
05	RFU
06	FeliCa

Set Protocol Command Response

Data Field	SW1SW2
empty	See following table

Set Protocol Return Codes

Type	SW1	SW2	Description
Normal	90	00	Successful
Execution Error	6A	81	Function not supported (invalid protocol type)

8.1.5 Set Speed

Set communication speed.

Set Speed Command APDU

CLA	INS	P1	P2	Lc	Data Field			Le
FF	68	0E	01	03	Tx Speed	Rx Speed	RFU = 00	–

Communication Speed (Baud Rate)

Tx Speed / Rx Speed	Description
00	106 kbps
01	212 kbps
02	424 kbps
03	848 kbps

Set Speed Command Response

Data Field	SW1SW2
empty	See following table

Set Speed Return Codes

Type	SW1	SW2	Description
Normal	90	00	Successful
Execution Error	6B	00	Invalid input parameters

8.1.6 Set Field

Turn on/off the RF field.

Set Field Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
FF	68	0E	02	01	RF Field State	–

RF Field State

RF Field State Value	SW1SW2
00	Turn RF field off
01	Turn RF field on

Set Field Command Response

Data Field	SW1SW2
empty	See following table

Set Field Return Codes

Type	SW1	SW2	Description
Normal	90	00	Successful
Execution Error	6B	00	Invalid input parameters

8.1.7 Transceive

Transceive command provides direct access to a contactless card. It sends Generic Card Command and returns card's response.

It is not recommended to use transparent communication unless no alternative is available. This is usually when the card supports functionality that is not accessible by using standard HID PC/SC API of the OMNIKEY 5022.

Transceive Command APDU

CLA	INS	P1	P2	Lc	Data Field	Le
FF	68	0E	03	10 + n	Header 10 bytes + Generic Card Command	00

Header, 10 Bytes				Generic Card Command
1 Byte	1 Byte	4 Bytes	4 Bytes	N Bytes
TxRxFlags	ValidBits	Timeout MSB first	RFU	Data to send

The **TxRxFlags** byte defines actions taken by reader when data is sent/received to/from card.

The **ValidBits** defines bit length of the data (may be not a multiple of eight).

The **Timeout** value specifies the amount of time reader should wait for a response from the card, in microseconds.

TxRxFlags Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Usage
–	–	–	–	–	–	–	1	Tx CRC Enable	CRC is computed by the reader and sent to the card, otherwise the user must do it.
–	–	–	–	–	–	1	–	Tx Parity Enable	Parity is computed by the reader and sent to the card, otherwise the user must do it.
–	–	–	–	–	1	–	–	Rx CRC Enable	CRC is checked by the reader and not passed to the user, otherwise the CRC bytes are passed to the user for checking.
–	–	–	–	1	–	–	–	Rx Parity Enable	Parity is checked by the reader and not passed to the user, otherwise the parity bit is passed to the user for checking.
–	–	–	1	–	–	–	–	EMD Suppression Enable	While receiving, EMD suppression algorithm is activated. May be used only for ISO144443-4 transfers.

ValidBits Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description	Status
00000000								Bit length is whole number of bytes	Supported
0xxx				----				Number of bits expected to be received in the last byte (rx data shifted left)	Not supported
1xxx				----				Number of bits expected to be received in the first byte (rx data shifted right - e.g. 14A Anticollision)	Not supported
----				-xxx				Number of bits to be transmitted in the last byte	Supported

Transceive Command Response

Data Out		
2 Bytes, RF Controller Status	**n Bytes, Card Response (Optional)**	**2 Bytes, SW1SW2**
STS, RxB	[XX, XX, ... XX]	9000h - successful 6700h – generic error 6F00h - transceive fail

STS = Status Byte.

RxB = Number of valid bits in last byte (RxBits)

Status Byte Codes

Code	Name	Description
00h	`HID_SUCCESS`	Transceive command succeeded
01h	`HID_FAIL`	Generic error
03h	`HID_BUFFER_FULL`	Receive buffer overflow
06h	`HID_TIMED_OUT`	Receive timeout expired or unrecognized media
07h	`HID_PARITY_ERR`	Parity error

8.2 Using Transparent Session

The `SCardConnect()` function establishes a connection between the calling application and a smart card contained by a specific reader. `dwShareMode` parameter describes whether other applications may form connections to the card. Transparent session can be used with two modes: `SCARD_SHARE_SHARED` or `SCARD_SHARE_DIRECT`.

`SCardTransmit()` can be used in `SCARD_SHARE_SHARED` mode (and `SCARD_PROTOCOL_T1`) when an ICC is present and the application has established an ICC handle. This usually means that the card has been "powered up" – air protocol type has been set along with other transmission parameters. The card is selected and ready to work.

`SCardControl()` can be used in `SCARD_SHARE_ DIRECT` mode (and `SCARD_PROTOCOL_UNDEFINED`) when an ICC is not present. Usually after entering the transparent session the card state is unknown. Application should first reinitialize the ICC (reset card). The air protocol type and speed should be set manually before sending any data to a card.

Basing scenario for using Transparent Session is as follows:

1. At first application executes the `Enter Transparent Session` command to establish a session.
2. If an ICC is present and the state of the IFD / ICC is known, then the application can immediately start communicating at this state using either "Transceive" command or standard APDU (as already defined or supported by the ICC) for communication to the ICC.

   During transparent session application is fully responsible for card state and if some invalid actions are taken it is possible that the reader will return error after exiting the session.

   For example transparent session should not turn RF field off before exiting transparent session.
3. To finish the session application executes `Exit Transparent Session` command. After this command reader restores all its autonomous activities as polling, tracking, etc.

   Certain actions performed with the ICC during the transparent session may result in invalid state of the card causing PC/SC errors. Application can handle such exceptions by calling `SCardDisconnect()` function with a disposition mode set to card reset or use transactions that will finish with card reset.

8.2.1 Example: Transparent Communication with MIFARE Ultralight EV1 Card (Get Version EV1 command)

1. Enter Transparent Session

   ```
   >> FF680E0600
   << 9000
   ```

2. Transceive: GetVersion EV1

   ```
   >> FF680E030B0F00040A8BC00000000060FE
   << 000000040301010000B03 9000
   ```

3. ExitTransparentSession

   ```
   >> FF680E070100
   << 9000
   ```

If GetVersion fails, card goes into a non-selected state – that is why user should disconnect PCSC with CardDispositionMode = RESET_CARD or do the session operations in PCSC transaction which ends with card reset

8.2.2 Example: Transparent Communication with MIFARE Classic 1K Including Anticollision

Connection made in Direct Mode

1. Enter session

   ```
   >> FF680E0600
   << 9000
   ```

2. Turn Field Off

   ```
   >> FF680E020100
   << 9000
   ```

3. Set Protocol to ISO14443A

   ```
   >> FF680E000102
   << 9000
   ```

4. Set Speed to 106 kbps

   ```
   >> FF680E0103000000
   << 9000
   ```

5. Transceive: REQA

   ```
   >> FF680E030B0A0700108D800000000026FE
   << 00000200 9000
   ```

6. Transceive - Anticollision

   ```
   >> FF680E030C0A00000A5870000000009320FE
   << 0000265C404A70 9000
   ```

7. Transceive: Anticollision continued

```
>> FF680E03110F00000A5870000000009370265C404A70FE
<< 000018 9000
```

Card found

8. Transceive: HaltA

```
>> FF680E030C0B00000034F8000000005000FE
<< 06006F00
```

9. Transceive: Try REQA again

```
>> FF680E030B0A0700108D800000000026FE
<< 06006F00
```

Chapter 9

ICAO Test Commands

9.1 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:

9.1.1 ICAO Commands

Instruction	Description	Comments
0x92	ISO/IEC 14443-2	Fully supported
0x94	Transmit Pattern	Partially supported
0x96	ISO/IEC 14443-3	Fully supported
0x98	ISO/IEC 14443-4	Fully supported
0x9A	Miscellaneous	Partially supported

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

9.1.2 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.

9.1.3 ISO/IEC 14443-2 P1 Coding

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

9.1.4 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)

9.1.5 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.

9.1.6 ICAO Transmit Pattern P1 Coding

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

9.1.7 ICAO Transmit Pattern P2 Coding

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

9.1.8 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)

9.1.9 0x96 - ISO/IEC 14443-3 Command APDU

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

9.1.10 ISO/IEC 14443-3 P1 Coding

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

9.1.11 ISO/IEC 14443-3 P2 Coding

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

9.1.12 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

9.1.13 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).

9.1.14 0x98 - ISO/IEC 14443-4 Command APDU

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

9.1.15 ISO/IEC 14443-4 P1 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description
RFU						0	0	ISO/IEC 14443 Type A RATS (FSDI+CID in P2)
						0	1	ISO/IEC 14443 Type A PPS (Bit rate in P2)
						1	0	T=CL transmit - Data In contains data to be sent, including the PCB and CID bytes
						1	1	T=CL transmit - Data In contains data to be sent (I-Block only), PCB and CID bytes handled by reader

9.1.16 ISO/IEC 14443-4 P2 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description
FSDI+CID for RATS command								
FSDI				CID				FSDI codes FSD as in ISO/IEC 14443-4
Bit rate for PPS command								
RFU		---		RFU		0	0	Tx - 106 kbps
		---				0	1	Tx - 212 kbps
		---				1	0	Tx - 424 kbps
		---				1	1	Tx - 848 kbps
		RFU	0	0	---			Rx - 106 kbps
			0	1	---			Rx - 212 kbps
			1	0	---			Rx - 424 kbps
			1	1	---			Rx - 848 kbps

9.1.17 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.

9.1.18 0x9A: ICAO Miscellaneous Command APDU

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

9.1.19 ICAO Miscellaneous P1 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description
0	0	0	0	0	0	0	1	Reader information (coded in P2)
0	0	0	0	0	0	1	0	ISO/IEC 14443 Type A trigger signal - NOT SUPPORTED
0	0	0	0	0	0	1	1	ISO/IEC 14443 Type B trigger signal - NOT SUPPORTED
0	0	0	0	0	1	0	0	Reader control (coded in P2)

Note: All other values of P1 are RFU.

9.1.20 ICAO Miscellaneous P2 Coding

b7	b6	b5	b4	b3	b2	b1	b0	Description
Coding for Reader information (Lc absent, Le present)								
0	0	0	0	0	0	0	1	Vendor name
0	0	0	0	0	0	1	0	Vendor ID
0	0	0	0	0	0	1	1	Product name
0	0	0	0	0	1	0	0	Product ID
0	0	0	0	0	1	0	1	Product serial number
0	0	0	0	0	1	1	0	Product firmware version
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)
0	0	0	0	0	0	0	1	Turn on polling for card (exit text mode)

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

9.1.21 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

Where Data Out for supported Reader information is as follows:

9.1.22 ICAO Miscellaneous Data Out for Supported Reader Information

Information	Data Out
Vendor name	“HID Global”
Vendor ID	0x076B
Product name	“5127 CK”
Product ID	0x5127
Product serial number	16-byte Unique ID
Product firmware version	4-byte number

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 = 'FF') 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), ...)
```

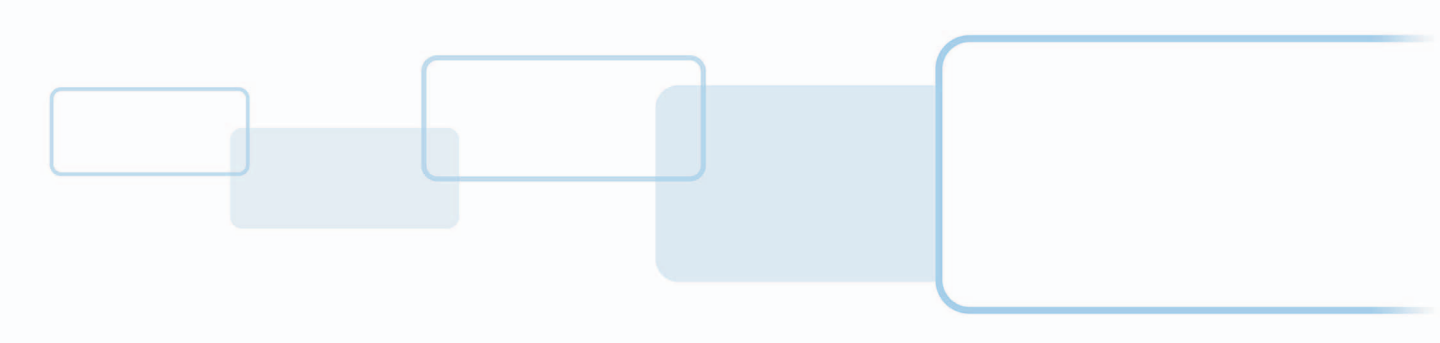
The data in the *lpInBuffer* parameter of the length given in *nInBufferSize* are copied to the abData field of the `PC_to_RDR_Escape` command and all the data in the response in `RDR_to_PC_Escape` *abData* field are copied back to the *lpOutBuffer*.

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.

Revision History

Date	Description	Version
October 2016	Initial Release.	A.0



hidglobal.com